

INF3400 Digital Mikroelektronikk

YNGVAR BERG

2 juni, 2010

Contents

1	Innledning	11
1.1	Transistor som bryter (Kapittel 1.3 side 9)	12
1.1.1	nMOS transistor	12
1.1.2	pMOS transistor	12
1.2	CMOS Inverter (Kapittel 1.4.1 side 10)	13
1.3	NAND port (Kapittel 1.4.2 side 10 - 11)	14
1.3.1	Opptrekk og nedtrekk i CMOS porter	14
1.4	Kombinatorisk logikk (Kapittel 1.4.3 side 11 - 12)	14
1.4.1	Serie- og parallellkobling av transistorer	15
1.5	NOR port (Kapittel 1.4.4 side 12 - 13)	16
1.6	Komplementær logikk (Compound gates) (Kapittel 1.4.5 side 13 - 14)	16
1.7	Pass transistorer og transmisjonsporter (Kapittel 1.4.6 side 14 - 15)	17
1.7.1	Pass transistorer	17
1.7.2	Transmisjonsporter	17
1.8	Tristate (Kapittel 1.4.7 side 17 - 18)	18
1.9	Multipleksere (Kapittel 1.4.8 side 18 - 20)	18
1.10	Latcher (Kapittel 1.4.9 side 20 - 21)	19
1.10.1	Latcher	19
1.11	Vipper (Kapittel 1.4.9 side 22 - 23)	20
2	MOS transistor teori	22
2.1	MOS transistor i tverrsnitt (Kapittel 1.3 side 7 - 9)	23
2.1.1	pn overgang	23
2.1.2	Tverrsnitt av MOS transistorer	23
2.2	Tverrsnitt av CMOS Inverter (Kapittel 1.5.1 side 23 - 24)	24
2.3	Akkumulasjon, deplesjon og inversjon (Kapittel 2.1 side 67 - 68)	24
2.4	Enkel beskrivelse av MOS transistor (Kapittel 2.1 side 68 - 71)	25
2.5	Enkel MOS transistor modell (Kapittel 2.2 side 71 - 75)	27
2.5.1	Utvikling av enkel strømmodell for nMOS transistor	27
2.5.2	Strømmodell for pMOS transistor	29
2.5.3	I-V karakteristikker	30

2.6	Kanallengdemodulasjon eller kanalforkortning	30
	<i>(Kapittel 2.4.2 side 86 - 89)</i>	
2.6.1	I-V karakteristikk for nMOS transistor med kanallengdemodulasjon	31
2.6.2	Transferkarakteristikk for inverter med kanallengdemodulasjon	31
2.6.3	Mål	31
2.7	Bodyeffekt	
	<i>(Kapittel 2.4.3 side 87 - 88)</i>	31
2.7.1	Mål	32
2.8	Svak inversjon	
	<i>(Kapittel 2.4.4 side 88 - 89)</i>	32
2.8.1	Mål	32
2.9	Lekkasje i pn-overganger	
	<i>(Kapittel 2.4.5 side 89 - 90)</i>	33
2.9.1	Mål	33
2.10	Tunnelering	
	<i>(Kapittel 2.4.6 side 90)</i>	33
2.10.1	Mål	33
2.11	Geometriavhengighet for MOS transistorer.	
	<i>(Kapittel 2.4.8 side 92)</i>	34
2.12	Temperaturavhengighet for MOS transistorer.	
	<i>(Kapittel 2.4.7 side 90 - 92)</i>	34
2.13	CMOS inverter DC karakteristikk	
	<i>(Kapittel 2.5.1 side 94 - 97)</i>	35
2.13.1	Område A, se Fig. 2.30	36
2.13.2	Område B, se Fig. 2.30	36
2.13.3	Område C, se Fig. 2.30	36
2.13.4	Område D, se Fig. 2.30	36
2.13.5	Område E, se Fig. 2.30	36
2.13.6	Oppsummering	37
2.13.7	Mål	37
2.14	Transistorstørrelser	
	<i>(Kapittel 2.5.2 side 97)</i>	37
2.14.1	Mål	37
2.15	Støymargin	
	<i>(Kapittel 2.5.3 side 98 - 99)</i>	38
2.15.1	Mål	38
2.16	Invertere med statisk last	
	<i>(Kapittel 2.5.4 side 100 - 101)</i>	39
2.16.1	Sammenligning med vanlig inverter	40
2.16.2	Mål	40
2.17	Pseudo nMOS inverter	
	<i>(Kapittel 2.5.4 side 101)</i>	40
2.17.1	Sammenligning med andre invertere	40
2.17.2	Mål	41
2.18	Pass transistor DC karakteristikk	
	<i>(Kapittel 2.5.6 side 101 - 102)</i>	41
2.18.1	Terskelfall	41
2.18.2	Transmisjonsport	42
2.18.3	Konduktans	43
2.18.4	Mål	43
2.19	Tristate inverter	
	<i>(Kapittel 2.5.6 side 102 - 103)</i>	44
2.19.1	Mål	44
3	CMOS prosess teknologi	45
3.1	Introduksjon til utleggsregler	
	<i>(Kapittel 1.5.3 side 28 - 32)</i>	46
3.1.1	Mål	46

3.2	Introduksjon til CMOS fabrikkasjonsprosess	47
	<i>(Kapittel 1.5.2 side 24 - 28)</i>	
3.2.1	Mål	48
3.3	Utlegg av port	48
	<i>(Kapittel 1.5.4 side 32 - 33)</i>	
3.3.1	Mål	49
3.4	Stick diagrammer	49
	<i>(Kapittel 1.5.4 side 33 - 35)</i>	
3.4.1	Mål	50
3.5	Bakgrunn for CMOS teknologier og fremstilling av wafer	50
	<i>(Kapittel 3.2.1 og 3.2.2 side 113 - 115)</i>	
3.5.1	Bakgrunn for CMOS teknologier	50
3.5.2	Fremstilling av wafer	51
3.5.3	Mål	51
3.6	Fotolitografi	51
	<i>(Kapittel 3.2.3 side 115 - 116)</i>	
3.6.1	Mål	51
3.7	Fremstilling av brønn og kanal	52
	<i>(Kapittel 3.2.4 side 117 - 118)</i>	
3.7.1	Mål	52
3.8	Silisium dioksid	52
	<i>(Kapittel 3.2.5 side 118 - 119)</i>	
3.8.1	Mål	52
3.9	Isolering	53
	<i>(Kapittel 3.2.6 side 119 - 120)</i>	
3.9.1	Mål	53
3.10	Gate oksid	54
	<i>(Kapittel 3.2.7 side 120 - 121)</i>	
3.10.1	Mål	54
3.11	Fremstilling av source og drain	54
	<i>(Kapittel 3.2.8 side 121 - 123)</i>	
3.11.1	Mål	55
3.12	Kontakter, metall og passivering	56
	<i>(Kapittel 3.2.9 og 3.2.10 side 124)</i>	
3.12.1	Kontakter og metall	56
3.12.2	Passivering	56
3.12.3	Mål	56
3.13	Bakgrunn for utleggsregler	56
	<i>(Kapittel 3.3.1 og 3.2.10 side 125 - 129)</i>	
3.13.1	Regler for brønner	56
3.13.2	Regler for transistorer	56
3.13.3	Regler for kontakter	56
3.13.4	Regler for metall	57
3.13.5	Mål	57
3.14	Enkle MOS kapasitans modeller	57
	<i>(Kapittel 2.3.1 side 75 - 77)</i>	
3.14.1	Mål	57
3.15	Gatekapasitans detaljer	58
	<i>(Kapittel 2.3.2 side 77 - 80)</i>	
3.15.1	Mål	59
3.16	Diffusjonskapasitans detaljer	60
	<i>(Kapittel 2.3.3 side 80 - 83)</i>	
3.16.1	Mål	60
3.17	Enkle RC modeller	61
	<i>(Kapittel 2.6 side 103 - 105)</i>	
3.18	RC forsinkelsesmodeller	62
	<i>(Kapittel 4.2 - 4.2.1 side 158 - 161)</i>	
3.18.1	Estimering av forsinkelse	62
	<i>(Kapittel 4.2 side 158 - 159)</i>	

3.18.2	RC forsinkelsesmodeller (Kapittel 4.2.1 side 159 - 161)	62
3.18.3	RC modell	63
3.18.4	Mål	63
3.19	Hastighetsmetning (Kapittel 2.4.2 side 84 - 86)	64
3.19.1	Mål	65
4	Kretskarakterisering	66
4.1	Elmore forsinkelsesmodell (Kapittel 4.2.1.3 side 161 - 164)	67
4.1.1	Eksempler	67
4.1.2	Parasittisk tidsforsinkelse	68
4.1.3	Elektrisk effort	68
4.1.4	Effort tidsforsinkelse	68
4.1.5	Mål	68
4.2	Lineær forsinkelsesmodell (Kapittel 4.2.2 side 165 - 166)	69
4.2.1	Mål	69
4.3	Logisk effort (Kapittel 4.2.3 side 166 - 167)	69
4.3.1	Mål	69
4.4	Parasittisk tidsforsinkelse (Kapittel 4.2.4 side 167 - 169)	70
4.4.1	Mål	70
4.5	Stige- og falltidsforsinkelse for inngang (Kapittel 4.2.5.1 side 169 - 170)	71
4.5.1	Mål	71
4.6	Ulik transisjonstidspunkt for innganger (Kapittel 4.2.5.2 side 170 - 171)	71
4.6.1	Mål	72
4.7	MOS kapasitanser for inverter ved transisjoner	72
4.7.1	Område a.	73
4.7.2	Område b.	73
4.7.3	Område c.	73
4.7.4	Område d.	73
4.7.5	Område e.	73
4.7.6	MOS kapasitanser	74
4.7.7	Mål	74
4.8	Gate source kapasitans (Kapittel 4.2.5.3 side 171 - 172)	74
4.8.1	Mål	74
4.9	Bootstrapping (Kapittel 4.2.5.4 side 172 - 173)	75
4.9.1	Mål	75
4.10	Tidsforsinkelse i en logisk port (Kapittel 4.3.2 side 173 - 174)	76
4.10.1	Mål	76
4.11	Tidsforsinkelse i en kjede av logiske porter (Kapittel 4.3.2 side 174 - 178)	76
4.11.1	Eksempel	78
4.11.2	Mål	79
4.12	Optimalt antall porter i en kjede (Kapittel 4.3.2 side 178 - 181)	79
4.12.1	0 porter i tillegg	79
4.12.2	1 port i tillegg	79
4.12.3	2 porter i tillegg	80
4.12.4	3 porter i tillegg	80
4.12.5	Optimalt antall porter i en kjede	80

4.12.6 Mål	80
4.13 Oppsummering av logisk effort (Kapittel 4.3.5 side 183 - 185)	81
4.13.1 Mål	81
4.14 Introduksjon til effektforbruk (Kapittel 4.4 side 187 - 188)	82
4.14.1 Mål	82
4.15 Statisk effektforbruk (Kapittel 4.4.1 side 188 - 190)	83
4.15.1 Mål	83
4.16 Introduksjon til interkonnekt (Kapittel 4.5 side 196 - 197)	84
4.16.1 Mål	84
4.17 Motstand i interkonnekt (Kapittel 4.5.1 side 198 - 200)	84
4.17.1 Mål	85
4.18 Dielektrikum	86
4.18.1 Lav-k (Low-k)	86
4.18.2 Høy-k (High-k)	86
4.18.3 Mål	86
4.19 Kapasitans i interkonnekt (Kapittel 4.5.2 side 200 - 205)	86
4.19.1 Mål	87
4.20 Forsinkelse i interkonnekt (Kapittel 4.5.3 side 205 - 207)	88
4.20.1 Eksempel	88
4.20.2 Mål	88
4.21 Crosstalk (Kapittel 4.5.4 side 207 - 210)	89
4.21.1 Forsinkelse	89
4.21.2 Crosstalk støy	89
4.21.3 Mål	90
4.22 Bredde og avstand for ledere (Kapittel 4.6.1 side 219 - 220)	90
4.22.1 Mål	91
4.23 Valg av metallag for ledere (Kapittel 4.6.2 side 219 - 221)	91
4.24 Beskyttelse av ledere (Kapittel 4.6.3 side 221)	92
4.25 Designmarginer og variasjoner (Kapittel 4.7.1-3 side 231 - 233)	92
4.25.1 Variasjon i forsyningsspenning	92
4.25.2 Variasjon i temperatur	93
4.25.3 Prosessvariasjoner	93
4.26 Designhjørner. (Kapittel 4.7.4 side 233-235)	93
4.27 Matching. (Kapittel 4.7.5 side 235 - 237)	94
4.28 Pålitelighet. (Kapittel 4.8.1 side 239 - 240)	94
4.29 Elektromigrasjon (Kapittel 4.8.2 side 240)	95
4.30 Varmeutvikling (self-heating) (Kapittel 4.8.3 side 241)	95
4.31 Hot carriers (Kapittel 4.8.4 side 241 - 242)	96
4.32 Latchup (Kapittel 4.8.5 side 242 - 244)	96

4.33	Skalering	
	(Kapittel 4.9 side 245 - 246)	98
4.33.1	Mål	98
4.34	Transistorskalering	
	(Kapittel 4.9.1 side 246 - 249)	98
4.35	Skalering av interkonnekt	
	(Kapittel 4.9.2 side 249 - 250)	99
4.36	Teknologiutvikling	
	(Kapittel 4.9.3 side 251)	99
4.37	Designpåvirkninger	
	(Kapittel 4.9.4 side 252-257)	100
4.37.1	Økt ytelse og kostnader	100
4.37.2	Interkonnekt	100
4.37.3	Effektforbruk	100
4.37.4	Produktivitet	101
4.37.5	Fysiske begrensinger	101
5	Kombinatoriske kretser	104
5.1	Bubble pushing	
	(Kapittel 6.2.1.1 side 321)	105
5.1.1	Mål	105
5.2	Komplementær logikk (compound gates)	
	(Kapittel 6.2.1.2 side 321 - 323)	106
5.2.1	Eksempel	106
5.2.2	Mål	108
5.3	Hvordan rekkefølgen på inngangssignaler påvirker tidsforsinkelse	
	(Kapittel 6.2.1.3 side 324)	108
5.3.1	Mål	109
5.4	Asymmetriske porter	
	(Kapittel 6.2.1.4 side 324 - 325)	109
5.4.1	Mål	109
5.5	Porter med skew	
	(Kapittel 6.2.1.5 side 325)	110
5.5.1	Mål	110
5.6	P/N forhold	
	(Kapittel 6.2.1.6 side 325 - 327)	110
5.6.1	Mål	110
5.6.2	Kortslutningseffekt	111
5.6.3	Mål	111
5.7	Pseudo nMOS	
	(Kapittel 6.2.2.1 side 327 - 330)	112
5.7.1	pseudo nMOS inverter	112
5.7.2	2inngangs pseudo nMOS NAND port	113
5.7.3	2inngangs pseudo nMOS NOR port	114
5.7.4	Pseudo nMOS design	115
5.7.5	Eksempel	115
5.7.6	Mål	116
5.8	Ganged CMOS	
	(Kapittel 6.2.2.2 side 330 - 331)	117
5.8.1	$A = B = 0$	117
5.8.2	A og B er forskjellige	117
5.8.3	$A = B = 1$	118
5.8.4	Mål	118
5.9	Source følger opptrekslogikk	
	(Kapittel 6.2.2.3 side 331)	118
5.9.1	Mål	118
5.10	Kaskode spenning Svitsj logikk	
	(Kapittel 6.2.3 side 331 - 332)	119
5.10.1	Eksempel	119

5.10.2	Mål	120
5.11	Introduksjon til dynamiske kretser	
	<i>(Kapittel 6.2.4 side 332 - 334)</i>	120
5.11.1	Footed inverter	121
5.11.2	Footed NAND2 port	122
5.11.3	Footed NOR2 port	122
5.11.4	Precharge og evalueringsdetaljer	123
5.11.5	Mål	124
5.12	Domino logikk	
	<i>(Kapittel 6.2.4.1 side 334 - 336)</i>	124
5.12.1	Mål	125
5.13	Dual-rail domino logikk	
	<i>(Kapittel 6.2.4.2 side 336 - 337)</i>	125
5.13.1	Mål	126
5.14	Ladningsdeling	
	<i>(side 340)</i>	126
5.14.1	Mål	127
5.15	Precharge av interne noder i nedtrekket	
	<i>(Kapittel 6.2.4.4 side 340 - 341)</i>	127
5.15.1	Mål	127
5.16	Blødere (keepers)	
	<i>(Kapittel 6.2.4.3 side 338 - 340)</i>	128
5.16.1	Mål	129
5.17	Logisk effort i dynamiske kjeder	
	<i>(Kapittel 6.2.4.5 side 341 - 343)</i>	129
5.17.1	Eksempel	130
5.17.2	Mål	130
5.18	NP og Zipper domino	
	<i>(Kapittel 6.2.4.7 side 343 - 345)</i>	131
5.18.1	Mål	131
5.19	CMOS med transmisjonsporter	
	<i>(Kapittel 6.2.5.1 side 347 - 348)</i>	131
5.19.1	Mål	132
5.20	Komplementær pass transistor logikk (CPL)	
	<i>(Kapittel 6.2.5.2 side 348 - 349)</i>	132
5.20.1	Mål	133
5.21	Lean Integration med pass transistorer (LEAP)	
	<i>(Kapittel 6.2.5.3 side 349 - 350)</i>	133
5.21.1	Mål	133
5.22	Andre pass transistor familier	
	<i>(Kapittel 6.2.5.4 side 350)</i>	134
5.22.1	DPL	134
5.22.2	EEPL	134
5.22.3	PPL	134
5.22.4	SRPL	134
5.22.5	DCVSPG	135
5.22.6	Mål	135
5.23	Differensielle kretser	
	<i>(Kapittel 6.4.1 side 359 - 360)</i>	135
5.23.1	Mål	136
5.24	Oversikt over kretsproblemer	
	<i>(Kapittel 6.3 side 350-351)</i>	136
5.25	Terskefall	
	<i>(Kapittel 6.3.1 side 351)</i>	136
5.26	Nivåfeil	
	<i>(Kapittel 6.3.2 side 352)</i>	137
5.27	Lekkasje	
	<i>(Kapittel 6.3.3 side 352-353)</i>	138

5.28	Støy som skyldes ladningsdeling (Kapittel 6.3.4 side 353)	138
5.29	Støy i forsyningsspenningen (Kapittel 6.3.4 side 353)	139
5.30	Hot spots (Kapittel 6.3.5 side 354-355)	139
5.31	Injeksjon av minoritetsbærere (Kapittel 6.3.7 side 355-356)	139
5.32	Back-gate (Kapittel 6.3.8 side 356)	140
5.33	Følsomhet for støy i diffusjon (Kapittel 6.3.9 side 357)	141
5.34	Prosessfølsomhet (Kapittel 6.3.10 side 357)	141
5.35	Domino støyregnskap (Kapittel 6.3.11 side 357-358)	141
5.36	Sense-amplifier kretser (Kapittel 6.4.2 side 360 - 362)	142
5.36.1	Mål	142
5.37	Sample set differensiell logikk (SSDL) (Kapittel 6.4.2.1 side 362 - 363)	142
5.37.1	Mål	143
5.38	Enable/Disable CMOS differensiell logikk (ECDL) (Kapittel 6.4.2.2 side 363 - 364)	143
5.38.1	Mål	144
5.39	Latched CMOS differensiell logikk (LCDL) (Kapittel 6.4.2.3 side 364)	145
5.39.1	Mål	145
5.40	Differensiell strøm svitsj logikk (DCSL) (Kapittel 6.4.2.4 side 364)	146
5.40.1	Mål	147
5.41	BiCMOS kretser (Kapittel 6.4.3 side 365)	147
5.41.1	Mål	147
5.42	Sammenligning av CMOS logikk typer (Kapittel 6.6 side 367 - 369)	148
5.42.1	Mål	148
6	Sekvensielle kretser	149
6.1	Introduksjon til sekvensielle kretser (Kapittel 7.1 side 383 - 384)	150
6.1.1	Mål	150
6.2	Introduksjon til sekvensering av statiske kretser (Kapittel 7.2 side 384)	150
6.2.1	Mål	150
6.3	Sekvenseringsmetoder (Kapittel 7.2.1 side 385 - 387)	151
6.3.1	Timing for kombinatorisk logikk	152
6.3.2	Timing for vippe	152
6.3.3	Timing for Latch	153
6.3.4	Mål	153
6.4	Begrensninger for maks forsinkelse (Kapittel 7.2.2 side 388 - 392)	154
6.4.1	Vipper	154
6.4.2	Latcher	154
6.4.3	Mål	155
6.5	Begrensninger for minimum forsinkelse (Kapittel 7.2.3 side 392 - 396)	155
6.5.1	Vipper	155

6.5.2	Latcher	156
6.5.3	Mål	157
6.6	Fordeling av tid mellom klokkefasene (Kapittel 7.2.4 side 396 - 399)	158
6.6.1	Mål	158
6.7	Klokke skew (Kapittel 7.2.5 side 399 - 402)	159
6.7.1	Mål	159
6.8	Konvensjonelle CMOS latcher (Kapittel 7.3.1 side 402 - 405)	160
6.8.1	Varianter av transparente latcher	161
6.8.2	Mål	161
6.9	Konvensjonelle CMOS vipper (Kapittel 7.3.2 side 405 - 407)	162
6.9.1	Mål	163
6.10	Latcher som styres av klokkepulser (Kapittel 7.3.3 side 407 - 408)	163
6.10.1	Puls generatorer	163
6.10.2	Puls latch	164
6.10.3	Mål	164
6.11	Latcher og vipper som kan resettes (Kapittel 7.3.4 side 408 - 409)	165
6.11.1	Vippe med asynkron set og reset	165
6.11.2	Detaljer	167
6.11.3	Mål	167
6.12	Latcher og vipper som kan enables (Kapittel 7.3.5 side 410)	167
6.12.1	Mål	168
6.13	Latcher med logikk (Kapittel 7.3.6 side 410 - 411)	168
6.13.1	Mål	169
6.14	Klass semidynamisk vippe (SDFF) (Kapittel 7.3.7 side 411 - 412)	169
6.14.1	Mål	170
6.15	Differensielle vipper (Kapittel 7.3.8 side 412 - 413)	170
6.15.1	Mål	171
6.16	Ekte en-fase (TSPC) latcher og vipper (Kapittel 7.3.9 side 414)	171
6.16.1	Mål	172
7	Indeks	174

Chapter 1

Innledning

1.1 Transistor som bryter

(Kapittel 1.3 side 9)

Definisjoner:

1. **Logisk 1** V_{DD} .
2. **Logisk 0** V_{SS} , GND.

1.1.1 nMOS transistor

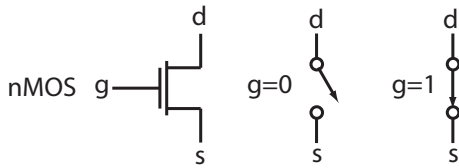


Figure 1.1: *nMOS transistor som bryter.*(FIG1.9)

En nMOS transistor som bryter er vist i Fig. 1.1, transistorens tre terminaler er **g**ate (inngang), **s**ource og **d**rain. En nMOS transistor kan betraktes som en bryter; avhengig av inngang (gate) vil det kunne gå strøm mellom drain og source. Når inngangen er 0 går det ingen strøm mellom drain og source, og vi sier at transistoren er **AV**. Når inngangen er 1 kan det gå strøm mellom drain og source, og vi sier at transistoren er **PÅ**.

Konvensjoner:

1. Den av drain og source terminalene på en nMOS transistor som har lavest spenning kalles *source*.
2. Den av drain og source terminalene på en nMOS transistor som har høyest spenning kalles *drain*.
3. En positiv strøm (strømretning) vil for en nMOS transistor alltid gå fra drain til source.

1.1.2 pMOS transistor

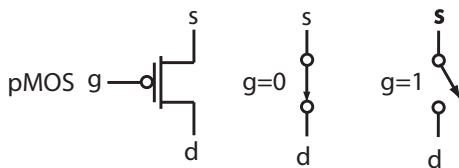


Figure 1.2: *pMOS transistor som bryter.*(FIG1.9)

En pMOS transistor som bryter er vist i Fig. 1.2. Transistorens tre terminaler er **g**ate (inngang), **s**ource og **d**rain. Når inngangen er logisk 0 kan det gå strøm mellom source og drain, og vi sier at transistoren er **PÅ**. Når inngangen er logisk 1 går det ingen strøm mellom source og drain, og vi sier at transistoren er **AV**.

Konvensjoner:

1. Den av drain og source terminalene på en pMOS transistor som har høyest spenning kalles *source*.

2. Den av drain og source terminalene på en pMOS transistor som har lavest spenning kalles *drain*.
3. En positiv strøm (strømretning) vil for en pMOS transistor alltid gå fra *source* til *drain*.

1.2 CMOS Inverter

(Kapittel 1.4.1 side 10)

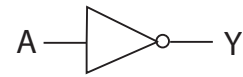


Figure 1.5: *Inverter symbol.*(FIG1.10b)

* Repetisjon fra INF1400.

Symbolet for en CMOS inverter er vist i Fig. 1.5.

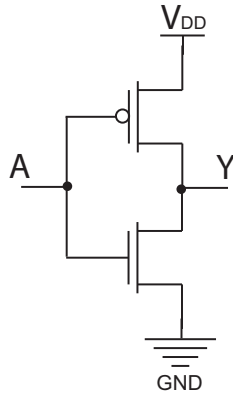


Figure 1.3: *Inverter skjematikk.*(FIG1.10a)

Dersom vi setter en pMOS- og en nMOS transistor sammen og kobler til spenningsreferansene V_{DD} og V_{SS} (GND) får vi en CMOS inverter som vist i Fig. 1.3. CMOS teknologi er grunnleggende inverterende, dvs. dersom man bruker pMOS transistorer mellom en utgang på en port og logisk 1 (V_{DD}), og tilsvarende nMOS transistorer mellom utgangen og logisk 0 (V_{SS}), vil utgangen danne en inverterende funksjon. Vi får typisk inverter, NAND, NOR eller generelle boolske funksjoner på formen $Y = \overline{A \cdot (B + C)}$.

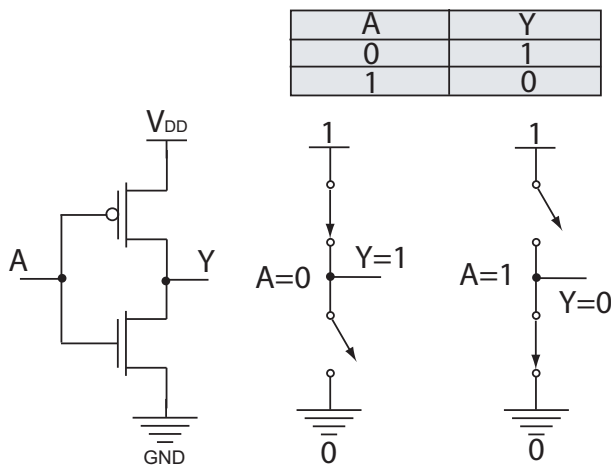


Figure 1.4: *Inverter skjematikk og sannhetstabell.*(FIG1.10a og Tabell 1.1)

Som vist i Fig. 1.4 vil utgangen på en inverter være 1 når inngangen er 0, og utgangen vil være 0 når inngangen er 1.

1.3 NAND port

(Kapittel 1.4.2 side 10 - 11)

* Repetisjon fra INF1400.

1.3.1 Opptrekk og nedtrekk i CMOS porter

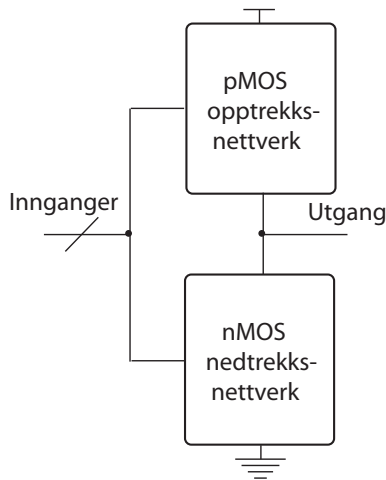


Figure 1.6: Generell logisk port med opptrekk bestående av pMOS transistorer og nedtrekk bestående av nMOS transistorer. (FIG1.13)

En generell port med generelt **opptreksnettverk** og **nedtreksnettverk** er vist i Fig. 1.6. Vi definerer et opp- eller nedtrekk som *på* dersom det finnes en strømvei (signalvei) mellom utgangen og en spenningsreferanse. Med andre ord et nedtrekk er *på* dersom det finnes en serie (kjede) av nMOS transistorer som alle er *på* og som forbinder utgangen med V_{SS} . I motsatt tilfelle er nedtrekket *av*. For et opptrekk som er *på* finnes det en serie (kjede) av pMOS transistorer som alle er *på* og som forbinder utgangen med V_{DD} . I motsatt tilfelle er opptrekket *av*. En kjede av transistorer i et nettverk kan bestå av en eller flere transistorer. I komplementær CMOS logikk (statisk CMOS) vil alltid en og bare en av opptrekk- og nedtreksnettverkene være *på*.

Skjematisk fremstilling av en to-inngangs CMOS NAND port (NAND2) er vist i Fig. 1.7. Den logiske funksjonen er $Y = \overline{A \cdot B}$. Opptrekket består av to pMOS transistorer i parallell og nedtrekket av to nMOS transistorer i serie. For at utgangen Y skal kunne trekkes til logisk 0 må begge nMOS transistorene være *på*, dvs. inngangene A og B må begge være logisk 1. Det er tilstrekkelig at en av inngangene er logisk 0 for å trekke utgangen Y til logisk 1. Vi sier at nedtrekket og opptrekket er komplementære, dvs. den ene utelukker den andre.

Som vist i Fig. 1.8 er det enkelt å utvide en to-inngangs NAND port til en tre-inngangs NAND port (NAND3). Symbolet for en NAND port med to innganger er vist i Fig. 1.9.

A	B	nedtrekk	opptrekk	Y
0	0	AV	PÅ	1
0	1	AV	PÅ	1
1	0	AV	PÅ	1
1	1	PÅ	AV	0

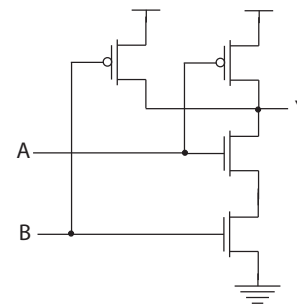


Figure 1.7: To-inngangs NAND port (NAND2) skjematikk og sannhetstabell. (FIG1.11a og Table 1.2)

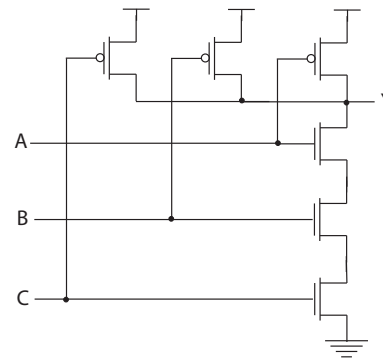


Figure 1.8: Tre-inngangs NAND port (NAND3) skjematikk. (FIG1.12)

1.4 Kombinatorisk logikk

(Kapittel 1.4.3 side 11 - 12)

* Repetisjon fra INF1400.

	opptrekk AV	opptrekk PÅ
nedtrekk AV	Z	1
nedtrekk PÅ	0	X (crowbarred)

Table 1.1: Utgangstilstander for en CMOS port.

Den generelle kombinatoriske porten i Fig. 1.6 vil enten trekke utgangen til 1 eller 0 avhengig av inngangssignalene. Dersom opptrekket og nedtrekket ikke var komplementære kan det forekomme tilfeller der hverken opptrekket eller nedtrekket er *på*, eller at begge er *på*. I tabell 1.1 er alle mulige utgangstilstander for en CMOS port vist. I tilfellet der hverken opptrekket eller nedtrekket er *på* vil utgangen ikke være drevet av porten, vi kaller denne tilstanden høyimpedant Z. I tilfeller der både opptrekket og nedtrekket er *på* får vi udefinert utgang X (crowbarred).

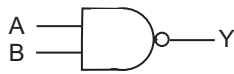


Figure 1.9: Symbol for NAND port med to innganger. (FIG1.11b)

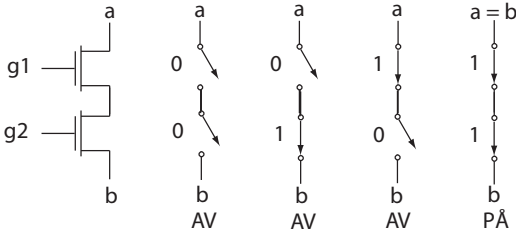


Figure 1.10: Seriekobling av nMOS transistorer. (FIG1.14a)

1.4.1 Serie- og parallellkobling av transistorer

Ulike tilstander for to seriekoblede nMOS transistorer er vist i Fig. 1.10. Begge transistorene må være på for at nettverket bestående av de to transistorene skal være på. Dette tilsvarer inngangssignaler $g1 = g2 = 1$ og dette vil medføre at $a = b$. For øvrige inngangsverdier vil det ikke kunne gå en strøm mellom a og b og nettverket er av. Vi ser at dersom $b = 0$ (GND) har vi at $a = b = 0$ når $g1 = g2 = 1$ eller $g1 \cdot g2 = 1$.

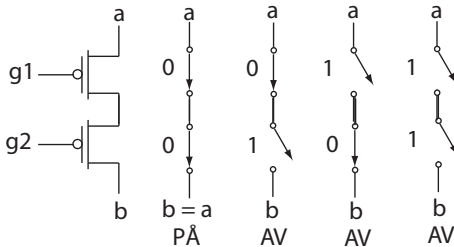


Figure 1.11: Seriekobling av pMOS transistorer. (FIG1.14b)

Ulike tilstander for to seriekoblede pMOS transistorer er vist i Fig. 1.11. Begge transistorene må være på for at nettverket bestående av de to transistorene skal være på. Dette tilsvarer inngangssignaler $g1 = g2 = 0$ og dette vil medføre at $b = a$. For øvrige inngangsverdier vil det ikke kunne gå en strøm mellom a og b og nettverket er av. Vi ser at dersom $a = 1$ (V_{DD}) har vi at $b = a = 1$ når $g1 = g2 = 0$ eller $g1 + g2 = 0$.

Ulike tilstander for to parallellkoblede nMOS transistorer er vist i Fig. 1.12. Minst en av transistorene må være på for at nettverket bestående av de to transistorene skal være på. Dette tilsvarer inngangssignaler $g1 + g2 = 1$ og dette vil medføre at $a = b$. For øvrige inngangsverdier, dvs. $g1 = g2 = 0$ eller $g1 + g2 = 0$, vil det ikke kunne gå en strøm mellom a og b og nettverket er av. Vi ser at dersom $b = 0$ (GND) har vi at $a = b = 0$ når $g1 + g2 = 1$.

Ulike tilstander for to parallellkoblede pMOS transistorer er vist i Fig. 1.13. Minst en av transistorene må være på for at nettverket bestående av de to transistorene

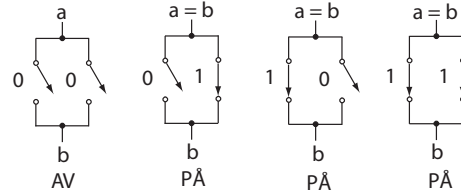
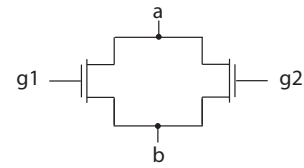


Figure 1.12: Parallellkobling av nMOS transistorer. (FIG1.14c)

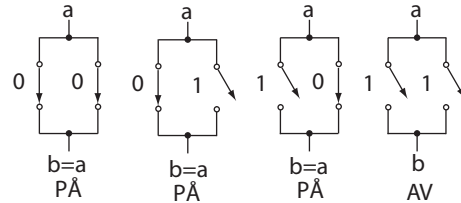
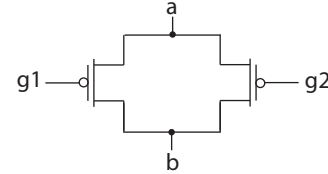


Figure 1.13: Parallellkobling av pMOS transistorer. (FIG1.14d)

skal være på. Dette tilsvarer inngangssignaler $g1 \cdot g2 = 0$ og dette vil medføre at $b = a$. For øvrige inngangsverdier, dvs. $g1 = g2 = 1$ eller $g1 \cdot g2 = 1$, vil det ikke kunne gå en strøm mellom a og b og nettverket er av. Vi ser at dersom $a = 1$ (V_{DD}) har vi at $b = a = 1$ når $g1 \cdot g2 = 0$.

Komplementære opptrekk og nedtrekk bestående av henholdsvis to pMOS- og to nMOS transistorer, og inngangene A og B , vil være:

1. $A \cdot B$ som tilsvarer to parallellkoblede pMOS transistorer som er på, ved at minst en av inngangene er 0, og to seriekoblede nMOS transistorer som er på, ved at begge inngangene er 1. Vi kaller utgangen Y og inngangene A og B , og får da $\bar{Y} = A \cdot B$ eller $Y = \overline{A \cdot B}$. Dette tilsvarer en NAND funksjon.
2. $A + B$ og som tilsvarer to seriekoblede pMOS transistorer som er på, ved at begge inngangene er 0, og to parallellkoblede nMOS transistorer som er på, ved at minst en av inngangene er 1. Vi kaller utgangen Y og inngangene A og B , og får da $\bar{Y} = A + B$ eller $Y = \overline{A + B}$. Dette tilsvarer en NOR funksjon.

1.5 NOR port

(Kapittel 1.4.4 side 12 - 13)

* Repetisjon fra INF1400.

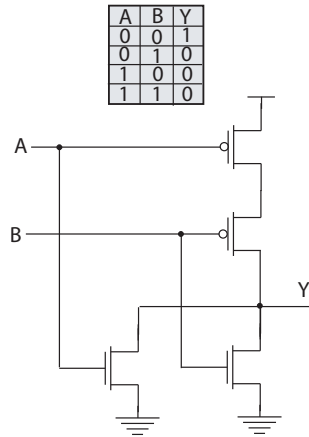


Figure 1.14: To-inngangs NOR port skjematikk og sannhetstabell. (FIG1.15a og Table 1.4)

Skjematikk for en CMOS NOR port er vist i Fig. 1.14. Den logiske funksjonen er $Y = \overline{A + B}$. Opptrekket består av to pMOS transistorer i serie og nedtrekket består av to nMOS transistorer i parallell. For at utgangen Y skal kunne trekkes til logisk 1 må begge pMOS transistorene være på, dvs. inngangene A og B må være logisk 0. Det er tilstrekkelig at en av inngangene er logisk 1 for å trekke utgangen Y til logisk 0. Vi ser at nedtrekket og opptrekket er komplementære.

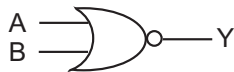


Figure 1.15: Symbol for NOR2 port med to innganger. (FIG1.15b)

Symbolet for en NOR2 port med to innganger er vist i Fig. 1.15.

1.6 Komplementær logikk (Compound gates)

(Kapittel 1.4.5 side 13 - 14)

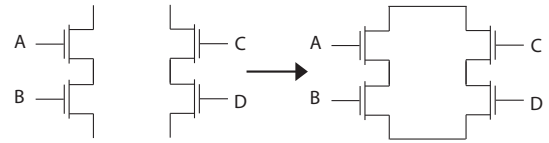


Figure 1.16: Nedtrekket for funksjonen $Y = \overline{(A \cdot B) + (C \cdot D)}$. (FIG1.17a og b)

Et eksempel på en boolsk funksjon implementert ved hjelp av en komplementær CMOS port kan uttrykkes på formen $Y = \overline{(A \cdot B) + (C \cdot D)}$.

Nedtrekket vil bestå av nMOS transistorer og vi har at Y bare kan bli 0 når $(A \cdot B) + (C \cdot D) = 1$. Dette forutsetter at $A \cdot B$ eller $C \cdot D$ er på. Vi ser da at nedtrekket består av to grener med seriekoblede nMOS transistorer, dvs. to transistorer med innganger henholdsvis A og B i serie, og to transistorer med innganger henholdsvis C og D i serie. I Fig. 6.26 er nedtrekket vist.

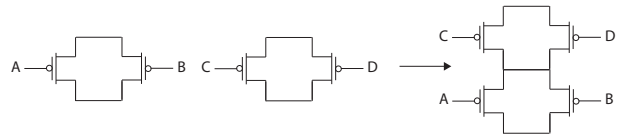


Figure 1.17: Opptrekket for funksjonen $Y = \overline{(A \cdot B) + (C \cdot D)}$. (FIG1.17c og d)

Opptrekket vil bestå av pMOS transistorer og vi har at Y bare kan bli 1 når $(A \cdot B) + (C \cdot D) = 0$. Dette forutsetter at A og/eller B ($A \cdot B = 0$) og C og/eller D ($C \cdot D = 0$) er på. Vi ser da at opptrekket består av to grener med parallellkoblede pMOS transistorer, dvs. to transistorer med innganger A og B i parallell, og to transistorer med innganger C og D i parallell. Til slutt må disse to parallellgrenene settes i serie slik at forutsetningen for opptrekket blir oppfylt. I Fig. 1.17 er opptrekket vist.

Den fullstendige komplementære CMOS kretsen som implementerer funksjonen $Y = \overline{(A \cdot B) + (C \cdot D)}$ er vist i Fig. 1.18.

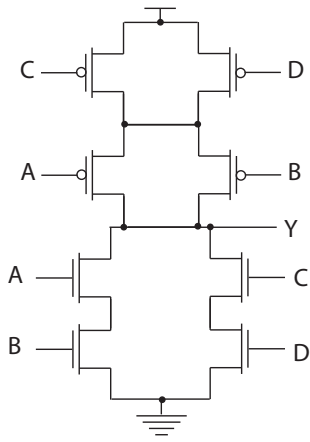


Figure 1.18: Komplementær CMOS port for funksjonen $Y = (A \cdot B) + (C \cdot D)$. (FIG1.17e)

1.7 Pass transistorer og transmisjonsporter

(Kapittel 1.4.6 side 14 - 15)

1.7.1 Pass transistorer

Styrken til et signal er et mål på hvor nær signalet er en spenningsreferanse, vanligvis 1 (V_{DD}) eller 0 (V_{SS} eller GND).

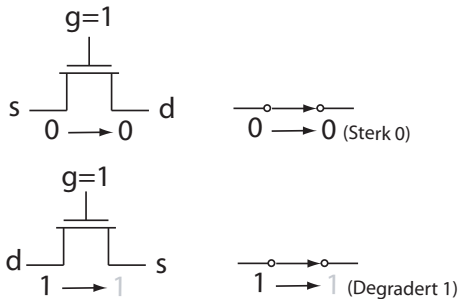


Figure 1.19: nMOS pass transistor. (FIG1.19a og c)

Det er vanlig å koble nMOS transistorer med source til 0. Dette skyldes at nMOS transistorer er effektive til å transmittere logisk 0. Vi sier at en logisk 0 transmitteres gjennom en nMOS transistor med stor styrke, dvs, en logisk 0 transmitteres fra source på en nMOS transistor til en like sterk 0 på drain terminalen. Dersom en logisk 1 skal transmitteres gjennom en nMOS transistor, dvs. fra drain til source, vil vi få en svak eller degradert logisk 1 på source terminalen. Dette skyldes elektriske egenskaper i nMOS transistoren. Transmisjonsegenskaper til nMOS transistoren er vist i Fig. 1.19.

For en pMOS transistor er det omvendt. Det er vanlig å koble pMOS transistorer med source til 1. Dette skyldes at pMOS transistorer er effektive til å transmittere logisk 1. Vi sier at en logisk 1 transmitteres gjennom en pMOS transistor med stor styrke, dvs, en logisk 1 transmitteres fra source på en pMOS transistor til en like sterk 1 på drain terminalen. Dersom en logisk 0 skal transmitteres

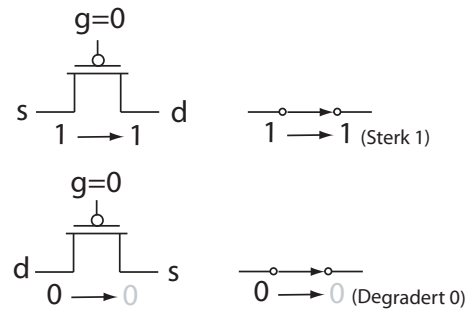


Figure 1.20: pMOS pass transistor. (FIG1.19d og f)

gjennom en pMOS transistor, dvs. fra drain til source, vil vi få en svak eller degradert logisk 0 på source terminalen. Dette skyldes elektriske egenskaper i pMOS transistoren. Transmisjonsegenskaper til pMOS transistoren er vist i Fig. 1.20.

1.7.2 Transmisjonsporter

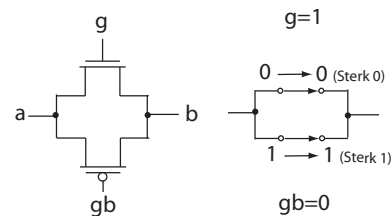


Figure 1.21: Transmisjonsport. (FIG1.20a, b og c). gb betyr g invertert, dvs. $gb = \bar{g}$.

Ved å kombinere en nMOS pass transistor og en pMOS pass transistor i parallell kan vi lage en transmisjonsport som kan brukes til å transmittere både logisk 0 og 1, som vist i Fig. 1.21. nMOS transistoren vil sørge for logisk 0 og pMOS transistoren vil sørge for logisk 1.

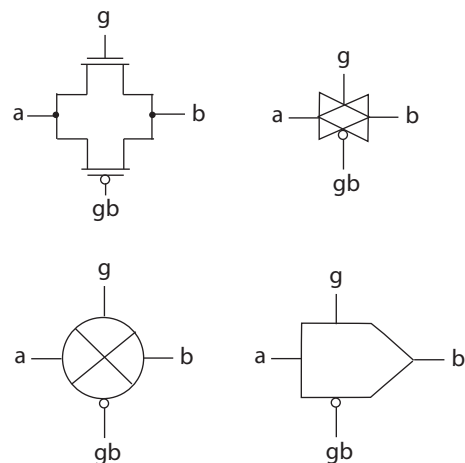


Figure 1.22: Transmisjonsport symboler. (FIG1.20d)

Symboler for transmisjonsporter i CMOS er vist i Fig. 1.22.

1.8 Tristate

(Kapittel 1.4.7 side 17 - 18)

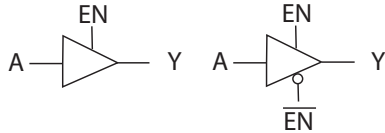


Figure 1.23: Tristate buffer symbols. (FIG1.24)

Symboler for tristate buffer er vist i Fig. 1.23. Tristate porter brukes ofte når flere enheter (porter) skal kunne drive en felles buss.

EN/\overline{EN}	A	Y
0/1	0	Z
0/1	1	Z
1/0	0	0
1/0	1	1

Table 1.2: Sannhetstabell for tristate buffer. EN/\overline{EN} er kontrollsignaler (enable), A er inngangen og Y er utgangen.

Et tristate buffer benytter et enable signal som bestemmer om porten skal drive utgangen eller ikke. Dersom utgangen ikke drives vil den representere en høy impedans (Z). Sannhetstabellen for tristate buffer er vist i tabell 1.2.

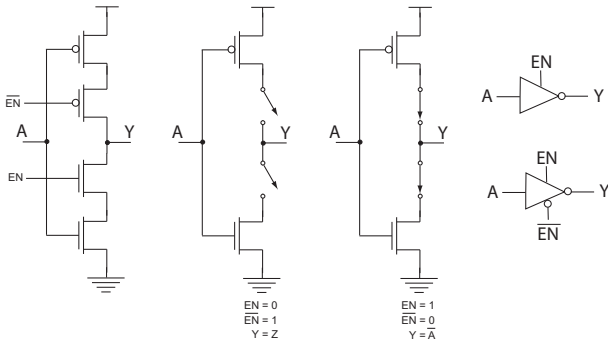


Figure 1.24: Tristate inverter. (FIG1.26)

En tristate inverter er vist i Fig. 2.16.

1.9 Multipleksere

(Kapittel 1.4.8 side 18 - 20)

S/\overline{S}	D1	D0	Y
0/1	X	0	0
0/1	X	1	1
1/0	0	X	0
1/0	1	X	1

Table 1.3: Sannhetstabell for to-inngangs (2:1) multiplakser. EN/\overline{EN} er kontrollsignaler (enable), D0 og D1 er innganger og Y er utgangen.

Multipleksere brukes i CMOS hukommelselementer og i en rekke andre kretser. En multiplexer brukes til å selektere en av flere innganger. Sannhetstabell for en to-inngangs multiplexer er vist i tabell 1.3 med inngangene er D0 og D1, kontrollsignalene er S og \overline{S} og utgang Y.

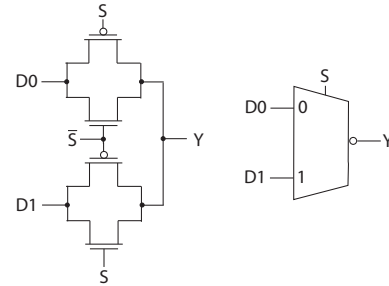


Figure 1.25: Transmisjonsport multiplexer med to innganger. (FIG1.27)

En enkel to-inngangs transmisjonsport multiplexer er vist i Fig. 5.66.

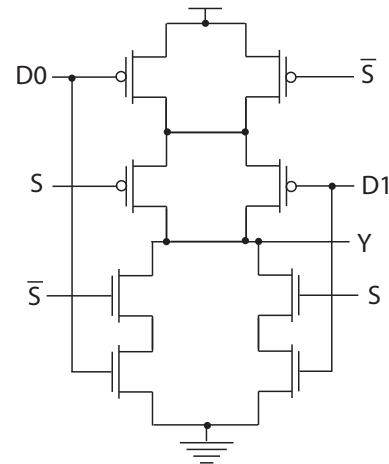


Figure 1.26: Inverterende to-inngangs multiplexer. (FIG1.28a)

En inverterende multiplexer med egenskaper til å gjenskape gode logiske verdier, dvs. styrking av signaler, kan lages ved å ta utgangspunkt i kretsen vist i Fig. 1.18 som implementerer funksjonen $Y = (A \cdot B) + (C \cdot D)$ som en

komplementær CMOS port. Dersom vi velger inngangene $D0 = C$, $D1 = B$, $S = A$ og $\bar{S} = D$ vil vi få funksjonen $Y = (S \cdot D1) + (D0 \cdot \bar{S})$ og kretsen vist i Fig. 1.26. Vi ser at dersom $S = 1$ får vi $Y = \bar{D1}$ og når $S = 0$ får vi $Y = D0$ som jo er den inverterende multiplekserens funksjon.

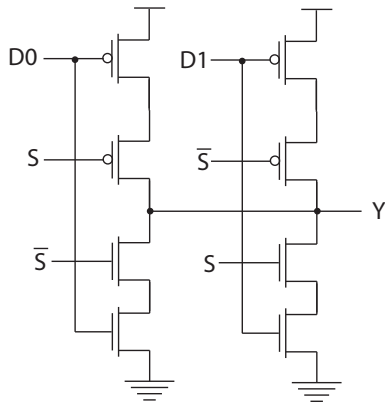


Figure 1.27: Inverterende to-inngangs multiplekser. (FIG1.28b)

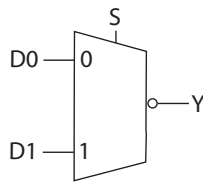


Figure 1.28: Inverterende to-inngangs multiplekser. (FIG1.28c)

En litt enklere og likeverdig implementasjon er vist i Fig. 1.27 og symbolet for den inverterende multiplekseren er vist i Fig. 1.28. Her ser vi tydelig at utgangen drives av to tristate kretser.

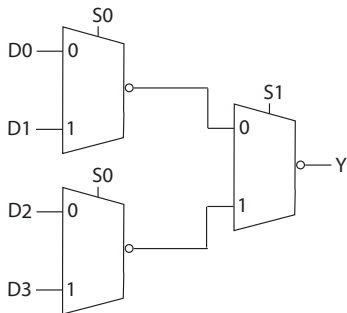


Figure 1.29: 4:1 multiplekser. (FIG1.29a)

En utvidelse til fire-inngangs multiplekser (4:1) er vist i Fig. 1.29.

Et eksempel på en 4:1 multiplekser bestående av invertere og tristate invertere er vist i Fig. 1.30.

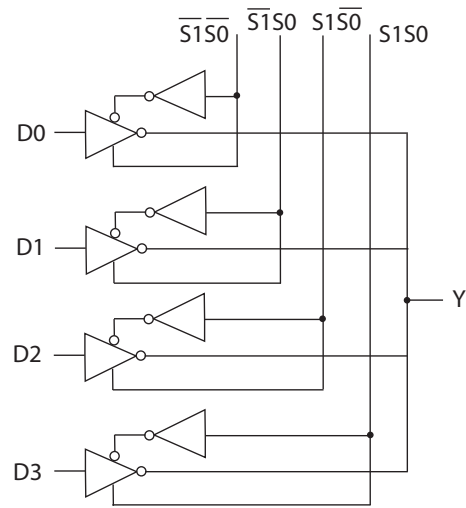


Figure 1.30: 4:1 multiplekser bestående av invertere og tristate invertere. (FIG1.29b)

1.10 Latcher

(Kapittel 1.4.9 side 20 - 21)

1.10.1 Latcher

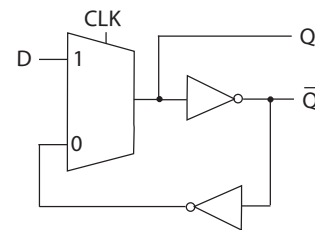


Figure 1.31: Positiv nivåfølsom latch bestående av en 2:1 multiplekser og invertere. (FIG1.30a)

Vi kan utnytte en multiplekser og to invertere til å lage en D latch som vist i Fig. 1.31. En implementasjon er vist i Fig. 1.32. En positiv nivåfølsom D latch vil være transparent, dvs. utgangen påvirkes direkte av inngangen, når $CLK = 1$, dvs. når $CLK = 1$ vil utgangen Q være lik inngangen D med en liten tidsforsinkelse. Når CLK er 0 vil inngangen D ikke kunne påvirke utgangen Q , men kretsen vil nå bestå av to invertere som er tilbakekoblet og sørge for at Q holdes konstant. Dette er vist i Fig. 1.33.

Symbolet for en positiv nivåfølsom D latch er vist i Fig. 1.34.

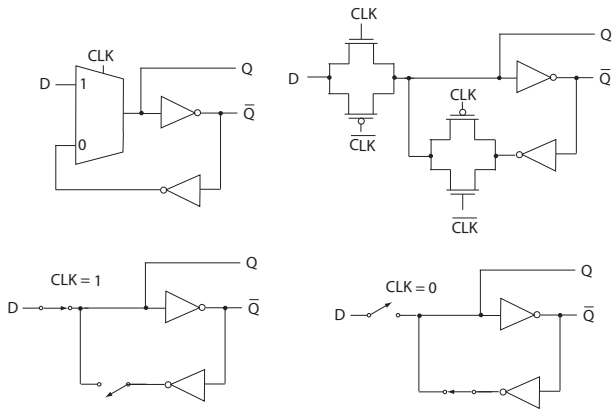


Figure 1.32: Implementasjon av positiv nivåfølsom latch. (FIG1.30a, b, c og d)

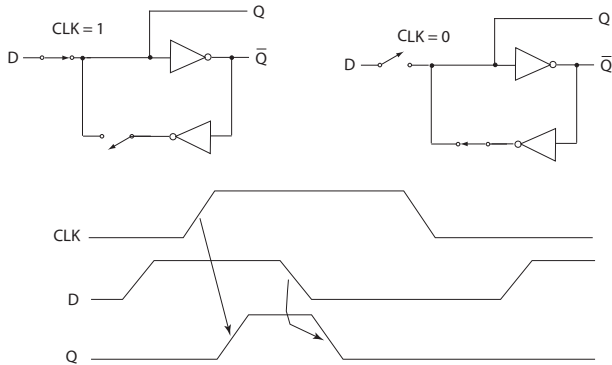


Figure 1.33: Virkemåten til positiv nivåfølsom D latch. (FIG1.30c, d, og e)

1.11 Vipper

(Kapittel 1.4.9 side 22 - 23)

En vippe består av to latches som klokkes i motsatt klokkefase som vist i Fig. 1.35. En implementasjon av positiv flankefølsom D vippe er vist i Fig. 1.36.

Inngangen D latches i den første latchen når $CLK = 0$ og utgangen på den første latchen \overline{QM} vil da følge D. Den andre latchen vil ikke transportere signalet videre fordi den er i tilbakekoblingsmodus når $CLK = 0$. Q er dermed upåvirket av inngangen D når $CLK = 0$. Når klokkesignalet svitsjer fra 0 til 1 vil den første latchen stenge og den andre latchen åpne. Utgangen på vippens Q vil da bli lik den siste verdien for \overline{QM} som er lik den siste verdien for D når CLK var 0. Q vil holde denne verdien inntil en eventuell endring inntreffer ved neste positive flanke på klokkesignalet CLK. Vippens virkemåte er vist i Fig. 1.37.

Et mulig problem med klokkingen som er vist i Fig. 1.37 er at akkurat når klokkesignaler svitsjer fra 0 til 1 vil begge latchene være delvis åpne i en kort periode og da vil vippens være nesten *transparent* slik at utgangen Q vil bli lik D direkte og ikke via latchet \overline{QM} . Dette vil være tydelig dersom det er synkroniseringsproblemer, for eksempel dersom \overline{CLK} skifter fra 0 til 1 før CLK skifter fra 1 til 0. Dette problemet er illustrert i Fig. 1.38. Problemet vil få effekt i etterfølgende porter som ikke forventer å få en

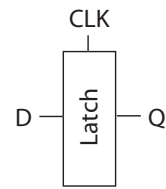


Figure 1.34: Symbol for positiv nivåfølsom D latch. (FIG1.30f)

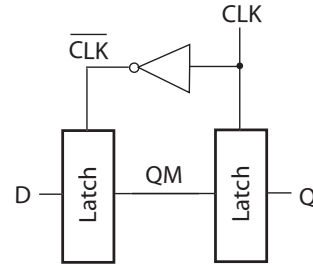


Figure 1.35: Positiv flankefølsom D vippe. (FIG1.31a)

inngangsendring i den tidsonen der begge klokkesignalene er 0 (rødt område).

En vanlig metode for å sikre seg mot problemer med usynkroniserte klokkesignaler er å benytte tofase ikkeoverlappende klokker som vist i Fig. 1.39. Her er det viktig at ϕ_1 og ϕ_2 ikke er logisk 1 samtidig.

Symbol for positiv flankefølsom D vippe er vist i Fig. 1.40.

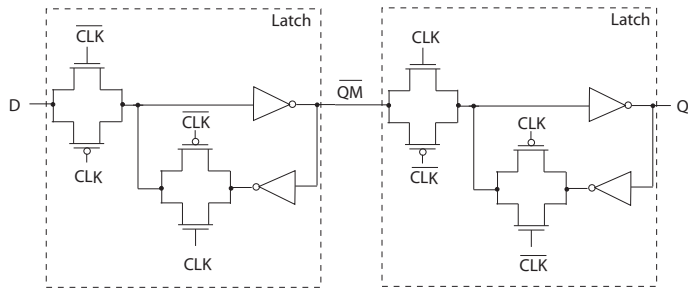
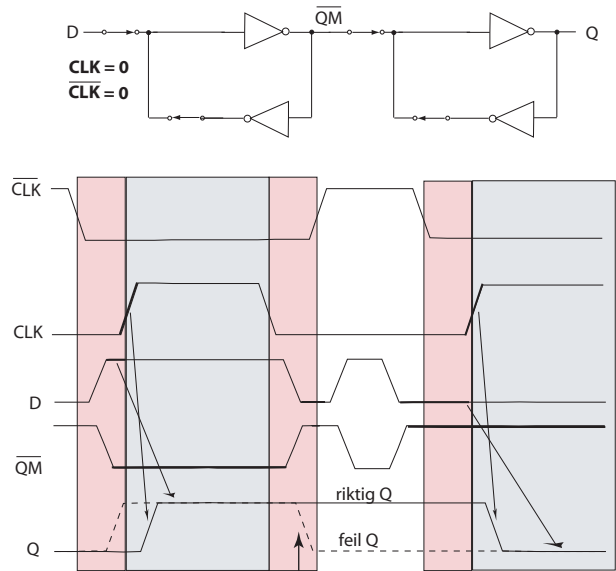


Figure 1.36: Implementasjon av positiv flankefølsom D vippe. (FIG1.31b)



Her vil vippen være transparent slik at Q ikke holder sin verdi som skulle være 1.

Figure 1.38: Potensielt problem med usynkronne klokkesignaler. Stiplet linje viser virkelig signalverdi for Q.

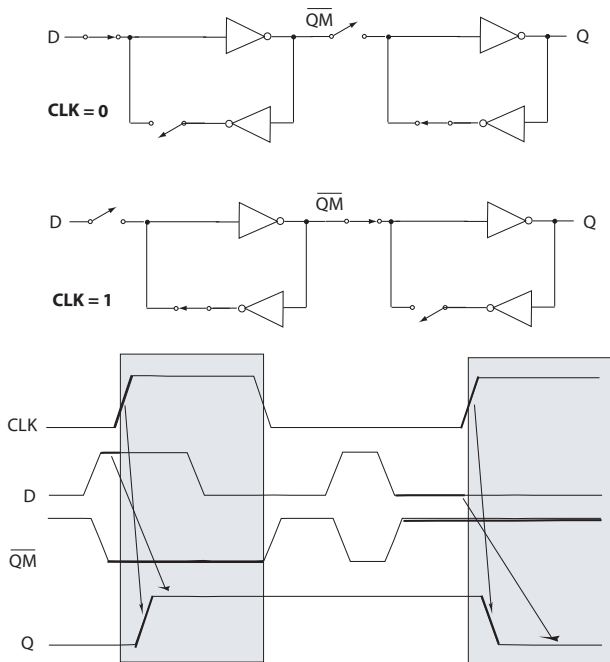


Figure 1.37: Implementasjon av positiv flankefølsom D vippe. (FIG1.31c, d og e)

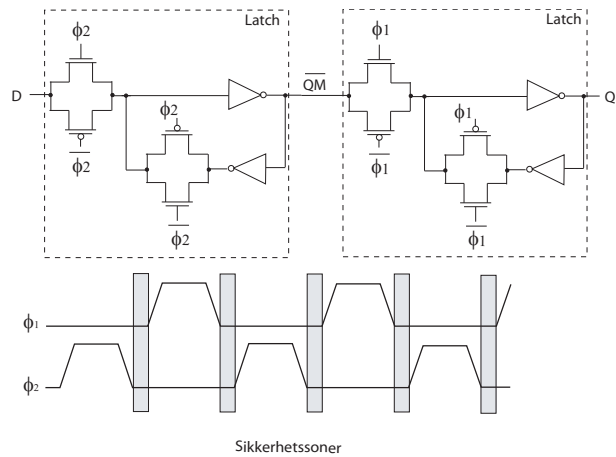


Figure 1.39: Dvippe med tofase ikkeoverlappende klokker. (FIG1.32)

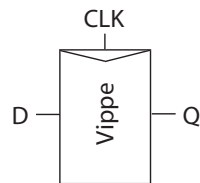


Figure 1.40: Symbol for positiv kantfølsom D vippe. (FIG1.31f)

Chapter 2

MOS transistor teori

2.1 MOS transistor i tverrsnitt

(Kapittel 1.3 side 7 - 9)

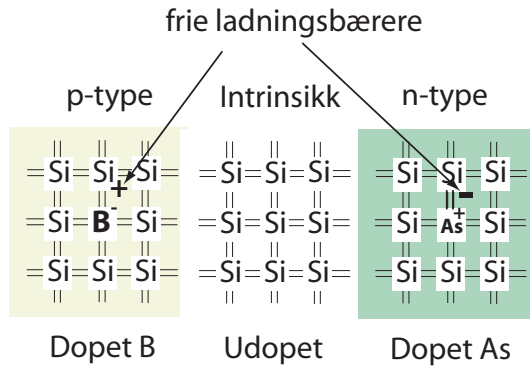


Figure 2.1: *Dopet silisium.*(FIG1.6)

Det grunnleggende materialet for CMOS halvleder er silisium (S_i). Udopet, eller intrinsikk, silisium vil ikke ha frie ladningsbærere. Ladningsbærere er negative elektroner og positive hull. Et *hull* kan betraktes som mangel på et elektron. Dersom vi forurensrer, eller doper, silisium kan vi få et halvledermateriale, fortsatt silisium, som vil ha frie ladningsbærere som vist i Fig. 2.1.

Eksempler på doping:

1. **Arsen** (A_s). Halvlederen vil ha frie elektroner og kalles derfor n-type (n står for negativ).
2. **Bor** (B). Halvlederen vil ha frie hull og kalles derfor p-type (p står for positiv).

2.1.1 pn overgang

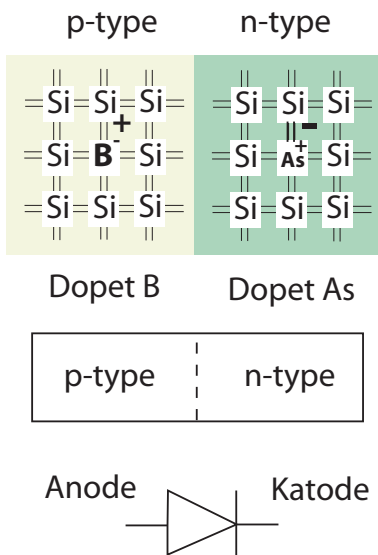


Figure 2.2: *pn overgang (diode).*(FIG1.7)

En pn-overgang vil oppstå der vi har p-type- og n-type halvledere inntil hverandre som vist i Fig. 2.2.

2.1.2 Tverrsnitt av MOS transistorer

Integrerte transistorer i CMOS teknologi kalles MOS-FET, som står for **Metal On Semiconductor Field Effect Transistor**. I moderne CMOS prosesser er det alltid polysilisium istedet for metall som former gaten.

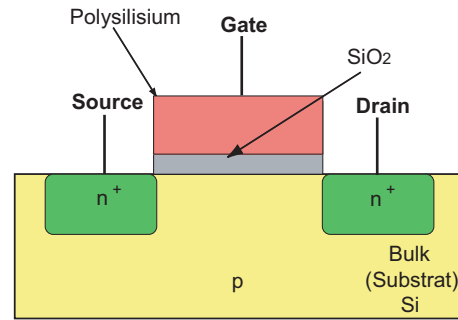


Figure 2.3: *Tverrsnitt av nMOS transistor.*(FIG1.8a)

Tverrsnitt av en nMOS transistor er vist i Fig. 2.3. Vi kaller transistoren nMOS fordi source og drain terminalene er koblet til n-type silisium. Disse områdene kalles for *diffusjon* og er av n^+ type, dvs. kraftig dopet med et stort antall frie elektroner. Diffusjonsområdene ligger i en svakt dopet silisium halvleder som kalles *substrat*. Mellom gaten og p-substrat er det et isolerende sjikt (SiO_2) som separerer gaten fra substrat slik at det ikke skal gå strøm fra gate til substrat.

Det vil bli dannet pn-overganger mellom n-type- og p-type silisium, dvs mellom source/drain og bulk (substrat). Diodene som vil oppstå her skal være revers forspent slik at det ikke vil gå strøm fra bulk til source/drain.

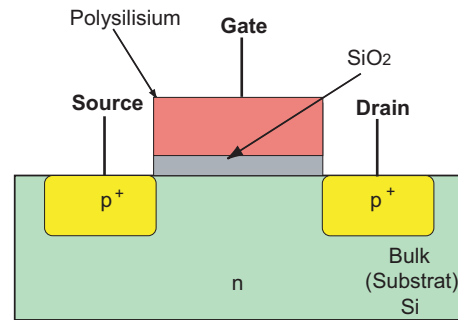


Figure 2.4: *Tverrsnitt av pMOS transistor.*(FIG1.8b)

Tverrsnitt av en pMOS transistor er vist i Fig. 2.4. Vi kaller transistoren pMOS fordi source og drain terminalene er koblet til p-type silisium. Disse områdene kalles for *diffusjon* og er av p^+ type, dvs. kraftig dopet med et stort antall frie hull.

Det vil bli dannet pn-overganger mellom n-type- og p-type silisium, dvs mellom bulk og source/drain. Diodene som vil oppstå her skal være revers forspent slik at det ikke vil gå strøm fra source/drain til bulk.

2.2 Tverrsnitt av CMOS Inverter

(Kapittel 1.5.1 side 23 - 24)

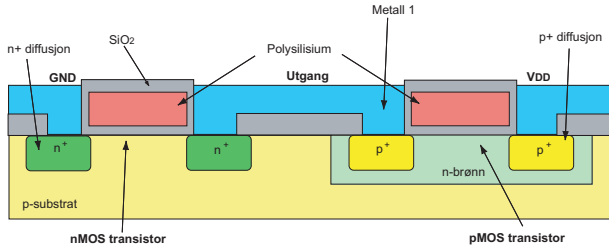


Figure 2.5: Tverrsnitt av CMOS inverter. (FIG1.33)

Dersom vi setter sammen en nMOS og en pMOS transistor og kobler sammen gate terminalene og drain terminalene på de to transistorene, og kobler source på nMOS transistoren til GND og source på pMOS transistoren til V_{DD} får vi en inverter. Tverrsnittet av en inverter er vist i Fig. 2.5.

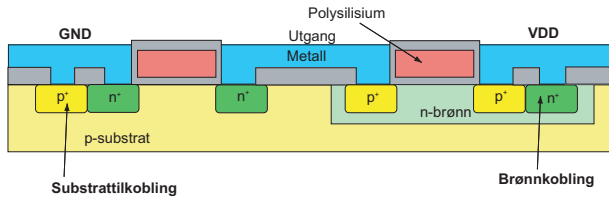


Figure 2.6: Tverrsnitt av CMOS inverter med substrat- og brønntilkoblinger. (FIG1.34)

Inverteren som er vist i figuren vil ikke kunne fungere korrekt fordi den mangler riktig tilkobling til substrat og n-brønn (n-substrat). I CMOS skal p-substrat kobles til GND for at nMOS transistoren skal fungere riktig. Likeledes må n-brønnen kobles til V_{DD} for at pMOS transistoren skal fungere riktig. Tverrsnitt av inverter med substrat- og brønntilkoblinger er vist i Fig. 3.3.

2.3 Akkumulasjon, deplesjon og inversjon

(Kapittel 2.1 side 67 - 68)

p-type silisium vil som nevnt ha frie ladningsbærere i form av hull. Bildet er noe mer nyansert, i p-type silisium vil det være et overskudd av frie ladningsbærere av form av hull som kalles majoritetsladningsbærere eller majoritetsbærere. I tillegg vil det være noen få frie ladningsbærere i form av elektroner. Vi kaller disse ladningsbærerne for minoritetsladningsbærere eller minoritetsbærere. I n-type silisium vil elektroner gjøre majoritetsladningsbærere og hull minoritetsladningsbærere.

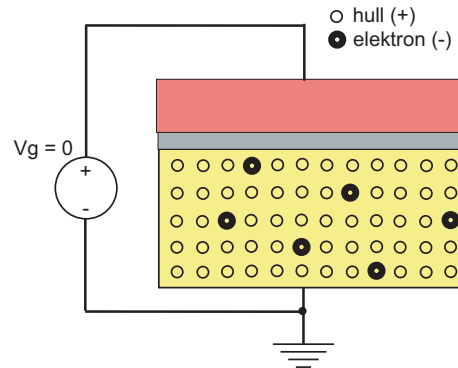


Figure 2.7: Ubiasert MOS struktur i p-substrat.

En ubiasert MOS struktur i p-substrat er vist i Fig. 2.7. Som figuren viser er det overskudd av frie hull i p-substratet og et lite antall frie elektroner.

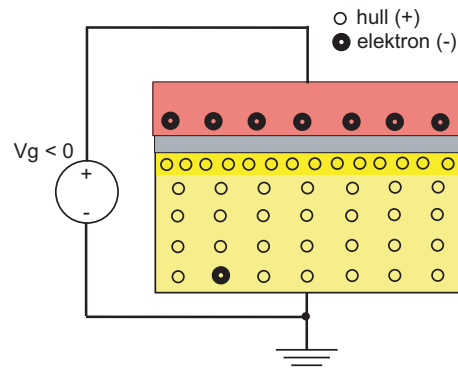


Figure 2.8: Negativt biasert MOS struktur i p-substrat. (FIG 2.2a)

Dersom vi biaserer slik at $V_g < 0$ (negativ biasering) vil vi tilføre negative ladning, dvs. elektroner, til gaten. Denne negative ladingen vil tiltrekke positive ladningsbærere, dvs. hull, fra substratet til overflaten av substratet rett under gaten. Halvleder materialet rett under gaten vil da bli sterkere p-type, dvs. større konsentrasjon eller overskudd på positive majoritetsladningsbærere. Dette kalles *akkumulasjon* og er vist i Fig. 2.8.

Hvis vi derimot biaserer slik at $0 < V_g < V_t$ (svak positiv biasering) vil vi tilføre positive ladning, dvs. hull, til

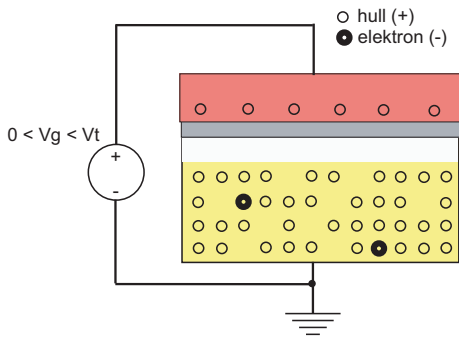


Figure 2.9: Svakt positivt biasert MOS struktur i p-substrat. (FIG 2.2b)

gaten. Den positive ladningen vil tiltrekke negative ladningsbærere fra substratet til toppen av substratet rett under gaten som vist i Fig. 2.9. Vi får da et sjikt rett under gaten som vil ha likt antall positive og negative ladningsbærere. Et slikt materiale tilsvarer intrinsikk, eller udopet, silisium og markeres som et hvitt område i figuren. Dette kalles *depleksjon*. I depleksjonssonen er det ikke en overvekt av en type frie ladningsbærere, dette er det samme som at det ikke er frie ladningsbærere i denne sonen. V_t kalles transistorens *terskelspenning*.

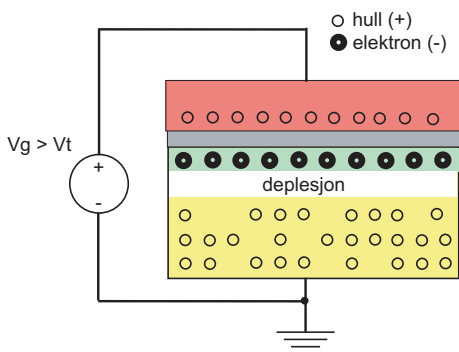


Figure 2.10: Sterkt positivt biasert MOS struktur i p-substrat. (FIG 2.2c)

Dersom vi øker den positive biaseringen slik at $V_g > V_t$ (sterk positiv biasering) vil vi tilføre en enda større positiv ladning til gaten som vil trekke enda flere elektroner til rett under gaten. Vi får da et sjikt med overskudd av elektroner som er markert som et grønt område i Fig.2.10. Dette sjiktet kalles *inversjon*, og mellom inversjon og resten av substratet vil det dannes en depleksjonssone som i praksis ikke vil inneholde frie ladningsbærere. Vi kaller det inverterte området under gaten for *kanal*.

2.4 Enkel beskrivelse av MOS transistor

(Kapittel 2.1 side 68 - 71)

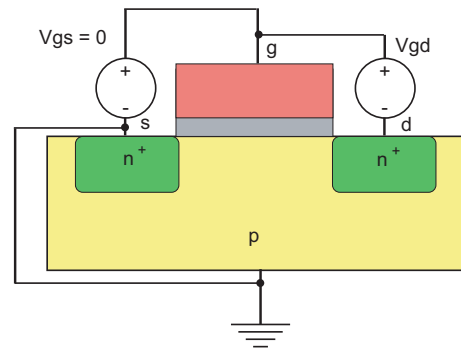


Figure 2.11: Ubiasert nMOS transistor. Transistoren vil være slått av fordi gate til source spenningen $V_{gs} \equiv V_g - V_s = 0$. Det vil ikke gå strøm mellom drain og source. $I_{ds} = 0$. (FIG 2.3a)

En ubiasert nMOS transistor er vist i Fig. 2.11. Transistoren vil være slått av fordi gate til source spenningen $V_{gs} \equiv V_g - V_s = 0$. Det vil ikke gå strøm mellom drain og source. $I_{ds} = 0$.

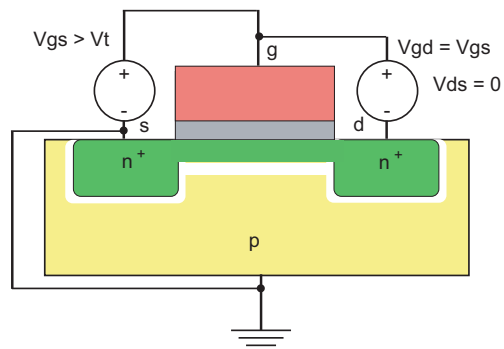


Figure 2.12: Biasert nMOS transistor. Transistoren vil være slått på fordi gate til source spenningen $V_{gs} \equiv V_g - V_s > V_t$. Det vil ikke gå strøm mellom drain og source fordi det ikke er spenningsforskjell mellom drain og source ($V_{ds} \equiv V_d - V_s = 0$). $I_{ds} = 0$. (FIG 2.3b)

En biasert nMOS transistor er vist i Fig. 2.12. Transistoren vil være slått på fordi gate til source spenningen $V_{gs} \equiv V_g - V_s > V_t$. Vi ser at det inverterte sjiktet strekker seg helt fra source til drain, og dermed sier vi at transistoren opererer i det *lineære området*. En forutsetning for å danne kanal på source siden av transistoren er at

$$V_{gs} > V_t. \quad (2.1)$$

Forutsetningen for at det skal dannes kanal på drain siden, dvs. at kanalen strekker seg fra source til drain og at transistoren dermed opererer i det lineære området er at

$$\begin{aligned}
V_{gd} &> V_t \\
V_g - V_d &> V_t \\
V_g - V_d - V_s &> V_t - V_s \\
V_g - V_s - (V_d - V_s) &> V_t \\
V_{gs} - V_{ds} &> V_t \\
V_{ds} &< V_{gs} - V_t.
\end{aligned} \tag{2.2}$$

Det vil ikke gå strøm mellom drain og source fordi det ikke er spenningsforskjell mellom drain og source ($V_{ds} \equiv V_d - V_s = 0$), $I_{ds} = 0$. Når $V_{ds} = 0$ vil det ikke være et elektrisk felt mellom drain og source slik at det ikke er elektrontransport fra source til drain. Det går da ingen strøm mellom drain og source.

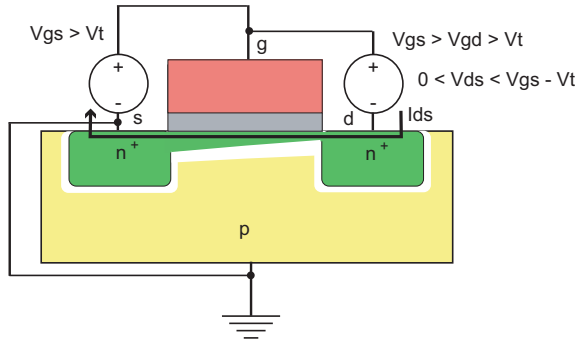


Figure 2.13: *Biasert nMOS transistor. Transistoren vil være slått på fordi gate til source spenningen $V_{gs} > V_t$. Det vil gå strøm mellom drain og source fordi det er spenningsforskjell mellom drain og source ($V_{ds} > 0$). $I_{ds} > 0$. (FIG 2.3c)*

En biasert NMOS transistor er vist i Fig. 2.13. Transistoren vil være slått på fordi gate til source spenningen $V_{gs} > V_t$. Vi ser at det inverterte sjiktet strekker seg helt fra source til drain, og dermed sier vi at transistoren opererer i det lineære området. I dette tilfellet har vi at $0 < V_{ds} < V_{gs} - V_t$.

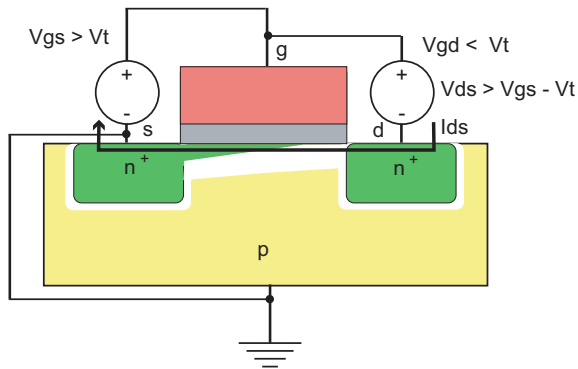


Figure 2.14: *Biasert nMOS transistor. Transistoren vil være slått på fordi gate til source spenningen $V_{gs} > V_t$. Det vil gå strøm mellom drain og source fordi det er spenningsforskjell mellom drain og source ($V_{ds} > V_{gs} - V_t$). $I_{ds} > 0$. (FIG 2.3d)*

Den biaserte transistoren som er vist i Fig. 2.14 har

så stor drain til source spenning V_{ds} at forutsetningen for at det skal dannes kanal på drain siden ikke er oppfylt. Det vil derfor ikke være kanal på drain siden, men det vil likevel gå en strøm mellom drain og source fordi det elektriske feltet mellom drain og source er sterkt. Vi sier at transistoren opererer i *metning* og $I_{ds} > 0$.

2.5 Enkel MOS transistor modell

(Kapittel 2.2 side 71 - 75)

Målet er å forstå enkle modeller for strøm i en MOS transistor. Vi skal modellere transistoren i områdene *av* (cut-off), *lineært område* og *metning*. Det forutsettes at transistorene er lengre enn $1\mu\text{m}$. For kortere transistorer må modellene utvides.

2.5.1 Utvikling av enkel strømmodell for nMOS transistor

Som nevnt kan vi se på transistoren som bryter, der transistoren er *AV* eller *PÅ* avhengig av gate til source spenning. Når transistoren er *AV* vil det ikke kunne gå strøm, mellom drain og source fordi det ikke er etablert en *kanal* ved *inversjon* mellom drain og source rett under gaten. En transistor som er *PÅ* vil kunne operere i to forskjellige modi avhengig av terminalspenninger V_d og V_s , eller mer konkret spenningsforskjellen mellom disse terminalene V_{ds} .

Transistorens operasjonsområder kan beskrives som

1. **AV**, eller cutoff. $V_{gs} < V_t$, som betyr at gate source spenningen ikke er tilstrekkelig til at det blir dannet en kanal. Det vil ikke gå strøm mellom drain og source, $I_{ds} = 0$.
2. **PÅ, lineær**. $V_{gs} > V_t$ og $0 < V_{ds} < V_{gs} - V_t$, som betyr at det er dannet kanal og at kanalen strekker seg helt fra source til drain. Transistoren er *lineær* i dette operasjonsområdet, dette indikerer at strømmen kan modelleres som en lineær funksjon av V_{ds} og V_{gs} .
3. **PÅ, metning**. $V_{gs} > V_t$ og $V_{ds} > V_{gs} - V_t$, som betyr at det er dannet kanal, men at det ikke er tilstrekkelig gate drain spenning V_{gd} til å danne kanal på drain siden av transistoren, som betyr at kanalen ikke strekker seg helt fra source til drain. Transistoren er i *metning* i dette operasjonsområdet, som indikerer at strømmen har gått i metning og ikke vil påvirkes av endring i V_{ds} . Transistorstrømmen vil i større grad påvirkes av V_{gs} enn for lineær operasjon fordi en økning i gate spenning og dermed V_{gs} og V_{gd} vil påvirke kanalen både på source og drain siden.

Førsteordens modell

En såkalt førsteordens modell for transistoren kalles *ideel Shockley* modell. Vi skal utvikle enkle modeller for transistorstrøm I_{ds} som funksjon av transistorens terminalspenninger V_{gs} og V_{ds} , en slik modell kan brukes til å lage en såkalt *I-V* karakteristikk eller transistorkarakteristikk.

Som vist i Fig. 2.15 er det et isolerende lag av silisiumdioksid SiO_2 mellom gate og substrat eller kanal. Dette isolerende laget skal forhindre at det går strøm mellom gate og kanal og vil representere en kapasitans C_g . Lading på hver side av kondensatoren (kapasitans) er gitt av

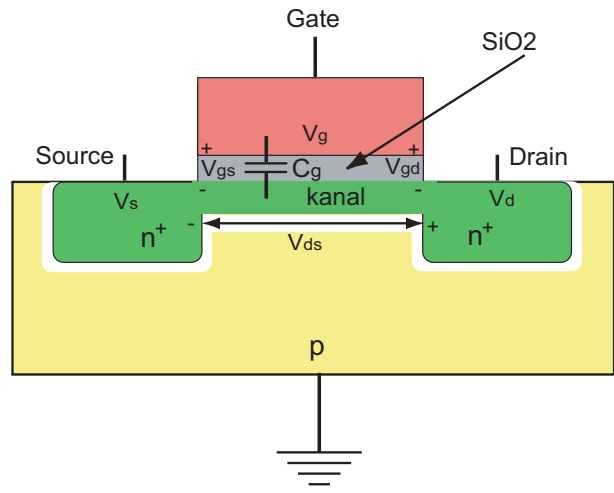


Figure 2.15: Tverrsnitt av nMOS transistor med gate kapasitans C_g . (FIG2.5)

$$Q = CV, \quad (2.3)$$

der C er kapasitansen og V er spenningen over kapasitansen.

Lineært område

Dersom det er etablert en kanal mellom source og drain ($V_{gs} > V_t$) kan vi beregne gjennomsnittelig spenning V_{gc} over gate kapasitansen C_g

$$\begin{aligned} V_{gc} &= V_g - \frac{(V_d - V_s)}{2} - V_s \\ &= V_{gs} - \frac{V_{ds}}{2} \end{aligned} \quad (2.4)$$

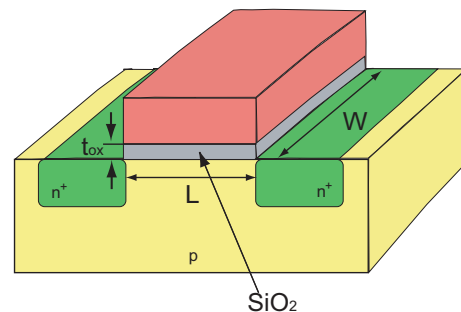


Figure 2.16: Tverrsnitt av nMOS transistor med transistorstørrelse bredde W og lengde L . (FIG1.26)

Gate kapasitansen C_g er avhengig av gate arealet, dvs. arealet av transistorens kanal, tykkelsen på det isolerende laget, tynnsid t_{ox} , og permitiviteten til det isolerende laget (SiO_2) ϵ_{ox} :

$$\begin{aligned} C_g &= \epsilon_{ox} \frac{WL}{t_{ox}} \\ &= C_{ox}WL, \end{aligned} \quad (2.5)$$

der W og L er bredde og lengde på transistoren som vist i Fig. 2.16 og $C_{ox} \equiv \epsilon_{ox}/t_{ox}$ er oskidkapsitans. Silisumdioksid har en permittivitet $\epsilon_{ox} = 3.9\epsilon_0$ der $\epsilon_0 = 8.85 \cdot 10^{-14} F/cm$ er permittivitet i vakum.

Det vil oppstå et elektrisk felt mellom drain og source der feltstyrken er avhengig av spenningeforskjellen mellom drain og source (V_{ds}). Ladningsbærere i kanalen vil ha en gjennomsnittelig hastighet som er proporsjonal med det elektriske feltet:

$$\nu = \mu E, \quad (2.6)$$

der μ er mobiliteten til ladningsbærere.

Det elektriske feltet er som tidligere nevnt avhengig av spenningen over feltet V_{ds} og transistorens lengde, dvs. avstanden mellom drain og source:

$$E = \frac{V_{ds}}{L}. \quad (2.7)$$

Tiden det tar for en ladningsbærer å krysse kanalen er gitt av kanalens lengde og ladningsbærernes hastighet:

$$\tau = \frac{L}{\nu}. \quad (2.8)$$

Vi har nå at strømmen mellom drain og source kan uttrykkes som den totale mengden ladning i kanalen dividert på tiden som behøves for å krysse kanalen:

$$\begin{aligned} I_{ds} &= \frac{Q_{kanal}}{\frac{L}{\nu}} \\ &= \mu C_{ox} \frac{W}{L} \left(V_{gs} - V_t - \frac{V_{ds}}{2} \right) V_{ds} \\ &= \beta \left(V_{gs} - V_t - \frac{V_{ds}}{2} \right) V_{ds}, \end{aligned} \quad (2.9)$$

der

$$\beta = \mu C_{ox} \frac{W}{L}. \quad (2.10)$$

Ligning 2.10 gjelder for en nMOS transistor i det lineære området¹. Vi ser av modellen at strømmen er lineært

¹Mange lærebøker operere med en litt anderledes modell $I_{ds} = \beta_n (V_{gs} - V_t - V_{ds}^2/2) V_{ds}$.

avhengig av V_{ds} . Dette tilsvarer en motstand og vi kaller det lineære området også for det resistive (eller triode) området.

En enkel *elektrisk eivalent* for MOS transistoren i det lineære eller resistive området er en motstand mellom drain og source, der motstandsverdien bestemmes av gate source spenningen:

$$I_{ds} = \frac{V_{ds}}{R_{gs}}, \quad (2.11)$$

der

$$R_{gs} = \left(\beta \left(V_{gs} - V_t - \frac{V_{ds}}{2} \right) \right)^{-1}. \quad (2.12)$$

Det er vanlig å forenkle kanal motstanden i nMOS- og pMOS transistorer til modellene

$$R_n = (\beta_n (V_{gsn} - V_{tn}))^{-1} \quad (2.13)$$

$$R_p = (\beta_p (V_{sgp} - |V_{tp}|))^{-1} \quad (2.14)$$

Metning

Dersom drain source spenningen blir tilstrekkelig høy, dvs. overstiger $V_{dsat} = V_{gs} - V_t$, vil transistoren være i metning og kanalen vil ikke strekke seg helt til drain. Vi erstatter V_{ds} med V_{dsat} og får:

$$\begin{aligned} V_{gc} &= V_{gs} - \frac{V_{dsat}}{2} \\ &= V_{gs} - \left(\frac{V_{gs} - V_t}{2} \right) \\ &= \frac{V_{gs} + V_t}{2}. \end{aligned} \quad (2.15)$$

Vi setter inn $(V_{gs} + V_t)/2$ for $V_{gs} - V_{ds}/2$ og $V_{ds} = V_{gs} - V_t$ og får modell for strøm for en transistor i metning:

$$I_{ds} = \frac{\beta}{2}(V_{gs} - V_t)^2. \quad (2.16)$$

Dersom vi setter inn $V_{ds} = V_{gs} - V_t$ i modellene for lineært område og metning bør vi få lik strøm:

$$\begin{aligned} \beta_n \left(V_{gs} - V_t - \frac{V_{ds}}{2} \right) V_{ds} &= \frac{\beta}{2}(V_{gs} - V_t)^2 \\ \left(V_{gs} - V_t - \frac{(V_{gs} - V_t)}{2} \right) (V_{gs} - V_t) &= \frac{(V_{gs} - V_t)^2}{2} \\ \frac{(V_{gs} - V_t)^2}{2} &= \frac{(V_{gs} - V_t)^2}{2}, \end{aligned} \quad (2.17)$$

som viser at modellen er kontinuerlig ved metningspunktet.

Vi kan oppsummere førsteordene modell for nMOS transistoren:

AV

$$I_{ds} = 0, \quad V_{gs} < V_t$$

LINEÆR

$$I_{ds} = \beta \left(V_{gs} - V_t - \frac{V_{ds}}{2} \right) V_{ds}, \quad V_{gs} > V_t, \quad V_{ds} < V_{dsat}$$

METNING

$$I_{ds} = \frac{\beta}{2}(V_{gs} - V_t)^2, \quad V_{gs} > V_t, \quad V_{ds} > V_{dsat}. \quad (2.18)$$

En enkel elektrisk modell eller ekvivalent for transistoren i metning er en strømkilde.

2.5.2 Strømmodell for pMOS transistor

Vi har definert positiv strømretning for nMOS transistoren fra drain til source, dvs. $I_{dsn} \equiv I_{ds}$. For pMOS transistoren blir positiv strømretning fra source til drain, dvs. $I_{sdp} \equiv -I_{ds}$. For pMOS transistoren vil strømmen øke når vi reduserer gatespenningen (i forhold til source, typisk V_{DD}). Vi kan da erstatte V_{gs} i modell for nMOS transistor med V_{sg} og V_{ds} med V_{sd} . Terskelspenningen på en pMOS transistor forholder seg til V_{gsp} (gate source spenning for pMOS transistor som jo alltid er 0 eller negativ). Vi erstatter derfor V_t i modellen med $|V_{tp}|$, dvs. vi ser på absoluttverdien for terskelspenningen for pMOS transistoren. Terskelspenningen for pMOS transistorer er negativ. Vi kan velge å vise I-V karakteristikk, dvs. strøm som funksjon av spenning for en pMOS transistor som I_{sdp} som funksjon av V_{sdp} , dette vil tilsvare $V_{DD} + V_{sdp}$.

Vi får da følgende modell for pMOS transistor strøm:
AV

$$I_{sdp} = 0, \quad V_{sgp} < |V_{tp}|$$

LINEÆR

$$I_{sdp} = \beta_p \left(V_{sgp} - |V_{tp}| - \frac{V_{sdp}}{2} \right) V_{sdp}, \quad V_{sgp} > |V_{tp}|, \quad V_{sdp} < V_{dsat}$$

METNING

$$I_{sdp} = \frac{\beta_p}{2}(V_{sgp} - |V_{tp}|)^2, \quad V_{sgp} > |V_{tp}|, \quad V_{sdp} > V_{dsat} \quad (2.19)$$

2.5.3 I-V karakteristikk

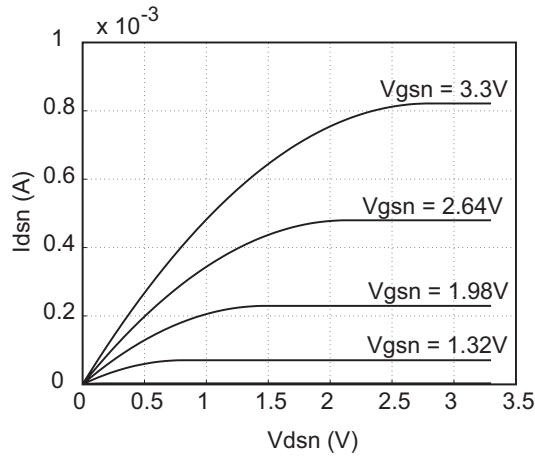


Figure 2.17: Strøm karakteristikk for nMOS transistor som funksjon av V_{dsn} .

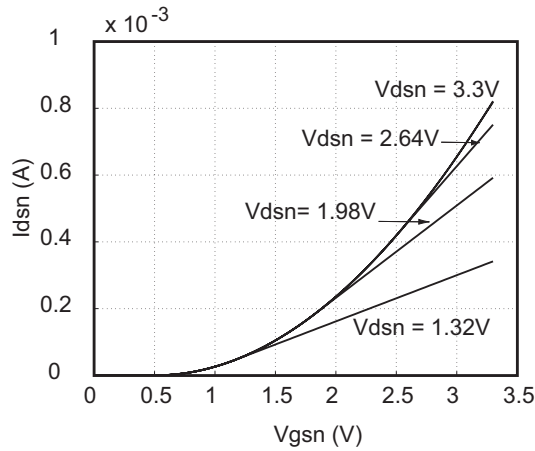


Figure 2.18: Strøm karakteristikk for nMOS transistor som funksjon av V_{gsn} .

I Fig. 2.17 og 2.18 er I-V karakteristikk for en nMOS transistor som funksjon av henholdsvis V_{dsn} og V_{gsn} vist.

I Fig. 2.19 og 2.20 er I-V karakteristikk for en pMOS transistor som funksjon av henholdsvis V_{sdp} og V_{sgp} vist.

2.6 Kanallengdemodulasjon eller kanalforkorting

(Kapittel 2.4.2 side 86 - 89)

Vi har beskrevet en transistor som en (ideell) strømkilde i metning, dvs. det er ingen endring i strømmen når vi øker drain source spenningen fra metningsspenningen ($V_{dsat} = V_{gsn} - V_{tn}$). Utgangsimpedansen (motstanden) blir da uendelig stor. Dette er en forenkling som ikke tar hensyn til at den *effektive kanallengden* blir redusert i metning. Som vist i Fig. 2.21 vil ikke kanalen strekke seg helt til drain siden. Effekten av dette er at den effektive kanallengden blir kortere og at strømmen øker i metning

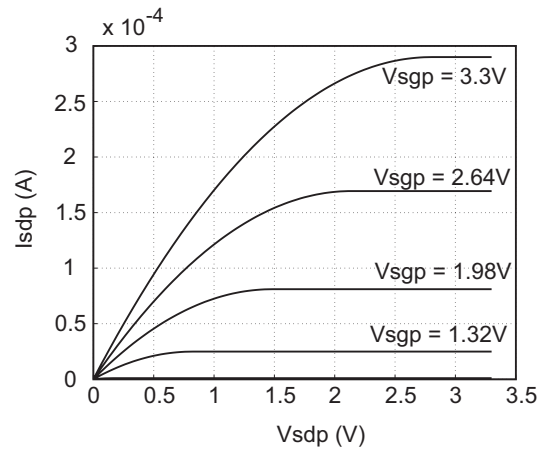


Figure 2.19: Strøm karakteristikk for pMOS transistor som funksjon av V_{sdp} .

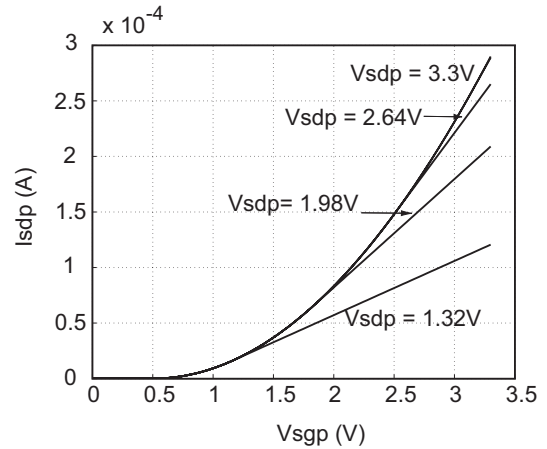


Figure 2.20: Strøm karakteristikk for pMOS transistor som funksjon av V_{sgp} .

når drain source spenningen øker. Vi kan modellere den effektive kanallengden som:

$$L_{eff} = L - L_d,$$

der L_d er avstanden fra drain til kanal og L nominell kanallengde. L_d vil øke når drain source spenningen øker. Vi kan modellere strømmen i metning, inkludert kanallengdemodulasjon:

$$I_{ds} = \frac{\beta}{2} (V_{gs} - V_t)^2 (1 + \lambda V_{ds}).$$

der λ er en empirisk verdi². For å få en kontinuerlig modell multipliseres kanalmodulasjonsleddet også med strømmen i lineært område slik at transistormodellen for nMOS transistor blir:

²I noen bøker beskrives kanalmodulasjonsfaktor λ analytisk.

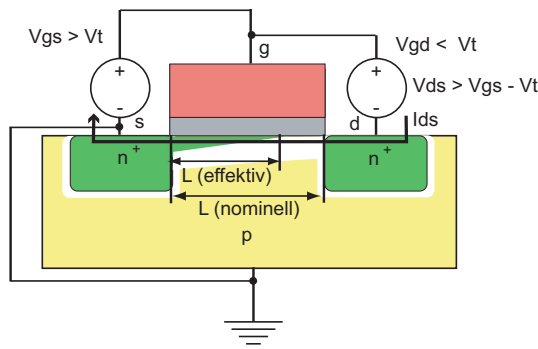


Figure 2.21: Tverrsnitt av transistor i metning. (FIG2.3d)

AV

$$I_{ds} = 0$$

LINEÆR

$$I_{ds} = \beta \left(V_{gs} - V_t - \frac{V_{ds}}{2} \right) V_{ds} (1 + \lambda_n V_{ds})$$

METNING

$$I_{ds} = \frac{\beta}{2} (V_{gs} - V_t)^2 (1 + \lambda_n V_{ds}) \quad (2.20)$$

Tilsvarende modeller for pMOS transistor med kanalforkortning er:

AV

$$I_{sdp} = 0$$

LINEÆR

$$I_{sdp} = \beta_p \left(V_{sgp} - |V_{tp}| - \frac{V_{sdp}}{2} \right) V_{sdp} (1 + \lambda_p V_{sdp})$$

METNING

$$I_{sdp} = \frac{\beta_p}{2} (V_{sgp} - |V_{tp}|)^2 (1 + \lambda_p V_{sdp}) \quad (2.21)$$

2.6.1 I-V karakteristikk for nMOS transistor med kanallengdemodulasjon

Fig. 2.22 viser transistorstrøm med og uten kanalmodulasjon.

2.6.2 Transferkarakteristikk for inverter med kanallengdemodulasjon

Fig. 2.23 viser DC karakteristikk for CMOS inverter med og uten modell for kanallengdemodulasjon. Som vi ser vil kanalmodulasjon redusere forsterkningen i forhold til en inverteren modellert med førsteordens transistormodeller.

2.6.3 Mål

Forstå og kunne modellere kanallengdemodulasjon.

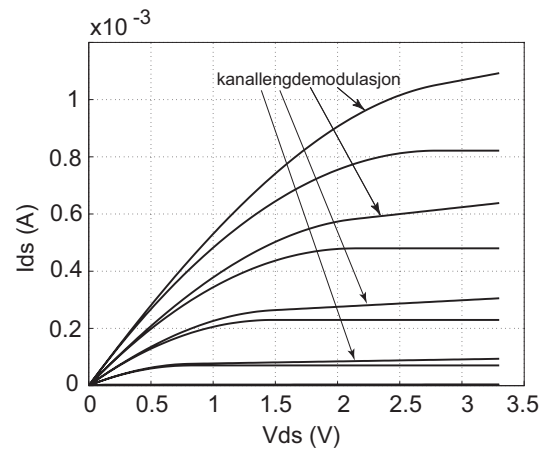


Figure 2.22: Transistorstrømm med og uten kanallengdemodulasjon. (FIG2.18)

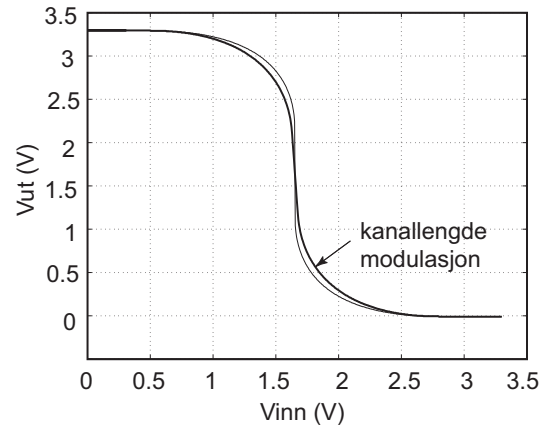


Figure 2.23: Inverter DC karakteristikk med og uten kanallengdemodulasjon.

2.7 Bodyeffekt

(Kapittel 2.4.3 side 87 - 88)

* Forutsetter en enkel forståelse av førsteordens transistor ligninger[?]. (INF3400 Del 2: Enkel MOS transistor modell, Kapittel 2.2 side 71 - 75).

Vi har til nå forutsatt at source og substrat (bulk) spenningene er like. Det kan selvsagt forekomme at source og substrat ikke har samme spenning, i mange tilfeller vil source på transistorene ikke være fysisk koblet til referansespenninger V_{DD} eller GND . I slike tilfeller vil source bulk spenning V_{sb} påvirke transistorstrømmen. En slik effekt modelleres som en endring i effektiv terskelspenning. For en nMOS transistor har vi at:

$$V_{tn} = V_{tn0} + \gamma \left(\sqrt{\Phi_s + V_{sb}} - \sqrt{\Phi_s} \right), \quad (2.22)$$

der V_{tn0} er nominell terskelspenning, dvs. terskelspenning når $V_{sb} = 0$, γ er *body effekt parameter* og *overflate potensialet* Φ_s er gitt av:

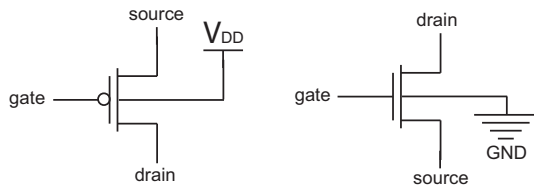


Figure 2.24: Transistorsymboler med substrat (body) terminaler. Substratet for nMOS transistor er normalt koblet til GND, og substratet for pMOS transistorer er normalt koblet til V_{DD} .

$$\Phi_s = 2V_T \ln \frac{N_A}{n_i}, \quad (2.23)$$

der N_A er *dopekonsentrasjon*, n_i er konsentrasjonen av frie ladningsbærere i udotet silisium, og *termisk spenning* er gitt av:

$$V_T = \frac{kT}{q}, \quad (2.24)$$

der k er bolzmanns konstant, T er temperatur i Kelvin og q er elementærladning. Termisk spenning er 0.026V ved romtemperatur. Bodyeffekt parameter kan uttrykkes som

$$\gamma = \frac{\sqrt{2q\epsilon_{Si}N_A}}{C_{ox}}, \quad (2.25)$$

der ϵ_{Si} er permitivitet for silisium og C_{ox} er oksidkapasitans.

2.7.1 Mål

Forstå hvordan bodyeffekt påvirker transistorer.

2.8 Svak inversjon

(Kapittel 2.4.4 side 88 - 89)

De enkle transistormodellene som vi har benyttet hittil modellerer strømmen gjennom transistoren lik 0A når gate source spenningen er lavere enn terskelspenningen. I virkeligheten er det ikke en skarp overgang fra ingen kanal til full kanal. Når det er etablert en kanal sier vi at transistoren opererer i *sterk inversjon* og de modellene som vi kjenner kan brukes. Når gate source spenningen er vesentlig mindre enn terskelspenningen er det vanlig å modellere transistorstrømmen som:

$$I_{ds} = I_{ds0} e^{\frac{V_{gs} - V_t}{nV_T}} \left(1 - e^{\frac{-V_{ds}}{V_T}} \right), \quad (2.26)$$

der n er *slope faktor* og

$$I_{ds0} = \beta V_T^2 e^{1.8}. \quad (2.27)$$

Som vi ser av modellen vil det være en eksponensiell økning i strømmen for økning i gate source spenning. Det er viktig å huske på at selv om strømmen stiger kraftig i dette området som kalles *svak inversjon*, så er strømmen svært liten³. I noen analoge kretser der det ikke er strenge krav til høy hastighet kan transistorer benyttes i svak inversjon. Liten strøm gir liten effekt som i noen analoge kretser er en stor fordel. Overgangen mellom svak og sterk inversjon kalles *moderat inversjon*. En annen ting som det er verdt å merke seg er at det vil gå en liten strøm gjennom transistoren selv om den er skrudd av.

Området rett under terskelspenningen, dvs. moderat inversjon eller *subterskel*, er mer aktuelt som følge av at korte transistorer med en høy drain source spenning får redusert effektiv terskelspenning. Denne effekten kalles *Drain-induced barrier lowering* (DIBL). DIBL modelleres som

$$V_t' = V_t - \eta V_{ds}, \quad (2.28)$$

der η er DIBL koeffisient med verdi i området 0.02 til 0.1.

2.8.1 Mål

Kunne modellere transistor i svak inversjon og kunne modellere drain-induced barrier lowering (DIBL).

³I svak inversjon er strømmen typisk i området fra pA til noen nA, selvom strømmen kan være vesentlig større dersom bredde/lengde forholdet er meget stort.

2.9 Lekkasje i pn-overganger

(Kapittel 2.4.5 side 89 - 90)

* Reversforspente dioder (pn-overganger). (FYS1210)

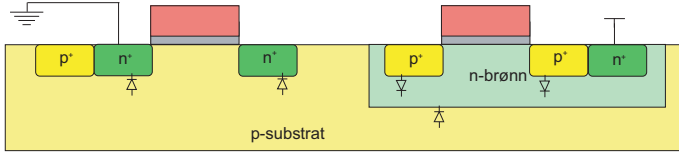


Figure 2.25: Reversforspente dioder i CMOS kretser. (FIG2.19)

I pn-overgangene mellom diffusjon og substrat eller brønn vil det oppstå reversforspente dioder som vist i Fig. 2.25. I tillegg er det dioder mellom brønn og substrat. Diodestrøm for reversforspent diode er gitt av:

$$I_D = I_S \left(e^{\frac{V_D}{V_T}} - 1 \right), \quad (2.29)$$

der I_S er avhengig av dopenivå, areal og omkrets på diffusjonsområdet, og diode spenning V_D som er gitt av V_{sb} eller V_{db} .

Når reversforspenningen er vesentlig høyere enn V_T (termisk spenning) vil reversstrømmen være lik I_S som typisk ligger i området 0.1 til $0.01 fA/\mu m^2$.

Historisk var lekkasje i pn-overganger den begrensende faktor for lagring av spenning i dynamiske noder. I moderne CMOS prosesser med lav terskelspenning er transistorstrømmer i svak inversjon langt viktigere som begrensende faktor.

2.9.1 Mål

Kunne modellere lekkasje i reversforspente pn-overganger (dioder).

2.10 Tunnelering

(Kapittel 2.4.6 side 90)

Det er en viss sannsynlighet for at ladningsbærere vil tunnelere gjennom gateoksid (tynnoksid). Denne sannsynligheten avtar eksponentielt med tykkelsen på gateoksidet (t_{ox}) og har inntil nylig vært neglisjerbar.

For prosesser med gateoksid tynnere enn 20Å vil tunneleringsstrøm bli en vesentlig faktor og sammenlignbar med transistorstrømmer i svak inversjon.

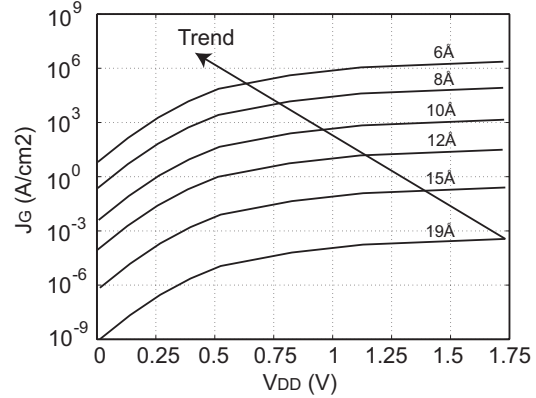


Figure 2.26: Lekkasje gjennom gateoksid. (FIG2.20)

Gate lekkasje eller strømtetthet J_G gjennom gateoksid er vist i Fig. 2.26 for forskjellige tykkelser på gateoksidet. Gatestrøm vil ikke bare påvirke dynamiske noder, men også øke effektforbruket. Hvert atomlag i silisiumdioksid er på ca. 3Å, og man antar derfor at den nedre grense for tykkelsen på gateoksid er ca. 8Å. Det jobbes med å utvikle en bedre isolator med høyere permittivitet.

Tunneleringsstrømmen er en orden høyere for nMOS transistorer enn for pMOS transistorer.

2.10.1 Mål

Etablere en oversikt over tunneleringsstrøm i moderne CMOS prosesser med tynt gateoksid.

2.11 Geometriavhengighet for MOS transistorer.

(Kapittel 2.4.8 side 92)

I design fasen av utlegg tegnes transistorene med bredde W_{tegned} og lengde L_{tegned} . De faktiske størrelsene ved produksjon kan variere på grunn av prosessvariasjoner med faktorene X_W og X_L for bredde og lengde. En av grunnene til at transistorenes faktiske lengde blir forskjellig fra tegnet eller nominelle lengde er *lateral diffusjon* L_D inn under gaten. Dette betyr at transistorene blir kortere enn nominell lengde. På tilsvarende måte vil diffusjon strekke seg inn under gaten og redusere transistorenes effektive bredde. Vi kan modellere transistorenes effektive lengde og bredde

$$\begin{aligned} L_{eff} &= L_{tegned} + X_L - 2L_D \\ W_{eff} &= W_{tegned} + X_W - 2W_D. \end{aligned}$$

For små transistorer kan effektiv lengde og bredde være betydelig forskjellig fra tegnede bredder og lengder. Transistorenes terskelspenning vil også variere med effektiv størrelse. Det er fornuftig å bruke identiske størrelser (tegnede) for transistorer som inngår i design med stor krav til presisjon. I moderne prosesser kan transistorlengde også variere med orientering på transistoren på en brikke.

2.12 Temperaturavhengighet for MOS transistorer.

(Kapittel 2.4.7 side 90 - 92)

Transistorkarakteristikker er påvirket av temperatur. Mobiliteten til ladningsbærere avtar når temperaturen stiger, dette kan modelleres som

$$\mu(T) = \mu(T_r) \left(\frac{T}{T_r} \right)^{-k_\mu},$$

der T er temperatur, T_r er romtemperatur ($300^\circ K$) og k_μ er en parameter i intervallet 1.2 – 2.0.

Terskelspenningen vil variere tilnærmet lineært med temperaturen

$$V_t(T) = V_t(T_r) - k_{vt}(T - T_r),$$

der k_{vt} typisk ligger i området 0.5 og 3.0 mV/K.

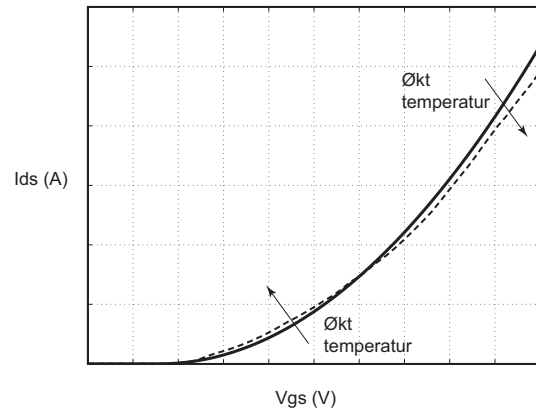


Figure 2.27: Transistor karakteristikk varierer med temperatur. (FIG2.21)

Effekten av endring i temperatur er vist i Fig.2.27, der maksimum (PÅ) strømmen reduseres med økt temperatur og minimum (AV) strømmen øker. Generelt vil ytelsen reduseres ved økt temperatur. Ytelsen kan økes ved å anvende kjøling.

2.13 CMOS inverter DC karakteristikk

(Kapittel 2.5.1 side 94 - 97)

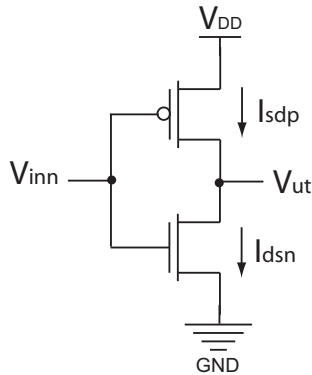


Figure 2.28: CMOS Inverter med positiv strømretning for pMOS- og nMOS transistor. Merk forskjellen fra læreboken ved at vi definerer positiv strømretning og kaller strømmen I_{sdp} istedet for I_{dsp} som vil være negativ. (FIG2.23)

En CMOS inverter er vist i Fig. 2.28, der positive strømretninger er vist. Merk forskjellen fra læreboken (FIG2.23) ved at vi definerer positiv strømretning for pMOS transistoren og kaller strømmen I_{sdp} istedet for I_{dsp} som vil være negativ.

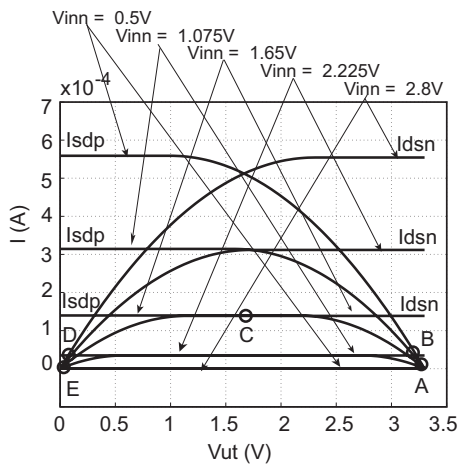


Figure 2.29: Transistor strømmen som funksjon av utgangsspenning for ulike inngangsspenninger.

Med utgangspunkt i førsteordens transistormodeller for pMOS- og nMOS transistorene skal vi finne inverters DC karakteristikk, dvs. utgangen som funksjon av inngangen eller V_{ut} som funksjon av V_{inn} . Vi lager et plott som viser strømmen gjennom transistorene som en funksjon av utgangen på inverteren for ulike inngangsspenninger. For nMOS transistoren tilsvarer dette I_{dsn} som funksjon av V_{dsn} fordi $V_{dsn} = V_{ut}$. For pMOS transistorene derimot må vi se på I_{sdp} som funksjon av $V_{ut} = V_{DD} - V_{sdp}$. Dette vil i praksis si at vi flytter kurven som vist i Fig. 2.29. Der hvor strømkurven for nMOS transistoren, gitt en spesifikk $V_{inn} = V_{gsn}$, krysser strømkurven for pMOS tran-

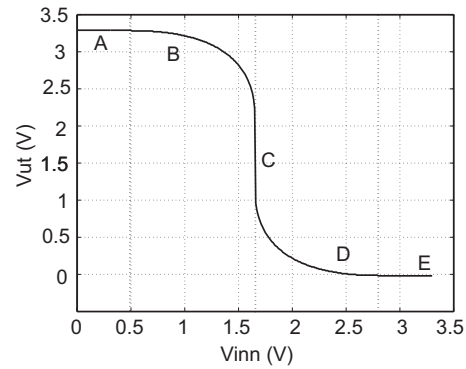


Figure 2.30: CMOS Inverter DC (V-V) karakteristikk

sistor med samme inngangsspenning $V_{inn} = V_{DD} - V_{sgp}$ får vi et DC punkt (V_{ut}, V_{inn}) . Med andre ord en gitt inngangsspenning vil gi ekvivalente strømmen i nMOS- og pMOS transistorene for en bestemt utgangsspenning. På denne måten kan vi finne transfer, eller V-V (DC) karakteristikk for inverteren som vist i Fig. 2.30.

For pMOS transistoren i inverteren har vi:

$$\begin{aligned} V_{sgp} &= V_{sp} - V_{gp} & (2.30) \\ &= V_{DD} - V_{inn} \\ V_{sdp} &= V_{sp} - V_{dp} \\ &= V_{DD} - V_{ut}. \end{aligned}$$

AV	LINEÆR	METNING
$V_{gsn} < V_{tn}$	$V_{gsn} > V_{tn}$	$V_{gsn} > V_{tn}$
$V_{inn} < V_{tn}$	$V_{inn} > V_{tn}$	$V_{inn} > V_{tn}$
	$V_{dsn} < V_{gsn} - V_{tn}$	$V_{dsn} > V_{gsn} - V_{tn}$
	$V_{ut} < V_{inn} - V_{tn}$	$V_{ut} > V_{inn} - V_{tn}$

Table 2.1: Operasjonsområder for nMOS transistor i en inverter (Tabell 2.2 side 94).

AV	LINEÆR	METNING
$V_{sgp} < V_{tp} $	$V_{sgp} > V_{tp} $	$V_{sgp} > V_{tp} $
$V_{inn} > V_{DD} + V_{tp}$	$V_{inn} < V_{DD} + V_{tp}$	$V_{inn} < V_{DD} + V_{tp}$
	$V_{sdp} < V_{dsat}$	$V_{sdp} > V_{dsat}$
	$V_{sdp} < V_{sgp} - V_{tp} $	$V_{sdp} > V_{sgp} - V_{tp} $
	$-V_{ut} < V_{tp} - V_{inn}$	$-V_{ut} > V_{tp} - V_{inn}$
	$V_{ut} > V_{inn} - V_{tp}$	$V_{ut} < V_{inn} - V_{tp}$

Table 2.2: Operasjonsområder for pMOS transistor i en inverter (Tabell 2.2 side 94).

Tabell I og II viser de tre operasjonsområdene for transistorene i en inverter. Vi kan anta at $V_{tp} \approx -V_{tn}$. Som vist i Fig. 2.30 får vi en inverter DC karakteristikk med forskjellige områder der forsterkningen varierer. Inverters forsterkning er definert i området C.

2.13.1 Område A, se Fig. 2.30

Inngangsspenningen er lavere enn terskelspenningen for nMOS transistoren, dvs. $0 \leq V_{inn} < V_{tn}$. nMOS transistoren vil da være AV og pMOS transistoren vil være PÅ fordi $V_{inn} < V_{DD} + V_{tp}$. Dette må resultere i at utgangen trekkes til V_{DD} på grunn av at $I_{dsn} = 0$ og dermed $I_{sdp} > I_{dsn}$. Med utgang $V_{ut} \approx V_{DD}$ vil pMOS transistoren åpenbart være i det lineære området fordi $V_{ut} > V_{inn} - V_{tp}$. Vi har da at:

$$\begin{aligned} I_{sdp} &= I_{dsn} \\ \beta_p \left(V_{sgp} - |V_{tp}| - \frac{V_{sdp}}{2} \right) V_{sdp} &= 0 \\ \beta_p \left(V_{sgp} - |V_{tp}| - \frac{V_{sdp}}{2} \right) \cdot (V_{DD} - V_{ut}) &= 0 \\ V_{ut} &= V_{DD}. \end{aligned}$$

2.13.2 Område B, se Fig. 2.30

Inngangsspenningen er høyere enn terskelspenningen og lavere enn $V_{DD}/2$ for nMOS transistoren, dvs. $V_{tn} \leq V_{inn} < V_{DD}/2$. nMOS transistoren vil da være PÅ og pMOS transistoren vil også være PÅ fordi $V_{inn} < V_{DD} + V_{tp}$. Begge transistorene er PÅ og det vil nå gå en strøm mellom V_{DD} og $V_{SS}(GND)$. Dersom vi antar at $\beta_n = \beta_p$, dvs transistorene er like sterke og $V_{inn} < V_{DD}/2$, som betyr at $V_{gsn}(= V_{inn}) < V_{sgp}(= V_{DD} - V_{inn})$, medfører dette at pMOS er kraftigere biasert og leverer mer strøm enn nMOS transistoren. Utgangen vil derfor ligge nærmere V_{DD} enn V_{SS} . Kriteriet for at nMOS transistoren skal være i metning er at $V_{ut} > V_{inn} - V_{tn}$ som betyr at $V_{inn} < V_{ut} + V_{tn}$. Det spesifikke punktet der nMOS transistoren går fra metning til lineært område er gitt av $V_{ut} = V_{inn} - V_{tn}$. For pMOS transistoren har vi at metning forutsetter $V_{ut} < V_{inn} - V_{tp}$ som betyr at $V_{inn} > V_{ut} + V_{tp}$ (husk at V_{tp} er negativ). Dette gir en spesifikt punkt for pMOS skifte mellom lineært område og metning $V_{ut} = V_{inn} - V_{tp}$. Som Fig. 2.30 viser vil nMOS transistoren være i metning og pMOS transistoren være i det lineære området. Vi har da følgende:

$$\begin{aligned} I_{dsn} &= I_{sdp} \\ V_{ut} &= V_{inn} + V_t + \sqrt{(V_{DD} - 2V_{inn})(V_{DD} - 2V_t)}, \end{aligned}$$

der $-V_{tp} = V_{tn} = V_t$ og $\beta_p = \beta_n$.

2.13.3 Område C, se Fig. 2.30

Inngangsspenningen er nær svitsjepunktet til inverteren, dvs. $V_{inn} = V_{DD}/2$. Vi har da at begge transistorene er i metning fordi $V_{inn} - V_{tn} \leq V_{ut} \leq V_{inn} - V_{tp}$. Her er det svært stor forsterkning fordi transistorstrømmene ikke er avhengig av utgangsspenningen i dette området.

Stor forsterkning betyr kraftig fall i utgangsspenningen. Vi har:

$$\begin{aligned} I_{dsn} &= I_{sdp} \\ \frac{\beta_n}{2} (V_{gsn} - V_{tn})^2 &= \frac{\beta_p}{2} (V_{sgp} - |V_{tp}|)^2 \\ (V_{inn} - V_t) &= (V_{DD} - V_{inn} - V_t) \\ V_{inn} &= \frac{V_{DD}}{2}, \end{aligned}$$

der $-V_{tp} = V_{tn} = V_t$ og $\beta_p = \beta_n$.

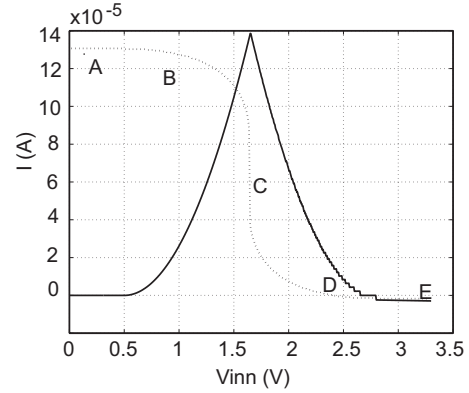


Figure 2.31: I en inverters utgangsstransisjon vil både pMOS- og nMOS transistoren være på samtidig slik at det vil gå en strøm mellom V_{DD} og GND .

Når en inverter endrer utgangsverdi vil det i løpet av transisjonen være korte perioder der både pMOS- og nMOS transistoren er på. I slike tilfeller vil det gå en strøm mellom V_{DD} og GND som vist i Fig. 2.31.

2.13.4 Område D, se Fig. 2.30

$V_{DD}/2 < V_{inn} < V_{DD} + V_{tp}$. nMOS transistoren er opplagt PÅ og pMOS transistoren er også PÅ fordi $V_{inn} < V_{DD} - V_{tp}$. nMOS transistoren er i lineært område fordi $V_{ut} < V_{inn} - V_{tn}$, og pMOS transistoren er i metning fordi $V_{ut} < V_{inn} - V_{tp}$, se Fig. 2.30. Vi har da at:

$$\begin{aligned} I_{dsn} &= I_{sdp} \\ \left(V_{inn} - V_{tn} - \frac{V_{ut}}{2} \right) V_{ut} &= \frac{1}{2} (V_{DD} - V_{inn} - |V_{tp}|)^2 \\ V_{ut} &= V_{inn} - V_t - \sqrt{(2V_{inn} - V_{DD})(V_{DD} - 2V_t)}, \end{aligned}$$

der $-V_{tp} = V_{tn} = V_t$ og $\beta_p = \beta_n$.

2.13.5 Område E, se Fig. 2.30

Inngangen er nå nær V_{DD} dvs. $V_{inn} > V_{DD} + V_{tp}$. nMOS transistoren er da PÅ og i metning, mens pMOS transistoren er AV. Utgangen vil da være svært nær V_{SS} , dvs. $V_{ut} \approx V_{ss}$. Vi har

2.14 Transistorstørrelser

(Kapittel 2.5.2 side 97)

$$\begin{aligned} I_{dsn} &= I_{dsp} \\ \beta_p \left(V_{inn} - V_{tn} - \frac{V_{ut}}{2} \right) V_{ut} &= 0 \\ V_{ut} &= 0 \end{aligned}$$

der $-V_{tp} = V_{tn} = V_t$ og $\beta_p = \beta_n$.

2.13.6 Oppsummering

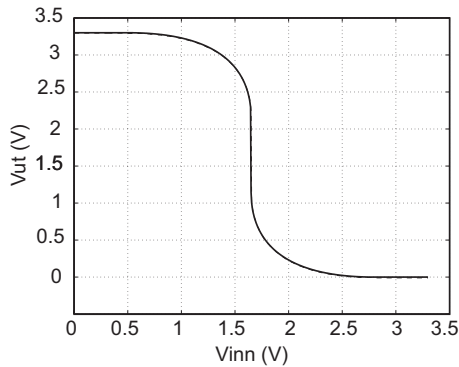


Figure 2.32: CMOS Inverter DC (V-V) karakteristik og DC karakteristik gitt av uttrykk for utgangsspenning i områdene A, B, C, D og E.

I Fig. 2.32 er en DC karakteristik modellert ved hjelp av ligningene for utgangsspenning i områdene A, B, C, D og E vist sammen med karakteristikken fra Fig. 2.30. Som vi ser gir dette selvfølgelig samme resultat.

		pMOS	nMOS	utgang
A	$0 \leq V_{inn} < V_{tn}$	lineær	av	$V_{ut} = V_{DD}$
B	$V_{tn} \leq V_{inn} < V_{DD}/2$	lineær	metning	$V_{ut} > V_{DD}/2$
C	$V_{inn} = V_{DD}/2$	metning	metning	V_{ut} faller
D	$V_{DD}/2 < V_{inn} \leq V_{DD} - V_{tp} $	metning	lineær	$V_{ut} < V_{DD}/2$
E	$V_{inn} > V_{DD} - V_{tp} $	av	lineær	$V_{ut} = 0$

Table 2.3: CMOS inverter (Tabell 2.3 side 96).

En oppsummering av inverterens operasjonsområder er vist i tabell 2.3. Inverterens *inngangsterskel* er definert som $V_{inv} = V_{inn} = V_{ut}$.

2.13.7 Mål

Forstå hvordan CMOS inverterens transfer-, eller DC (V-V), karakteristik ser ut, og kunne lokalisere områder der nMOS-, og pMOS transistorene er AV og PÅ, i lineært område eller metning.

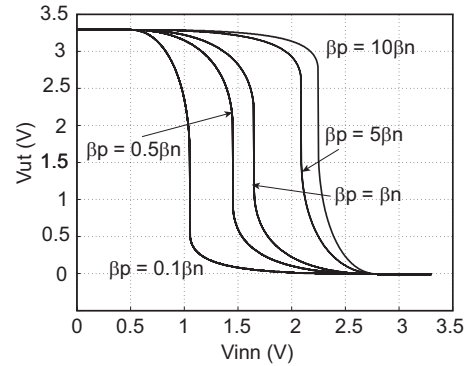


Figure 2.33: CMOS Inverter DC (V-V) karakteristik for forskjellige transistorstørrelser.

Transfer karakteristikker for invertere med ulike transistorstørrelser er vist i Fig. 2.33. Vi kan se hvordan forholdet i størrelse mellom de to transistorene påvirker inverterens transfer karakteristik. Dersom vi gjør pMOS transistoren sterkere enn nMOS transistoren vil svitsjepunktet, inngangsterskel, flyttes mot høyre fordi det blir tyngre å dra utgangen ned til 0 gjennom en svak nMOS transistor. Likeledes vil inngangsterskelen flyttes mot venstre når nMOS transistoren styrkes i forhold til pMOS transistoren.

2.14.1 Mål

Forstå hvordan transistorstørrelse, spesielt forholdet mellom nMOS- og pMOS transistorer, påvirker inverterens transfer karakteristik.

2.15 Støymargin

(Kapittel 2.5.3 side 98 - 99)



Figure 2.34: CMOS Invertere der utgang X skal drive neste inverter og tolkes som inngang Y.

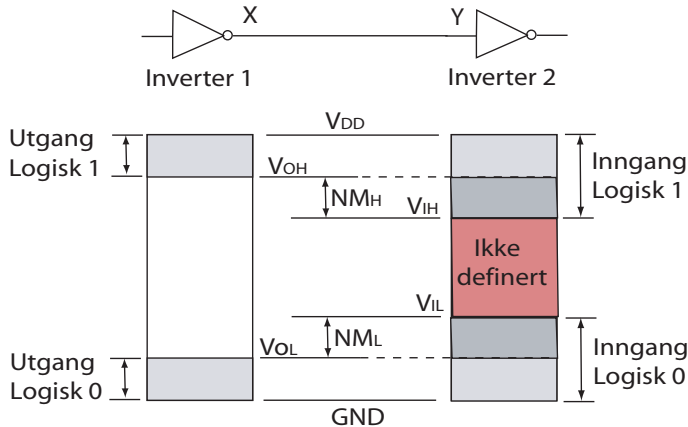


Figure 2.35: Støymarginer. (FIG2.27)

Støymargin har med robusthet å gjøre. Utgangen på en port må være tilstrekkelig nær de logiske verdiene 0 eller 1 (GND eller V_{DD}) slik at porter som skal tolke signalet i form av en inngang, tolker signalet riktig. Vi må definere sikkerhetsmarginer slik at alle porter fungerer riktig logisk. I Fig. 2.34 skal inverter 1 med utgang X drive inverter 2 med inngang Y. Vi forutsetter at $Y = X$. Det kritiske for inverter 2 er at inngangen Y er definert som et lovlig signal som vil gi en riktig logisk utgang på inverter 2. Dersom inverter 1 alltid produserer gode signaler som kan defineres som godkjent 0 eller 1 er situasjonen tilfredstillende. For å være sikker på å ikke generere gale logiske verdier er det fornuftig å legge et sterkere krav på utgangssignaler enn på inngangssignaler, dvs. dersom inverter 2 kan tolke signaler av dårlige kvalitet (svak 0 eller 1) enn inverter 1 kan produsere får vi en *støymargin*.

Definisjoner, se Fig. 2.35:

$$NM_L = V_{IL} - V_{OL},$$

der

$$V_{IL} = \text{høyeste inngang tolkes som 0,}$$

$$V_{OL} = \text{høyeste utgang definert som 0}$$

$$NM_H = V_{OH} - V_{IH},$$

der

$$V_{OH} = \text{laveste utgang definert som 1,}$$

$$V_{IH} = \text{laveste inngang tolkes som 1}$$

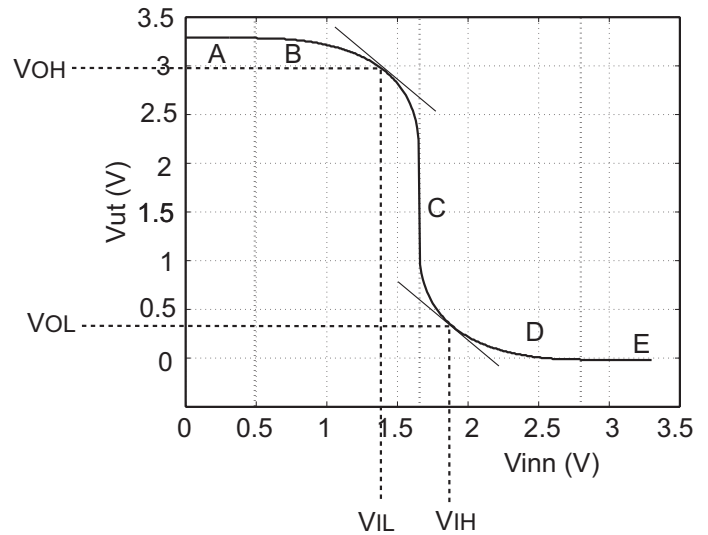


Figure 2.36: CMOS Inverter DC (V-V) karakteristikk med støymarginer. Der hvor forsterkningen er -1, dvs. der den deriverte av $V_{ut} = -1$ får vi V_{IH} , V_{IL} , V_{OH} og V_{OL} . (FIG2.28)

Støymarginer er vist i Fig. 2.35 og for en inverter i Fig. 2.36. Der hvor forsterkningen er -1, dvs. der den deriverte av $V_{ut} = -1$ får vi V_{IH} , V_{IL} , V_{OH} og V_{OL} .

2.15.1 Mål

Forstå hvordan støymarginer defineres og hvorfor det er nødvendig med støymarginer

2.16 Invertere med statisk last

(Kapittel 2.5.4 side 100 - 101)

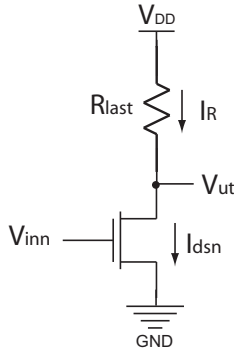


Figure 2.37: Inverter med lastmotstand. (FIG2.29a venstre)

Det er to grunnleggende typer statiske eller passive laster som er aktuelle i CMOS:

1. Motstand som kalles R_{last} .
2. Strømkilde som kalles I_{last} .

I Fig. 2.37 er det vist en inverter med en lastmotstand. Strømmen gjennom nMOS transistoren vil ved et stabilt DC punkt alltid være lik strømmen gjennom lastmotstanden. Med andre ord; for en gitt inngangsspenning vil det finnes en utgangsspenning som tilfredstiller kravet $I_R = I_{dsn}$. I motsetning til en vanlig inverter vil denne inverteren ikke ha en aktiv last som endres med inngangssignalet (pMOS transistoren i vanlig inverter når inngangssignalet stiger), men vil ha en last som ikke er avhengig av inngangssignalet og som vil variere lineært mhp. på utgangen, dvs.

$$R_{last} = \frac{(V_{DD} - V_{ut})}{I_{dsn}} \quad (2.31)$$

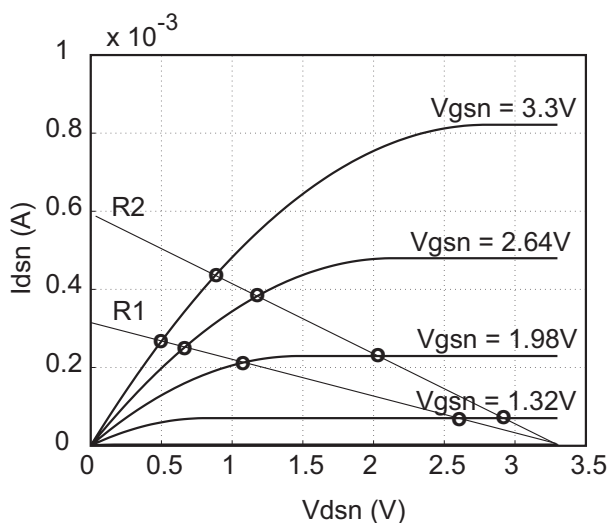


Figure 2.38: Transistorstrømmer for inverter med lastmotstand. (FIG2.29b)

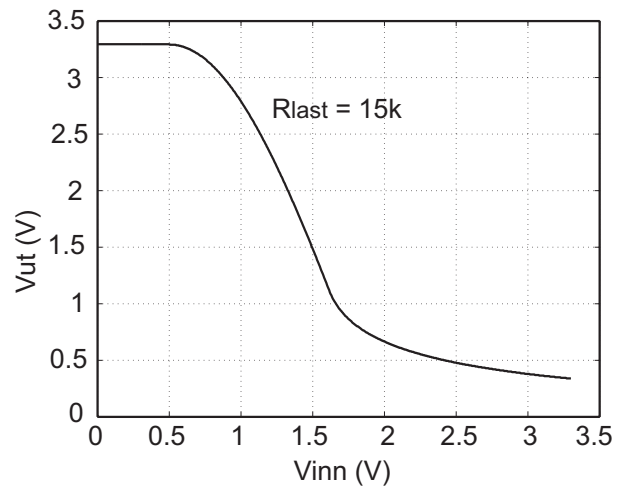


Figure 2.39: DC karakteristikk for inverter med lastmotstand. (FIG2.29c)

Transistorstrømmer i en inverter med en motstand som last er vist i Fig. 2.38, og en DC karakteristikk for inverter med lastmotstand lik $15k\Omega$ er vist i Fig. 2.39. Som vi ser er det tyngre å dra utgangen lav gjennom nMOS transistoren fordi strømmen i lastmotstanden øker når utgangen faller. I en komplementær inverter vil pMOS strømmen avta når utgangen faller fordi et fall i utgangen er avhengig av en økning i inngangsspenningen som direkte reduserer pMOS strømmen.

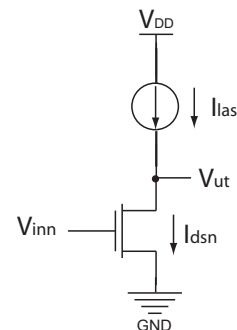


Figure 2.40: Inverter med ideell strømkilde som last. (FIG2.29a høyre)

I Fig. 2.40 er det vist en inverter med en laststrøm. Strømmen gjennom nMOS transistoren vil ved et stabilt DC punkt alltid være lik laststrømmen. Med andre ord; for en gitt inngangsspenning vil det finnes en utgangsspenning som tilfredstiller kravet $I_{dsn} = I_{last}$. I motsetning til en vanlig inverter vil denne inverteren ikke ha en aktiv last som endres med inngangssignalet (pMOS transistoren i vanlig inverter), men vil ha en last som er uavhengig av inngangssignalet og utgangssignalet. For en ideell strømkilde vil vi anta uendelig impedans (motstand), som betyr at laststrømmen er konstant og dermed uavhengig av utgangs- og inngangsspenningen. En DC karakteristikk for en inverter med strømlast lik $200\mu A$ er vist i Fig. 2.41.

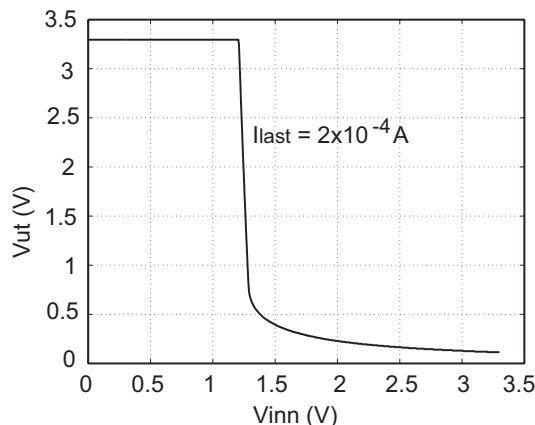


Figure 2.41: DC karakteristik for inverter med ideell strømkilde ($200\mu\text{A}$) som last.

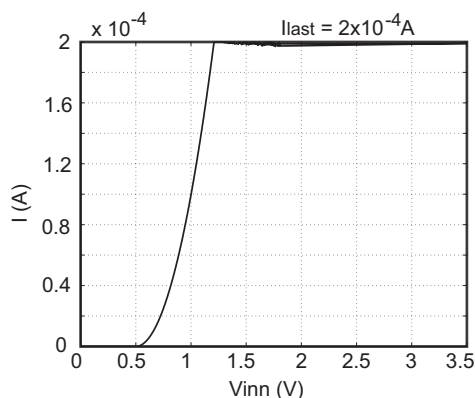


Figure 2.42: Strømforbruk for inverter med ideell strømkilde ($200\mu\text{A}$) som last.

2.16.1 Sammenligning med vanlig inverter

For invertere med statisk last vil det alltid gå en strøm mellom V_{DD} og GND som vil medføre at porten vil ha en betydelig statisk strøm som gir et statisk *effektforbruk* eller *strømforbruk* som vist i Fig. 2.42. Dette er en klar ulempe i forhold til en vanlig (komplementær) CMOS inverter der den statiske strømmen er neglisjerbar fordi pMOS transistoren (aktiv) vil skrus av når inngangen er 1.

2.16.2 Mål

Forstå hvordan inverter med ulike typer statisk last, motstand, strømkilde, fungerer.

2.17 Pseudo nMOS inverter

(Kapittel 2.5.4 side 101)

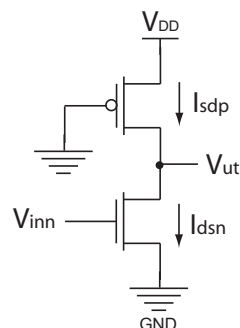


Figure 2.43: Pseudo nMOS inverter. (FIG2.30a)

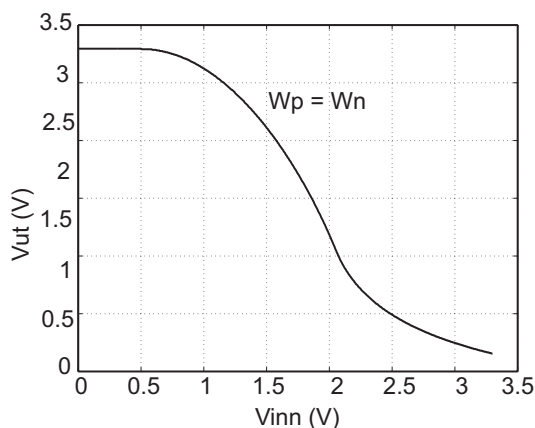


Figure 2.44: Pseudo nMOS inverter DC karakteristik med $W_p/L_p = W_n/L_n$. (FIG2.30c)

En såkalt pseudo nMOS inverter⁴ er vist i Fig. 2.43. Lasttransistoren (pMOS) vil alltid være på, men vil være i lineært område eller metning avhengig av utgangsspenningen. I lineært område kan pMOS transistoren modelleres som en motstand og i metning kan pMOS transistoren forenklet⁵ modelleres som en ideell strømkilde. Lasten vil her virke som en kombinasjon av motstand og ideell strømkilde.

En DC karakteristik for en pseudo nMOS inverter er vist i Fig. 2.44. Vi ser at det er tyngre å dra utgangen lav enn for en komplementær inverter.

2.17.1 Sammenligning med andre invertere

En vesentlig forskjell på pseudo nMOS inverter og komplementær inverter er strømforbruket når utgangen er lav. Vi ser at strømmen som trekkes av kretsen ikke går ned mot 0 for pseudo nMOS inverteren som vist i Fig. 2.45. Forsterkningen er i tillegg mye mindre og dermed er støymarginer dårligere enn for komplementær inverter. En fordel mel pseudo nMOS inverteren er redusert areal, primært gjennom redusert ruting, og

⁴Opprinnelsen til dette navnet er at en pMOS transistor som alltid er på har lignende karakteristik som en depletion transistor i NMOS teknologi som er forløperen til CMOS teknologi.

⁵Vi ser da bort ifra kanallengdemodulasjon.

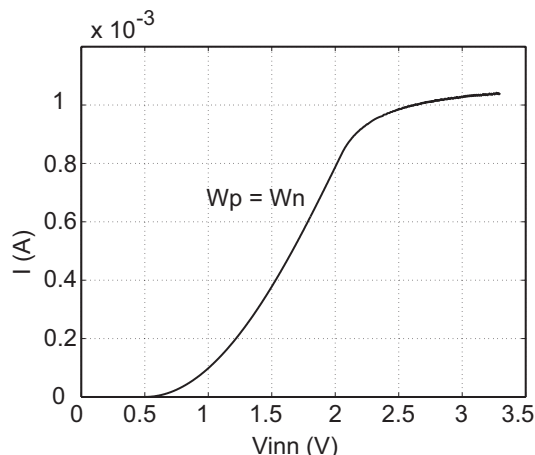


Figure 2.45: Strømforbruk for pseudo nMOS inverter som funksjon av inngangsspenningen med $W_p/L_p = W_n/L_n$. (FIG2.30d)

redusert inngangslast fordi inngangen bare skal drive en transistor.

2.17.2 Mål

Forstå hvordan pseudo nMOS inverter fungerer.

2.18 Pass transistor DC karakteristikk

(Kapittel 2.5.6 side 101 - 102)

* Forutsetter en enkel forståelse av pass transistorer og transmisjonsporter. (INF3400 Del 1: Pass transistorer og transmisjonsporter, Kapittel 1.4.6 side 14 - 15).

2.18.1 Terskelfall

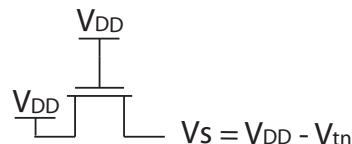


Figure 2.46: Terskelfall for nMOS transistor (FIG2.31a).

En nMOS pass transistor vil flytte en sterk logisk 0 fra source til en sterk logisk 0 på drain, og en sterk 1 fra drain flyttes til en svak 1 på source siden av nMOS transistoren. Dersom vi antar at det er logisk 1 på gaten (V_{DD}) vil transistoren skrus av når $V_{gs} < V_{tn0}$, der V_{tn0} er nominelle terskelspenning for nMOS transistor, som betyr at $V_{DD} - V_s$ må være større enn V_{tn0} . Vi har altså at:

$$\begin{aligned} V_{gs} &\geq V_{tn} \\ V_g - V_s &\geq V_{tn} \\ V_{DD} - V_s &\geq V_{tn} \\ V_s &\leq V_{DD} - V_{tn} \end{aligned}$$

En svak 1'er (V'_{DD}) kan være en logisk 1 med et terskelfall, som vist i Fig. 2.46:

$$V'_{DD} = V_{DD} - V_{tn}. \quad (2.32)$$

Vi har til nå ikke tatt hensyn til bodyeffekten som vil påvirke pass transistorens transmisjonsegenskaper ytterligere. For en nMOS transistor med en logisk 1 på drain og der source på transistoren ikke er koblet til 0 vil source på transistoren og dermed også V_{sb} ikke bli 0V. Vi kan benytte modellen for effektiv terskelspenning for en nMOS transistor med body effekt til å finne en førsteordens tilnærming for den reelle spenningen på source siden av en nMOS pass transistor

$$\begin{aligned} V''_{DD} &= V_{DD} - V_{t_{effektive}} \\ &= V_{DD} - V_{tn0} + \gamma (\sqrt{\Phi_s + V_{sb}} - \sqrt{\Phi_s}) \end{aligned} \quad (2.33)$$

der vi antar $V_{sb} = V_{DD} - V_{tn0}$. For $V_{DD} = 3.3V$, $V_{tn0} = 0.5V$, $\gamma = 0.58$ og $\Phi_s = 0.93V$ får vi $V_s = 3.3V - 1.06V = 2.24V$ mot 2.8V uten bodyeffekt.

Dersom vi anvender en svak logisk 1 på drain på en kjede (serie) av nMOS transistorer, som vist i Fig. 2.47, vil signalet flyttes gjennom kjeden av nMOS transistorer til en like svak, men ikke svakere, logisk 1 på på source terminalen på den siste nMOS transistoren.

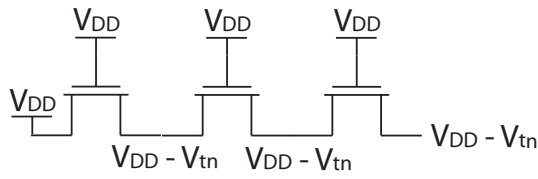


Figure 2.47: Terskelfall for kjede av nMOS transistorer (FIG2.31c).

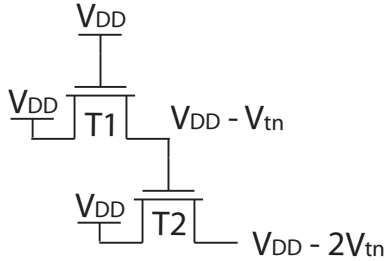


Figure 2.48: Terskelfall for to nMOS transistorer (FIG2.31d).

Dersom vi kobler to nMOS transistorer som vist i Fig. 2.48 vil den øvre transistoren få et terskelfall på source som vil styre gaten til den nedre transistoren. Gate spenningen på den nedre transistoren blir da lik

$$\begin{aligned} V_{gs} &\geq V_{tn2} \\ V_g - V_s &\geq V_{tn2} \\ V_{DD} - V_{tn1} - V_s &\geq V_{tn2} \\ V_s &\leq V_{DD} - (V_{tn1} + V_{tn2}), \end{aligned}$$

der V_{tn1} og V_{tn2} er effektive terskelspenninger for henholdsvis T1 og T2. Man kan forenkle uttrykket ved å sette $V_{tn2} = V_{tn1} = V_{tn}$ slik at $V_{DD}' = V_{DD} - 2V_{tn}$. De to transistorene vil ha forskjellig effektiv terskelspenning fordi de har forskjellig source bulk spenning som i praksis betyr at det blir et større effektivt terskelfall for T1 enn for T2.

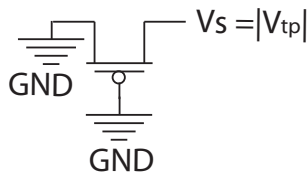


Figure 2.49: Terskelfall for pMOS transistor (FIG2.31b).

Tilsvarende vil en pMOS transistorer brukes til å flytte en sterk logisk 1 fra source terminalen til en sterk 1 på drain terminalen, mens en sterk 0 flyttes fra drain til en svak 0 på source terminalen. En svak 0'er (V_{SS}') kan være en logisk 0 med et terskelspenning i tillegg, som vist i Fig. 2.49:

$$V_{SS}' = |V_{tp}|, \quad (2.34)$$

der V_{tp} er effektiv terskelspenning gitt bulk source spenningen for pMOS transistoren $V_{b_{sp}} = V_{DD} - |V_{tp0}|$.

2.18.2 Transmisjonsport

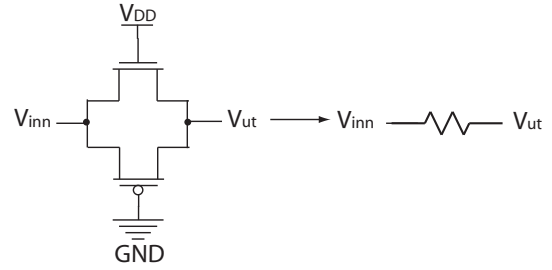


Figure 2.50: Ekvivalent motstand i transmisjonsport (FIG2.32 øvre).

En transmisjonsport bestående av en nMOS- og en pMOS pass transistor er vist i Fig. 2.50. Fra [?] har vi at en nMOS transistor i lineært område kan modelleres som en motstand

$$\begin{aligned} R_n &= \left(\beta_n \left(V_{gsn} - V_{tn} - \frac{V_{dsn}}{2} \right) \right)^{-1} \quad (2.35) \\ &= \left(\beta_n \left(V_{DD} - V_{inn} - V_{tn} - \frac{(V_{ut} - V_{inn})}{2} \right) \right)^{-1} \\ &= \left(\beta_n \left(V_{DD} - \frac{V_{inn}}{2} - V_{tn} - \frac{V_{ut}}{2} \right) \right)^{-1} \\ &\approx \left(\beta_n (V_{DD} - V_{inn} - V_{tn}) \right)^{-1}, \end{aligned}$$

der V_{tn} er effektiv terskelspenning for nMOS transistoren og vi antar at $V_{ut} \approx V_{inn}$ (følger). Vi har to forskjellige muligheter for signaltransport for transmisjonsporten:

1. Positiv transisjon på inngangen, dvs. V_{inn} endres fra 0 til 1.
2. Negativ transisjon på inngangen, dvs. V_{inn} endres fra 1 til 0.

Vi ser først på tilfellet der inngangen skifter fra 0 til 1. nMOS transistoren er i lineært område når

$$\begin{aligned} V_{dsn} &< V_{gsn} - V_{tn}, \\ V_{ut} - V_{inn} &< V_{DD} - V_{inn} - V_{tn} \\ V_{ut} &< V_{DD} - V_{tn}. \end{aligned} \quad (2.36)$$

Vi ser at nMOS transistoren sannsynligvis er i lineært område for alle inngangsspenninger ved en positiv transisjon. Husk at utgangen vil starte en transisjon fra 0 til 1 før transisjonen på inngangen er ferdig, dvs. $V_{ut} \approx V_{inn}$. Modell for motstandsekivalenten for nMOS transistoren i lineært område viser at motstandsverdien øker når inngangsspenningen øker som vist i Fig. 2.51. Tilsvarende argumentasjon gjelder for pMOS transistoren der vi kan anta at transistoren er i lineært område for alle inngangsspenninger, og en forenklet pMOS motstandsekivalent kan uttrykkes som

$$\begin{aligned} R_p &= (\beta_p (V_{sgp} - |V_{tp}|))^{-1} \quad (2.37) \\ &= (\beta_p (V_{ut} - |V_{tp}|))^{-1}, \end{aligned}$$

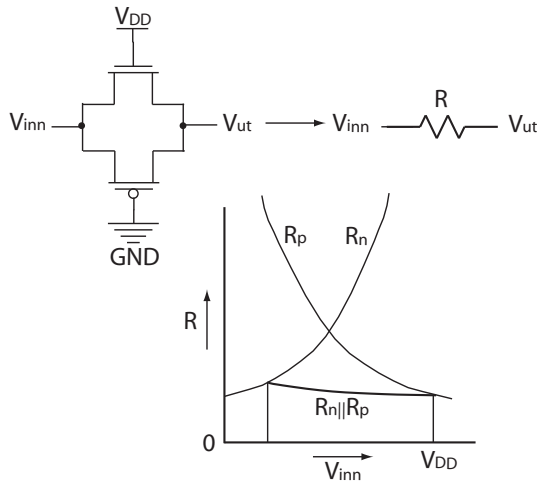


Figure 2.51: Ekvivalent motstand i transmisjonsport (FIG2.32 nedre).

der V_{tp} er effektiv terskelspenning for pMOS transistoren. Vi ser at R_p vil avta med økende inngangsspenning som vist i Fig. 2.51. Den effektive motstanden vil være gitt av motstandsekvivalenter for en nMOS- og en pMOS transistor i parallell:

$$\begin{aligned} R &= R_n || R_p \\ &= \frac{R_n R_p}{R_n + R_p}. \end{aligned}$$

2.18.3 Konduktans

En konduktans kan defineres som

$$\begin{aligned} G &= R^{-1} \\ &= \frac{I}{V}. \end{aligned}$$

For nMOS og pMOS transistorer i lineært område får vi

$$\begin{aligned} G_n &= R_n^{-1} \\ &= \beta_n (V_{DD} - V_{inn} - V_{tn}) \\ G_p &= R_p^{-1} \\ &= \beta_p (V_{ut} - |V_{tp}|) \end{aligned}$$

Parallellkobling av to konduktanse er gitt av

$$\begin{aligned} G_{0 \rightarrow 1} &= G_n + G_p \\ &= \beta_n (V_{DD} - V_{inn} - V_{tn}) + \beta_p (V_{ut} - |V_{tp}|) \\ &= \beta (V_{DD} - 2V_t + (V_{ut} - V_{inn})) \\ &\approx \beta (V_{DD} - 2V_t), \end{aligned}$$

der $V_t = V_{tn} = -V_{tp}$, $\beta = \beta_n = \beta_p$ og $V_{ut} \approx V_{inn}$. I en transmisjonsport vil utgangen følge inngangen med en liten tidsforsinkelse slik at utgangen i de fleste tilfeller vil være nesten

lik inngangen. Vi ser at med de forenklinger vi har gjort vil ekvivalent konduktansen være konstant $\approx \beta V_{DD}$, dette tilsvarer en ekvivalent motstand $R_{0 \rightarrow 1} = (\beta V_{DD})^{-1}$ som tilsvarer en strøm $I = \beta V_{DD} (V_{ut} - V_{inn})$. Transmisjonsporten er symmetrisk som betyr at for en motsatt transisjon på inngangen vil vi få tilsvarende forenklet ekvivalent konduktans (og motstand)

$$\begin{aligned} G_{1 \rightarrow 0} &= \beta (V_{DD} - 2V_t + (V_{inn} - V_{ut})) \\ &\approx \beta (V_{DD} - 2V_t) \\ R_{1 \rightarrow 0} &\approx (\beta (V_{DD} - 2V_t))^{-1}. \end{aligned} \quad (2.38)$$

Vi ser at med forenklet ekvivalent får vi samme konduktans og motstandverdi $G = G_{1 \rightarrow 0} = G_{0 \rightarrow 1}$ og $R = R_{1 \rightarrow 0} = R_{0 \rightarrow 1}$.

2.18.4 Mål

Forstå hvordan pass transistorer virker med terskelfall og ekvivalent motstand.

2.19 Tristate inverter

(Kapittel 2.5.6 side 102 - 103)

* Forutsetter en enkel forståelse av tristate inverter[?].
(INF3400 Del 1: , Kapittel 1.4.7 side 17 - 18).

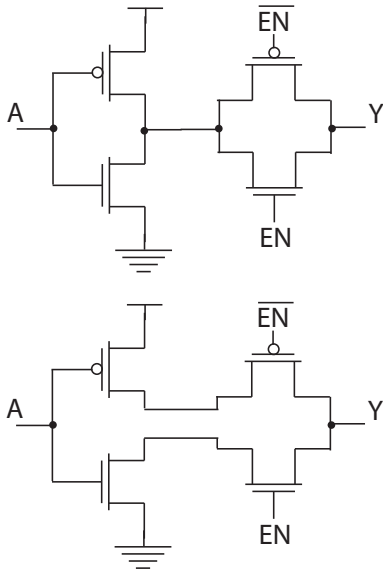


Figure 2.52: Tristate inverter med transmisjonsport (FIG2.33a og b).

Vi kan lage en tristate inverter ved å kombinere en inverter og en transmisjonsport som vist i Fig. 2.52 (øverste figur). Enable signalet EN brukes til å kontrollere om utgangen skal drives av inverteren eller ikke. Ser vi nærmere på kretsen vil vi se at kretsene i Fig. 2.52 vil være logisk ekvivalente. En fordel med den nederste kretsen er at vi får en mer effektiv tristate inverter.

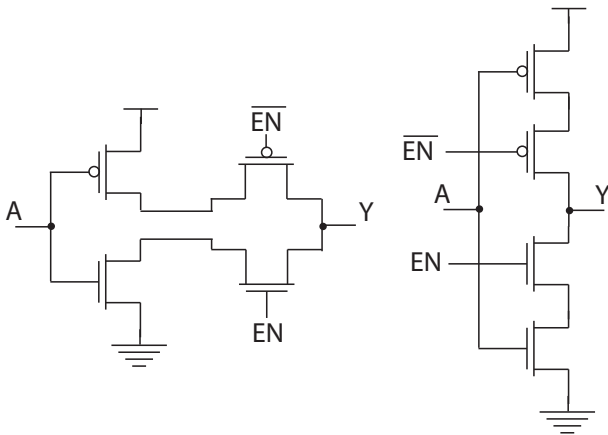


Figure 2.53: Tristate inverter (FIG2.33b og c).

En elektrisk ekvivalent implementasjon av den nederste figuren i Fig. 2.52 er vist i Fig. 2.53. Legg merke til at figuren på høyre side er identisk med tristate inverteren i kapittel 1.

2.19.1 Mål

Forstå hvordan tristate inverter virker.

Chapter 3

CMOS proses teknologi

3.1 Introduksjon til utleggsregler

(Kapittel 1.5.3 side 28 - 32)

3.1.1 Mål

Enkel forståelse av utleggsregler.

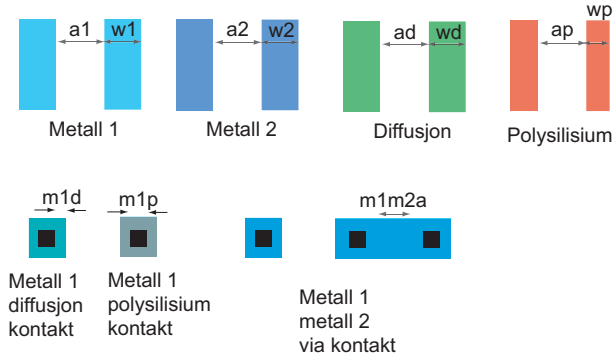


Figure 3.1: Utleggsregler (FIG1.39)

Noen sentrale minimumsavstander og størrelser er vist i Fig. 3.1. Ulike CMOS prosesser vil ha forskjellige utleggsregler. Det er slik at moderne prosesser tillater generelt mindre avstander og strukturer enn eldre prosesser. Dette kommer av framskritt i prosesseteknologien. Årsaken til at man krever en viss avstand mellom ulike signalførende noder, for eksempel avstand mellom metalledere, er fare for elektrisk interferens (crosstalk) mellom signaler som ikke skal påvirkes av hverandre. Det er alltid en avveining mellom avstander, dvs. areal, og elektrisk interferens.

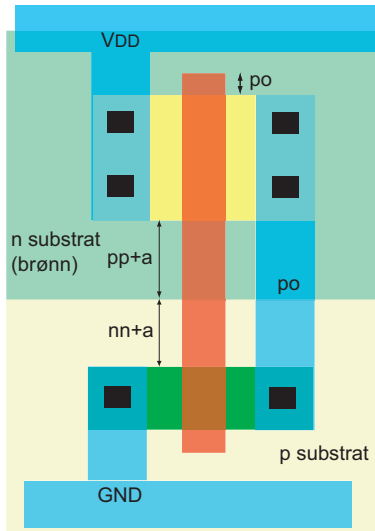


Figure 3.2: Noen utleggsregler for inverter (FIG1.39)

Noen sentrale regler for avstander og størrelser i tilknytning til en inverter er vist i Fig. 3.2. I dette eksemplet har vi et ptype substrat og ntype brønn. Det er vanlig i digital CMOS å benytte minimumsstørrelser på ulike strukturer, typisk transistorer. Dette medfører en gevinst i redusert areal, men også i redusert kapasitans som vil medføre at kretsene vil ha liten tidsforsinkelse. Liten tidsforsinkelse gir raske kretser som kan fungere med svært høye klokkefrekvenser.

3.2 Introduksjon til CMOS fabrikkasjonsprosessen

(Kapittel 1.5.2 side 24 - 28)

En fabrikkasjon av en CMOS krets består av en serie av steg der *lag i en krets* defineres gjennom en prosess som kalles *fotolitografi*.

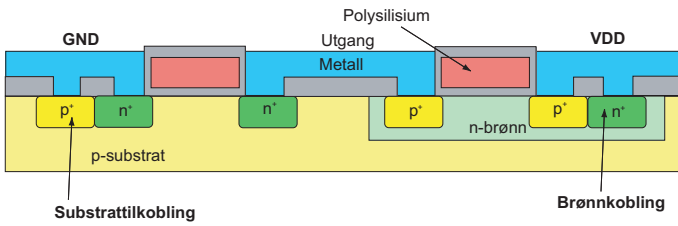


Figure 3.3: Tverrsnitt av en inverter. (FIG1.34)

En CMOS inverter, som vist i Fig. 3.3 kan defineres ved hjelp av 6 hypotetiske masker:

1. n-brønn.
2. Polysilium.
3. n+ diffusjon.
4. p+ diffusjon.
5. Kontakter.
6. Metall.

I Fig. 3.4 de hypotetiske maskene for en CMOS inverter vist.

I utgangspunktet starter vi med et p-type substrat som er relativt svakt dopet p-type silisium som vist i Fig. 3.5 (a). Neste prosesseringssteg er å legge på et lag med silisiumdioksid (tykt) SiO_2 (b) som brukes som isolering til substratet. Dette gjøres ved at en wafer blir oksidert ved høy temperatur (typisk 900–1200°C) slik at Si og O_2 reagerer til SiO_2 på waferens overflate. Silisiumdioksidlaget må merkes slik at n-brønnen kan bli definert.

I neste steg legges et lag fotoresist (c) på brikken (wafer) som vil forandre struktur (soften) når den blir utsatt for lys. For å kunne fjerne silisiumdioksidlaget der n-brønner skal være må man bruke en maske for n-brønn som vist i Fig. 3.6. En slik maske er identisk med n-brønn regionen vist i Fig. 3.4. n-brønnmasken anvendes til å utsette fotoresist for lys i områder der n-brønnen skal være (Fig. 3.6 (c)), deretter fjernes fotoresisten (d) som er påvirket av lys og deretter fjernes silisiumdioksid der det ikke er fotoresist (e). p-substratet er nå eksponert i det området der n-brønnen skal ligge og all fotoresist fjernes (f).

n-brønnen kan nå dannes, som vist i Fig. 3.7, enten med diffusjon eller ioneimplantering. Legg merke til at n-brønnen blir større enn masken tilsier (g). Tilslutt fjerner vi silisiumdioksid (h) med hydrogenfluoridsyre (HF) slik at vi ender opp med et p-type substrat med en n-type brønn.

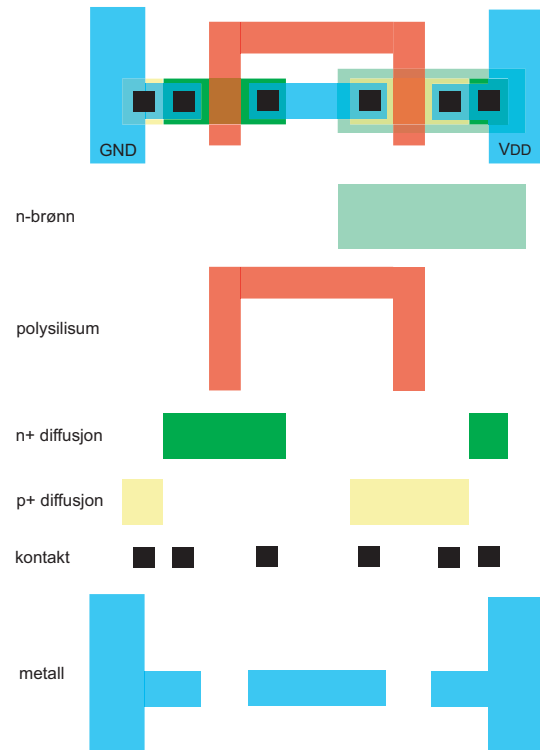


Figure 3.4: Masker for en inverter. (FIG1.35)

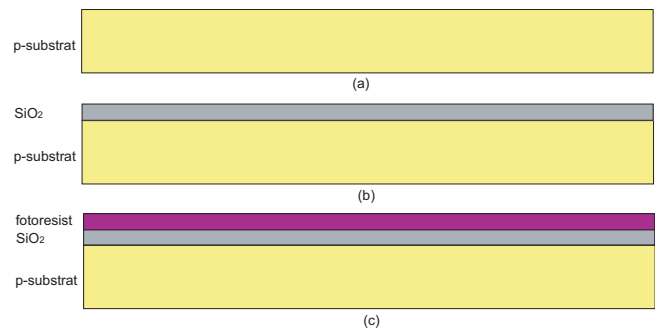


Figure 3.5: Tverrsnitt ved prosessering av n-brønn. (FIG1.36 a, b og c)

Det neste steget er å lage gatene som består av polysilium over et meget tynt lag med silisiumdioksid som kalles tynnoksid. Først groes et tynt lag med silisiumdioksid og deretter groes polysilium ved hjelp av SiH_4 gass som vist i Fig. 3.8 (a). Polysilium er kraftig dopet slik at motstanden i polysilium blir lav, dvs. polysilium blir en rimelig god leder. Nå legges et lag med fotoresist og ved å bruke polysilium maske fra Fig. 3.4 får vi laget gatene som består av polysilium som vist i Fig. 3.8 (b).

n+ områdene blir laget ved diffusjon for å danne aktive områder for nMOS transistorer og brønnkontakter. Dette gjøres med en tilsvarende prosess som ved dannelse av n-brønnen som vist i Fig. 3.5, 3.6 og 3.7. n+ områdene er vist i Fig. 3.8 (c) til (e). Selv om n+ diffusjon typisk blir dannet ved hjelp av ioneimplantering ble det historisk gjort ved hjelp av diffusjon og derav navnet *n-diffusjon*. Tilslutt fjernes silisiumdioksid som vist i (f).

Samme prosedyre gjentas for p+ diffusjon som vist i Fig.

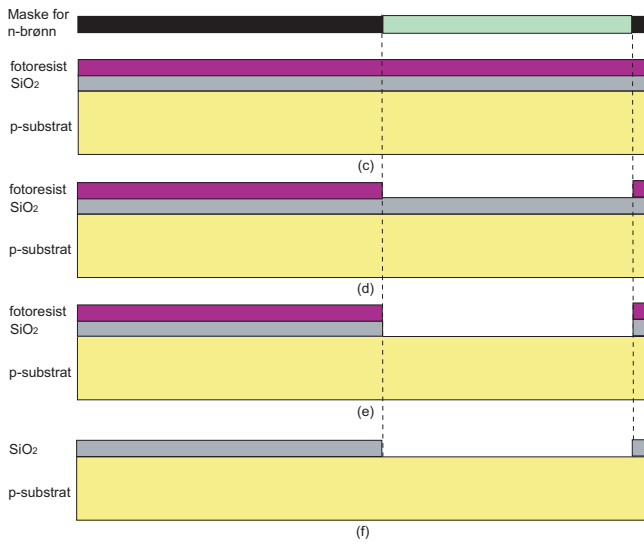


Figure 3.6: Tverrsnitt ved prosessering av n-brønn. (FIG1.36 c, d, e og f)

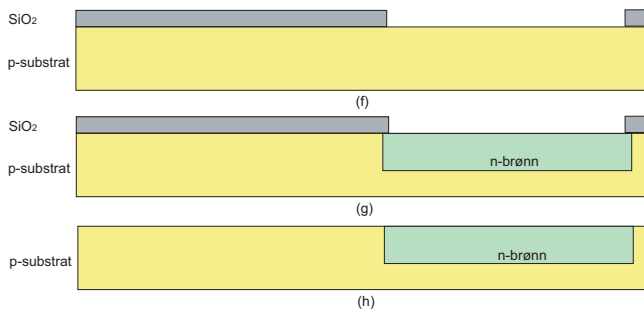


Figure 3.7: Tverrsnitt ved prosessering av n-brønn.. (FIG1.36 f, g og h)

3.9 (a). Etter at p+ diffusjon er dannet groes et tykt lag med silisiumdioksid på hele waferen og kontaktåpninger til de aktive diffusjonsområdene blir laget ved hjelp av fotoresist som vist i (b). Til slutt legges metall, aluminium, eller kobber i moderne prosesser, over hele waferen. Metall masken som er vist i Fig. 3.4 blir brukt til å fjerne metall ved hjelp av etsing der det ikke skal være metall som vist i Fig. 3.9 (d).

Det er viktig å bemerke at moderne CMOS prosesser kan være langt mer komplekse, med mange metallag og andre steg for å forbedre de elektriske egenskapene til transistorer og kretser.

3.2.1 Mål

Kunne forstå enkle grunnleggende fabrikasjonssteg for en typisk CMOS prosess.

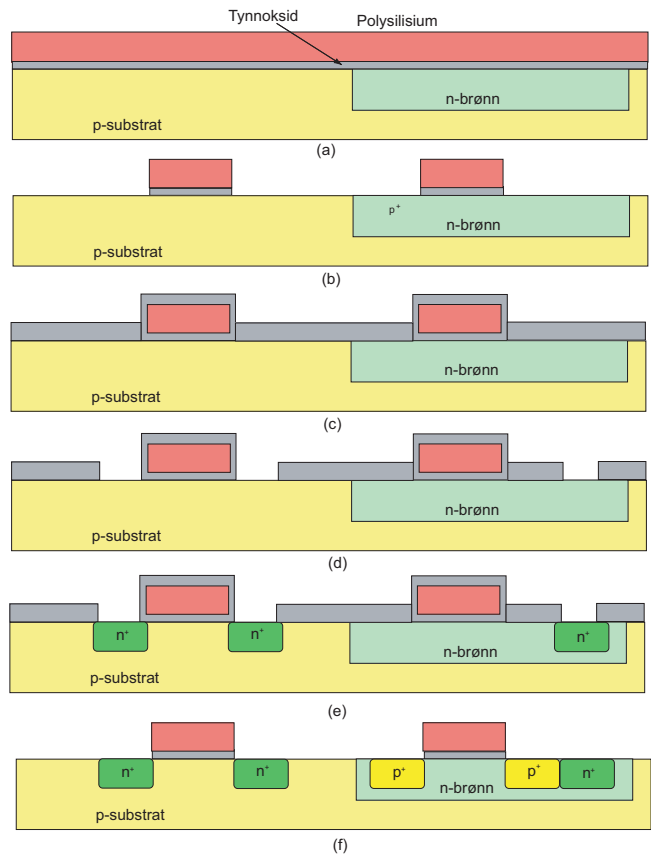


Figure 3.8: Tverrsnitt ved prosessering av polysilicium og p+ diffusjon. (FIG1.37)

3.3 Utlegg av port

(Kapittel 1.5.4 side 32 - 33)

I noen tilfeller er det viktig å bruke mye tid på utleggsdetaljer for en port dersom det er spesielle krav til portens ytelse, eller porten skal brukes svært mange ganger. I de fleste tilfeller vil en standard port, eller *standard celle* være tilstrekkelig effektiv for applikasjonen. I en standard celle vil vi velge noen retningslinjer for referanser *GND* og *V_{DD}*, og innganger og utganger. Det er vanlig å legge polysilicium vertikalt og diffusjon horisontalt som vist i Fig. 3.10.

I Fig. 3.11 er standard celle utlegg for en 3inngangs NAND port vist. Vi ser at høyden på cellen (H) er uavhengig av antall innganger og at alle innganger går vertikalt som for standard celle inverteren. Et standard celle utlegg for en NOR port vil ha en tilsvarende høyde som NAND porten og invertere. Ved å bruke standard celle utlegg kan man forenkle implementasjonen av en digital krets. Konvensjonen er at cellene er like høye, *V_{DD}* ligger øverst og *GND* ligger nederst og at inngangene går vertikalt. I moderne prosesser er det vanlig å redusere bruk av polysilicium til et minimum på grunn av stor egenmotstand. Kritiske signaler kan legges i metall av kobber som vil være meget gode ledere.

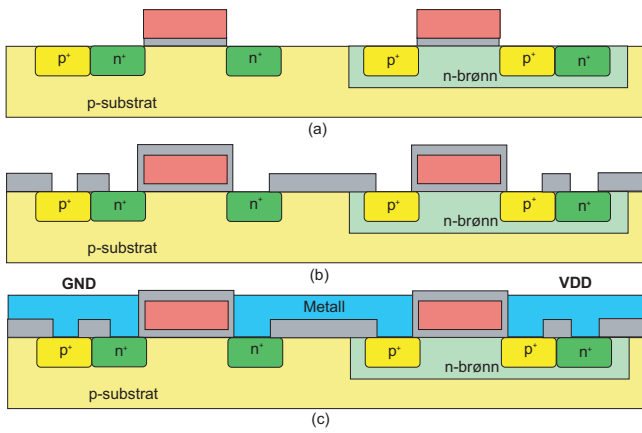


Figure 3.9: Tverrsnitt ved prosessering av p+ diffusjon, kontakter og metall. (FIG1.38)

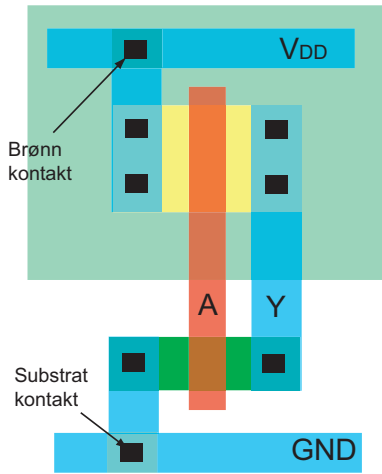


Figure 3.10: Standard celle utlegg av inverter. (FIG1.41)

3.3.1 Mål

Forstå prinsippet med standard celle utlegg.

3.4 Stick diagrammer

(Kapittel 1.5.4 side 33 - 35)

Det å lage utlegg for hånd tar mye tid. Det er hensiktsmessig å kunne planlegge et utlegg og beregne arealbehov før utleggsdetaljene bestemmes. Vi kaller et slikt mellomnivå *stick diagram*. Et stick diagram for en inverter og 3innngangs NAND port er vist i Fig. 3.12.

Det er enkelt å beregne arealet for utlegget for en gitt prosess ved å se på noen få grunnleggende utleggsregler. De konkrete utleggsreglene for en gitt prosess bestemmer de faktiske avstandene mellom lag eller aktive områder som er aktuelle. Det er verdt å merke seg at utleggsregler kan variere mye mellom ulike prosesser. Vi kan se på transistorene som komponenter med et utlegg for en minimumsstørrelse som vist i Fig. 3.13. Arealet som en transistor trenger kan da beregnes til transistor areal (A_T) = (2 Metall bredder + 2 metall til polysilisium avstander + transistor bredde) x (transistor lengde + 2 gate overlapp).

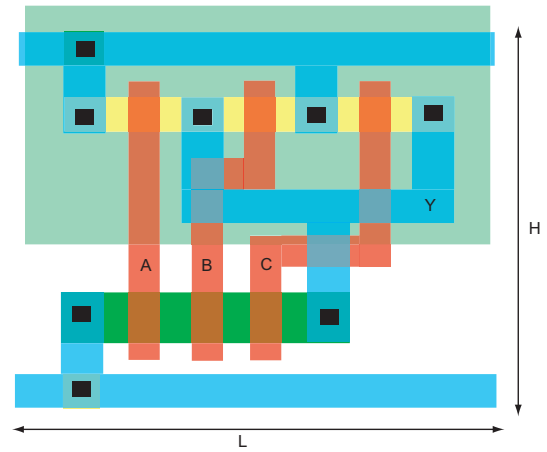


Figure 3.11: Standard celle utlegg av NAND3 port. (FIG1.42)

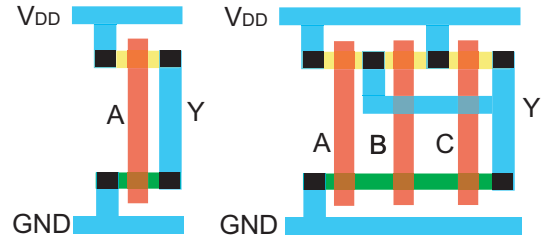


Figure 3.12: Stick diagram for inverter og 3innngangs NAND port. (FIG1.43)

Vi ser at utstrekningen for en transistor vil være: Lengde (L_T) = 2 Metall bredder + 2 metall til polysilisium avstander + transistor bredde. Transistoren vil ha et behov i høyden gitt som: Lengde (H_T) = transistor lengde + 2 gate overlapp. Dersom vi skal legge flere transistorer i samme diffusjonsområde, som for eksempel for NAND port, er det hensiktsmessig å definere arealbehovet mellom transistorene som felles. Vi ser da at en transistor i tillegg vil medføre en lengdeøkning i utlegg: Lengde' = (L'_T) = Metall bredde + 2 metall til polysilisium avstander + transistor bredde. Vi kan definere et *horisontalt utleggsspør* som lengden $L_s = L'_T$. De spesifikke utleggsregler for minimums størrelser og avstander representerer arealbehovet for en minimums transistor i en gitt prosess.

I tillegg til areal til transistorer vil det være noen sentrale utleggsregler som vi må ta hensyn til. Dette er blant annet minimumsavstand mellom polysilisium og avtand mellom metalledere som vist i Fig. 3.14. Vi kan definere noen størrelser i høyderetningen som er praktisk for beregning av areal for en port.

Utleggsavstander for brønn og diffusjoner er vist i Fig. 3.15.

I tillegg til avstander mellom ulike lag i en port må vi også definere minimumsavstand mellom standard celler i vertikal retning som vist i Fig. 3.16. Standard celler kan settes helt inntil hverandre i horisontal retning, men må ha en avstant i vertikal retning. Vi ser av figuren at den totale lengden for en 3innngangs NAND port vil være $4L_s$ og høyden vil være $5L_s$.

En implementasjon av den boolske funksjonen $Y =$

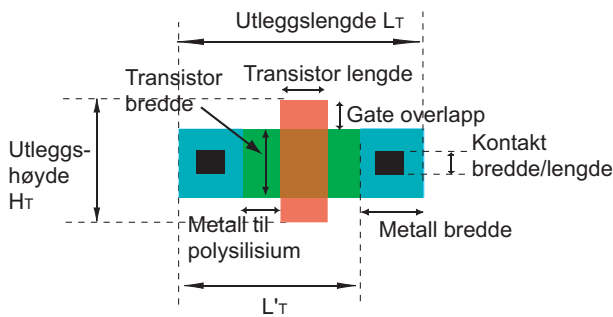


Figure 3.13: Standard utlegg for minimums nMOS transistor. (FIG1.44b)

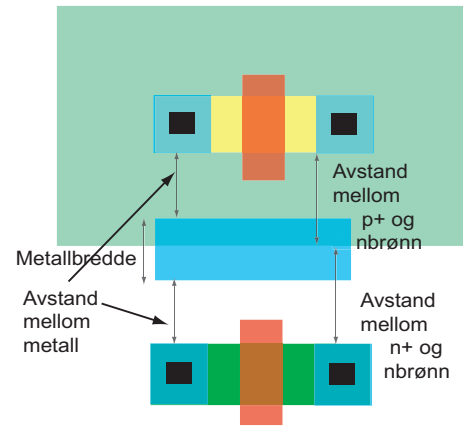


Figure 3.15: Utleggsregler for diffusjon og brønn. (FIG1.45b)

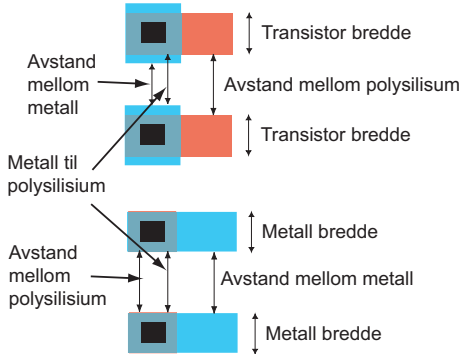


Figure 3.14: Utleggsregler for polysilisium og metall med kontakter. (FIG1.44a)

$(A + B + C) \cdot D$ er vist i Fig. 3.17. Denne skiller seg fra 3inngangs NAND port ved spesielt to signifikante forhold. Det er en inngang mer som betyr at det blir en nMOS- og en pMOS transistor i tillegg. Dette får betydning for cellens lengde som økes med L_s . I tillegg må diffusjonsområdene mellom transistorene styrt av A og B kobles sammen med diffusjonsområdet mellom transistorene styrt av C og D. I praksis gjøres dette ved hjelp av metall som vil kunne komme i konflikt med metalleder for utgangen. Vi må derfor i dette tilfellet øke høyden med L_s slik at portens areal blir lik $30L_s^2$.

3.4.1 Mål

Kunne estimere arealbehov ved implementasjon av standard celle ved hjelp av forenklet stick diagram.

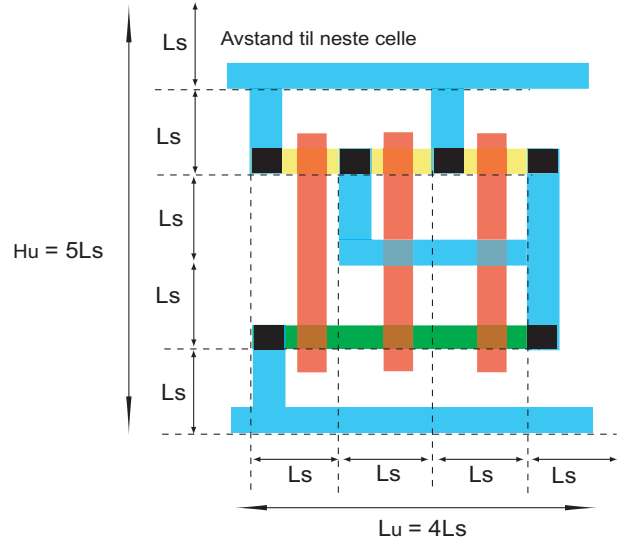


Figure 3.16: Estimert utleggsareal for 3inngangs NAND port. (FIG1.46)

3.5 Bakgrunn for CMOS teknologier og fremstilling av wafer

(Kapittel 3.2.1 og 3.2.2 side 113 - 115)

3.5.1 Bakgrunn for CMOS teknologier

Dagens CMOS teknologier kan deles inn i følgende kategorier:

- n-brønn prosess.
- p-brønn prosess.
- Twin-brønn prosess.
- Triple-brønn prosess.

Vi tar for oss vanlige CMOS prosesser og ikke såkalte BiCMOS eller SOI (Silicon On Insulator) prosesser.

De vanligste CMOS prosessene er n-brønn- eller twin-brønn prosesser. De opprinnelige CMOS prosessene var

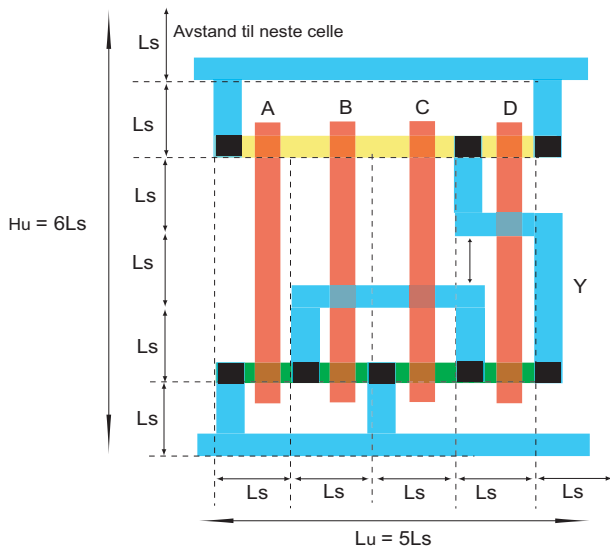


Figure 3.17: *Estimat for utleggsareal for Y = (A + B + C) · D. (FIG1.47)*

typisk p-brønn prosesser. Svært moderne CMOS prosesser er basert på triple-brønn prosesser.

3.5.2 Fremstilling av wafer

Råmaterialet ved fremstilling av moderne halvledere er skive (*wafer*) av silisium som har en diameter fra 75mm til 300mm og er ca. 1mm tykk. En-krystall silisium blir forenset (dopet) slik at de elektriske egenskapene til substratet møter en spesifikkasjon.

3.5.3 Mål

Kunne karakterisere vanlige CMOS prosesser på et enkelt nivå.

3.6 Fotolitografi

(Kapittel 3.2.3 side 115 - 116)

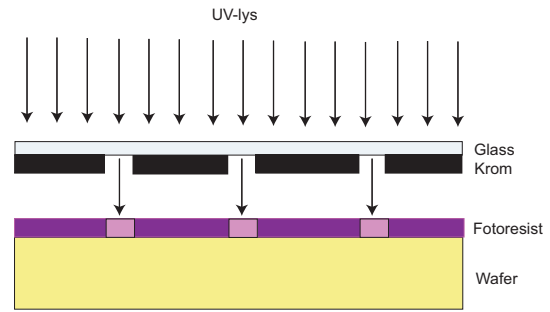


Figure 3.18: *Fotomaske for fremstilling av ulike områder på en wafer ved hjelp av fotolitografi (FIG3.1)*

Som kjent brukes masker til å definere områder der diffusjon, polysilisium, metall og kontakter skal være. Maskene genereres fra utleggsmaskene spesifisert av en utleggsdesigner. Ved markeringen av disse områdene legges først et *dielektrikum*, som er tykkoksid (SiO_2) på hele waferen. I neste omgang legges et lag av fotoresist og tilslutt en maske som brukes til å blottlegge områder som skal defineres. Masken består av glass som vil fungere som et UV stop filter. Vi utsetter waferen for UV lys slik at de blottlagte områdene blir påvirket av UV lys og strukturen i fotoresisten endres. En fremkallingvæske brukes til å fjerne fotoresist som har fått endret struktur. Denne prosessen minner svært mye om fremkalling av fotografier.

Bølgelengden på lys som blir brukt har betydning for minstestørrelser som kan lages. På 80 tallet ble lys med bølgelengde 436nm eller 365nm brukt. For en 0.25μ prosess brukes laser med bølgelengde lik 248nm (dyp ultrafiolett), som også har blitt brukt for prosesser ned til 180nm. I fremtiden vil ekstrem ultrafiolett lys med bølgelengde ned til 13.4nm brukes, men foreløpig medfører dette for store kostnader.

3.6.1 Mål

Forstå hvordan forskjellige lag fremstilles ved hjelp av fotolitografi.

3.7 Fremstilling av brønn og kanal

(Kapittel 3.2.4 side 117 - 118)

Ulike grader av donor eller akseptor doping kan oppnås ved å bruke *epitaksi*, *avsetning* (deposition) eller *implantering*.

Epitaksi innebærer groing av en en-krystall film på en silisium skive ved å la silisium waferen bli utsatt for et dopemateriale ved høy temperatur.

Avsetning betyr plassering av dopet materiale på silisium og deretter tvinge materialet inn i substratet ved hjelp av termisk diffusjon. Denne teknikken brukes for å lage dype diffusjonsområder.

Ioneimplantering medfører å utsette silisium substrat for donor eller akseptor atomer med meget høy energi. Når disse atomene treffer overflaten av silisiumskiven vil de trenge inn i substratet. Ioneimplantering brukes som standard metode for fremstilling av brønner og source/drain områder.

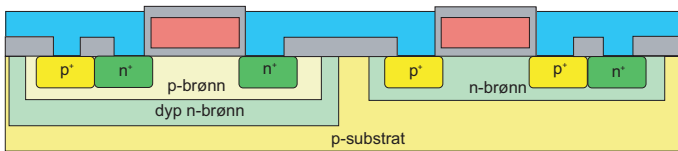


Figure 3.19: Brønnstruktur i triple-brønn prosess (FIG3.2)

Det første steget i de fleste CMOS prosesser innebærer definisjon av brønnområder ved hjelp av ioneimplantering. En inverter i en triple-brønn prosess er vist i Fig. 3.19.

3.7.1 Mål

Forstå hvordan brønn og kanal fremstilles i en CMOS prosess.

3.8 Silisium dioksid

(Kapittel 3.2.5 side 118 - 119)

Mange av strukturene som skal fremstilles og fremstillingsprosesser som brukes er avhengig av egenskapene til *silisiumdioksid* (SiO_2). Derfor er fremstilling av silisiumdioksid svært viktig. Det er behov for ulik tykkelse på silisiumdioksid avhengig av den spesifikke prosessen. Tynnoksid brukes under gaten på transistorer, mens tykkoksid brukes til generell isolering i en krets.

Oksidering av silisium gjøres ved å varme opp en silisium wafer i en oksideringsatmosfære. Vanlige metoder er:

1. *Våtoksidering*. Oksideringsatmosfæren inneholder vandamp. Temperaturen er vanligvis mellom $900^{\circ}C$ og $1000^{\circ}C$. Våtoksidering er en rask prosess.
2. *Tørroksidering*. Oksideringsatmosfæren inneholder ren oksygen. Temperaturen er omtrent $1200^{\circ}C$ for å oppnå en akseptabel oksidering. Tørroksidering er en mer presis metode en våtoksidering og brukes til fremstilling av tynnoksid.
3. *Avsetning av atomlag (ALD)*. Et tynt kjemisk lag avsettes på en overflate, dernest anvendes et kjemikalie slik at det fremstilles et tynt lag med silisiumdioksid. Prosessen gjentas slik at lag på lag dannes. Dette er en ny metode som antas å bli mer brukt i fremtiden.

Oksideringsprosesser forbruker vanligvis silisium (unntak er ALD).

3.8.1 Mål

Forstå hvordan silisiumdioksid avsettes på silisium.

3.9 Isolering

(Kapittel 3.2.6 side 119 - 120)

3.9.1 Mål

Forstå poenget med isolering og hvordan isolering anvendes i en CMOS prosess.

Et viktig kriterium for at CMOS skal fungere er at komponenter, dvs. transistorer, plasseres tilstrekkelig langt fra hverandre slik at de ikke påvirker hverandre utilsiktet. Det er også viktig å forhindre uønskede, eller parasittiske kanaler. Dette gjøres vanligvis ved hjelp av tykt oksid eller enda tykkere oksid som kalles *feltoksid*. I tillegg til feltoksid bruker man implantat for å forhindre uønskede kanaler eller transistorer, slike diffusjonsområder kalles *kanal stop diffusjon*.

Historisk ble lokal oksidering av silisium, *LOCOS*, brukt for å produsere varierende oksidtykkelse. Et problem med LOCOS var at overganger mellom tykt oksid og tynt oksid, som fikk en betydelig lateral (horisontal) utstrekning, ikke tillot høy tetthet av transistor.

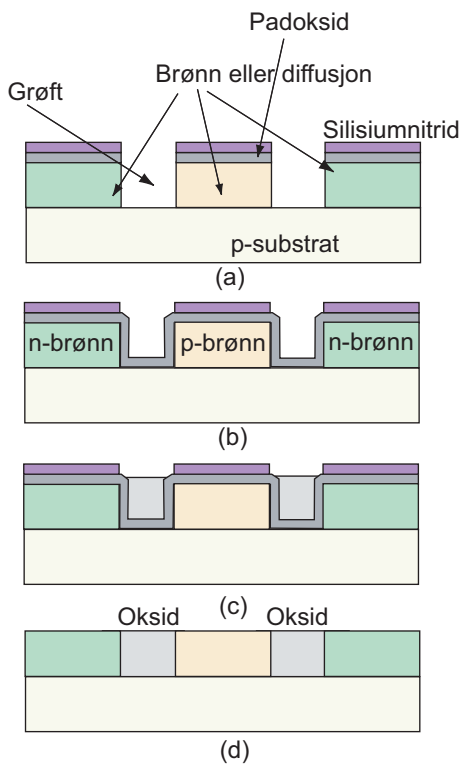


Figure 3.20: *Shallow trench isolations (FIG3.3)*

Isoleringsmetoder som anvendes for teknologier under 180nm består av å lage en grøfter av silisiumdioksid som omringer aktive områder. Typiske grøfter i en 90nm prosess vil være 140nm brede og 400nm dype. Slike grøfter kalles *shallow trench isolation* som vist i Fig. 3.20. I tillegg kan det være dypere grøfter.

Fra en designers side kan isoleringsgrøfter brukes til å isolere støyfølsomme kretser, for eksempel analoge delkretser, fra støygenerende strukturer, for eksempel svitsjende porter.

3.10 Gate oksid

(Kapittel 3.2.7 side 120 - 121)

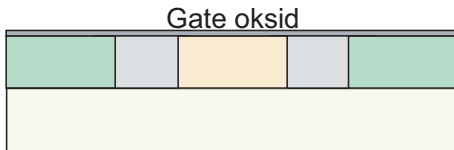


Figure 3.21: Fremstilling av gate oksid. (FIG3.4)

Neste steg i prosesseringen er fremstilling av gate oksid. Fra utgangspunktet i Fig. 3.20 (d) kan det legges et tynt lag med silisiumdioksid som vist i Fig. 3.21. Moderne prosesser har minst to ulike tykkelser på tynnoksid. Svært tynt oksid brukes under gaten på transistor, og noe tykkere oksid brukes for såkalte I/O transistorer, dvs. transistorer som er koblet til eksterne innganger og utganger på en chip. Noen prosesser tillater ulike tykkelser for tynnoksid for transistorer som inngår i logikk.

3.10.1 Mål

Forstå hvordan gate oksid fremstilles.

3.11 Fremstilling av source og drain

(Kapittel 3.2.8 side 121 - 123)

Historisk ble MOS (Metal On Semiconductor) laget ved at source og drain diffusjon ble fremstilt først og deretter gate i form av et metallag.

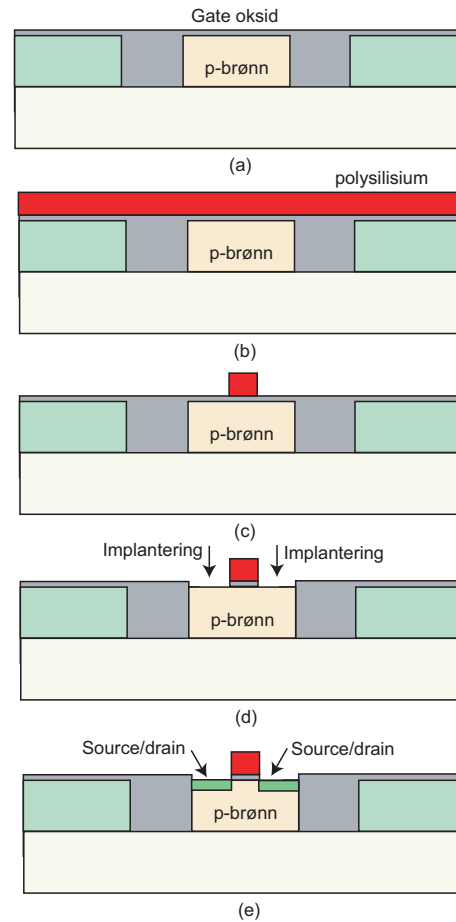


Figure 3.22: Fremstilling av source og drain. (FIG3.5)

Når silisium avsettes på silisiumdioksid uten bestemt orientering av krystallstrukturen, dannes polycrystalline silisium som kalles *poly* eller polysilium. Polysilium kan brukes for en mer presis fremstilling av transistorer enn ved bruk av metall for gaten. Fremstilling av poly for transistorer kalles *self-aligned polysilicon gate prosess* og følger noen grunnleggende steg:

1. Groing av gate oksid der transistoren skal ligge. Fig. 3.22 (a).
2. Avsetning av polysilium på wafer. Fig. 3.22 (b).
3. Utforming av polysilium. Fig. 3.22 (c).
4. Etsing av gate oksid, dvs. de områder som ikke er dekket av polysilium. Fig. 3.22 (d).

5. Implantering av pMOS og nMOS source/drain områder. Fig. 3.22 (e).

3.11.1 Mål

Forstå hvordan source og drain fremstilles, herunder lett dopet drain (LDD).

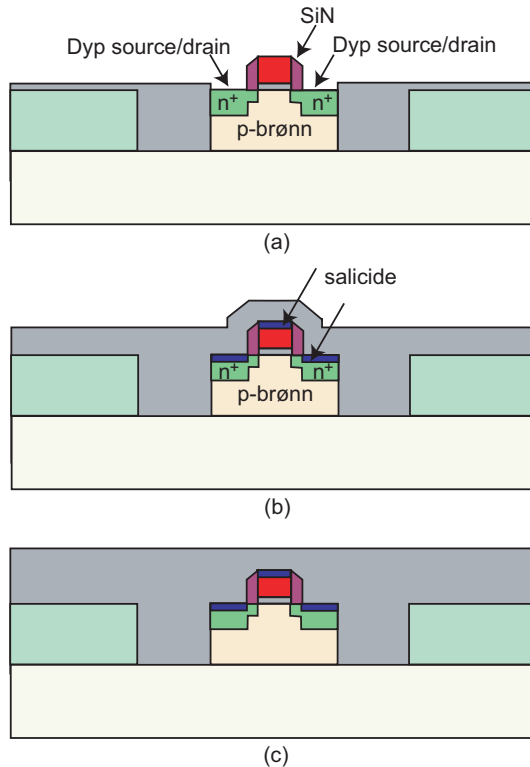


Figure 3.23: Svakt dopet Drain (LDD). (FIG3.6)

Konsentrasjonen av ladningsbærere i source/drain er relativt lavt, typisk i området $10^{18} - 10^{20} \text{ cm}^{-3}$. Vi kaller et slikt område for *lett dopet drain (LDD)*, en slik struktur reduserer det elektriske feltet i transistoren. Et lett dopet område representerer liten kapasitans, men stor motstand. Dette betyr en redusert ytelse for en transistor. Det er derfor behov for et kraftigere dopet source/drain implantat. Et *silisium nitrid* (Si_3N_4) materiale plassert langs kanten på gater benyttes for å markere områder hvor det skal avsettes dypere og kraftige dopet source/drain områder som vist i Fig. 3.23.

Som nevnt har polysilisium og LDD strukturer en viss motstand. Moderne prosesser anvender derfor et overflatemateriale for å redusere motstanden som vist i Fig. 3.23. Dette materialet er metall med høyt smeltepunkt som ikke vil bli påvirket av senere prosesssteg.

Polysilisium over diffusjon vil vanligvis gi en transistor. Utgangen på en port vil vanligvis være knyttet til en drain diffusjon. Dersom utgangen skal rett inn som inngang til en ny transistor vil det være hensiktsmessig å kunne koble diffusjon rett til polysilisium. I eldre prosesser må dette gjøres via et metallag. Moderne prosesser tillater en direkte kontakt mellom diffusjon og polysilisium. Dette kalles *lokal interkonnekt*.

3.12 Kontakter, metall og passivering

(Kapittel 3.2.9 og 3.2.10 side 124)

3.12.1 Kontakter og metall

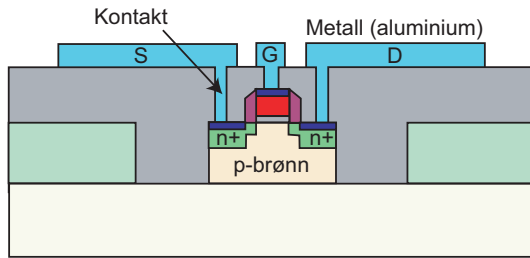


Figure 3.24: Aluminium metall og kontakter. (FIG3.7)

Vi trenger kontakter til source, drain og gate på alle transistorer. En kontakt er bare en etsning, dvs. fjerning, av tykkoksid ned til source, drain eller gate (eller til et annet metallag) slik at man kan få en fysisk kontakt med typisk en leder (metall) og det området man ønsker å koble seg til som vist i Fig. 3.24. Aluminium er ofte brukt for signalføring.

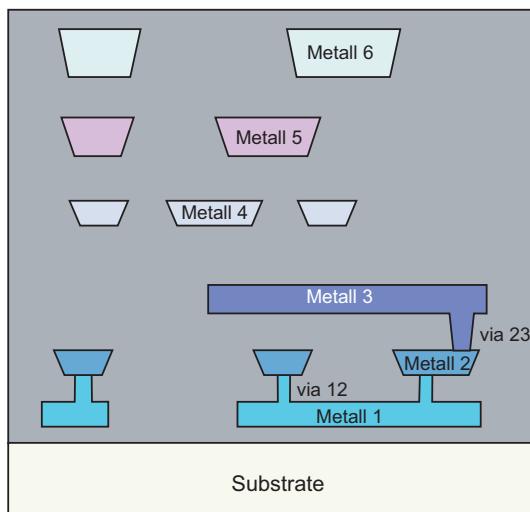


Figure 3.25: Tversnitt med metallag. (FIG3.8)

I Fig. 3.25 er det vist et tversnitt av metallag i en seks metall CMOS prosess. Kontakter mellom metallag kalles via, for eksempel kontakt mellom metall 2 og metall 3 kalles via 12 osv.

3.12.2 Passivering

Siste steg ved fremsilling av en krets er å legge på et lag av beskyttende glass. Dette kalles *passivering* eller *overglass*. Åpninger i passivering tillater kobling til innganger og utganger, I/O, til en krets (chip).

3.12.3 Mål

Forstå hvordan kontakter og passivering fremstilles.

3.13 Bakgrunn for utleggsregler

(Kapittel 3.3.1 og 3.2.10 side 125 - 129)

Formålet med utleggsregler er å kunne bygge funksjonelle kretser på et så lite areal som mulig. Reglene vil alltid representere et kompromiss mellom ytelse og *yield*. Konservative regler vil øke sannsynligheten for at kretsen virker (*yield*), mens aggressive regler vil medføre økt ytelse.

Det er viktig å være klar over at man ikke har noen garanti for at en krets virker etter spesifikasjon selv om man følger utleggsregler, men det vil øke sannsynligheten for at kretsen virker.

3.13.1 Regler for brønner

Brønner er ofte en et dypere implantat en source/drain områder til transistorer. Derfor er det viktig at strukturer som ligger i nærheten har tilstrekkelig avstand til kanten på en brønn.

3.13.2 Regler for transistorer

CMOS transistorer er generelt definert av minst fire fysiske lag. Dette er *aktiv* som også kalles diffusjon, *n-select* som også kalles n-implantat, *p-select* som også kalles p-implantat og polysilisium. Aktiv maske definerer alle områder med n- eller p-type diffusjon, og plasseringen av gate. Transisorenes gate defineres av en logisk AND funksjon mellom maske for polysilisium og maske for aktive. Select masken definerer hvilken type diffusjon som ønskes. n-select definerer områder der n-type diffusjon ønskes og p-select definerer områder for p-type diffusjon. Noen prosesser har bare n- og p-diffusjon (ndiff og pdiff) for å redusere kompleksiteten i prosessen.

3.13.3 Regler for kontakter

Det er en rekke kontakttypen tilgjengelig i de fleste prosesser:

- Metall til p-aktiv (p-diffusjon).
- Metall til n-aktiv (n-diffusjon).
- Metall til polysilisium.
- Metall til brønn eller substrat.
- Metall til metall (vias).

Metall til substrat og metall til n-aktiv kontakter er vist i figure 3.26. Det er god design praksis å inkludere så mange kontakter som det er plass til. Dette gir best elektrisk kontakt mellom to lag.

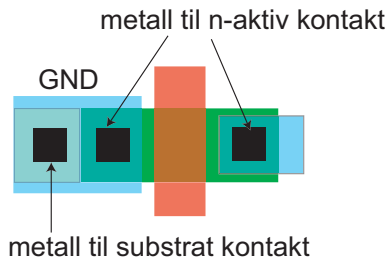


Figure 3.26: Metall til substrat kontakt og metall til n-aktiv kontakter for en nMOS transistor. (FIG3.10)

3.13.4 Regler for metall

Avstand mellom metallag og mellom metalledere i samme lag vil variere fra prosess til prosess. Det vil være regler som setter grenser for både minimum bredde og maksimum bredde på metalledere. Dersom man ønsker bredere metalledere enn prosessen tillater må man lage parallelle metalledere.

3.13.5 Mål

Kunne redegjøre for grunnleggende regler i en generell CMOS prosess.

3.14 Enkle MOS kapasitans modeller

(Kapittel 2.3.1 side 75 - 77)

Fra [?] har vi en enkel modell for *gatekapasitans*:

$$C_g = C_{ox}WL, \quad (3.1)$$

der oksidkapasitans $C_{ox} = \epsilon_{ox}/t_{ox}$.

Som kjent vil transistorkanalen ikke alltid strekke seg helt til drain siden av transistoren. Dette betyr at den enkle gate kapasitans modellen, der kanalens areal antas å være lik WL (dvs. strekker seg helt til drain), ikke er særlig nøyaktig. Det er vanlig å tilpasse modellen for gatekapasitans ved å bruke følgende modell

$$C_g = C_{permicron}W, \quad (3.2)$$

der

$$C_{permicron} = C_{ox}L. \quad (3.3)$$

Det vi nå har er gjort er å dele opp kapasitansen i to deler, der vi har en del som er en funksjon av den effektive kanallengden L .

I Fig. 3.27 er det vist ulike geometrier for diffusjonsområder. Utlegget viser tre ulike versjoner av to nMOS transistorer i serie. Som vi ser av figuren er det stor forskjell i areal og omkrets på diffusjonsområdet som deles

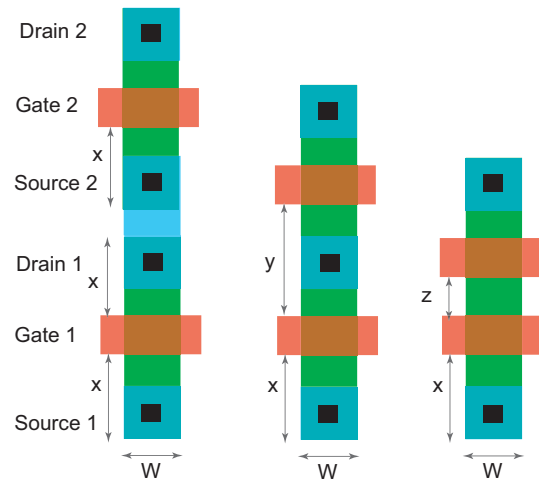


Figure 3.27: Ulike geometrier for diffusjonsområder. (FIG2.9)

av drain på transistor 1 (Drain 1) og source på transistor 2 (Source 2). Det er ingen logisk forskjell på funksjonen som dette utlegget representerer. I versjonen lengst til venstre har de to transistorene hvert sitt diffusjonsområde som er koblet sammen ved hjelp av metall og to kontakter. I dette tilfellet blir arealet og omkrets på diffusjonen mellom transistor 1 og 2 gitt av summen av de to områdene. I versjonen i midten deler transistor 1 og 2 samme diffusjonsområde med en kontakt i midten. I versjonen til høyre deler transistorene også diffusjonsområdet mellom transistorene, og her er arealet og omkrets på diffusjonsområdet redusert til et minimum ved at kontakten er fjernet og gatene er flyttet så nær hverandre som mulig. Minimumsavstand mellom gatene er bestemt av utleggsregler for en bestemt prosess.

3.14.1 Mål

Modellere gatekapasitans med enkle modeller.

3.15 Gatekapasitans detaljer

(Kapittel 2.3.2 side 77 - 80)

* Forutsetter kjennskap til enkel MOS transistor modell[?]. (INF3400 Del 2: , Kapittel 2.2 side 71 - 75)

For å beregne riktig verdi på gatekapasitanser må vi ta hensyn til transistorenes operasjonsområde. Den enkle modellen for gate kapasitans tar ikke hensyn til at kanalen (substratet) rett under gaten endres avhengig av transistorens operasjonsområde. Vi kaller den *intrinsikk ubiasert gatekapasitans* C_0 , som er gitt av

$$C_0 = WLC_{ox}. \quad (3.4)$$

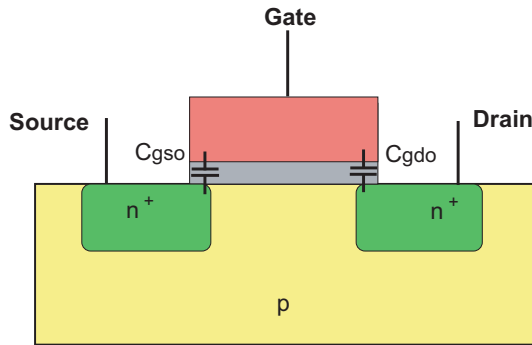


Figure 3.28: Overlappskapasitanser. (FIG2.10)

Det er naturlig å først se på en statisk kapasitans som ikke er avhengig av terminalspenninger på transistoren. Ser vi nærmere på transistoren vist i Fig. 3.28 ser vi (noe overdrevet på figuren) at source og drain regionene ligger litt under gaten. Dette er nødvendig for å sikre at transistoren vil fungere. Dette overlappet mellom gate og source, og gate og drain kommer av *lateral diffusjon*. Overlappet gir opphav til to kapasitanskomponenter som kalles *overlappskapasitanser* C_{gs0} og C_{gd0} :

$$\begin{aligned} C_{gs0} &= C_{gsol}W \\ C_{gd0} &= C_{gdol}W, \end{aligned}$$

der C_{gsol} og C_{gdol} er prosessparametre for overlappskapasitanser per mikrometer, typiske verdier er $0.2 - 0.4 fF/\mu m$.

Det er vanlig å dele opp gatekapasitans i ulike komponenter fordi substratet under kanalen varierer; gate til bulk kapasitans C_{gb} når transistoren er AV, gate til source kapasitans C_{gs} når vi har kanal på source siden av kanalen og gate til drain kapasitans C_{gd} når vi har kanal på drain siden av transistoren. Vi ser nærmere på transistorens operasjonsområder:

1. **AV.** Transistoren er AV som medfører at det ikke er kanal under gaten. Vi kaller kapasitansen mellom

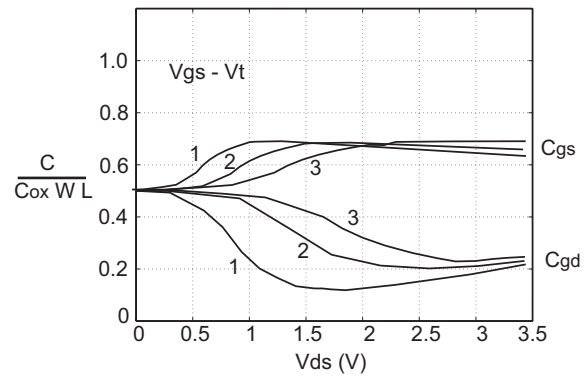


Figure 3.29: Gate source- og gate drain kapasitanser. (FIG2.11)

gate og substrat (bulk) *gate bulk kapasitans* C_{gb} . Når gate spenningen økes til rett under terskelspenningen får vi en depleksjonsone under gaten som medfører at gate bulk kapasitansen reduseres¹. Det er vanlig å modellere gate bulk kapasitansen som $C_{gb} = C_0$.

2. **Lineært område.** I lineært område har vi kanal som strekker seg fra source til drain. Vi fordeler da kapasitansen sett fra gate til kanal ved source og drain. Vi kaller disse kapasitansene *gate source kapasitans* C_{gs} og *gate drain kapasitans* C_{gd} . Vi har da $C_{gs} = C_{gd} = C_0/2$. Dersom vi øker drain spenningen vil kanalen på drain siden reduseres, noe som medfører at gate drain kapasitansen reduseres.
3. **Metning.** I metning har vi kanal bare ved source slik at gate drain kapasitansen er nær 0. I tillegg vil gate source kapasitansen øke noe. Vi har $C_{gd} = 0$ og $C_{gs} = 2C_0/3$.

I Fig. 3.29 er C_{gs} og C_{gd} vist som funksjon av drain source spenning for ulike gate source spenninger. Som vi ser er kapasitansene avhengig av drain source spenningen.

Parameter	AV	LINEÆR	METNING
C_{gb}	C_0	0	0
C_{gs}	0	$C_0/2$	$2C_0/3$
C_{gd}	0	$C_0/2$	0
$C_g = C_{gb} + C_{gs} + C_{gd}$	C_0	C_0	$2C_0/3$

Table 3.1: Gatekapasitans i ulike operasjonsområder for transistor (Tabell 2.12 side 78).

I tabell 3.1 vises forenklet modeller for gatekapasitans i ulike operasjonsområder for transistoren.

Det vil i tillegg være stor variasjon på gatekapasitans for ulike spenninger og transisjoner på terminalene. I Fig. 3.30 vises gatekapasitans for en nMOS transistor som har

¹Vi får da to kapasitanser i serie, dvs. en kapasitans mellom gate og toppen av substratet rett under gaten og en kapasitans fra toppen av substratet og over depleksjonsonen ned i substratet.

en positiv inngangstransisjon (slås på) i ulike situasjoner. I eksempel **3** der både source og drain er 0 vil gatekapasitansen være lik C_0 . Dersom source og/eller drain får en lik transisjon som inngangen vil gatekapasitansen bli redusert. I motsatt tilfeller, dvs. der transisjonene på source og/eller drain går i motsatt retning vil gatekapasitansen bli økt. Som vi ser er det betydelig variasjon på gatekapasitansen.

3.15.1 Mål

Modellere gatekapasitans, gate source kapasitans og gate drain kapasitans. Forstå hvordan transistorenes terminalspenninger påvirker kapasitansene knyttet til transistorene.

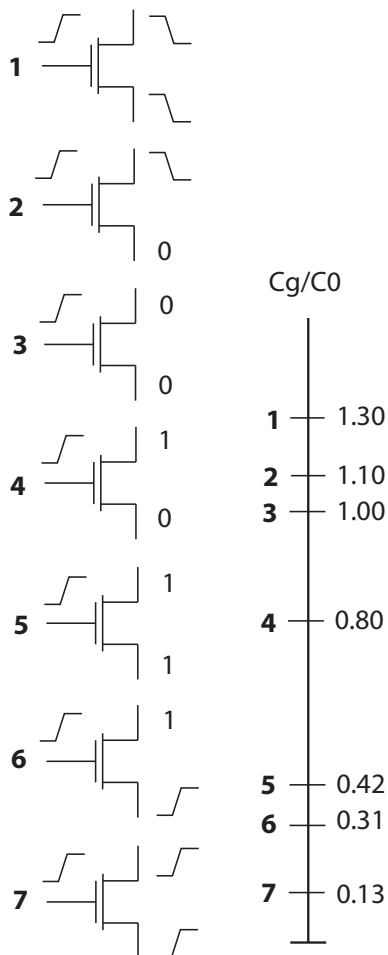


Figure 3.30: Gate kapasitans for ulike signalnivåer og signaltransisjoner på source og drain. (FIG2.12)

3.16 Diffusjonskapasitans detaljer

(Kapittel 2.3.3 side 80 - 83)

* Antar kjennskap deplesjonsutstrekning i reversforspenning *pn* overganger (dioder) som er pensum i INF1410/FYS1210.

Som kjent vil *pn* overgangen mellom et diffusjonsområde (sterkt dopet silisium) og substrat danne en deplesjonsone som vil fungere som en kapasitans som vil være knyttet til diffusjonsområdet. En slik kapasitans kalles *diffusjonskapasitans*. Dette vil i praksis ha betydning for drain/source områder på transistorer. Det er bare elektriske noder som endrer spenning som vil merke en slik kapasitans, dvs. diffusjonsområder knyttet til spenningsreferansene V_{DD} og GND vil ikke bidra med kapasitans i en logisk port som skifter signalnivå.

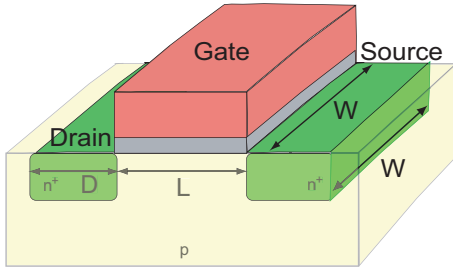


Figure 3.31: Geometrier for diffusjonsområder. (FIG2.13)

Som vist i Fig. 3.31 vil diffusjonsområdene knyttet til en transistor bestå av source og drain. Bredden på transistoren vil bestemme bredden på diffusjonsområdet og utstrekningen av diffusjonsområdene vil være avhengig av prosess og design som for eksempel vist i Fig. 3.27. Vi kan kalle diffusjonsområdets utstrekning for D som vist i Fig. 3.31 og har da modellen for *diffusjonskapasitans på source*:

$$C_{sb} = AS \cdot C_{jbs} + PS \cdot C_{jbsw}, \quad (3.5)$$

der diffusjonsområdets areal $AS = WD$, diffusjonsområdets omkrets $PS = 2W + 2D$, C_{jbs} er en prosessavhengig parameter i kapasitans/areal og C_{jbsw} også er prosessavhengig og oppgitt i kapasitans/lengde.

Kapasitansen er avhengig av deplesjonsdybden og derfor avhengig av reversforspenningen:

$$C_{jbs} = C_j \left(1 + \frac{V_{sb}}{\Psi_0}\right)^{-M_J}, \quad (3.6)$$

der C_j er deplesjonskapasitans uten reversforspenning ($V_{sb} = 0$), M_J er deplesjonskoeffisient (“junction grading coefficient”) og Ψ_0 er innebygd potensiale (“built-in potential”). C_j er en empirisk verdi gitt for spesifikke prosesser og M_J varierer fra 0.5 til 0.33 avhengig av hvor brå *pn* overgangen er. Det *innebygde potensialet* er gitt av:

$$\Psi_0 = V_T \ln \frac{N_A N_D}{n_i^2}, \quad (3.7)$$

der V_T er *termisk spenning*, N_A er *dopekonsentrasjonen i ptype silisium*, N_D er *dopekonsentrasjonen i ntype silisium* og n_i er konsentrasjonen av frie ladningsbærere i intrinsikk eller udopet silisium. Termisk spenning er som navnet tilsier avhengig av temperatur; $V_T = kT/q$, der k er *boltzmanns konstant*, T er temperatur i Kelvin og q er ladning til en ladningsbærer. For en nMOS transistor er ladningsbærere elektroner og konsentrasjonen av frie ladningsbærere i ntype silisium vil være lik dopekonsentrasjonen N_D . Konsentrasjonen av frie ladningsbærere (hull) i p substratet vil være lik N_A .

Kapasitansbidraget fra sideveggene i diffusjonsområder uttrykkes på tilsvarende måte:

$$C_{jbsw} = C_{JSW} \left(1 + \frac{V_{sb}}{\Psi_0}\right)^{-M_{JSW}}, \quad (3.8)$$

der C_{JSW} og M_{JSW} er henholdsvis deplesjonskapasitans for sideveggene uten forspenning og deplesjonskoeffisient for diffusjonsområdets sidevegger.

På tilsvarende måte kan diffusjonskapasitans for drain områder uttrykkes, der AD erstatter AS , PD erstatter PS og V_{bd} erstatter V_{sb} .

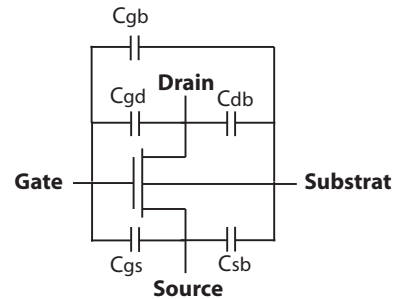


Figure 3.32: Kapasitanser knyttet til en nMOS transistor. (FIG2.14)

En oppsummering av kapasitansene knyttet til en nMOS transistor er vist i Fig. 3.32.

3.16.1 Mål

Modellere diffusjonskapasitanser og forstå hvordan kapasitansene er avhengig av terminalspenninger og utleggsgeometri.

3.17 Enkle RC modeller

(Kapittel 2.6 side 103 - 105)

Formålet med *RC modeller* er å utvikle enkle modeller for beregning av tidsforsinkelser i en port eller krets. Det er vanlig å uttrykke *tidsforsinkelse* på formen RC , der R er effektiv motstand og C er lastkapasitans. Som kjent kan en MOS transistor i lineært område modelleres (forenklet) som som en spenningsstyrt motstand:

$$\begin{aligned}
 R &= \left(\frac{\partial I_{ds}}{\partial V_{ds}} \right)^{-1} \\
 &\approx (\beta(V_{gs} - V_t))^{-1} \\
 &\approx \frac{1}{\mu C_{ox}} \frac{L}{W} (V_{gs} - V_t)^{-1}.
 \end{aligned}$$

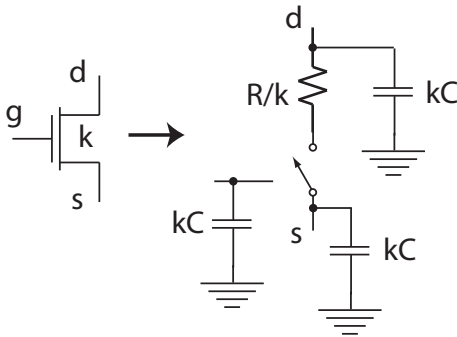


Figure 3.33: *Ekvivalent RC krets for en nMOS transistor med bredde lik k . (FIG2.34 øverst)*

I Fig. 3.33 er det vist en *RC ekvivalent* for en nMOS transistor med bredde lik k . Her er kapasitansene koblet mot *GND* fordi substratet som regel er koblet til *GND*. Spenningsreferansen har ingen betydning for *RC ekvivalent*kretsen fordi det antas at både *GND* og V_{DD} ligger på faste potensialer, dvs. spenningene endres ikke.

Vi antar at en enhetstransistor, dvs. med bredde lik 1 og minimum lengde, har motstand lik R , gatekapasitans lik C og diffusjonskapasitans lik C .

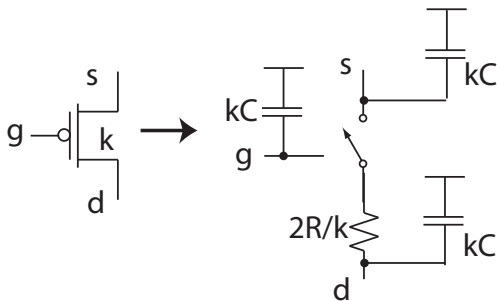


Figure 3.34: *Ekvivalent RC krets for en pMOS transistor med bredde lik k . (FIG2.34 nederst)*

I Fig. 3.34 er det vist *RC ekvivalent* for en pMOS transistor. Vi modellerer motstanden som $2R/k$ som utgjør

dobbelt så stor motstand som for en like stor nMOS transistor. Dette kommer av mobilitetsforskjell mellom transistorene $\mu_n \approx 2\mu_p$. Det er verdt å bemerke at forskjell i mobilitet kan være vesentlig større og typisk større i moderne prosesser. Det er alltid slik at mobiliteten for nMOS transistorer er større enn mobiliteten i pMOS transistorer fordi elektroner er mer mobile enn hull. Her er kapasitansene koblet mot V_{DD} for å indikere at substratet er koblet til V_{DD} . Dette har ingen elektrisk betydning for *RC ekvivalenten*. Legg merke til at motstanden er plassert på drain siden tilsvarende som for nMOS transistoren i motsetning til [1]. Dette har heller ingen elektrisk betydning for *RC ekvivalenten* fordi motstanden fungerer som en motstand fra drain til source.

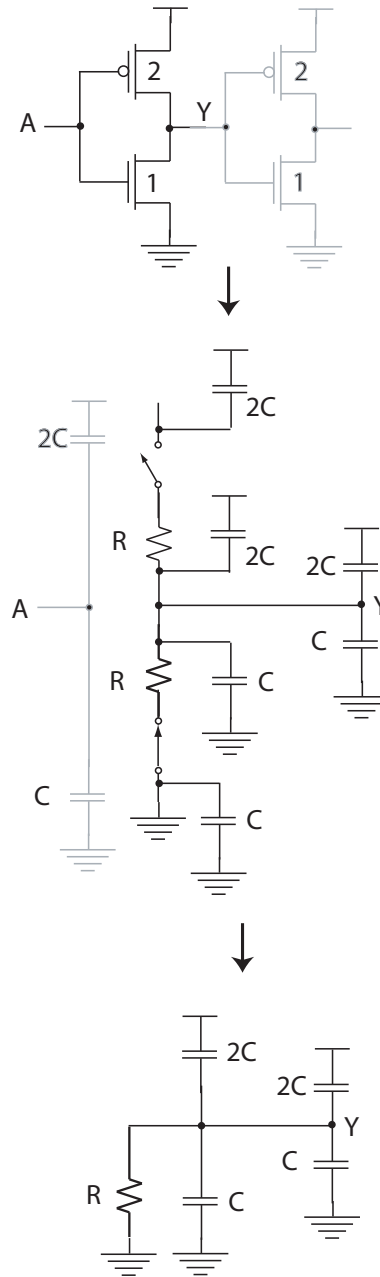


Figure 3.35: *Ekvivalent RC krets for en inverter. (FIG2.35)*

Vi benytter *RC ekvivalentene* til å beregne tids-

forsinkelse i en logisk port. Et eksempel er vist i Fig. 3.35 der en inverter har en annen inverter som last, dvs. skal drive en annen inverter. Vi antar at inngangen A er logisk 1 slik at nMOS transistoren er PÅ og pMOS transistoren er AV. Motstanden for pMOS transistoren vil ikke inngå i ekvivalenten som vist i den nederste figuren fordi det ikke går strøm gjennom pMOS transistoren. Husk at motstanden er drain source motstand. Vi kan modellere tidsforsinkelsen for inverteren ved $t_d = R \cdot (6C) = 6RC$. Vi har uttrykt alle kapasitanser og motstander i forhold til en enhetsinverter, dvs. til en inverter med enhets transistorer.

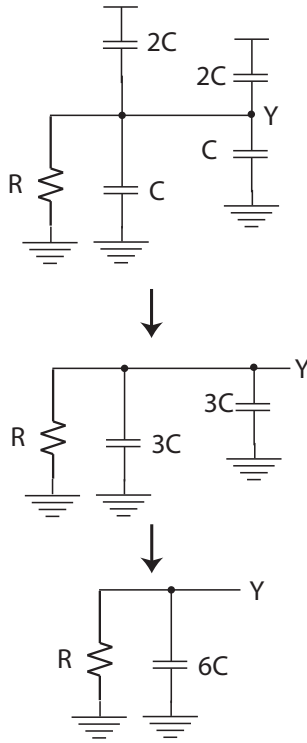


Figure 3.36: Forenklet ekvivalent RC krets for en inverter.

En forenklet RC ekvivalent krets av inverteren er vist i Fig. 3.36.

For passtransistorer vil ekvivalentmotstanden bli litt anderledes. I Fig. 3.37 er det vist en transmisjonsport der en nMOS transistorer som bidrar til å transmittere en 1er vil ha en økt motstand fra R til $2R$. Tilsvarende vil en pMOS transistorer som bidrar til å transmittere en 0er få økt motstand fra $2R$ til $4R$. Transmisjonsporten vil da ha to motstander i parallell der vi kan finne ekvivalent motstanden for transisjon fra 0 til 1 på inngangen ($a = 1$) $R = 2R || 2R = R$ og ekvivalent motstanden for transisjon fra 1 til 0 på inngangen som $R = R || 4R = (4/5)R$.

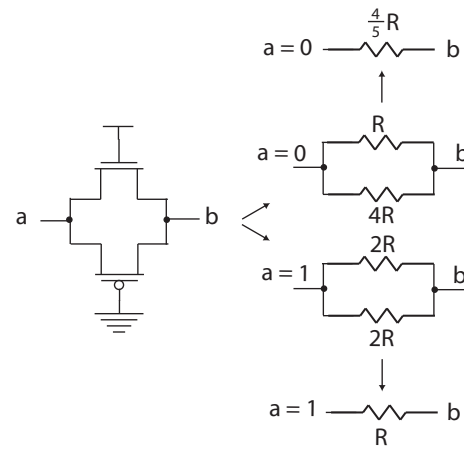


Figure 3.37: Ekvivalent RC krets for transmisjonsport. (FIG2.36)

3.18 RC forsinkelsesmodeller

(Kapittel 4.2 - 4.2.1 side 158 - 161)

Vi ser på modeller for tidsforsinkelse på portnivå.

3.18.1 Estimering av forsinkelse

(Kapittel 4.2 side 158 - 159)

Definisjoner for tidsforsinkelse:

- *Stigetid t_r* . Defineres som tiden det tar for et signal (utgang av port) å stige fra 20% til 80% av stabil verdi. Dette vil i praksis si fra $0.2V_{DD}$ til $0.8V_{DD}$.
- *Falltid t_f* . Defineres som tiden det tar for et signal (utgang av port) å falle fra 80% til 20% av stabil verdi. Dette vil i praksis si fra $0.8V_{DD}$ til $0.2V_{DD}$.
- *Gjennomsnittstid t_{rf}* . (Edge rate). Gjennomsnittet av stige- og falltid for en port.
- *Tidsforsinkelse t_{cd} eller t_d* (contamination delay). Minimum tid fra inngang krysser 50% til utgang krysser 50% av stabil verdi. Dette vil i praksis si $V_{DD}/2$.

Definisjoner av tidsforsinkelser er vist i Fig. 3.38.

3.18.2 RC forsinkelsesmodeller

(Kapittel 4.2.1 side 159 - 161)

Med utgangspunkt i den enkle RC modellen i avsnitt 3.17 skal vi utvikle RC forsinkelsesmodeller for mer komplekse porter.

Seriekobling av transistorer

Seriekobling av transistorer vil medføre en *effektiv motstand* eller *ekvivalent motstand* gjennom kjeden som

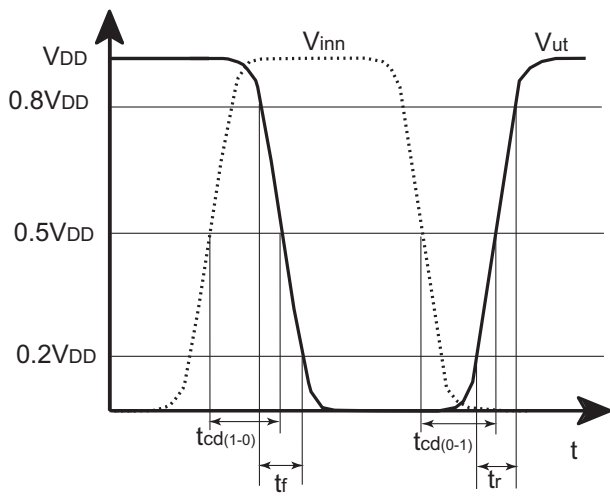


Figure 3.38: Definisjon av tidsforsinkelser.

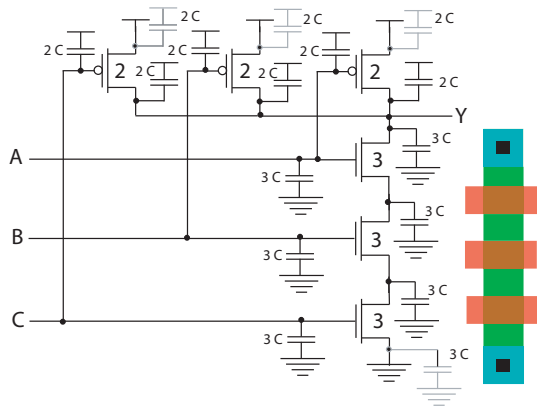


Figure 3.39: 3inngangs NAND port med kapasitanser.

$$R_{effektiv} = \sum_{i=1}^n \frac{R}{k_i}, \quad (3.9)$$

der k_i er bredden på i te transistor og n er antall transistorer i kjeden.

Parallellkobling av transistorer

To transistorer, med ekvivalent motstand R , i parallell som vi vet er PÅ vil modelleres som en ekvivalent motstand $R_{effektiv} = R||R = R/2$. Dette betyr at parallelle transistorer som er PÅ vil bidra til å redusere tidsforsinkelser. Det er imidlertid vanlig å beregne såkalt “worst case” tidsforsinkelse, som betyr at vi antar at kun en av de parallelle transistorene er PÅ. I praksis vil vi se bort fra parallelle transistorer ved beregning av effektiv motstand.

Eksempel

I Fig. 3.39 er det vist en 3inngangs NAND port med kapasitanser. Kapasitanser som er knyttet til spenningsreferansene

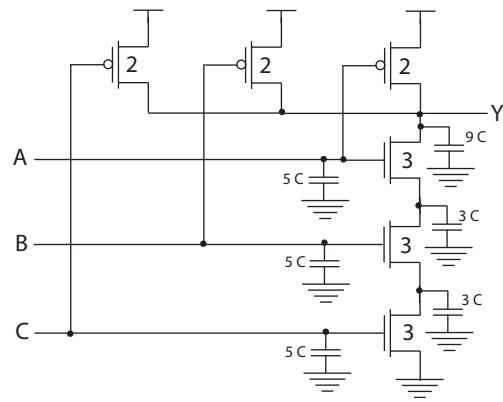


Figure 3.40: 3inngangs NAND port med forenklet kapasitansmodell.

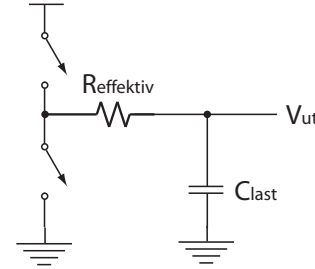


Figure 3.41: Enkel RC modell består av en effektiv motstand og en lastkapasitans.

ansene kan vi se bort ifra, det er markert med grått i figuren. nMOS transistorene i serie vil dele drain og source diffusjon som vist til venstre i figuren slik at det er naturlig å ikke ta med kapasitansbidrag fra to diffusjonsområder mellom serie koblede transistorer.

I Fig. 4.1 er det vist en forenklet kapasitans ekvivalent for 3inngangs NAND porten. Kapasitanser koblet til spenningsreferansene er fjernet og andre kapasitanser tilhørende samme elektriske node er slått sammen.

Ekvivalentmotstand for $0 \rightarrow 1$ transisjon vil være gitt av $R_{effektiv} = 2R/2 = R$ og ekvivalentmotstand for $1 \rightarrow 0$ transisjon vil være gitt av $R_{effektiv} = (1/3+1/3+1/3)R = R$. Vi har valgt transistorstørrelser, eller bredder, som vil gi lik motstand i “worst case” opptrekk og nedtrekk.

3.18.3 RC modell

En enkel RC modell vil bestå av en effektiv motstand $R_{effektiv}$ og en lastkapasitans C_{last} som vist Fig. 3.41. Lastkapasitansen vil være lik summen av alle kapasitanser i noder mellom en spenningsreferanse og utgangen med unntak av kapasitanser knyttet til selve spenningsreferansene.

3.18.4 Mål

Modellere lastkapasitans og ekvivalent motstand i komplekse logiske porter. Finne tidsforsinkelse i komplekse logiske porter.

3.19 Hastighetsmetning

(Kapittel 2.4.2 side 84 - 86)

* Forutsetter kjennskap til enkel MOS transistor modell[?].
(INF3400 Del 2: , Kapittel 2.2 side 71 - 75)

Transistorer som er mindre enn ca. $1\mu m$ vil oppføre seg noe anderledes enn enkle MOS transistor modeller tilsier. En av grunnene til dette er at det elektriske feltet over kanalen blir svært kraftig pga. kort kanallengde, vi sier da at transistoren går raskere i metning, eller mer presist i *hastighetsmetning*. Hastighetsmetning inntreffer når ladningsbærere får så stor energi, pga påvirkning fra et svært kraftig felt, at ladningsbærerne kolliderer forholdsvis ofte. Når dette skjer vil ikke en styrking av det elektriske feltet, ved økning i drain source spenning, bidra til å øke strømmen i transistoren. Hastighetsmetning inntreffer raskere, dvs for mindre drain source spenning, når kanallengden blir redusert.

Vi uttrykker hastigheten til ladningsbærere som:

$$\nu = \frac{\mu E_{lat}}{1 + \frac{E_{lat}}{E_{sat}}}, \quad (3.10)$$

der $E_{lat} = V_{ds}/L$ er lateralt elektrisk felt mellom source og drain², E_{sat} er feltstyrken når hastighetsmetning inntreffer og μ er mobiliteten til ladningsbærere. Vi modellerer transistorstrømmen når transistoren er i hastighetsmetning som:

$$I_{ds} = C_{ox} W (V_{gs} - V_t) V_{dsat}, \quad (3.11)$$

der V_{dsat} er *metningsspenningen for hastighetsmetning*. For enkelthets skyld kan man anta at transistorer som er korte, dvs. $L < 1\mu m$, vil bli hastighetsmettet ved tilstrekkelig høy drain source spenning.

For å få med effekten av at hastighetsmetningen, eller metningsspenningen for hastighetsmetning, er avhengig av transistorlengden utvider vi transistormodellen til:

$$\begin{aligned} I_{ds} &= 0 & V_{gs} < V_t & & AV, \\ I_{ds} &= P_c \frac{\beta}{2} (V_{gs} - V_t)^\alpha \frac{V_{ds}}{V_{dsat}} & LINEÆR & & V_{ds} < V_{dsat}, \\ I_{ds} &= P_c \frac{\beta}{2} (V_{gs} - V_t)^\alpha & METNING & & V_{ds} > V_{dsat}, \end{aligned}$$

der P_c og α er empiriske verdier avhengig av transistorlengde og metningsspenning V_{dsat} er gitt av:

$$V_{dsat} = P_v (V_{gs} - V_t)^{\frac{\alpha}{2}}, \quad (3.12)$$

der P_v er en empirisk verdi avhengig av transistorlengde.

²Lateralt elektrisk felt er det samme som elektrisk felt som beskrevet i [?].

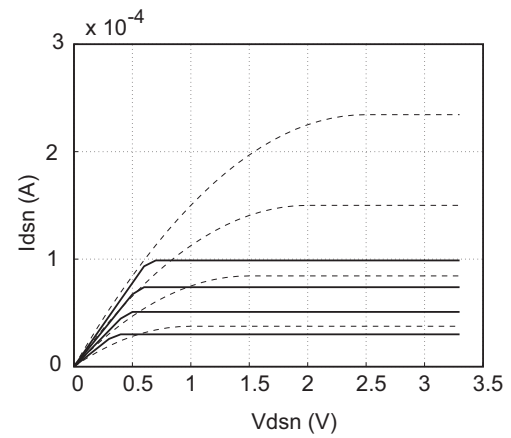


Figure 3.42: *I-V karakteristik for lang og kort transistor med enkle modeller. (FIG2.17)*

I Fig. 3.42 er det vist transistorstrømmer i to transistorer med samme W/L , men ulik lengde. Stiplet linje viser transistorstrøm i en transistor med $W/L = 2\mu m/2\mu m$ som er vist sammen med en kort transistor med $W/L = 0.25\mu m/0.25\mu m$ for samme gate source spenninger. Modellene som er benyttet i denne figuren er gitt av og enkle førsteordens modell uten kanallengde modulasjon.

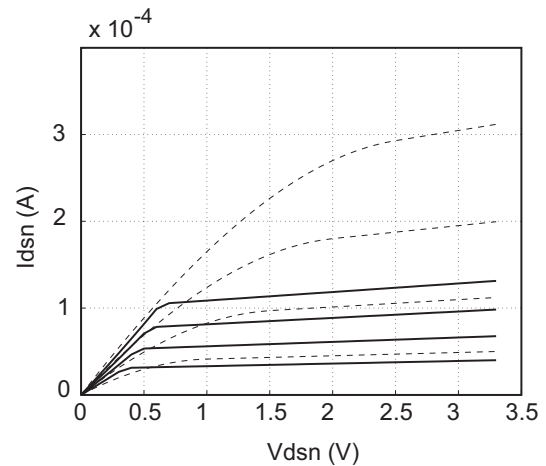


Figure 3.43: *I-V karakteristik for lang og kort transistor med kanallengdemodulasjon. (FIG2.17)*

Vi kan utvide modeller for transistorstrøm for korte transistorer ved å inkludere kanallengdemodulasjon. Ved hastighetsmetning vil kanalen ikke bli kortere, men det vil være en økning i strømmen for økende drain source spenning. Det å benytte samme modell for kanallengde modulasjon for en transistor i hastighetsmetning som i metning er ikke basert på samme fysiske forklaring, men gir en fornuftig modell. Modellen blir da

$$\begin{aligned} I_{ds} &= 0 & V_{gs} < V_t & & AV, \\ I_{ds} &= P_c \frac{\beta}{2} (V_{gs} - V_t)^\alpha \frac{V_{ds}}{V_{dsat}} (1 + \lambda V_{ds}) & LINEÆR & & V_{ds} < V_{dsat}, \\ I_{ds} &= P_c \frac{\beta}{2} (V_{gs} - V_t)^\alpha (1 + \lambda V_{ds}) & METNING & & V_{ds} > V_{dsat}, \end{aligned}$$

Transistorstrømmer for transistor i hastighetsmetning er vist sammen med en lengre transistor med samme forhold W/L i Fig. 3.43.

3.19.1 Mål

Forstå, og kunne modellere transistorer i hastighetsmetning.

Chapter 4

Kretskaracterisering

4.1 Elmore forsinkelsesmodell

(Kapittel 4.2.1.3 side 161 - 164)

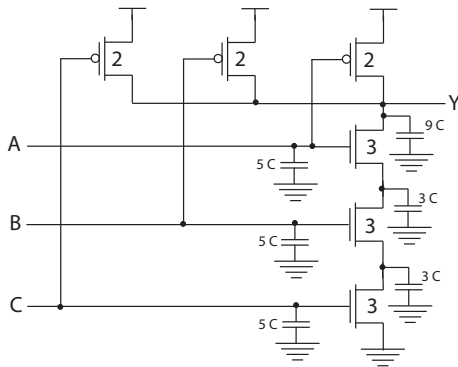


Figure 4.1: 3inngangs NAND port med kapasitanser. (Figur 4.2)

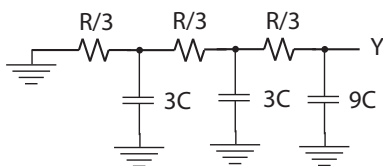


Figure 4.2: RC kjede for Elmore forsinkelse for 3inngangs NAND port.

Dersom vi ser på nMOS transistorene i 3inngangs NAND porten i Fig. 4.1 kan vi modellere nedtrekket som en kjede av nMOS transistorer (3) som vist i Fig. 4.2.

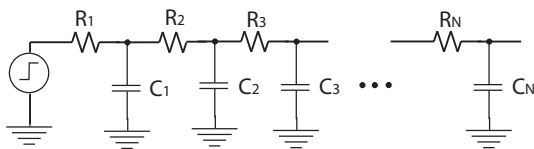


Figure 4.3: RC kjede for Elmore forsinkelse. (Figur 4.3)

Elmore forsinkelsesmodell modellerer tidsforsinkelsen på en detaljert måte. Vi snakker her om tidsforsinkelse uten ekstern last, dvs. tidsforsinkelse som bare er avhengig av interne- eller diffusjonskapasitanser i selve porten. En slik tidsforsinkelse kaller vi *parasittisk tidsforsinkelse*:

$$\begin{aligned} t_{pd} &= \sum_{i=1}^N C_i \sum_{j=1}^i R_j \\ &= \sum_{i=1}^N \left(C_i \sum_{j=1}^i R_j \right), \end{aligned} \quad (4.1)$$

der N er antall RC elementer i en RC kjede, som vist i Fig. 4.3. For 3inngangs NAND porten vil Elmore forsinkelsesmodell gi $t_{pd} = 3C \cdot (R/3) + 3C \cdot (R/3 + R/3) + 9C \cdot (R/3 + R/3 + R/3) = 12RC$. Benytter vi den enkle

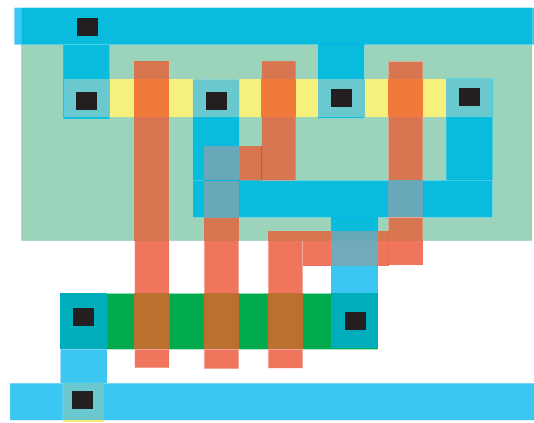


Figure 4.4: Utlegg av 3inngangs NAND port. (Figur 4.4)

RC modellen[?] får vi en tidsforsinkele $t_{pd} = 15RC$, som vil representere et mer konservativt (forsiktig) estimat på tidsforsinkelse. Et eksempel på utlegg av en 3inngangs NAND port er vist i Fig. 4.4.

4.1.1 Eksempler

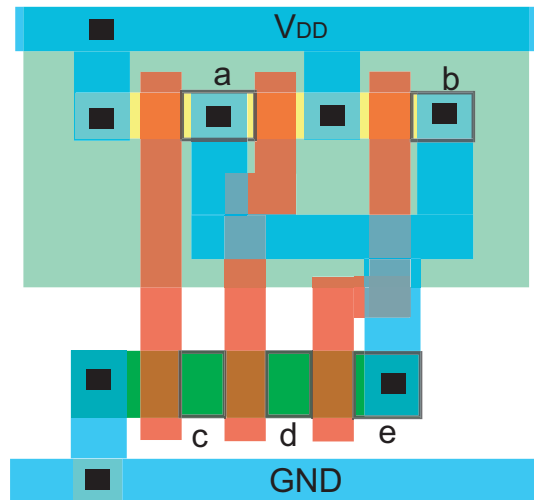


Figure 4.5: 3inngangs NAND port. (Figur 4.5)

Ser vi nærmere på 3inngangs NAND porten i Fig. 4.5 kan vi se at arealene på diffusjonsområdene er helt avhengig av hvordan utlegget ser ut. Det er god praksis å redusere diffusjonsarealene til et minimum slik at kapasitansen som diffusjonsområdene bidrar med reduseres mest mulig. Vi ser at diffusjonsarealene for pMOS transistorene som er knyttet til utgangen består av to arealer a og b , der a er delt av to pMOS transistorer. Disse arealene er omtrent like store og vil representere en kapasitans lik $k_p C$ der k_p er bredden til transistorene. I vårt eksempel er bredden lik 2 slik at diffusjonskapasitansen er $2C$. For nMOS transistorene i kjede ser vi at for å nå GND må alle tre nMOS transistorer være PÅ som betyr at alle diffusjonsområdene for nMOS transistorene, med unntak av diffusjonsområdet knyttet til GND, tas med i lastkapasitansen. Områdene c , d og e er i dette tilfellet omtrent

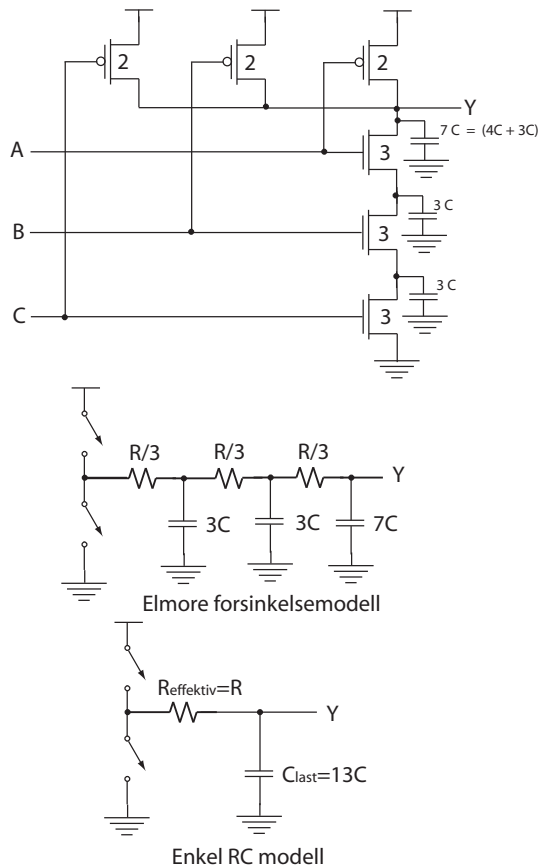


Figure 4.6: Elmore og enkel RC forsinkelsesmodeller for 3 inngangs NAND port. (Figur 4.5b)

like store, som betyr at hver diffusjonskapasitans blir lik $k_n C$ der k_n er bredden på nMOS transistorene. I vårt eksempel er bredden lik 3 slik at diffusjonskapasitansen er $3C$. Elmore forsinkelses modell og enkel RC modell for 3inngangs NAND port er vist i Fig. 4.6. Med enkel RC modell får vi $t_{pd} = 13RC$ mens Elmore forsinkelsesmodell gir $t_{pd} = 10RC$.

Det er vanlig å betrakte diffusjonskapasitans som *parasittisk kapasitans* som skiller seg fra gatekapasitanser fordi diffusjonskapasitans er utelukkende avhengig av portens eget utlegg og ikke porter som skal drives. Vi kaller gatekapasitans for *ekstern kapasitans* eller *ekstern last*, mens parasittisk kapasitans kalles *intern kapasitans* eller *intern last*¹.

4.1.2 Parasittisk tidsforsinkelse

Som nevnt skiller vi på gate- og diffusjonskapasitanser. Vi kaller diffusjonskapasitanser parasittiske kapasitanser og videre kaller vi den andelen av forsinkelsen som avhenger kun av diffusjonskapasitanser for *parasittisk tidsforsinkelse*. I eksemplet med 3inngangs NAND port vil tidsforsinkelsen som er beregnet være parasittisk tidsforsinkelse. Dersom vi antar at porten skal drive en tilsvarende port vil den eksterne kapasitansen være gitt av

¹Et annet uttrykk som brukes for intern kapasitans er diffusjonskapasitans.

$C_{ekstern} = 2C + 3C = 5C$ (dette er vist som inngangskapasitanser i Fig. 4.1). Tidsforsinkelsen vil da bli med enkel RC modell bli $t_{pd} = 18RC$, der parasittisk tidsforsinkelse utgjør $13RC$. Med Elmore forsinkelsesmodell blir tidsforsinkelsen $15RC$, der parasittisk tidsforsinkelse utgjør $10RC$.

I eksemplet med 3inngangs NAND port kan vi finne tidsforsinkeler når h er 4, der h er antall identiske porter som skal drives. Vi får da med den enkle RC modellen $t_{pd} = (13C + h5C)R = 33RC$ og med Elmore modellen får vi $t_{pd} = (1 + 2 + 27)CR = 30RC$. Merk at den relative forskjellen på de to modellene blir liten når den eksterne lasten er stor i forhold til den interne. Dersom vi antar at $1RC = 20ps$, vil dette medføre henholdsvis $660ps$ og $600ps$ med enkel RC modell og Elmore modell.

4.1.3 Elektrisk effort

Vi kan skille mellom inngangslast og ekstern last og definerer forholdet mellom lastene som $C_h = C_{ekstern}/C_{inngang}$, der $C_{ekstern}$ er ekstern kapasitans og $C_{inngang}$ er inngangskapasitans. Vi definerer *elektrisk effort* eller *fanout* som C_h . Forholdet som indikerer en logisk ports kompleksitet kalles *logisk effort*.

4.1.4 Effort tidsforsinkelse

Videre definerer vi h som antallet identiske porter som en spesifikk port skal drive.

Anta at vi lar 3inngangs NAND porten drive samme last som før, dvs. 4 NAND porter med transistor bredder lik den opprinnelige porten. Dersom vi øker bredden på transistorene i porten som skal drive lasten med en faktor k uten å endre lasten vil lasten forbli den samme $5h'C$, der h' er lik h uten endring. Vi får da en ny h som kan uttrykkes som $h = h'/k$. Vi definerer *effort tidsforsinkelse* som $5(h'/k)C$. Dette tilsvarer $5hC$.

4.1.5 Mål

Kunne anvende Elmore forsinkelsesmodell på ulike logiske porter for å estimere tidsforsinkelse.

4.2 Lineær forsinkelsesmodell

(Kapittel 4.2.2 side 165 - 166)

Generelt kan tidsforsinkelse, eller *normalisert tidsforsinkelse* i en port skrives på formen:

$$d = f + p, \quad (4.2)$$

der p er parasittisk tidsforsinkelse uten ekstern last, og f er *effort tidsforsinkelse* eller *port effort* (stage effort). Vi ser at p er avhengig av porten selv, mens f er avhengig av en ekstern last eller fanout:

$$f = gh, \quad (4.3)$$

der g er logisk effort. En inverter, med pMOS transistor bredde lik 2 og nMOS transistor med bredde lik 1, defineres til å ha en logisk effort lik 1. Mer komplekse porter har større logisk effort som tilsier at de trenger lenger tid til å drive en gitt last.

For treinngangs NAND porten i Fig. 4.1 er logisk effort $g = 5/3$, fordi en inverter som last vil representere en last lik $C_{inverter} = 3C = 3C_g$, mens en 3inngangs NAND port vil representere en last lik $C_{NAND3} = 5C = 5C_g$.

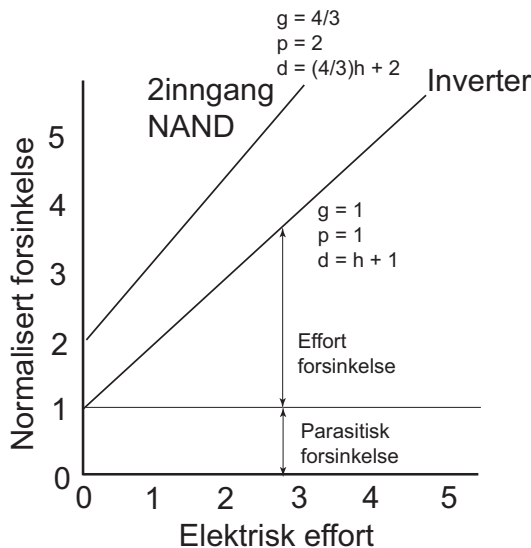


Figure 4.7: *Normalisert forsinkelse som funksjon av fanout.* (Figur 4.8)

I Fig. 4.7 er normalisert forsinkelse vist som funksjon av elektrisk effort eller fanout. Vi ser at $g = 4/3$ for en 2inngangs NAND port som gir $d = (4/3)h + 2$.

4.2.1 Mål

Kunne anvende lineær forsinkelsesmodell for ulike logiske porter.

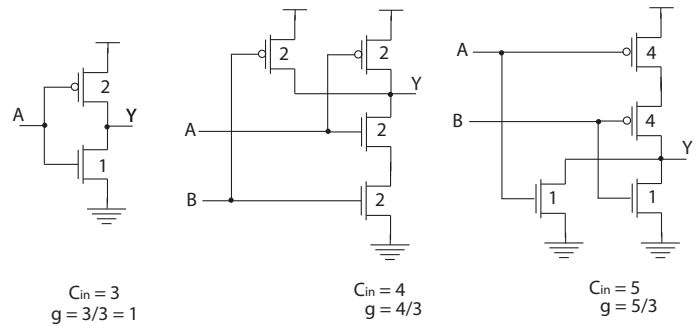


Figure 4.8: *Logiske porter der ekvivalent eller effektiv motstand er lik i opptrekk og nedtrekk.* (Figur 4.9)

4.3 Logisk effort

(Kapittel 4.2.3 side 166 - 167)

Logisk effort defineres som forholdet mellom en ports inngangskapasitans og inngangskapasitans til en inverter som levere samme utgangsstrøm. Alternativt sier vi at logisk effort uttrykker hvor mye dårligere en port er til å levere utgangsstrøm sammenlignet med en inverter.

Logisk effort (g) er vist for inverter, 2inngangs- og 3inngangs NAND port i Fig. 4.8, der transistorene er dimensjonert slik at den effektive eller ekvivalent motstanden blir lik for opptrekk og nedtrekk. Generelt kan man uttrykke logisk effort for en n inngangs NAND port dimensjonert etter samme betingelser som $g = (n + 2)/3$ og tilsvarende for en n inngangs NOR port gir $g = (2n + 1)/3$. Vi ser at logisk effort er høyere for NOR porter enn for NAND porter med samme antall innganger. Dette skyldes at pMOS transistorene har lavere mobilitet enn nMOS transistorene, som betyr at bredden på pMOS transistorene må økes betydelig når de seriekobles.

4.3.1 Mål

Kunne beregne logisk effort i ulike logiske porter.

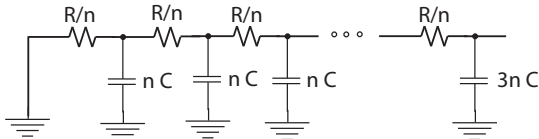
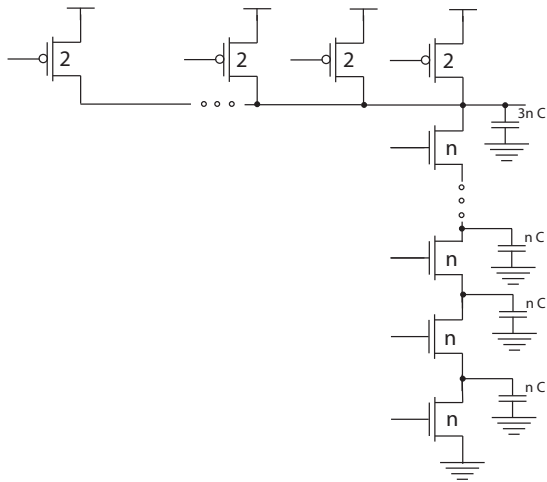


Figure 4.9: Parasittisk kapasitans og motstand i n -inngangs NAND port. (Figur 4.10)

4.4 Parasittisk tidsforsinkelse

(Kapittel 4.2.4 side 167 - 169)

Vi definerer *parasittisk tidsforsinkelse* som tidsforsinkelse i en port som ikke har ekstern last. All last som da bidrar til tidsforsinkelse i porten kommer av diffusjonskapasitanser internt i porten.

Port	1	2	3	4	n
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tristate	2	4	6	8	2n

Table 4.1: Parasittisk tidsforsinkelse i vanlige porter (relativt til en inverter) (Tabell 4.3 side 168).

Parasittisk tidsforsinkelse i vanlige logiske porter er vist i tabell 4.1. Metoden som ble brukt for å komme fram til resultatene er enkle, slik at reell parasittisk tidsforsinkelse evaluert ved hjelp av en simulator kan vise et forskjellig resultat. Vi kan forbedre modellen eller estimatet for parasittisk tidsforsinkelse ved å anvende Elmore modellen og dele opp en port i flere elektriske noder avhengig av den logiske portens kompleksitet.

En n -inngangs NAND port er vist i Fig. 4.9. Dersom vi anvender Elmore modellen 4.2 får vi:

$$t_{pd} = \sum_{i=1}^n C_i \sum_{j=1}^i R_j$$

$$\begin{aligned}
 &= \sum_{i=1}^{n-1} C_i \sum_{j=1}^i R_j + 3nCn \frac{R}{n} \\
 &= 3nRC + \sum_{i=1}^{n-1} nC \frac{R \cdot i}{n} \\
 &= 3nRC + RC \sum_{i=1}^{n-1} i \\
 &= 3nRC + RC \left(\frac{n(n-1)}{2} \right) \\
 &= (n+5) \frac{n}{2} RC, \tag{4.4}
 \end{aligned}$$

der n er antall innganger. Vi ser at den parasittiske tidsforsinkelsen øker kvadratisk med antall innganger. Dette betyr at man vil få mindre total tidsforsinkelse ved å dele opp en port med mange innganger i en kjede av to porter med færre innganger når antall innganger blir stort. Videre er NAND porter bedre enn NOR porter, særlig når porten har mange innganger.

4.4.1 Mål

Kunne beregne parasittisk tidsforsinkelse i ulike logiske porter.

4.5 Stige- og falltidsforsinkelse for inngang

(Kapittel 4.2.5.1 side 169 - 170)

Selv om den lineære modellen for tidsforsinkelse som ble presentert i avsnitt 4.2 gir fornuftig estimat for tidsforsinkelser, er det noen viktige forhold som det ikke tas hensyn til. Vi har så langt antatt at inngangssignaler for en port er stabile før utgangen endres. I virkeligheten vil utgangen på en port begynne en transisjon fra 0 til 1 eller 1 til 0 mens inngangene forandres. Sett fra utgangen betyr dette at vi må anta at inngangene forandres i det utgangen svitsjer. Dette vil medføre en større tidsforsinkelse. Vi kaller den tidsforsinkelsen vi har modellert så langt for $t_{Pd-step}^2$ som står for propageringsforsinkelse (P) med svært raske inngangstransisjoner. Dette betyr i praksis at inngangene stabiliserer seg før utgangen rekker å starte en transisjon.

Dersom vi antar at inngangen(e) til en port har 1/4 eller mer av lastkapasitansen til selve porten, vil porten starte sin utgangstransisjon før inngangstransisjonene er avsluttet. En annen måte å uttrykke dette på er at porten har en fanout som er 4 eller mindre. For å få med denne faktoren i et uttrykk for propageringsforsinkelse bruker vi modellen:

$$t_{Pd} = t_{Pd-step} + t_{kant} \cdot \left(\frac{1 + 2 \frac{|V_t|}{V_{DD}}}{6} \right), \quad (4.5)$$

der $t_{Pd-step}$ er propageringsforsinkelse for svært raske inngangstransisjoner og t_{kant} er stige- eller fall tidsforsinkelse (t_r eller t_f) for innganger.

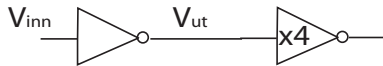


Figure 4.10: Inverter med en fanout lik 4. (Figur 4.11)

I Fig. 4.10 er det vist en inverter med fanout lik 4. Propageringsforsinkelsen for utgangsspenningen V_{ut} påvirkes av stige/falltidsforsinkelse t_{kant} som vist i Fig. 4.11. Vi ser at stige/falltid ikke påvirker propageringsforsinkelse på utgangen dramatisk, men stort sett er knyttet til inngangen. Dette vises tydelig i Fig. 4.12 der propageringsforsinkelse er vist relativt til stige/fall tidsforsinkelse på inngang.

4.5.1 Mål

Forstå hvordan stige- og falltidsforsinkelse på inngangen påvirker tidsforsinkelse i en inverter.

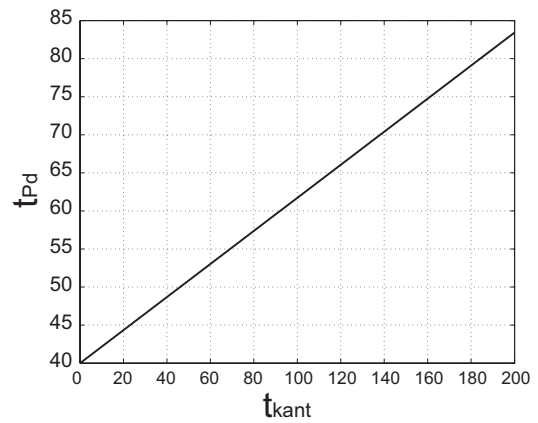


Figure 4.11: Propageringsforsinkelse som funksjon av stige/fall tidsforsinkelse t_{kant} på inngang. (Figur 4.11b)

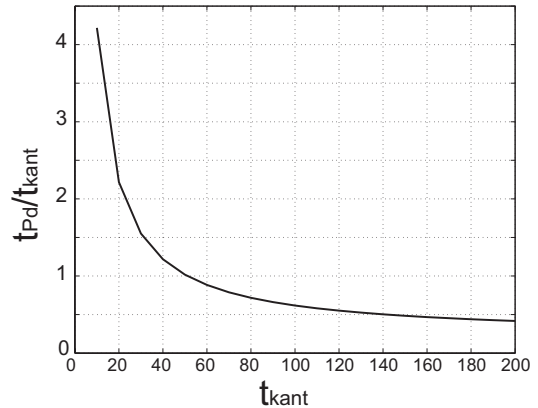


Figure 4.12: Propageringsforsinkelse relativt til stige/fall tidsforsinkelse på inngang som funksjon av stige/fall tidsforsinkelse t_{kant} på inngang.

4.6 Ulik transisjonstidspunkt for innganger

(Kapittel 4.2.5.2 side 170 - 171)

En antagelse som ligger til grunn for den lineære tidsforsinkelsesmodellen som ble presentert i avsnitt 4.2 er at bare en av inngangene endres mens øvrige inngangene er stabile. Ofte vil en port ha innganger som endres samtidig slik at portens tidsforsinkelse vil bli noe større enn estimert med den lineære modellen.

I Fig. 4.13 er det vist propageringsforsinkelse for en 3inngangs NAND port som funksjon av transisjonstidspunkt for innganger. Inngang A svitsjer ved tidspunkt 0 mens B svitsjer i samme retning, dvs. samme transisjon, ved tidspunkt t_b . Propageringsforsinkelsen som vises er referert til den siste stigende ($0 \rightarrow 1$) inngang for fallende utgang, og den tidligste fallende inngang for stigende utgang. Når en inngangstransisjon kommer vesentlig tidligere enn den andre inngangstransisjonen, vil $|t_b|$ bli stor og propageringsforsinkelsen blir nesten uavhengig av t_b . Når inngangene svitsjer omtrent samtidig, vil $|t_b|$ bli liten. Propageringsforsinkelse for transisjon fra 1 til 0 t_{pdf} øker på grunn

²I Weste et. al [1] kalles propageringsforsinkelsen t_{pd} som vil være samme uttrykk som brukes for parasittisk tidsforsinkelse. Derfor kaller vi propageringsforsinkelse t_{Pd} for å skille de to uttrykkene.

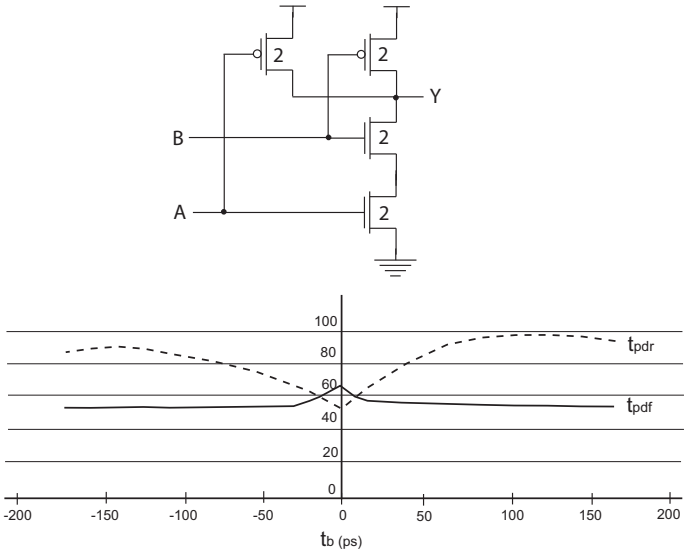


Figure 4.13: Påvirkning av transisjonsstidspunkt for innganger på propageringsforsinkelse for en logisk port. (Figur 4.12)

av seriekobling av to nMOS transistorer mens propageringsforsinkelse for transisjon fra 0 til 1 t_{pdr} reduseres på grunn av de to parallelle pMOS transistorene.

4.6.1 Mål

Forstå hvordan svitsjetidspunkt på innganger påvirker tidsforsinkelse i en logisk port.

4.7 MOS kapasitanser for inverter ved transisjoner

En antagelse som ligger til grunn for del lineære tidsforsinkelsesmodellen som ble presentert i avsnitt 4.2 er at kapasitansene som inngår i modellen er statiske, dvs. ikke endres på grunn av endringer i spenninger i ulike noder i kretsen. Vi vet at både gate source- og drain source kapasitanser er avhengig av transistorenes terminalspenninger. I tillegg vil effektive kapasitanser (ekvivalent kapasitanser) være avhengig av transisjonsretninger på gater i forhold til transisjonsretninger på drain og source.

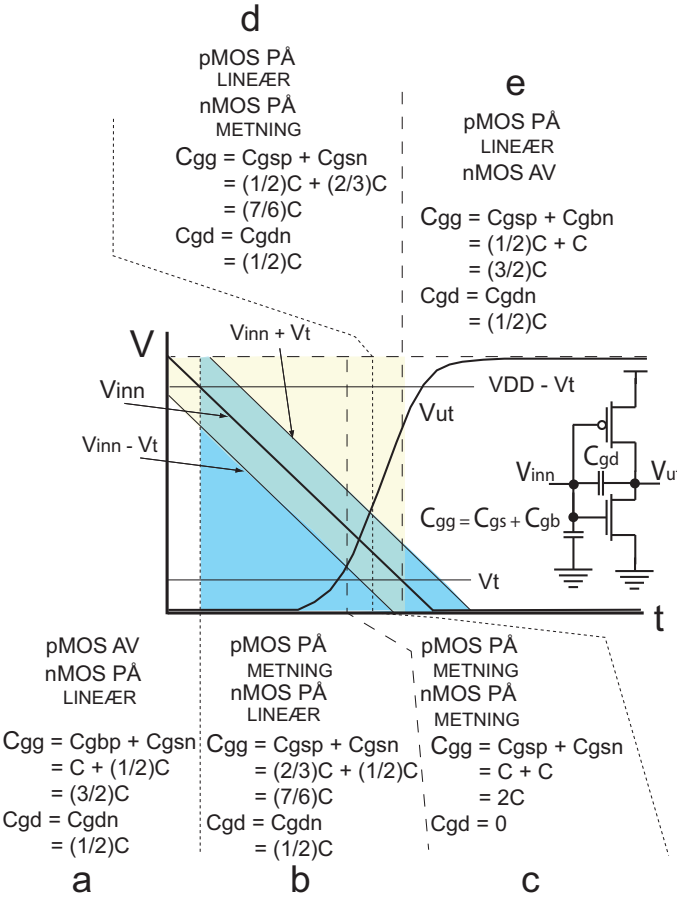


Figure 4.14: MOS kapasitanser for en inverter i transisjon. Vi antar at $-V_{tp} = V_{tn} = V_t$, $W_p = W_n$ og $L_p = L_n$.

Vi kan modellere MOS kapasitansene for inverteren som kapasitans til statisk spenning (GND), dvs. gate til GND kapasitans C_{gg} , og kapasitans til varierende spenning V_{ut} som C_{gd} . MOS kapasitanser for en inverter i transisjon er vist i Fig. 4.14. Vi kan uttrykke gate til GND og gate til drain kapasitans i de ulike områdene, der vi antar at $-V_{tp} = V_{tn} = V_t$, $W_p = W_n$ og $L_p = L_n$ og at transistorene ikke er i hastighetsmetning. Gult område i figuren indikerer området der nMOS transistoren kan være i metning, mens det blå området viser hvor pMOS transistoren kan være i metning.

4.7.1 Område a.

pMOS transistoren er AV fordi $V_{inn} > V_{DD} - V_t$. Kapasitansen sett fra inngangen (gate) som pMOS transistoren vil bidra med er gate bulk kapasitans $C_{gb} = C$. nMOS transistoren er PÅ fordi $V_{inn} > V_t$ og i lineært område fordi $V_{ut} < V_{inn} - V_t$. nMOS transistoren bidrar da med gate source kapasitans $C_{gs} = (1/2)C$ til gate til *GND* kapasitans. I tillegg vil transistoren bidra med gate til drain kapasitans som er kapasitansen mellom inngang og utgang. Vi har da

$$\begin{aligned}
 C_{gg} &= C_{gbp} + C_{gsn} \\
 &= C + \frac{1}{2}C \\
 &= \frac{3}{2}C \\
 C_{gd} &= C_{gdn} \\
 &= \frac{1}{2}C \\
 C_{total} &= C_{gg} + C_{db} \\
 &= 2C.
 \end{aligned} \tag{4.6}$$

4.7.2 Område b.

pMOS transistoren er PÅ fordi $V_{inn} \leq V_{DD} - V_t$. Utgangen er fortsatt lav og dette betyr at pMOS transistoren er i metning, dvs. $V_{ut} \leq V_{inn} + V_t$. pMOS transistoren bidrar da bare med kapasitans til *GND*, dvs. vi har $C_{gs} = (2/3)C$ og $C_{gd} = 0$. nMOS transistoren er også PÅ fordi $V_{inn} > V_t$ og fortsatt i lineært område fordi $V_{ut} < V_{inn} - V_t$. nMOS transistoren bidrar da med gate source kapasitans $C_{gs} = (1/2)C$ til gate til *GND* kapasitans. I tillegg vil transistoren bidra med gate til drain kapasitans som er kapasitansen mellom inngang og utgang. Vi har da

$$\begin{aligned}
 C_{gg} &= C_{gsp} + C_{gsn} \\
 &= \frac{2}{3}C + \frac{1}{2}C \\
 &= \frac{7}{6}C \\
 C_{gd} &= C_{gdn} \\
 &= \frac{1}{2}C \\
 C_{total} &= C_{gg} + C_{db} \\
 &= \frac{10}{6}C.
 \end{aligned} \tag{4.7}$$

4.7.3 Område c.

pMOS transistoren er PÅ og i metning fordi $V_{inn} \leq V_{DD} - V_t$ og $V_{ut} \leq V_{inn} + V_t$. pMOS transistoren bidrar da bare med kapasitans til *GND*, dvs. vi har $C_{gs} = (2/3)C$ og $C_{gd} = 0$. nMOS transistoren er også PÅ og i metning fordi

$V_{inn} \geq V_t$ og $V_{ut} \geq V_{inn} - V_t$. nMOS transistoren bidrar da bare med kapasitans til *GND*, dvs. vi har $C_{gs} = (2/3)C$ og $C_{gd} = 0$. Vi har da

$$\begin{aligned}
 C_{gg} &= C_{gsp} + C_{gsn} \\
 &= \frac{2}{3}C + \frac{2}{3}C \\
 &= \frac{4}{3}C \\
 C_{gd} &= 0 \\
 C_{total} &= C_{gg} + C_{db} \\
 &= \frac{4}{3}C.
 \end{aligned} \tag{4.8}$$

4.7.4 Område d.

pMOS transistoren er PÅ, men nå i lineært område fordi $V_{inn} \leq V_{DD} - V_t$ og $V_{ut} > V_{inn} + V_t$. pMOS transistoren bidrar da med kapasitans til *GND*, $C_{gs} = (1/2)C$, og kapasitans til utgang, $C_{gd} = (1/2)C$. nMOS transistoren er fortsatt på og i metning fordi $V_{inn} \geq V_t$ og $V_{ut} \geq V_{inn} - V_t$. nMOS transistoren bidrar da bare med kapasitans til *GND*, dvs. vi har $C_{gs} = (2/3)C$ og $C_{gd} = 0$. Vi har da

$$\begin{aligned}
 C_{gg} &= C_{gsp} + C_{gsn} \\
 &= \frac{1}{2}C + \frac{2}{3}C \\
 &= \frac{7}{6}C \\
 C_{gd} &= C_{gdp} \\
 &= \frac{1}{2}C \\
 C_{total} &= C_{gg} + C_{db} \\
 &= \frac{10}{6}C.
 \end{aligned} \tag{4.9}$$

4.7.5 Område e.

pMOS transistoren er PÅ og i lineært område fordi $V_{inn} \leq V_{DD} - V_t$ og $V_{ut} > V_{inn} + V_t$. pMOS transistoren bidrar da med kapasitans til *GND*, $C_{gs} = (1/2)C$, og kapasitans til utgang, $C_{gd} = (1/2)C$. nMOS transistoren er AV fordi $V_{inn} < V_t$ som betyr at transistoren bidrar med kapasitansen $C_{gb} = C$ til *GND*. Vi har da

$$\begin{aligned}
 C_{gg} &= C_{gsp} + C_{gbn} \\
 &= \frac{1}{2}C + C \\
 &= \frac{3}{2}C \\
 C_{gd} &= C_{gdp} \\
 &= \frac{1}{2}C \\
 C_{total} &= C_{gg} + C_{db}
 \end{aligned}$$

$$= 2C.$$

(4.10) 4.8 Gate source kapasitans

(Kapittel 4.2.5.3 side 171 - 172)

4.7.6 MOS kapasitanser

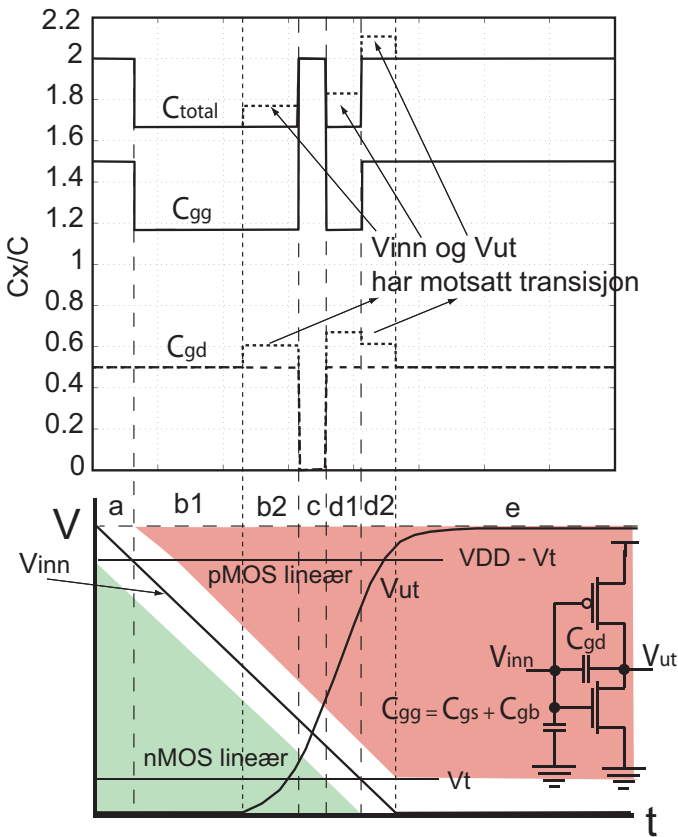


Figure 4.15: MOS kapasitanser for en inverter i transisjon. Vi antar at $-V_{tp} = V_{tn} = V_t$, $W_p = W_n$ og $L_p = L_n$.

MOS kapasitansene for en inverter ved en inngangstransisjon fra 1 til 0 er vist i Fig. 4.15, der grønn farge indikerer spenningsområdet der nMOS transistoren er i lineært område og rødt område viser der pMOS transistoren er i lineært område. I områdene b2, d1 og d2 vil inngangen og utgangen svitsje motsatt vei. Kapasitansen mellom inngang og utgang vil da virke som en større kapasitans. Dette kalles *Miller effekt*. I området d1 stiger utgangen forholdsvis bratt slik at Miller effekten bidrar ekstra til å øke C_{gd} og den totale kapasitansen i dette området.

4.7.7 Mål

Forstå hvordan MOS transistor kapasitanser er avhengig av inn- og utgangsspenning i en inverter under transisjon på inngang og utgang.

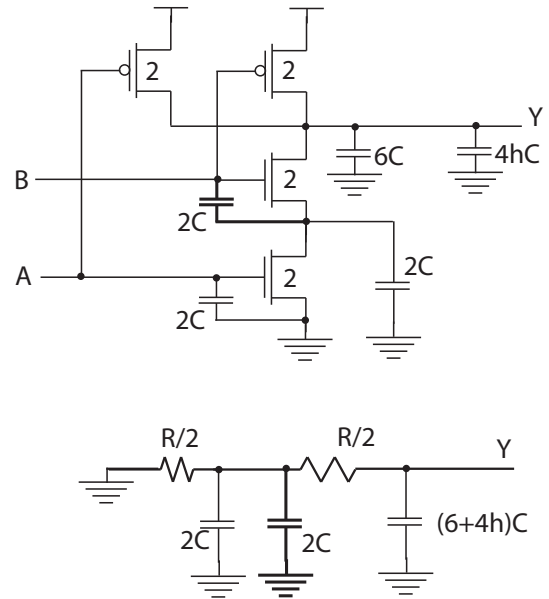


Figure 4.16: Modifisert Elmore tidsforsinkelse der gate source kapasitans for interne noder (source) er tatt med. (Figur 4.13)

De enkle modellene for beregning av tidsforsinkelser modellerer gate source kapasitanser som gate til GND kapasitans. Dette blir inkludert som kapasitans for gate eller inngang, dvs. den vil være en inngangskapasitans. I porter der source ikke alltid er koblet til en fast spenningsreferanse, typisk V_{DD} eller GND , vil gate source kapasitans bidra med kapasitans for interne noder i porten også. En inverter vil i praksis alltid ha source koblet til faste spenningsreferanser, men mer komplekse porter vil ha interne source noder skal skal lades opp eller ut. Disse nodene vil bidra ikke bare med diffusjonskapasitanser men også gate source kapasitans som vist i Fig. 4.16.

4.8.1 Mål

Forstå hvordan gate source kapasitans knyttet til interne source terminaler som ikke er koblet til faste spenningsreferanser påvirker tidsforsinkelse i en port.

4.9 Bootstrapping

(Kapittel 4.2.5.4 side 172 - 173)

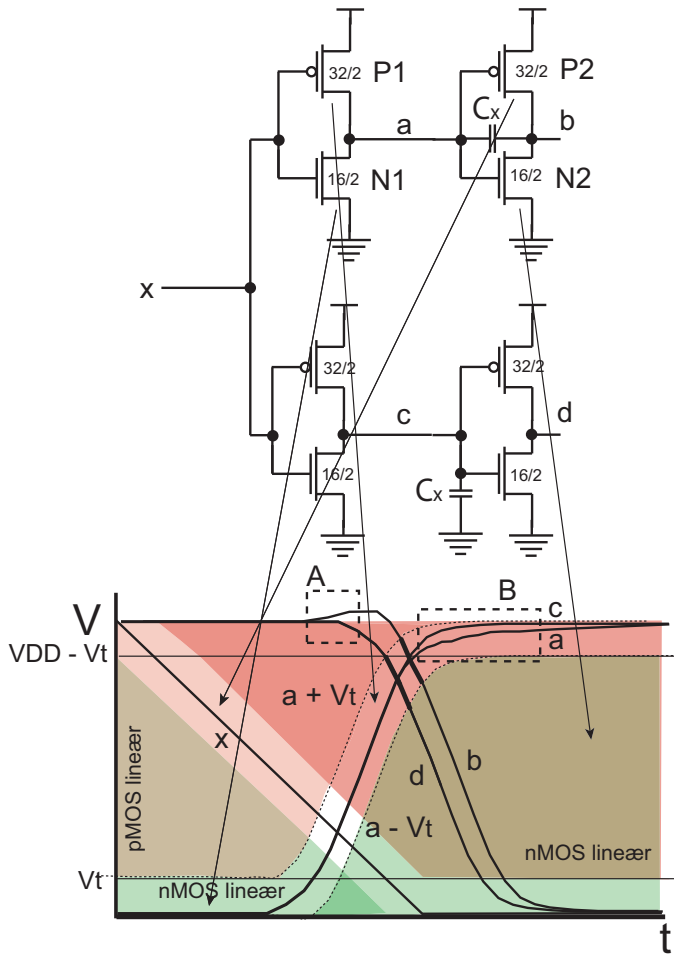


Figure 4.17: Bootstrapping i inverter. (Figur 4.24 b og c)

I det en inverter svitsjer vil transistorene være i lineært område i deler av svitsjetiden. Dette er avhengig av inn- og utgangsspenninger, dvs gate og drain spenninger for transistorene. Gate drain kapasitanser virker som en kapasitiv kobling mellom inngang og utgang som vist i Fig. 4.17. I tillegg vil denne kapasitansen øke i verdi når inngang og utgang endres (svitsjer) i ulik retning. I figuren er det vist to par invertere i serie der den øvre kjeden er modellert med gate drain kapasitans i siste inverter, mens denne kapasitansen er lagt til GND i nederste kjede. Den grunnleggende forskjellen i de to kjedene er at den øverste har en kapasitiv kobling mellom nodene a og b i motsetning til den nederste der det ikke er kapasitive koblinger mellom inngang og utgang på inverterne.

En spenningsendring på inngangen vil kunne påvirke utgangen direkte via kapasitansen mellom inngang og utgang:

$$\Delta V_{ut} = \frac{C_{gd}}{C_{last}} \Delta V_{inn}, \quad (4.11)$$

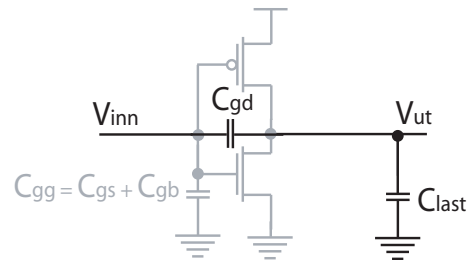


Figure 4.18: Kapasitiv divisjon via flytende kondensator.

der C_{last} er den totale kapasitans sett fra utganger, inkludert intern kapasitans og C_{gd} (C_x i øvre kjede). Kapasitiv påvirkning er vist i Fig. 4.18. I området A i Fig. 4.17 vil inngangen endres mens nMOS transistoren leverer lite strøm fordi inngangen a er relativt lav og pMOS transistoren er i lineært område med svært liten drain source spenning. Inverteren leverer derfor lite strøm som betyr at utgangen kan påvirkes fra inngang via C_{gd} . Dette vil føre til en endring på utgangsspenningen b i positiv retning, dvs. b vil dras over V_{DD} . Når inngangen a er nær V_{DD} som vist i område B kan vi anta at inverteren som driver a ikke leverer mye strøm fordi nMOS transistoren $N1$ er AV og pMOS transistoren $P1$ vil være i lineært område med liten drain source spenning. I dette tilfellet vil en endring av på utgangen b derfor påvirke a på tilsvarende måte. Dette vil slå ut som en reduksjon i spenningen for a som vist i figuren. Vi kaller denne effekten *bootstrapping*.

4.9.1 Mål

Forstå hvordan gate drain kapasitans, dvs. mellom inngang og utgang på en inverter, påvirker transisjoner og tidsforsinkelse for en inverter.

4.10 Tidsforsinkelse i en logisk port

(Kapittel 4.3.2 side 173 - 174)

Som kjent er tidsforsinkelse i en logisk port avhengig av mange faktorer; kompleksitet (logisk effort g), kapasitiv fanout (elektrisk effort h) og parasittisk tidsforsinkelse p .

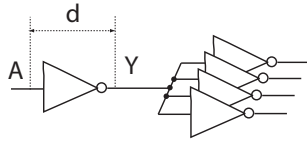


Figure 4.19: Inverter med fanout lik 4 (FO4). (Figur 4.15)

Som eksempel kan vi se på en inverter med fanout lik 4 som vist i Fig. 4.19. I dette eksemplet er alle inverterne like. Vi kan definere en referanse for tidsforsinkelse som tidsforsinkelse for en enhetsinverter der lasten bare representerer ekstern kapasitans gitt av en identisk inverter. Dersom μ_n er $2\mu_p$ har vi at $W_p/L_p = 2W_n/L_n$. Den eksterne lasten representerer i dette tilfellet $3C$ og effektiv motstand er definert som R . Referanse tidsforsinkelse τ er da lik $3RC$. Elektrisk effort h er lik 4 fordi fanout er 4 med identiske invertere. Logisk effort g for inverteren er 1. Parasittisk tidsforsinkelse p for en inverter er 1, og dermed får vi følgende uttrykk for tidsforsinkelse:

$$\begin{aligned} t_{pd} &= d \cdot \tau \\ &= (f + p) \cdot \tau \\ &= (gh + p) \cdot \tau \\ &= (1 \cdot 4 + 1) \cdot \tau \\ &= 5\tau. \end{aligned} \quad (4.12)$$

Som eksempel kan vi anta at referanse tidsforsinkelse τ er lik $15ps$ som vil gi en tidsforsinkelse $t_{pd} = 75ps$.

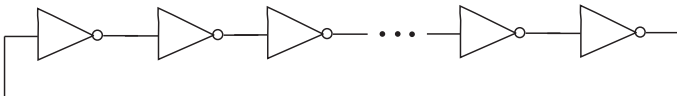


Figure 4.20: Inverter ring oscillator. (Figur 4.17)

Et annet eksempel er vist i Fig. 4.20 der N enhetsinvertere er koblet sammen som en ringoscillator. For hver inverter får vi; logisk effort $g = 1$, elektrisk effort $h = 1$ og parasittisk tidsforsinkelse $p = 1$. I dette eksemplet vil tidsforsinkelse for hver inverter være $t_{pd} = (gh + p)\tau = 30ps$. En ringoscillator med N invertere vil gi total tidsforsinkelse i en halvperiode $t_{oscillator} = Nt_{pd}$. For å få samme polaritet i samme node må vi ha to halvperioder og to halvperioder blir en periode. Frekvens er definert som $1/(\text{tidsforsinkelse i en periode})$;

$$\text{frekvens} = \frac{1}{2Nt_{pd}}, \quad (4.13)$$

der N er antall invertere i oscillatoren og t_{pd} er tidsforsinkelse i hver inverter. For $N = 31$ og $t_{pd} = 2\tau = 30ps$ får vi $\text{frekvens} = 1/(2 \cdot 31 \cdot 30ps) = 1/1860ps = 538MHz$.

4.10.1 Mål

Kunne finne tidsforsinkelse i logiske porter med forskjellig ekstern last.

4.11 Tidsforsinkelse i en kjede av logiske porter

(Kapittel 4.3.2 side 174 - 178)

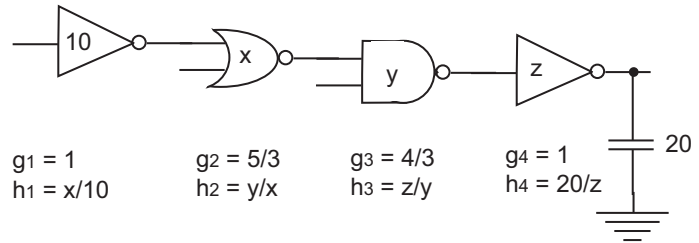


Figure 4.21: Kjede av logiske porter med logisk- og elektrisk effort for hver port. (FIG4.16)

Det er enkelt å generalisere logisk effort til en kjede av logiske porter som vist i Fig. 4.21. Som kjent representerer logisk effort g kompleksiteten til en logisk port relativt til en inverter. Den første og siste porten i kjeden er inverter og vil derfor ha logisk effort $g_1 = g_4 = 1$. Port 2 i kjeden er en 2inngangs NOR port som vil ha en logisk effort $g_2 = 5/3$ per definisjon. Den neste porten i kjeden er en 2inngangs NAND port med $g_3 = 4/3$. Logisk effort er ikke avhengig av de faktiske transistorstørrelsene i de logiske portene som inngår i kjeden. Elektrisk effort h derimot er direkte avhengig av transistorstørrelser i porten selv og porter som skal drives. Elektrisk effort eller fanout er gitt av $h = C_{ekstern}/C_{inngang}$ for en logisk port. I kjeden får vi for den første inverteren $h_1 = x/10$ der x størrelsen på transistorene, dvs. en pMOS og en nMOS, i NOR porten som skal drives av inverteren og der størrelsen på transistorene i inverteren er 10. På tilsvarende måte kan vi uttrykke elektrisk effort for de andre portene; $h_2 = y/x$, $h_3 = z/y$ og $h_4 = 20/z$. Vi legger merke til at logisk effort er uavhengig av transistorstørrelser, mens elektrisk effort er avhengig av transistorstørrelser.

Vi kan definere *kjede logisk effort* G som:

$$G = \prod_{i=1}^n g_i, \quad (4.14)$$

der n er antall porter i kjeden.

Vi definerer *kjede elektrisk effort* H som forholdet mellom kjedens eksterne kapasitans og kjedens inngangskapasitans:

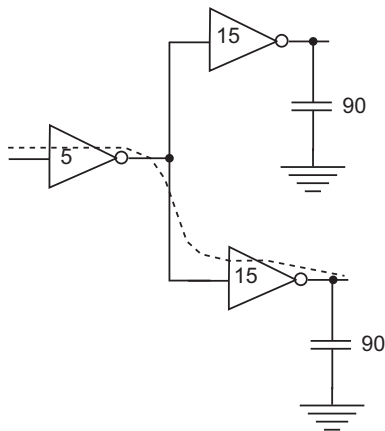


Figure 4.22: Kjede av to grener. (FIG4.18)

$$H = \frac{C_{ekstern(kjede)}}{C_{inngang(kjede)}}. \quad (4.15)$$

For kjeden i Fig. 4.21 får vi $G = 1 \cdot 5/3 \cdot 4/3 \cdot 1 = 2.22$ og $H = 20/10 = 2$.

For en kjede med logiske porter kan beregning av *kjede effort* F bli en utfordring. Grunnen til det er at metodene vi her benyttet så langt for en port vil medføre at vi vil ta hensyn til kapasitiv last knyttet til porter som ikke er med i kjeden. Der hvor vi får en forgrening vil det bli tatt med last for porter som ikke inngår i selve kjeden. Dette er vist i Fig. 4.22 der kjeden går via den nederste av de to inverterne som drives av den første porten (til venstre). Dette kan illustreres ved at vi forsøker å beregne kjede effort med modellen $F = GH$. Vi finner først logisk effort for kjeden $G = 1 \cdot 1 = 1$ og elektrisk effort for kjeden $H = 90/5 = 18$. Vi får da kjedens effort $F = 18$. Dersom vi beregner kjedens effort med modellen $F = \prod g_i h_i = (1 \cdot 30/5) \cdot (1 \cdot 90/15) = 36$. Som vi ser blir kjedens effort forskjellig med de to modellene. I den enkle modellen $F = GH$ tar vi ikke hensyn til lasten som porter som ikke inngår i kjedens signalvei representerer. Vi endrer derfor den enkle modellen ved å inkludere lasten som en forgrening representerer ved å introdusere en *forgreningseffort* b som er forholdet mellom total kapasitans sett av en port i kjeden og portens kjedekapasitans:

$$b = \frac{C_{P\dot{A}-kjede} + C_{AV-kjede}}{C_{P\dot{A}-kjede}}. \quad (4.16)$$

For kjeden i Fig. 4.22 gir dette $b = (15 + 15)/15 = 2$. I tillegg definerer vi *kjede forgreningseffort* B som:

$$B = \prod b_i. \quad (4.17)$$

Vi kan nå definere kjede effort:

$$F = GBH. \quad (4.18)$$

I eksemplet fra Fig. 4.22 gir dette $F = 1 \cdot 2 \cdot 18 = 36$ som er identisk med kjedens effort modellert som produkt av portenes effort. Poenget med modellene er å forenkle beregninger av tidsforsinkelse.

Vi kan nå beregne tidsforsinkelse i en kjede av logiske porter. *Kjedeforsinkelse* D er lik summen av tidsforsinkelse i hver port:

$$\begin{aligned} D &= \sum d_i \\ &= D_F + P, \end{aligned} \quad (4.19)$$

der

$$\begin{aligned} D_F &= \sum f_i \\ P &= \sum p_i, \end{aligned} \quad (4.20)$$

der D_F er *kjede effort tidsforsinkelse* og P er *kjede parasittisk tidsforsinkelse*.

I praksis vil en minimumsverdi for kjedens elektriske effort forutsette at hver port har lik elektrisk effort. Dette betyr at tidsforsinkelse i en kjede har et minimum når den elektriske efforten i hver port er lik. Vi har da at:

$$\begin{aligned} f' &= f_i \\ &= g_i h_i \\ &= F^{1/N}, \end{aligned} \quad (4.21)$$

der N er antall porter i kjeden. Dette betyr at minimum tidsforsinkelse i en kjede med N porter med kjede effort lik F og kjede parasittisk tidsforsinkelse P blir:

$$D_{minimum} = NF^{1/N} + P. \quad (4.22)$$

Dette er et viktig resultat for logisk effort, som viser at minimum tidsforsinkelse i en kjede av logiske porter kan bli estimert med kjennskap bare til antall porter i kjeden, kjedens effort og parasittiske tidsforsinkelse uten å bestemme transistorstørrelser.

Det er også enkelt å bestemme transistorstørrelser slik at tidsforsinkelsen blir minst mulig. Ved å kombinere ligningene:

$$\begin{aligned} f &= gh \\ h &= \frac{C_{ekstern}}{C_{inngang}}, \end{aligned}$$

får vi:

$$C_{innangi} = \frac{C_{eksterni} g_i}{f'} \quad (4.23)$$

I praksis starter man med kjedens ende og arbeider seg mot kjedens inngang, og bestemmer transistorenes størrelse for hver port.

4.11.1 Eksempel

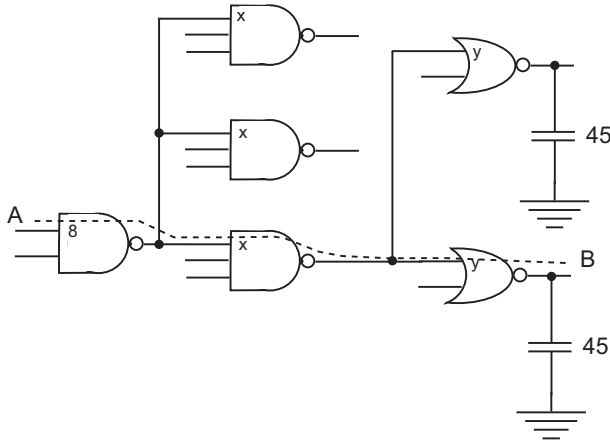


Figure 4.23: Kjede med porter. (FIG4.19)

En kjede med logiske porter og forgreninger er vist i Fig. 4.23. Kjeden består av 2inngangs NAND port, med logisk effort lik $4/3$, 3inngangs NAND port, med logisk effort $5/3$ og 2inngangs NOR port med logisk effort $5/3$. Vi finner først kjedens logiske effort:

$$G = \frac{4}{3} \cdot \frac{5}{3} \cdot \frac{5}{3} = 100/27.$$

Kjedens elektriske effort blir:

$$H = 45/8,$$

og kjedens forgreningseffort blir:

$$B = \left(\frac{x + 2x}{x} \right) \cdot \left(\frac{y + y}{y} \right) = 6.$$

Kjedens effort blir da:

$$F = GBH = \frac{100}{27} \cdot 6 \cdot \frac{45}{8} = 125.$$

V kan beregne optimal port effort:

$$f' = 125^{\frac{1}{3}} = 5.$$

Parasittisk tidsforsinkelse for kjeden blir:

$$P = 2 + 3 + 2 = 7,$$

slik at minimum kjedeforsinkelse blir:

$$D = N \cdot f' + P = 3 \cdot 5 + 7 = 22,$$

uttrykt i enheter av referanse tidsforsinkelse τ .

Vi skal nå finne transistorstørrelse som gir minimum tidsforsinkelse i kjeden. Vi starter ved utgangen i kjeden og beregner transistorstørrelse for 2inngangs NOR port:

$$y = \frac{C_{ekstern} \cdot g}{f'} = \frac{45 \cdot \frac{5}{3}}{5} = 15.$$

Videre beregner vi x :

$$x = \frac{(15 + 15) \cdot \frac{5}{3}}{5} = 10,$$

og ved kontroll på 2inngangs NAND porten ser vi at størrelsen på transistorene blir $((10+10+10) \cdot (4/3))/5 = 8$, som stemmer med Fig. 4.23. Vi har så langt omtalt transistorstørrelse uten å skille på nMOS- pMOS transistorer. Det er viktig å bestemme transistorstørrelser slik at vi får samme effektive motstand i "worst-case" opptrekk og nedtrekk for de enkelte portene i kjeden. For en 2inngangs NOR port vil vi det være 2 pMOS transistorer i serie i opptrekket, mens det bare er en nMOS transistor i nedtrekket. Vi kan uttrykke forholdet mellom pMOS- og nMOS transistorene ved å anta at lengden³ på transistorene er like:

$$W_p = e \cdot u \cdot W_n,$$

³Typisk minimums lengde i digital mikroelektronikk for å redusere lasten mest mulig.

der e er forholdet mellom serietransistorer i opptrekk og nedtrekk, og $u = \mu_n/\mu_p$ er forholdet mellom mobilitet i nMOS- og pMOS transistorer. For en 2inngangs NOR port blir derfor transistorstørrelsene, gitt at den optimale størrelsen er 15 (som betyr at det er en nMOS og en pMOS transistor som utgjør lasten for noden i kjeden):

$$\begin{aligned} W_p + W_n &= 15 \\ e \cdot u \cdot W_n + W_n &= 15 \\ (2 \cdot 2 + 1) W_n &= 15 \\ W_n &= 3, \end{aligned}$$

der $\mu_n = \mu_p$. For 2inngangs NOR porten har vi at $W_n = 3$ og $W_p = 2 \cdot 2 \cdot 3 = 12$. For 3inngangs NAND porten har vi

$$\begin{aligned} W_p + W_n &= 10 \\ \left(\frac{1}{3} \cdot 2 + 1\right) W_n &= 10 \\ W_n &= 6, \end{aligned}$$

og $W_p = (1/3) \cdot 2 \cdot 6 = 4$.

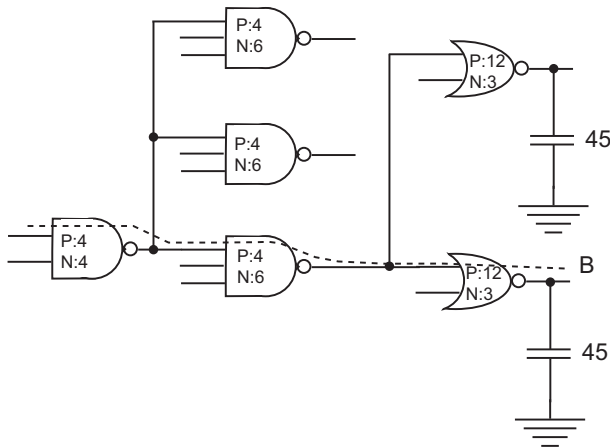


Figure 4.24: *Kjede med porter og transistorstørrelser. (FIG4.20)*

Transistorstørrelser er vist i Fig. 4.24.

4.11.2 Mål

Kunne beregne logisk effort og elektrisk effort i en kjede. Kunne inkludere forgreninger som forgreningseffort og beregne tidsforsinkelse i en kjede. I tillegg skal man kunne finne optimal port effort for minimum tidsforsinkelse i en kjede med en gitt ekstern last.

4.12 Optimalt antall porter i en kjede

(Kapittel 4.3.2 side 178 - 181)



Figure 4.25: *Ekstern last som skal drives av et antall invertere. (FIG4.21)*

I Fig. 4.25 har vi en enhetsinverter som skal drive en ekstern last lik 64 (tilsvarende 64 enhetsinvertere). Vi skal finne ut det optimale antallet invertere som skal settes inn mellom enhetsinverteren og den eksterne lasten. I tillegg til tidsforsinkelse gjennom kjeden skal vi bestemme størrelsen på transistorene i de ulike inverterene i kjeden som gir minst tidsforsinkelse i kjeden. Vi ser bort fra polariteten på signalet i dette eksemplet. Vi kan beregne kjedens effort, som er uavhengig av antall porter i kjeden:

$$\begin{aligned} F &= GBH \\ &= 1 \cdot 1 \cdot \frac{C_{ekstern}}{C_{inngang}} \\ &= 64. \end{aligned} \tag{4.24}$$

4.12.1 0 porter i tillegg

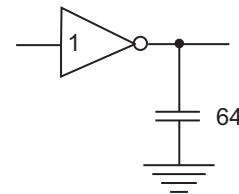


Figure 4.26: *Ekstern last som skal drives av en enhetsinverter. (FIG4.21)*

I dette tilfellet lar vi enhetsinverteren drive den eksterne lasten direkte som vist i Fig. 4.26. I tillegg vil den logiske efforten for enhetsinverteren være lik logisk effort for hele kjeden. Tidsforsinkelsen blir da $D = F + 1 = 65\tau$.

4.12.2 1 port i tillegg

Ved å sette inn en port i tillegg til enhetstransistoren får vi kretsen som er vist i Fig. 4.27. Vi har nå:

$$\begin{aligned} N &= 2 \\ f' &= F^{\frac{1}{2}} \\ &= 8 \\ D &= Nf' + P \end{aligned}$$

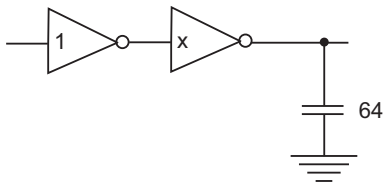


Figure 4.27: Ekstern last som skal drives av to invertere. (FIG4.21)

$$\begin{aligned} &= 2 \cdot 8 + 1 + 1 \\ &= 18\tau. \end{aligned} \quad (4.25)$$

Tidsforsinkelsen for en kjede med to invertere er redusert fra 64τ til 18τ . Transistorstørrelsene er gitt av $n = f' = 8$ som gir bredde på nMOS lik $n = 8$ og bredde på pMOS transistoren $p = 16$ som gir samme stige og falltid for utgangen. Breddene er størrelse i forhold til enhetstransistorene. Inverteren størrelse er 8 ganger enhetsinverter, dvs. $x = 8$.

4.12.3 2 porter i tillegg

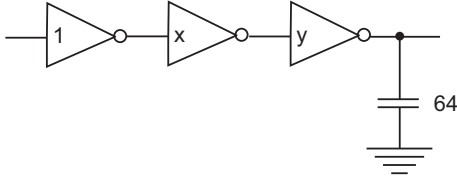


Figure 4.28: Ekstern last som skal drives av tre invertere. (FIG4.21)

Ved å sette inn to porter i tillegg til enhetstransistoren får vi kretsen som er vist i Fig. 4.28. Vi har nå:

$$\begin{aligned} N &= 3 \\ f' &= F^{\frac{1}{3}} \\ &= 4 \\ D &= Nf' + P \\ &= 3 \cdot 4 + 1 + 1 + 1 \\ &= 15\tau. \end{aligned} \quad (4.26)$$

Tidsforsinkelsen for en kjede med tre invertere, i forhold til kjede med to invertere, er redusert fra 18τ til 15τ . Vi finner først størrelsen på den siste inverteren $y = C_{ekstern}p/f' = 64 \cdot 1/4 = 16$ som gir bredde på nMOS transistor lik 16 og bredde på pMOS transistor lik 32. Størrelsen på inverter nummer 2 er gitt av $x = 16 \cdot 1/4 = 4$ som gir bredde på nMOS transistor lik 4 og bredde på pMOS transistor lik 8.

4.12.4 3 porter i tillegg

Ved å sette inn tre porter i tillegg til enhetstransistoren får vi kretsen som er vist i Fig. 4.29. Vi har nå:

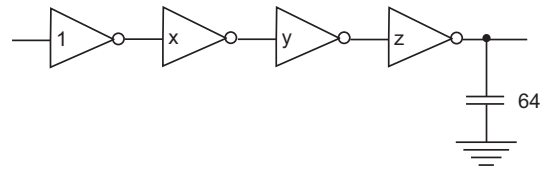


Figure 4.29: Ekstern last som skal drives av fire invertere. (FIG4.21)

$$\begin{aligned} N &= 4 \\ f' &= F^{\frac{1}{4}} \\ &= 2.8 \\ D &= Nf' + P \\ &= 4 \cdot 2.8 + 1 + 1 + 1 + 1 \\ &= 15.2\tau. \end{aligned} \quad (4.27)$$

Når vi øker kjedens lengde fra 3 til 4 vil tidsforsinkelsen øke litt. Dette betyr at det optimale antall ligger mellom 3 og 4. Størrelsen på inverterne blir fra utgangen $z = 64/2.8 \approx 23$, $y = 23/2.8 \approx 8$ og $x = 8/2.8 \approx 2.8$.

4.12.5 Optimalt antall porter i en kjede

Vi ser av eksemplet at optimalt antall porter er mellom 3 og 4. Vi kan uttrykke tidsforsinkelsen i kjeden som

$$D = NF^{\frac{1}{N}} + Np_{inv}, \quad (4.28)$$

der p_{inv} er parasittisk tidsforsinkelse for en inverter. Der som vi deriverer D med hensyn på antall porter i kjeden N og setter lik 0 får vi optimalt antall porter:

$$\begin{aligned} \frac{\partial D}{\partial N} &= -F^{\frac{1}{N}} \ln F^{\frac{1}{N}} + F^{\frac{1}{N}} + p_{inv} \\ &= \rho(1 - \ln \rho) + p_{inv}, \end{aligned} \quad (4.29)$$

der $\rho = F^{\frac{1}{N}}$. En numerisk løsning for ρ er 3.59 som betyr at det vil lønne seg å øke inverterstørrelsen i hvert trinn med 3.59.

Vi kan uttrykke optimalt antall invertere i en kjede som

$$N' = \log_{\rho} F. \quad (4.30)$$

4.12.6 Mål

Kunne beregne optimalt antall invertere i en kjede som skal drive en gitt ekstern last.

4.13 Oppsummering av logisk effort

(Kapittel 4.3.5 side 183 - 185)

<i>Terminologi</i>	<i>Port</i>	<i>Kjede</i>
<i>Antall porter</i>	1	N
<i>Logisk effort</i>	g	$G = \prod g_i$
<i>Elektrisk effort</i>	$h = \frac{C_{ekstern}}{C_{inngang}}$	$H = \frac{C_{ekstern(kjede)}}{C_{inngang(kjede)}}$
<i>Forgreings-effort</i>	$b = \frac{C_{PA-kjede} + C_{AV-kjede}}{C_{PA-kjede}}$	$B = \prod b_i$
<i>Effort</i>	$f = gh$	$F = GHB$
<i>Effort tidsforsinkelse</i>	f	$D_F = \sum f_i$
<i>Parasittisk tidsforsinkelse</i>	p	$P = \sum p_i$
<i>Tidsforsinkelse</i>	$d = f + p$	$D = D_F + P$

simulering og analyse av logiske kjeder. Det er noen nyttige tommelfinger regler som man skal huske på:

- Ideen med logisk effort er å kunne på en enkel måte sammenligne ulike topologier med hensyn på tidsforsinkelse.
- NAND porter er raskere enn NOR porter..
- Tidsforsinkelsen i en kjede er minst når effort forsinkelse er omtrent lik for hver port i kjeden.
- Tidsforsinkelsen i en kjede er relativt lite påvirket av moderate forandringer rundt et optimalt punkt.
- Porter med høyere port effort enn 4 vil gi kjeder med mindre areal og mindre effektforbruk, men dersom vi øker port efforten til 6-8 vil dette medføre betydelig redusert hastighet.
- Nøyaktigheten ved beregning av tidsforsinkelse ved hjelp av logisk effort er begrenset.
- RC forsinkelsesmodeller tar ikke hensyn til hastighetsmetning og bodyeffekt.
- Logisk effort tar ikke i betraktning signalføring mellom porter (interkonnekt).

4.13.1 Mål

Kunne anvende logisk effort, elektrisk effort og parasittisk tidsforsinkelse til å designe logiske kjeder med liten tidsforsinkelse.

Når man anvender logisk effort er det vanlig å arbeide etter følgende steg:

1. Beregn kjede effort: $F = GHB$.
2. Estimer det optimale antall porter: $N' = \log_4 F$.
3. Skisser en kjede med: N' porter.
4. Estimer minimum tidsforsinkelse: $D = N' F^{\frac{1}{N'}} + P$.
5. Bestem den beste port effort: $f' = F^{\frac{1}{N'}}$.
6. Start ved kjedens utgang og beregn bakover transistor størrelser: $C_{inngang_i} = \frac{C_{ekstern_i} \cdot g_i}{f'}$.

Bruk av logisk effort er praktisk for små kjeder av logiske porter. For store systemer er det vanlig å bruke CAD (Computer Aided Design) verktøy ved konstruksjon,

4.14 Introduksjon til effektforbruk

(Kapittel 4.4 side 187 - 188)

Statisk eller komplementær CMOS porter er svært effektive med hensyn på effektforbruk fordi når utgangen på en port har stabilisert seg på enten 1 eller 0 så går det nesten ikke strøm gjennom transistorene. Dette betyr at effektforbruket er tilnærmet lik 0 når utgangen er stabil. Historisk har effektforbruk vært mindre viktig enn hastighet og arealforbruk for en krets. For moderne CMOS prosesser, der antall transistorer og porter er svært høyt og klokkefrekvensen øker, blir effektforbruk stadig viktigere.

Vi kan definere *effektforbruk* som trekkes fra spenningsforsyningen V_{DD} som:

$$P(t) = i_{DD}(t) \cdot V_{DD}, \quad (4.31)$$

der $i_{DD}(t)$ er strømmen som trekkes fra spenningsforsyningen.

Energiforbruket over en tidsperiode T kan modelleres ved å integrere effektforbruket:

$$E = \int_0^T i_{DD}(t) \cdot V_{DD} dt. \quad (4.32)$$

Gjennomsnittelig effektforbruk over dette intervallet er gitt av:

$$\begin{aligned} P_{avg} &= \frac{E}{T} \\ &= \frac{1}{T} \int_0^T i_{DD}(t) \cdot V_{DD} dt. \end{aligned} \quad (4.33)$$

Det er to hovedkomponenter i effektforbruk:

- *Statisk effektforbruk.*

- Svak inversjonsstrøm i transistorer som er skrudd av.
- Tunnelingsstrøm gjennom gate oksid (tynnoksid).
- Lekkasje i reversforspente dioder (pn-overganger).
- Strøm i transistorer som skal overstyres av andre transistorer, i for eksempel pseudo nMOS logikk.

- *Dynamisk effektforbruk.*

- Opp- og utlading av kapasitanser.
- Kortslutningsstrøm i korte tidsperioder når både opp- og nedtrekk er PÅ.

4.14.1 Mål

Forstå enkle modeller for effektforbruk og grunnleggende forskjeller på statisk- og dynamisk effektforbruk.

4.15 Statisk effektforbruk

(Kapittel 4.4.1 side 188 - 190)

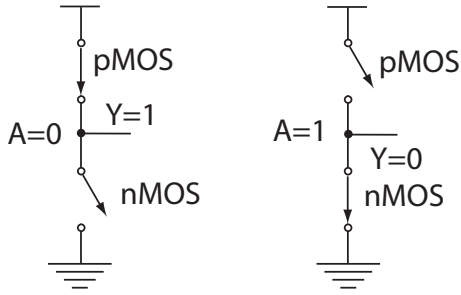


Figure 4.30: Statisk effektforbruk i en CMOS inverter. (FIG4.26)

En komplementær eller statisk inverter er vist i Fig. 4.30. Når inngangen er 0 vil pMOS transistoren være PÅ og nMOS transistorene være AV, og utgangen vil være høy (1). Når inngangen er 1 vil pMOS transistoren være AV og nMOS transistorene være PÅ, og dermed vil utgangen være lav. Dette er stabile tilstander for en CMOS port og ideelt vil det da ikke gå strøm mellom V_{DD} og GND . En transistor som er skrudd AV vil likevel levere noe strøm som er gitt av modellen for transistorstrøm i svak inversjon med $V_{gs} = 0$:

$$I_{statisk} = I_{ds0} e^{\frac{-V_t}{nV_T}} \left(1 - e^{\frac{-V_{DD}}{V_T}} \right), \quad (4.34)$$

der n er slope faktor og

$$I_{ds0} = \beta v_T^2 e^{1.8}. \quad (4.35)$$

Som vi ser er den statiske strømmen som går gjennom en transistor som er skrudd av eksponensielt avhengig av terskelspenningen. Etterhvert som teknologien utvikles vil terskelspenningen bli redusert og dermed vil den statiske strømmen øke. Det *statistiske effektforbruket* kan da modelleres som:

$$P_{statisk} = I_{statisk} V_{DD}. \quad (4.36)$$

Som en konsekvens av stadig tynnere gateoksid vil det for moderne CMOS prosesser (med tynnere gateoksid enn 20Å) være en lekkasje mellom kanal og gate gjennom gateoksidet. Denne tunnerlingen som kalles gate lekkasje er merkbar for CMOS prosesser fra gatelengder lik 130nm, med tynnsid mindre enn 20Å, og under.

I tillegg vil det være en liten lekkasje i pn-overganger, særlig mellom diffusjonsområder og substrat eller brønn. I moderne prosesser er denne lekkasjen mindre enn lekkasjestrømmer i transistorer. Det er derfor vanlig å neglisjere denne lekkasjeeffekten.

Noen logikkstiler, som for eksempel pseudo nMOS, vil ha et betydelig innslag av statisk effektforbruk.

4.15.1 Mål

Kunne modellere statisk effektforbruk, med lekkasjestrømmer via transistor som er skrudd AV, gate lekkasje og lekkasje i pn-overganger.

4.16 Introduksjon til interkonnekt

(Kapittel 4.5 side 196 - 197)

Interkonnekt, dvs. sammenkobling av delkretser, porter og transistorer er av stor betydning for ytelsen til en brikke. I moderne CMOS prosesser har en designer tilgang på mange metallag som typisk brukes til å koble sammen kretsmoduler globalt, porter relativt lokalt og transistorer lokalt. Ulike metallag har ulike elektriske egenskaper i form av egenmotstand og kapasitans, og vil derfor egne seg for å transportere ulike signaler. Kritiske signaler, for eksempel spenningsreferanser V_{DD} og GND og klokkesignaler, har høy prioritet og rutes typisk med gode ledere med liten motstand og kapasitans. Signaler som skal rutes globalt, dvs. fra en del av brikken til en annen del, må rutes slik at tidsforsinkelsen i *interkonnekt* (*ruting*) blir kritisk. Det er vanlig å bruke buffer for å drive signaler over en viss avstand (typisk mm).

w(nm)	s(nm)	t(nm)	Lag
800	800	1700	6
		1000	
750	750	1600	5
		800	
600	600	1200	4
		750	
400	400	800	3
		700	
300	300	700	2
		700	
250	250	500	1
		800	
Substrat			

Figure 4.31: Metallag i en hypotetisk CMOS prosess 200nm. (FIG4.31)

Ulike metallag i en typisk 200nm CMOS prosess er vist i Fig. 4.31. De ulike lagene vil variere med hensyn til kapasitiv last, men først og fremst egenmotstand. Dessuten vil moderne CMOS prosesser typisk tilby metalledere i ulike metaller som har liten egenmotstand.

4.16.1 Mål

Få en enkel oversikt over bredde på metalledere og avstand mellom metallag.

4.17 Motstand i interkonnekt

(Kapittel 4.5.1 side 198 - 200)

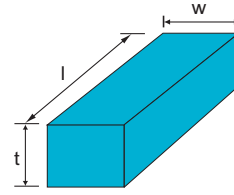


Figure 4.32: Rektangulær metalleder. (FIG4.32)

Motstand i et ledende materiale med tykkelse t , lengde l og bredde w som vist i Fig. 4.32, kan uttrykkes som:

$$R = \frac{\rho l}{t w}, \quad (4.37)$$

der ρ er materialets *egenmotstand*. Vi kan uttrykke motstanden på formen:

$$R = R_{\square} \frac{l}{w}, \quad (4.38)$$

der $R_{\square} = \rho/t$ er enhets egenmotstand i Ω/square .

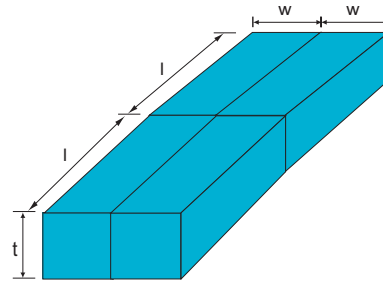


Figure 4.33: Sammensatte rektangulær metalledere. (FIG4.32)

Dersom vi setter sammen 4 rektangulære metalledere får vi en metalleder med dobbel bredde og dobbel lengde som vist i Fig. 4.33. Vi kan beregne den effektive motstanden for lederen i Fig. 4.33 ved:

$$\begin{aligned} R_{effektiv} &= R_{\square} \frac{2l}{2w} \\ &= R_{\square} \frac{l}{w} \end{aligned} \quad (4.39)$$

Vi ser at den effektive motstanden for en metalleder vil dobles dersom vi dobler langden og halveres dersom vi dobler bredden. En dobling av både bredde og lengde gir derfor ingen endring i den effektive motstanden, men vil resultere i fire ganger så stort arealet og dermed en betydelig økning i kapasitans.

Metall	Egenmotsand ($\mu\Omega \cdot cm$)
Sølv (Ag)	1.6
Kobber (Cu)	1.7
Gull (Au)	2.2
Alluminium (Al)	2.8
Tungsten (W)	5.3
Molybden (Mo)	5.3
Titanium (Ti)	43.0

Table 4.2: Egenmotstand for ulike metaller ved 22°C.



Figure 4.34: Mange viakontakter for å redusere motstand. (FIG4.33)

I tabell 4.2 er egenmotstand for ulike metaller oppgitt. Tradisjonelt har metallet som har vært anvendt i CMOS vært utelukkende aluminium. Vi ser at kobber har vesentlig mindre egenmotstand enn aluminium. Dette er grunnen til at man i moderne CMOS prosesser har begynt å bruke kobber for enkelte metallag. Signaler som er kritiske med hensyn på tidsforsinkelse bør rutes i kobber dersom det er tilgjengelig.

Metall	Egenmostand (Ω/\square)
Diffusjon (salicid)	3-10
Diffusjon	50-200
Polysilisium (salicid)	3-10
Polysilisium	50-400
Metall 1	0.08
Metall 2	0.05
Metall 3	0.05
Metall 4	0.03
Metall 5	0.02
Metall 6	0.02

Table 4.3: Effektiv motstand for ulike lag i en CMOS prosess.

Typisk effektiv motstand for ulike lag i en typisk 180nm CMOS prosess er vist gitt i tabell 4.3. De øvre metallagene har lavere effektiv motstand fordi de er tykkere (t). Effektiv motstand i polysilisium, diffusjon og brønner er avhengig av dopenivået.

Kontakter og viakontakter (kontakter mellom ulike metallag) har effektiv motstand som er avhengig av materialet som kobles sammen og kontaktens størrelse. Typiske verdier for kontakter er 2 – 20 Ω . Det er mer effektivt, dvs. resulterer i mindre total motstand, å bruke mange små kontakter enn en stor kontakt som vist i Fig. 4.34.

4.17.1 Mål

Kunne beregne effektiv motstand i ulike lag i en CMOS prosess.

4.18 Dielektrikum

Et *dielektrikum* fungerer som en elektrisk isolator, dvs. et materiale som har meget høy motstandsverdi. Et dielektrikum mellom to ledende lag vil utgjøre en kapasitans per arealenhet som kan uttrykkes som

$$C = \frac{k\epsilon_0}{d}, \quad (4.40)$$

der k^4 er *relativ permittivitet* til dielektrikum, ϵ_0 er permittivitet i vakum og d er tykkelsen på det dielektriske materialet. Ofte blir relativ permittivitet uttrykt som *dielektrisk konstant*. Dielektrisk konstant og relativ permittivitet er synonyme begreper.

For en MOS transistor har vi $k = \epsilon_{ox}/\epsilon_0$ og $d = t_{ox}$.

4.18.1 Lav-k (Low-k)

Et *lav-k dielektrikum* har lav dielektrisk konstant. Lav dielektrisk konstant vil gi raskere kretser, både fordi kapasitans knyttet til transistorer (gate kapasitans) og kapasitans i ledere vil bli redusert, men dette vil gå på bekostning av lekkasjestrømmer som for eksempel tunnelering fra og til gate. Det eksperimenteres med ulike dielektrikum med lavere dielektrisk konstant for å øke hastigheten til digitale systemer.

4.18.2 Høy-k (High-k)

For å redusere lekkasjestrømmer i dielektriske materialer eksperimenteres det med dielektriske materialet som har høyere dielektrisk konstant. Dette kalles *høy-k dielektrikum*. Alternativ til høy-k vil være å øke dybden på det dielektriske materialet for å redusere lekkasjestrømmer, men da vil det bli vanskeligere å redusere størrelsen på komponenter og ledere.

4.18.3 Mål

Forstå behovet for dielektrikum og hvorfor det er behov for lav-k- og høy-k dielektrikum.

4.19 Kapasitans i interkonnekt

(Kapittel 4.5.2 side 200 - 205)

En isolert leder over substrat kan modelleres som en leder over jordplan. *Kapasitans* i ledere har to hovedkomponenter:

1. Parallell plate kapasitans fra lederen mot jordplanet.
2. Sideveis kapasitans for metalleder (fringing fields).

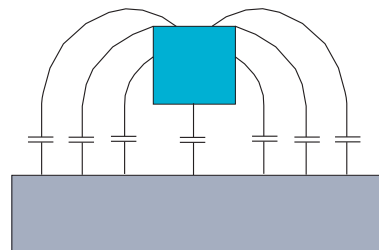


Figure 4.35: *Kapasitans knyttet til en metalleder over substrat. (FIG4.34)*

I tillegg vil en metalleder ha kapasitans mot andre metalledere i samme lag. Kapasitans for metalleder mot jordplan er illustrert i Fig. 4.35. En enkel modell for parallell plate kapasitans er gitt av:

$$C = \frac{\epsilon_{ox}}{h}wl, \quad (4.41)$$

der ϵ_{ox} er permittivitet i silisiumdioksid, w og l er bredde og lengde på metallederen og h er avstanden fra substratet til metallederen.

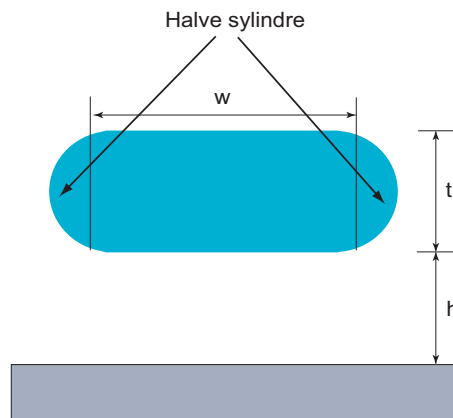


Figure 4.36: *Kapasitans modell (Yuan & Trick). (FIG4.35)*

En kapasitansmodell der metallkantene blir modellert som halve sylindre er vist i Fig. 4.36. Halvsylinder vil ha en radius lik $t/2$ og bredden på metallederen beregnes til $w - t/2$. En forholdsvis kompleks, men relativt nøyaktig modell (ca. 10% for $t/w < 2$) for kapasitans er gitt av:

⁴Merk at dette ikke er Boltzmanns konstant.

$$C = \epsilon_{ox} l \left(\frac{w - \frac{t}{2}}{h} + \frac{2\pi}{\ln \left(1 + \frac{2h}{t} + \sqrt{\frac{2h}{t} \left(\frac{2h}{t} + 2 \right)} \right)} \right) \quad (4.42)$$

der vi antar at $t \approx h$.

En empirisk modell for interkonnekt kapasitans er gitt av:

$$C = \epsilon_{ox} l \left(\frac{w}{h} + 0.77 + 1.06 \left(\frac{w}{h} \right)^{0.25} + 1.06 \left(\frac{t}{h} \right)^{0.5} \right) \quad (4.43)$$

der feilen er mindre enn 6% for $t/w < 3.3$.

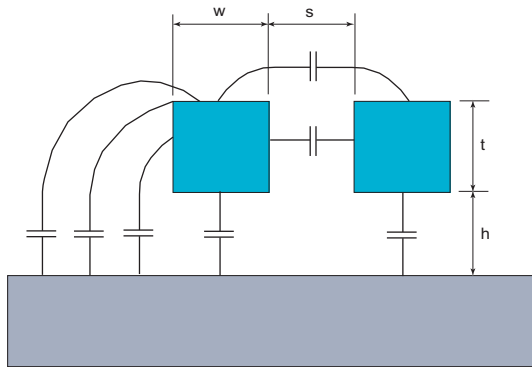


Figure 4.37: Kapasitans knyttet til en metalleder over substrat og mellom metalledere i samme lag. (FIG4.34)

Modellene som er presentert tar ikke hensyn til nærliggende metalledere i samme lag eller nærliggende metalledere i lag rett over selve lederen. Vi kan utvide modellene ved å betrakte lag (ledere) over og under som parallelle plater. Dette gir en konservativ verdi for kapasitans. I Fig. 4.37 er det vist to metalledere i samme lag med avstand s .

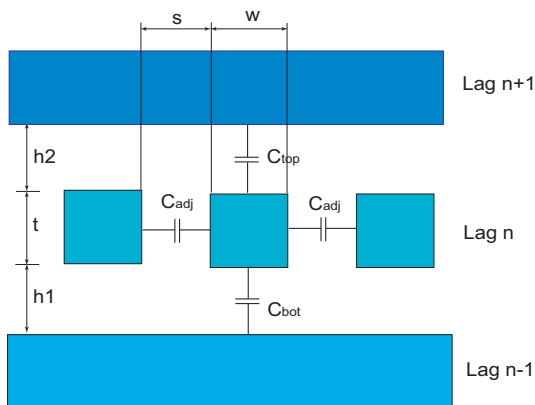


Figure 4.38: Kapasitansmodell som tar hensyn til nærliggende metalledere. (FIG4.36)

En modell for kapasitans som tar hensyn til nærliggende metalledere er vist i Fig. 4.38. Vi kan bruke en konservativ

modell for beregning av kapasitansen ved å dele opp i topp, bunn og sideveis kapasitanser:

$$\begin{aligned} C_{gnd} &= C_{bot} + C_{top} \\ C_{total} &= C_{gnd} + C_{adj}. \end{aligned} \quad (4.44)$$

Den totale kapasitansen knyttet til en metalleder vil være avhengig av avstand til andre metalledere i samme lag og metalledere i nabolag.

4.19.1 Mål

Kunne modellere kapasitans i interkonnekt.

4.20 Forsinkelse i interkonnekt

(Kapittel 4.5.3 side 205 - 207)

Det er to grunner til at interkonnekt bidrar til å øke tidsforsinkelse i en krets:

1. Ruting av signaler (i metall) vil legge last til utgangen på en port.
2. Lange ledere har signifikant motstand.

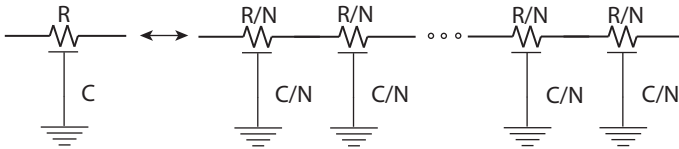


Figure 4.39: Oppdeling av en leder i N deler. (FIG4.38)

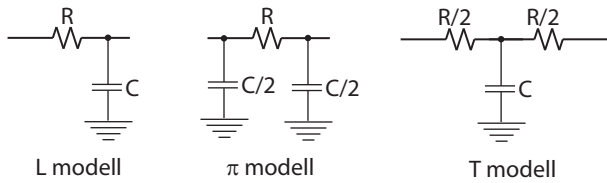


Figure 4.40: Ulike modeller for forsinkelse i interkonnekt. (FIG4.38)

Det er enkelt å utvide Elmore forsinkelsesmodell med forsinkelse i interkonnekt. Motstand og kapasitans i en leder kan approksimeres ved å dele opp lederen i små avdelinger som vist i Fig. 4.39. Det er tre standard metoder for approksimasjon som benyttes; *L modell*, *π modell* og *T modell* som vist i Fig. 4.40. *L modellen* krever et høyt antall avdelinger for å produsere et nøyaktig resultat og anvendes derfor ikke så ofte. *π modellen* gir god nøyaktighet (3% avvik) for 3 eller flere avdelinger. *L modellen* kan sammenlignes med *π modellen* men vil være mer krevende å benytte fordi antallet elektriske noder er større. Vi ser at både kapasitans og motstand i en metalleder vil øke med lengde som medfører at forsinkelse i lederen øker kvadratisk.

Det er vanligst å bruke metallag til å rute signaler (interkonnekt) på grunn av liten egenmotstand.

4.20.1 Eksempel

Gitt en 5mm lang og $0.32\mu\text{m}$ bred leder i metall 2 i en 180nm prosess med egenmotstand $0.05\Omega/\square$ og kapasitans $0.2\text{fF}/\text{cm}$. Bruk *π modell* med tre segmenter (avdelinger) og lage en modell for lederen.

3-segment *π modell* for leder er vist i Fig. 4.41. Lederen er $5000\mu\text{m}/0.32\mu\text{m}$ som utgjør 15625 arealenheter. Total motstand er $(0.05\Omega/\square \cdot 15625\square = 781\Omega)$. Total kapasitans er $(0.2\text{fF}/\mu\text{m}) \cdot (5000\mu\text{m}) = 1\text{pF}$. Hvert *π*-segment har en tredjedel av den totale motstanden og kapasitansen.

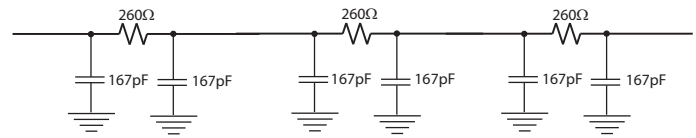


Figure 4.41: *π* modell av leder. (FIG4.39a)

4.20.2 Mål

Kunne modellere forsinkelse i en en metalleder ved hjelp av oppdeling av lederen i avdelinger med kapasitans og motstand.

4.21 Crosstalk

(Kapittel 4.5.4 side 207 - 210)

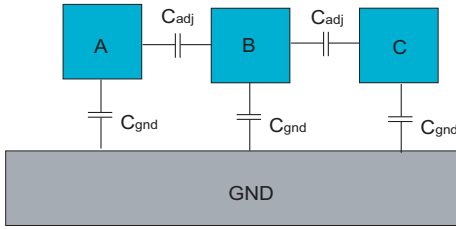


Figure 4.42: Kapasitans mellom naboledere i samme lag og til GND. (FIG4.41)

I Fig. 4.42 er det vist kapasitans mellom naboledere i samme lag og til GND. Når A svitsjer⁵ vil dette påvirke nabolederen B som også vil få en spenningsendring i samme retning. Den kapasitive påvirkningen kalles *crosstalk*. Crosstalk kan påvirke nabolederen slik at nabolederen får økt eller redusert sin egens svitsjetid. Påvirkningsgraden er avhengig av kapasitans mellom lederne og den totale kapasitans knyttet til lederen som påvirkes av crosstalk.

4.21.1 Forsinkelse

B	ΔV	$C_{eff}(A)$	MCF
Konstant	V_{DD}	$C_{gnd} + C_{adj}$	1
Svitsjing i samme retning	0	C_{gnd}	0
Svitsjing i motsatt retning	$2V_{DD}$	$C_{gnd} + 2C_{adj}$	2

Table 4.4: Crosstalk avhengighet av svitsjeretninger.

Dersom en leder og nabolederen svitsjer i samme retning vil lederne påvirke hverandre positivt, dvs. redusert, med hensyn på forsinkelse. I tabell 4.4 er det vist hvordan crosstalk påvirkes av svitsjeretninger. Ladning som overføres til en koblingskondensator er gitt av

$$Q = C_{adj}\Delta V, \quad (4.45)$$

der ΔV er spenningsendringen mellom de elektriske nodene (ledere). Dersom for eksempel A svitsjer og B ligger fast blir $\Delta V = V_{DD}$. Dersom nodene A og B svitsjer i motsatt retning blir $\Delta V = 2V_{DD}$, dette kalles *Miller effekt*. *Miller koblingsfaktor (MCF)* modellerer kapasitansen mellom to elektriske noder (ledere). En vanlig verdi for MCF er 1.5.

En konservativ modell for MCF er 2 ved beregning av propageringsforsinkelse og 0 ved beregning av contamination forsinkelse.

⁵Transisjon fra 0 til 1 eller fra 1 til 0.

Eksempel

To ledere med lengde $1mm$ har kapasitans $0.1fF/\mu m$ til jord og $0.1fF/\mu m$ til nabolederen. Hver leder blir drevet av en inverter med effektiv motstand lik $1k\Omega$. Hva blir contamination- og propageringsforsinkelsen til lederne?

Vi kan beregne contamination- og propageringsforsinkelse ved å finne de relevante kapasitansene; $C_{gnd} = C_{adj} = (0.1fF/\mu m) \cdot (1000\mu m) = 0.1pF$. Tidsforsinkelsen er gitt av RC_{eff} . Contamination forsinkelse kan beregnes ved at vi antar at nodene svitsjer i samme retning slik at $C_{eff} = C_{gnd}$ og dermed $t_{cd} = (1k\Omega) \cdot (0.1pF) = 100ps$. Ved beregning av propageringsforsinkelse antar vi at lederne svitsjer i motsatt retning slik at $C_{eff} = C_{gnd} + 2C_{adj} = 0.3pF$ som gir $t_{pd} = (1k\Omega) \cdot (0.3pF) = 300ps$.

4.21.2 Crosstalk støy

Når to ledere ligger forholdsvis nær hverandre vil de kunne påvirke hverandre elektrisk gjennom parasittiske (crosstalk) kapasitanser. En slik påvirkning er derfor kapasitiv. Anta et en leder B skal ligge på en fast spenningen og at en leder A svitsjer. Dersom A påvirker spenningen på B gjennom crosstalk kaller vi dette for *crosstalk støy*. I dette tilfellet kaller vi A for *aggressor* og B for *victim*.

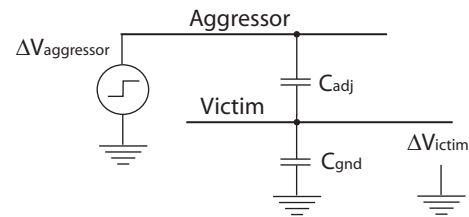


Figure 4.43: Aggressor og victim. (FIG4.42)

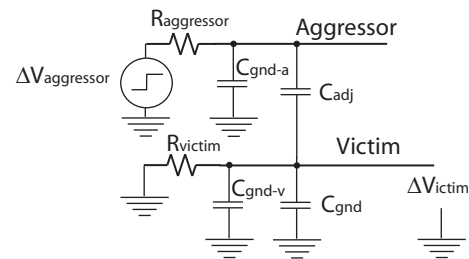


Figure 4.44: Aggressor og victim med drivere. (FIG4.43)

I Fig. 4.43 ser vi to ledere med kapasitansen C_{adj} mellom lederne. Den ene lederen (victim) påvirkes av spenningsendring på den andre lederen (aggressor):

$$\Delta V_{victim} = \frac{C_{adj}}{C_{gnd} + C_{adj}} \Delta V_{aggressor}, \quad (4.46)$$

der $\Delta V_{aggressor}$ er spenningsendring på aggressor lederen. Dersom victim lederen drives vil strømmen som driveren leverer redusere crosstalk støy for victim. Dette kan modelleres som

4.22 Bredde og avstand for ledere

(Kapittel 4.6.1 side 219 - 220)

$$\Delta V_{victim} = \left(\frac{C_{adj}}{C_{gnd} + C_{adj}} \right) \left(\frac{1}{1+k} \right) \Delta V_{aggressor} \quad (4.47)$$

der

$$\begin{aligned} k &= \frac{\tau_{aggressor}}{\tau_{victim}} \\ &= \frac{R_{aggressor} (C_{gnd-a} + C_{adj})}{R_{aggressor} (C_{gnd-v} + C_{adj})}, \end{aligned} \quad (4.48)$$

der C_{gnd-a} og C_{gnd-v} er henholdsvis kapasitans for aggressor- og victim til jord som vist i Fig. 4.44. Crosstalk støy er mest dominerende når victim er udrevet eller svakt drevet i forhold til aggressor, dette medfører at $k < 1$.

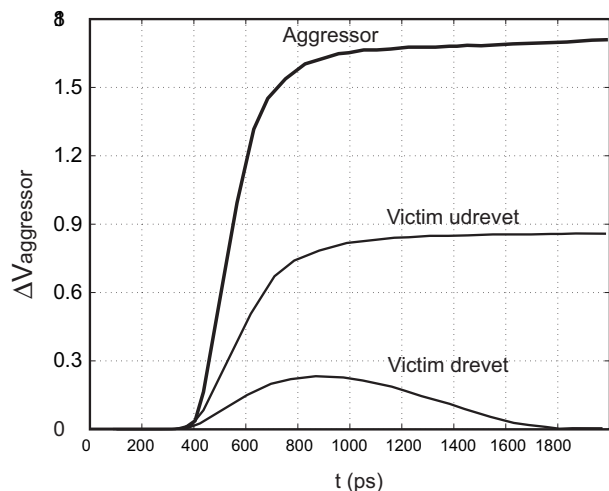


Figure 4.45: Crosstalk. (FIG4.44)

Effekten av crosstalk er vist i Fig. 4.45.

4.21.3 Mål

Kunne estimere contamination forsinkelse og propageringsforsinkelse inkludert crosstalk kapasitanser. Etablere en enkel forståelse for hvordan crosstalk introduserer støy i integrerte kretser.

Tidsforsinkelse i porter avtar for moderne prosesser på grunn av redusert kapasitans som følge av redusert areal på transistorer. Lange ledere derimot vil ikke gi reduksjon i tidsforsinkelse som følge av teknologiutviklingen i CMOS. Tidsforsinkelsen i lange ledere kan til og med øke fordi lederne blir smalere og grunnere som vil resultere i større egenmotstand. Dette betyr at design av ledere blir et stadig viktigere felt innenfor design av integrerte kretser. Det er viktig å planlegge en krets med hensyn på å finne lange ledere. I alle design vil det forekomme lange ledere. Designeren kan velge metallag, bredde og avstand på lederne for å minimere problemer med forsinkelse i lange ledere. Videre er det viktig å vurdere shielding, eller beskyttelse, av ledere for å minimere crosstalk.

Designeren velger *bredde på ledere, avstand mellom ledere* og hvilket metallag som skal brukes. Vanligvis brukes minstebredder og minsteavstander for ikke-kritiske signaler, som vil gi størst mulig tetthet og minst mulig areal. Dersom vi velger bredere ledere for å redusere motstanden får vi en økning i kapasitansen som er noe mindre enn reduksjonen i motstanden, slik at tidsforsinkelsen går noe ned. Økt bredde på ledere kan også redusere crosstalk fordi det blir en mindre andel av kapasitansen som går til metalledere i lag over og under. Dersom vi øker avstanden til andre ledere (samme lag) vil kapasitansen til de nærliggende lederne i samme lag bli redusert uten endring i motstanden.

En prosess er ofte karakterisert gjennom wire *pitch*:

$$pitch = w + s, \quad (4.49)$$

der w er bredde på lederen og s er avstanden til andre ledere.

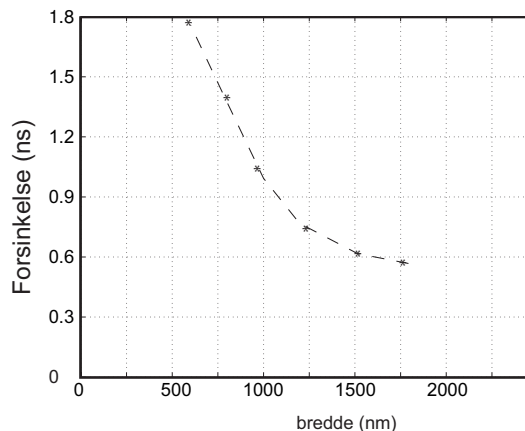


Figure 4.46: Tidsforsinkelse i metalleder på 10mm som funksjon av pitch. (FIG4.51(a))

I Fig.4.46 er tidsforsinkelse for en 10mm metalleder som funksjon av bredden på lederen vist. Relativ kapasitans til naboledere i samme lag for en 10mm metalleder er vist

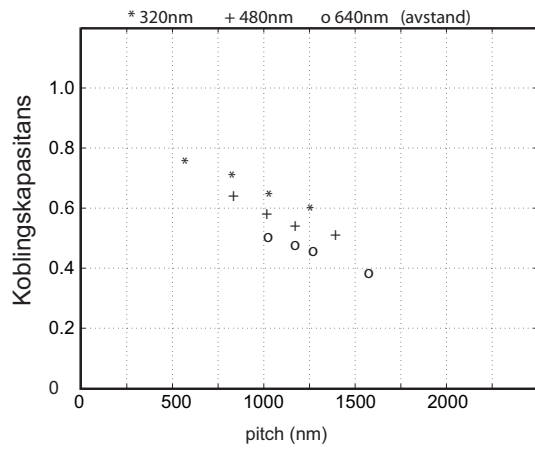


Figure 4.47: Koblingskapasitans, $2C_{adj}/(2C_{adj} + C_{gnd})$, mellom metalleder på 10mm som funksjon av pitch. (FIG4.51(b))

i Fig.4.47. Som hovedregel øker vi bredden på ledere for å redusere tidsforsinkelsen og øker avstanden til naboledere for å redusere crosstalk.

4.22.1 Mål

Forstå hvordan bredde på ledere og avstand mellom ledere påvirker tidsforsinkelse i lange ledere og crosstalk mellom lange ledere.

4.23 Valg av metallag for ledere

(Kapittel 4.6.2 side 219 - 221)

De første MOS prosessene hadde bare ett metallag i motsetning til moderne prosesser som har minst seks metallag. De nederste lagene gir grunne ledere som er optimalisert for lokal ruting. De midtre lagene er noe tykkere som gir mindre motstand og tåler mer strøm. De øverste lagene har lav motstand og brukes til distribusjon av forsyningsspenninger, klokkesignaler og raske globale signaler. På grunn av meget stor kompleksitet i moderne integrerte kretser, behov for distribusjon av forsyningsspenninger og klokkesignaler, er det et behov for mange metallag og det er svært viktig at lagene brukes fornuftig. Utnyttelse av metallagene bør gjøres ved planleggingen av en krets organisert med moduler.

Lag	Anvendelse
Metall 1	Lokal interkonnekt
Metall 2/3	Interkonnekt mellom små moduler
Metall 4/5	Interkonnekt mellom større moduler og kritiske signaler
Metall 6	I/O, klokkesignaler og spenningsforsyninger

Table 4.5: Typisk bruk av metallag.

Typisk bruk av metallag er gitt i tabell 4.5. Spenningsforsyninger blir i praksis distribuert over flere lag avhengig av strømtrekk og lokal interkonnekt. Det er fornuftig å dedikere to metallag til henholdsvis V_{DD} og GND for å redusere crosstalk problemer.

4.24 Beskyttelse av ledere

(Kapittel 4.6.3 side 221)

Som kjent kan crosstalk mellom naboledere representere et betydelig problem med hensyn på tidsforsinkelse og signalverdier. Dette problemet kan reduseres dersom to naboledere ikke svitsjer. Det er vanlig praksis å beskytte (*shield*) ledere som ligger i nærheten av hverandre. Det legges i faste spenningsreferanser, typisk V_{DD} eller GND , tett mot de signalførende lederne. Dette vil medføre økt arealbehov men mindre støy og redusert tidsforsinkelse på grunn av mindre crosstalk. Klokkesignaler beskyttes for å motvirke forsinkelse (clockskew). I mixed-signal, dvs. analoge og digitale signaler på samme integrerte krets, er det svært viktig å beskytte de analoge signalene mot digital støy. Man kan også utnytte spesifikk kjennskap til de aktuelle signalene som skal rutes i nærheten av hverandre. For eksempel dersom signal A alltid er konstant når nabosignal B svitsjer og omvendt kan disse signalene legges forholdsvis nær hverandre uten at det blir signifikant crosstalk. Nabolederen vil da kunne fungere som beskyttelse mot andre signaler. Vi antar at alle signaler er drevet slik at en kapasitiv påvirkning fra naboer ikke representerer et støyproblem.

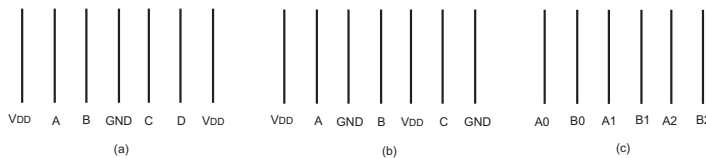


Figure 4.48: Ulike strategier for beskyttelse av ledere. (FIG4.52)

Ulike strategier for beskyttelse av signaler mot crosstalk er vist i Fig. 4.48. Til venstre (a) er det lagt inn spenningsreferanse mellom hvert andre signal, i midten (b) er det lagt spenningsreferanse mellom hvert signal, for eksempel klokkesignaler, og til høyre (c) signaler som ikke svitsjer samtidig lagt ved siden av hverandre.

4.25 Designmarginer og variasjoner

(Kapittel 4.7.1-3 side 231 - 233)

Responsten til integrerte kretser kan variere som følge av omgivelser og fabrikasjon. Det er vanlig å ta hensyn til noen viktige variasjonskilder ved design:

1. Forsyningsspenning.
2. Temperatur.
3. Prosessvariasjoner.

Det er viktig å designe kretser slik at de gir korrekt respons over et intervall av ekstremverdier for variasjonskildene. Dette gir et robust og pålitelig design som vil hindre et system i å feile katastrofalt.

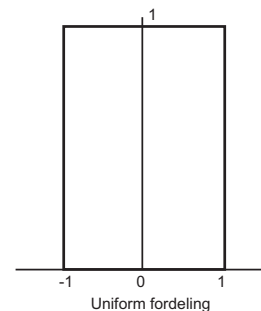
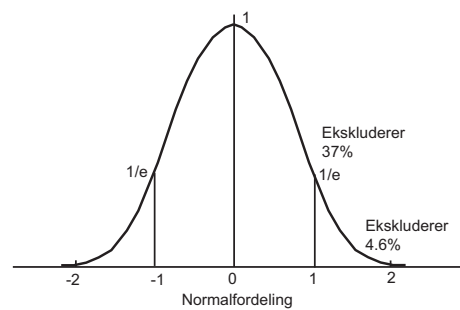


Figure 4.49: Uniform og normal fordeling. (FIG4.58)

Variasjoner kan modelleres med uniform eller normal (Gauss) statistiske fordelinger som vist i Fig. 4.49. For uniform fordeling er det vanlig å definere feks. variasjon i forsyningsspenning $\pm 10\%V_{DD}$ som vil gi et meget robust design. Det er da viktig at hele kretsen fungerer korrekt med den spesifiserte variasjonen. Normalfordeling spesifiseres med et standardavvik ρ . Prosess variasjoner er vanligvis modellert som en normalfordeling. Dersom kretseksemplarer som ligger utenfor 3ρ skal forkastes, tilsvarer dette 0.26% av eksemplarene. En grense på 2ρ tilsvarer 4.6%. Grensene 2ρ og 3ρ er vanlig å bruke.

4.25.1 Variasjon i forsyningsspenning

Når man designer en krets bruker man normalt en ideel forsyningsspenning som er tilpasset den prosessen man

skal realisere designet i. Det er mange årsaker til at forsyningsspenningen kan variere globalt, gjennom mismatch i spenningsregulatorer, og lokalt gjennom spenningsstap i ledere. Det er fornuftig å designe kretsene slik at logikken fungerer for en spesifikk klokkefrekvens når forsyningsspenningen variere med 10%. Hastigheten er proporsjonal med forsyningsspenningen slik at det er vanlig å designe med en god margin.

4.25.2 Variasjon i temperatur

Effektforbruk vil resultere i temperaturstigning i en krets. Når temperaturen øker vil transistorstrømmene minke. Den faktiske temperaturen lokalt i en integrert krets er avhengig av både omgivelsestemperatur og lokal påvirkning pga effektforbruk. Det er vanlig å spesifisere temperaturgrenser for omgivelsene som kretsene skal kunne fungere i. For kommersielle produkter er vanlige grenser $0^{\circ}C$ og $70^{\circ}C$, for industrielle produkter er grensene $-40^{\circ}C$ og $85^{\circ}C$ og for militære produkter er grensene $-55^{\circ}C$ og $125^{\circ}C$.

4.25.3 Prosessvariasjoner

Transistorer og ledere i en integrert krets vil bli fremstilt ved prosessering med en viss variasjon rundt nominelle verdier. Dette gjelder særlig selve størrelsene eller geometriene og dopekonsentrasjoner. Disse variasjonene kan oppstå innenfor en enkelt integrert krets og på en wafer.

De mest kritiske variasjonene er kanallengde L , tykkelsen på tynnsid t_{ox} og terskelspenning V_t . Terskelspenningene varierer blant annet på grunn av variasjon i dopekonsentrasjoner. For interkonnekt er de mest kritiske variasjonene lederbredde og avstand mellom ledere.

4.26 Designhjørner.

(Kapittel 4.7.4 side 233-235)

Det er vanlig å samle alle variasjoner og simulere med tre ulike parametersett som representerer *designhjørner*

- *Typisk*, eller nominell. Her brukes nominelle eller typiske parameterverdier.
- *Rask*. Parameterverdier som gir raskest mulig krets. Liten transistorlengde, stor bredde, lav terskelspenning ol.
- *Treg*. Omvendt av rask.

Design- eller prosesshjørner defineres av parameterverdier for transistorer og interkonnekt. Det kan ofte være lurt å simulere for rask nMOS og treg pMOS eller omvendt for å se på responsen for ulike situasjoner som vist i Fig. 4.50, der T stå for nominell eller typisk, F for rask (fast) og S står for treg (slow).

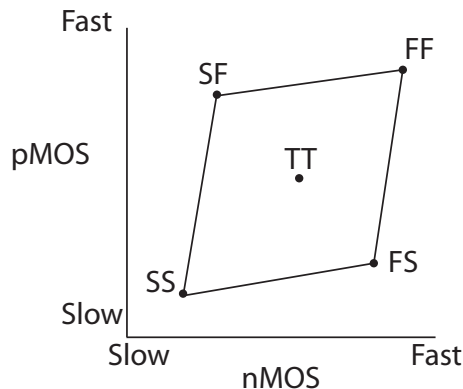


Figure 4.50: Designhjørner. (FIG4.59)

nM.	pM.	Led.	V_{DD}	T.	Anvendelse
T	T	T	S	S	Timing (spesifikasjon) (typisk)
S	S	S	S	S	Timing (spesifikasjon) (konservativ)
F	F	F	F	F	DC effektforbruk Timing og støy
F	F	F	F	S	Lekasje og støy
S	S	F	S	S	Treg logikk og raske ledere
F	F	S	F	F	Rask logikk og trege ledere
S	F	T	F	F	Pseudo-nMOS, støy-margin og pMOS mot nMOS
F	S	T	F	F	nMOS mot pMOS

Table 4.6: Designhjørner. $pM = pMOS$, $nM = nMOS$, $Led = leder$ og $T = temperatur$

Typiske simuleringsoppsett er vist i Tabell 4.6. Det er viktig å simulere med alle relevante hjørner for å verifisere at et design er robust. Det er viktig å kjenne til de enkelte parameterverdiene for hjørnesimuleringer. For forsyningsspenning V_{DD} vil F typisk bety 10% over nominell verdi og 10% under representerer S . For temperatur kan T bety $70^{\circ}C$, F $0^{\circ}C$ og S $125^{\circ}C$

4.27 Matching.

(Kapittel 4.7.5 side 235 - 237)

Som regel ønsker man at to identiske transistorer, dvs. tegnet identiske, skal ha like karakteristikker. Dette gjelder spesielt inngangstrinn på sense-amplifiser. Andre kritiske kretselementer er nettverk som distribuerer klokkesignaler fordi det er kritisk at det ikke er for stor variasjon i timing av klokkeflanker. I praksis modellerer man ikke to identiske nærliggende transistorer som henholdsvis rask og treg. Dette vil gi et urealistisk bilde og gjøre det nesten umulig å designe funksjonelle kretser. På den andre siden er det viktig å være klar over at to transistorer som er tegnet like ikke vil være helt like elektrisk. Det er viktig å få en oversikt over den realistiske spredningen eller *mismatch* som gjelder for den aktuelle prosess. Det er forskjellige grunner til at relevante data om matching ikke foreligger. Det er ofte slik at fabrikanten ikke ønsker å spre slik informasjon av hensyn til andre aktører i markedet.

Mismatch forekommer både i form av systematisk variasjon og usikkerhet eller tilfeldig variasjon. Systematisk variasjon har utgangspunkt i et kvantitativt forhold til en kilde. Eksempler på systematiske variasjoner er variasjon i doping ved fabrikasjon av kretser. Tilfeldige variasjoner forekommer, men uten at man kan finne kilden på en fornuftig måte. Systematiske variasjoner kan modelleres og i noen grad kompenseres, mens tilfeldige variasjoner må man ta høyde for ved design.

Variasjoner i terskelspenning for transistorer og transistorstrøm modellerer man med skalleringsfaktoren $1/\sqrt{WL}$. Transistorparametre er avhengig av størrelse og orientering. Det er derfor fornuftig å designe identiske transistorer som skal ligge i nærheten av hverandre for å oppnå maksimal matching. Lange transistorer har mindre variasjon eller mismatch enn korte transistorer. Et annet forhold som påvirker matching er tettheten av polysilisium som ligger i nærheten av transistorene. Der matching er spesielt kritisk, som for eksempel for klokke drivere, kan man legge identiske mønstre av polysilisium i nærheten av transistorene. Variasjon i terskelspenning skyldes en statistisk fordelt variasjon av dopeatomer i kanalen. Når transistorene blir mindre vil det være færre dopeatomer i kanalen og den relative matchingen vil da forverres.

Problemer med matching kan karakteriseres som systematiske, tilfeldige, drift eller jitter. Systematisk mismatch kan modelleres, simuleres og kompenseres i designfasen. Tilfeldige variasjoner skyldes forhold som ikke kan detekteres eller er for kostbart å modellere. Variasjoner som skyldes drift, spesielt endring i temperatur, er langsomme i forhold til klokkefrekvensen i et system. Drift kan modelleres og kompenseres. Jitter skyldes variasjoner i spenningsforsyningen og er den alvorligste kilden til mismatch. Jitter er spenningsendringer som kan komme like raskt eller raskere enn systemets klokkefrekvens og kan derfor ikke fjernes ved å bruke tilbakekoblinger.

4.28 Pålitelighet.

(Kapittel 4.8.1 side 239 - 240)

Et hyppig forekommende design problem er synkronisering av klokkesignaler. Det er avgjørende at designeren kan konstruere systemer slik at tidsforsinkelser blir like eller er riktig i forhold til timingkrav i kretsene. Den beste metoden for å få mest mulig lik tidsforsinkelse er å duplisere delkretser.

Det å konstruere pålitelige kretser involverer kunnskaper om potensielle situasjoner som kan forårsake feil. En rekke feil kan medføre permanente feil

- *Elektromigrasjon.*
- *Self-heating.*
- *Hot carriers.*
- *Latchup.*
- *Overspenningsfeil.*

Dette er såkalte *harde feil* som vil sørge for at systemet får permanente alvorlige feil og mister data.

Det er ulike begreper som brukes for å beskrive pålitelighet. *Gjennomsnittelig tid mellom feil* er det vanligste begrepet. I tillegg brukes antall feil over tid, dvs. antall feil i løpet av 1000 timer per million transistorer. For eksempel vil et system med en integrert krets som har 1000 feil hver 10^6 time feile en gang hvert 114 år. Et system med 100 integrerte kretser vil få en feil hvert 1.14 år (420 dager).

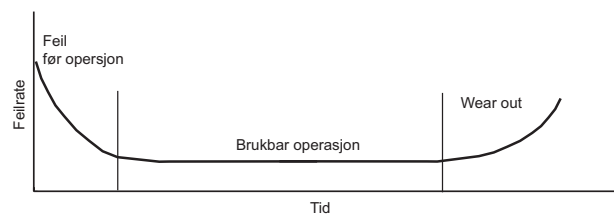


Figure 4.51: *Badekarkurven for pålitelighet. (FIG. 4.61)*

De fleste systemer gjennomgår den såkalte badekarkurven som er vist i Fig. 4.51. Før et system er i operativ drift detekteres komponenter som ikke fungerer. De fungerende komponentene vil deretter inngå i operative systemer i systemets naturlige levetid der feilraten er lav. Etterhvert som systemets levealder blir høy vil hyppigheten av feil øke, dette kalles *wear out*. Man kan estimere levealder ved å foreta tester der man stresser systemet (*burn in*), for eksempel med høy temperatur, for å simulere aldringsprosessen for komponenten eller systemet.

4.29 Elektromigrasjon

(Kapittel 4.8.2 side 240)

Elektromigrasjon skaper utarming av metalledere. Høy strømtetthet fører til såkalt elektronvind som forårsaker at metallatomene migrerer over tid og kan bidra til at lederen ødelegges. Problemet er spesielt stort for ledere av aluminium og vesentlig mindre for kobber. Elektromigrasjon er avhengig av strømtetthet $J = I/wt$ og er vanligere i ledere som fører DC-signaler (strømmer) enn AC-signaler (strømmer). Gjennomsnittelig tid før feil (MTTF) er svært avhengig av temperatur og kan modelleres som

$$MTTF = \frac{e^{\frac{E_a}{kT}}}{J_{dc}^n},$$

der E_a er aktiveringsenergien som kan finnes eksperimentelt ved å teste systemet under stress ved høy temperatur, J_{dc} er maksimal elektromigrasjonstrøm og $n \approx 2$. J_{dc} er avhengig av materialer og prosessering, og ligger typisk i området $1 - 2 \text{ mA}/\mu\text{m}^2$ for aluminium ved 110°C og $10 \text{ mA}/\mu\text{m}^2$ for kobber.

Elektromigrasjon er først og fremst et problem for unidireksjonale dc ledere på grunn av vedvarende og enrettet strøm.

4.30 Varmeutvikling (self-heating)

(Kapittel 4.8.3 side 241)

For bidireksjonale ledere vil elektromigrasjon ikke representere et stort problem, men strømtettheten må likevel begrenses. Årsaken til dette er *varmeutvikling (self-heating)* som følge av effektforbruk i lederen. Stort effektforbruk vil heve temperaturen lokalt ved lederen og dermed øke motstanden og tidsforsinkelsen. Stor varmeutvikling i ledere kan forårsake eller styrke elektromigrasjonsproblemer i bidireksjonale ledere. Korte, men høye strømpulser kan smelte ledere. Varmeutvikling i ledere er avhengig av strømtetthet og modelleres som

$$I_{rms} = \sqrt{\frac{\int_0^T I(t)^2 dt}{T}}.$$

En vanlig regel for kontroll av pålitelighetsproblemer som følge av varmeutvikling i ledere er å holde $J_{rms} < 15 \text{ mA}/\mu\text{m}^2$ for bidireksjonal aluminiumsledere i silisiumsubstrat.

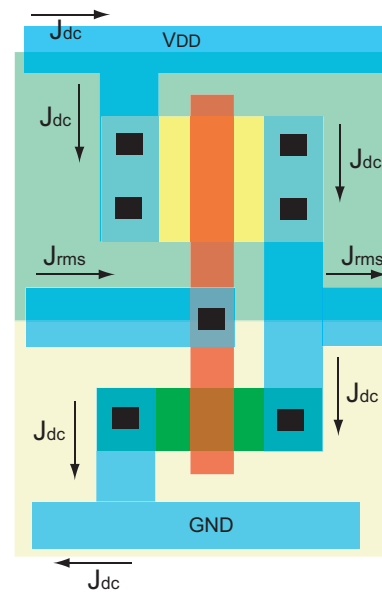


Figure 4.52: Begrensinger for strømtetthet i en inverter.

Begrensing for strømtetthet på grunn av varmeutvikling og elektromigrasjon for en inverter er vist i Fig. 4.52.

4.31 Hot carriers

(Kapittel 4.8.4 side 241 - 242)

Når transistorer skrus på kan noen ladningsbærerne, dvs. elektroner, med tilstrekkelig høy energi injiseres inn i gateoksidet og bli fanget der. Dette er såkalte *hot carriers* som vil forårsake skader i tynnoksidet og dermed endre transistorens strøm karakteristikk. Endringen vil være reduksjon i strøm for en nMOS transistor og økning i strøm for en pMOS transistor. Hot carriers forårsaker slitasje fordi nMOS transistorene blir for trege sammenlignet med pMOS transistorene. Dett er spesielt merkbart for NOR porter. Et beslektet problem er *negativ bias temperatur instabilitet (NBTI)* der hull fanges i tynnoksidet i pMOS transistorer og dermed reduserer strømmen i pMOS transistorene. Problemer med hot carriers er påvirket av feltstyrken i transistorkanalene og dermed direkte påvirket av drain-source spenning V_{ds} .

4.32 Latchup

(Kapittel 4.8.5 side 242 - 244)

* Bipolare transistorer. (FYS1210)

Ved introduksjonen av CMOS teknologien hadde kretsene en tendens til å danne forbindelser med liten motstand mellom V_{DD} og gnd som forårsaket en katastrofal kortslutning i kretsene. Fenomenet kalles *latchup* og er forårsaket av biaserte latente bipolare transistorer mellom substrat, brønn og diffusjon.

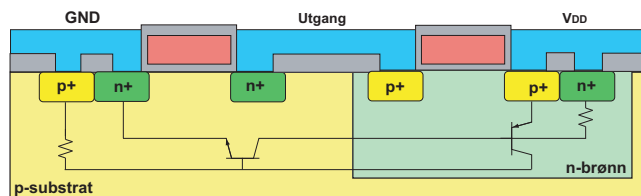


Figure 4.53: Latchup.

Latchup i en inverter er vist i Fig. 4.53. Som vi ser er det mange mulige npn- og pnp overganger som kan resultere i bipolare transistorer dersom biaseringsbetingelsene for det er tilstede. Normalt vil substratspenningen $V_{sub} = GND(OV)$ og $V_{brønn} = V_{DD}$ være tilstrekkelig til å holde npn- og pnp overganger ubiasert. Det er ulike årsaker til at de bipolare transistorene blir biasert slik at de begynner å levere strøm og påvirke substrat- og brønnspenninger. Ulike støypulser i form av uønskede spenningsving kan oppstå, gjerne via eksterne innganger. Støypulsene kan biasere de bipolare transistorene og dermed påvirke svake spenningsreferanser.

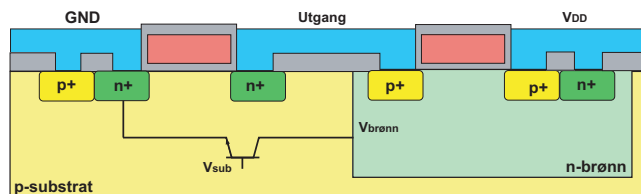


Figure 4.54: Situasjon som kan fremprovosere latchup.

Et eksempel på en situasjon som kan resultere i latchup er vist i Fig. 4.54. Anta for eksempel at spenningen i substratet økes, for eksempel ved tilførsel av ekstern spenning ved oppstart. Vi kan anta at V_{sub} økes slik at npn transistoren mellom n^+ diffusjon, p-substrat og n-brønn skrus på. Normalt skal substratet ligge til GND, men det er fysisk mulig at spenningen lokalt kan stige dersom kretsen lokalt blir påvirket av uheldige spenningsendringer. npn transistoren vil levere strøm fra kollektor (n-brønn) til emitter (n^+) diffusjon.

Dersom spenningsforsyningen til brønnen er svak vil npn transistoren kunne trekke brønnspenningen $V_{brønn}$ noe ned fra V_{DD} . Vi kan modellere dette som en motstand i n-brønnen som vist i Fig. 4.55.

Vi kan nå tenke oss at det trekkes strøm fra n-brønnen slik at spenningen i brønnen $V_{brønn}$ faller i forhold V_{DD} slik at det genereres en pnp bipolar transistor med base

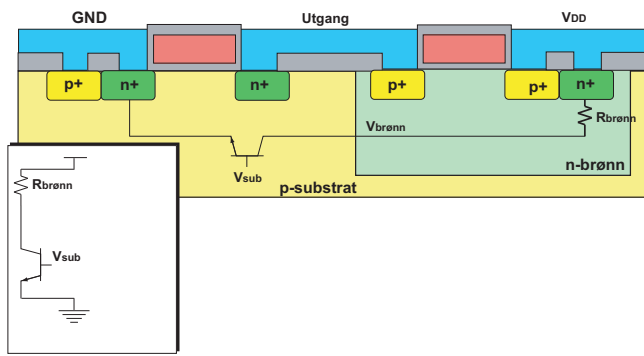


Figure 4.55: *Latchup*.

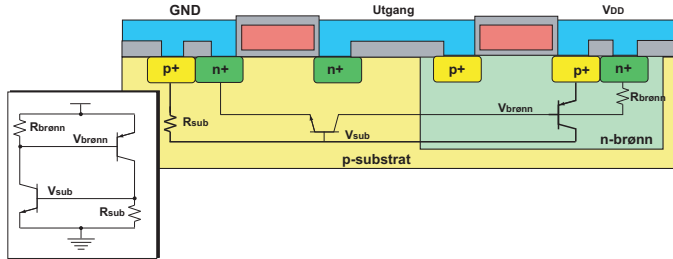


Figure 4.56: *Latchup*.

i n-brønnen som vist i Fig. 4.56, emitterer i p^+ diffusjon (V_{DD}) og kollektor i p-substratet. Den nye transistoren vil trekke strøm fra spenningsforsyningen (emitter) og tilføre strøm til substratet og dermed bidra til at substratspenningen V_{sub} stiger ytterligere og bidrar til at npn transistoren med kollektor i n-brønnen trekker mer strøm, som igjen vil bidra til at $V_{brønn}$ faller ytterligere. Vi har nå fått to bipolare transistorer som virker sammen som en aktiv kortslutning mellom V_{DD} og GND .

Latchup kan lett unngås ved å minimere $R_{brønn}$ og R_{sub} slik at det er mindre spenningsfall i brønn og substrat mot henholdsvis V_{DD} og GND . Det er svært viktig at designeren lager god kontakt til brønn og substrat slik at spenningsreferansene er sterke. Det er en god strategi å plassere så mange substrat og brønnkontakter som arealet tillater. Følgende strategier bør følges for å unngå problemer med latchup:

- Alle brønner skal ha minst en brønnkontakt.
- Alle substrat- og brønner bør kobles direkte til passende referanser tilført i gode metalledere.
- Brønn- og substratkontakter bør plasseres for hver 5-10 transistor og med maksimalt $25 \mu m$ avstand. Som regel er det fornuftig å plassere så mange brønn- og substratkontakter som arealet tillater.
- nMOS transistorer bør plasseres så nær spenningsforsyningen GND som mulig, og tilsvarende pMOS transistorer så nær V_{DD} som mulig.

I/O, dvs. inngangs- og/eller utgangspadder eller eksterne tilkoblinger, spiller en spesielt viktig rolle i latchup

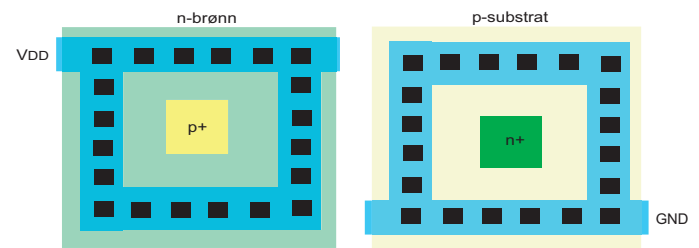


Figure 4.57: *Guard rings*.

problemer. Eksterne signaler har man liten kontroll over slik at det er fornuftig å beskytte disse padderne spesielt for å redusere risikoen for uønskede spenninger internt i den integrerte kretsen. Det er viktig å legge *beskyttelsesringer* (guard rings) rundt kontaktpunkter som er knyttet til kretser som er spesielt utsatt for latchup. Dette kan typisk være brønn- og substratkontakter i I/O padderne eller transistorer som ligger svært nær padderne som vist i Fig. 4.57.

4.33 Skalering

(Kapittel 4.9 side 245 - 246)

Det har i mer enn 30 år vært en utvikling i integrert teknologi som har preg av en stabil forbedring. I 1965 forutså Gordon Moore en eksponensiell vekst i antall transistorer på en integrert krets med en gitt størrelse. Antallet transistorer dobles hver 18nde måned. Veksten avtar noe med moderne teknologier på grunn av meget stort effektforbruk med tilhørende problemer med varmeutvikling. I de siste 10 årene har antall transistorer blitt doblet med noe lengre tidsintervall. Det er forventet at denne utviklingen vil fortsette i 10 år til. Denne utviklingen kalles *Moore's lov*.

Når transistorstørrelsen reduseres vil dette bidra til raskere kretser, flere transistorer, redusert forsyningspenninger, mindre effektforbruk i porter, men større effektforbruk per arealenheter fordi tettheten øker betydelig. Selv om hver transistor bidrar med mindre effektforbruk vil effekten som forbrukes øke fordi antall transistorer i en integrert krets øker kraftig. Andre effekter ved nedskalering er økt støy, økt mismatch, redusert forsterkning i porter og en rekke elektriske detaljer som bidrar med betydelig lekkasjestrømmer, for eksempel gate lekkasje.

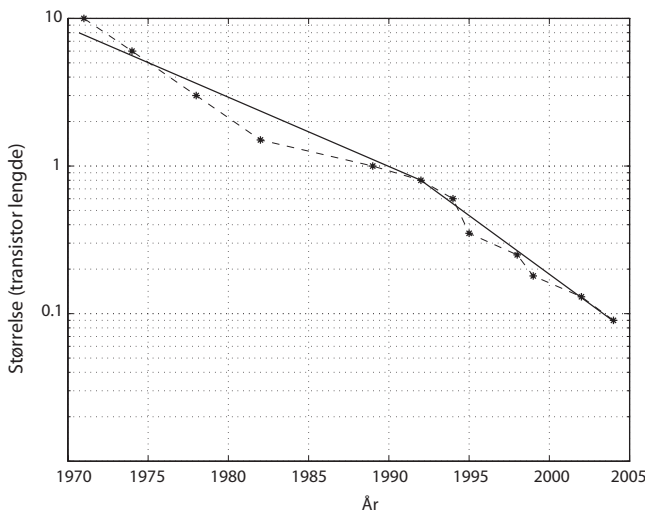


Figure 4.58: *Introduksjonstidspunkt for ulike prosesser (FIG4.65)*

Introduksjon av ulike prosesser er vist i Fig.4.58

4.33.1 Mål

Forstå konsekvenser ved skalering av teknologi.

4.34 Transistorskalering

(Kapittel 4.9.1 side 246 - 249)

Et begrep som er viktig i skalering av MOS transistorer er såkalt *konstant felt skalering*. For å bevare transistorkarakteristikker til neste generasjon har man tenkt at en lik skalering av noen viktige parametere med en faktor S . Aktuelle parametere er

- Alle dimensjonene skaleres (W , L og T_{ox}).
- Spenninger, dvs. spenningsforsyninger.
- Dopekonsentrasjoner.

En annen tilnærming er *lateral skalering* der bare transistorenes gate lengde skaleres.

Effekten av konstant felt og lateral skalering for transistorer er vist i tabell 4.7.

Prosessindustrien skalere en ny generasjon med $S = \sqrt{2}$ som vil resultere i en dobling av antall transistorer. Transistorenes ytelse vil dobles for hver annen generasjon. Det er vanlig å gjøre små endringer på modne prosesser ved å skalere med $S = 1.05$ (5%) for å øke kretsytelsen.

Ved skalering med konstant felt vil alle dimensjoner skaleres med en faktor $1/S$. Dette inkluderer transistorlengde og bredde og tykkelse på tynnoksid t_{ox} . Forsyningspenninger V_{DD} og terskelspenninger vil også skaleres med en faktor $1/S$, mens dopekonsentrasjonen for substratdoping skaleres med en faktor S . Fordi både avstand (L) og spenning (V_{DD}) skaleres likt vil elektrisk feltstyrke forbli nesten konstant. Gatekapasitans forblir også nesten upåvirket av skaleringen, dvs. $1.5 - 2fF/\mu m$.

Dersom man bare skalere transistorenes lengde og lar alle andre størrelser, spenninger og dopekonsentrasjoner være uendret kan man i utgangspunktet forvente en kvadratisk økning i ytelse. På grunn av hastighetsmetning vil forbedring i ytelsen likevel bare være omvendt lineært avhengig av L . Det var vanlig å skalere med såkalt *konstant spenning*, som eksempel var det vanlig å opprettholde en forsyningspenninger på 5V for transistorlengder mellom $6\mu m$ til $1\mu m$ for å få best mulig ytelse. Dette er nå uvanlig på grunn av frykt for uønskede lekkasjer, ødeleggelse av tynnoksid (hot carriers) som kan føre til at transistorene blir ødelagt.

4.35 Skalering av interkonnekt

(Kapittel 4.9.2 side 249 - 250)

Det er to vanlige tilnærminger til skalering av interkonnekt; skalering av alle dimensjoner og holde høyden på ledere konstant.

Parameter	Red.høyde	Konst.høyde
w Bredder	$1/S$	$1/S$
s Avstand	$1/S$	$1/S$
t Tykkelse (høyde)	$1/S$	1
h Oksidtykkelse	$1/S$	$1/S$

Table 4.8: Skalering av interkonnekt. Skaleringsparametere.

Skaleringsparametere for skalering av interkonnekt er vist i tabell 4.8.

Karakteristikker per enhetslengde ved skalering av interkonnekt er vist i tabell 4.9.

Lokal/skalerte interkonnekt karakteristikk for skalering av interkonnekt er vist i tabell 4.10.

Globale interkonnekt karakteristikk for skalering av interkonnekt er vist i tabell 4.11. D_c står for en skalert chip (die) størrelse, tpsik 1.1.

Dersom interkonnekt skaleres tilsvarende som transistorer vil mer og mer av tidsforsinkelsen ligge i interkonnekt og ikke i portene. I gamle prosesser der avstand mellom ledere og bredden på ledere var mye større enn tykkelsen på lederne, var det fordelaktig å skalere bredden og ikke tykkelsen på lederne. Dette betyr at det ikke var en kvadratisk økning i motstand per lengdeenhet og som samtidig var samtidig akseptabelt med hensyn på kantkapasitans (fringe) som er avhengig av tykkelsen på lederne. I moderne prosesser er kantkapasitans (fringe) for ledere en viktig del av den totale kapasitans i en krets. I tillegg er crosstalk et betydelig større problem i moderne prosesser. Det er derfor vanlig å redusere tykkelsen på metalledere.

4.36 Teknologitviking

(Kapittel 4.9.3 side 251)

Internasjonal teknologi roadmap for halvledere (ITRS) er utviklet og blir oppdatert av *Semiconductor Industry Association*. Poenget med dette er å holde en oversikt over teknologitvikingen og å bidra til at systemer er kompatible.

ITRS spår generasjonsskifte i halvlederteknologi hvert tredje år. Et slikt generasjonsskifte kalles også *teknologi node*. Skalering mellom generasjoner er tradisjonelt $S = \sqrt{2}$, som betyr at antall transistorer dobles for hver generasjon.

Forventet utvikling fra ITRS er vist i tabell 4.12.

4.37 Designpåvirkninger

(Kapittel 4.9.4 side 252-257)

Det er en rekke forhold som endres ved skalering av integrerte kretser.

4.37.1 Økt ytelse og kostnader

Det er i første rkke økt ytelse som er viktig ved skalering av teknologi. Transistorene blir billigere fordi man kan integrere langt flere transistorer på samme areal. Dette forholdet bør i sterkere grad påvirke systemdesignere til å utnytte både økt ytelse og økt kompleksitet gjennom nye produkttyper og designmetoder.

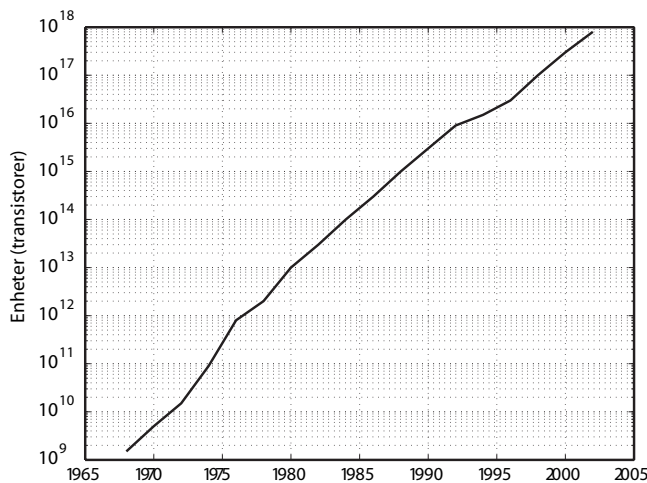


Figure 4.59: Antall transistorer som produseres på integrerte kretser.

I Fig. 4.59 er antall transistorer som produseres vist. Som vi ser er det en stabil eksponensiell økning i antall integrerte transistorer fra 1968.

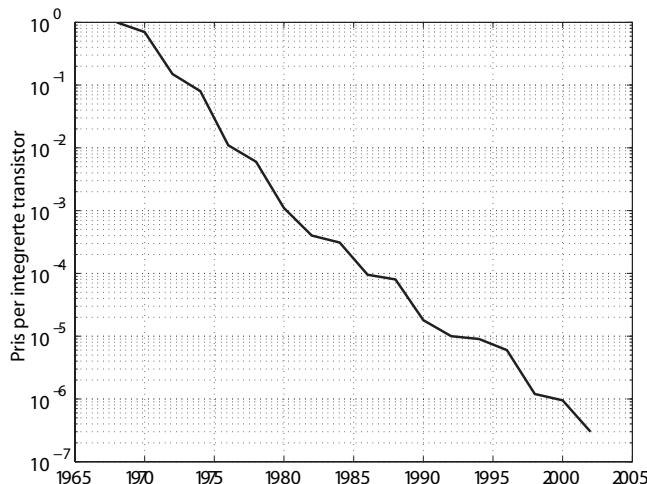


Figure 4.60: Pris per integrert transistor.

Pris per integrert transistor er vist i Fig. 4.60. Selv om prisen for en integrert krets kan stige ved overgang til en ny generasjon vil prisen per transistor gå ned.

4.37.2 Interkonnekt

Mens transistorene og portene får en betydelig økt ytelse ved nedskalering, gjelder dette ikke i samme grad for interkonnekt. Det motsatte kan faktisk være tilfellet.

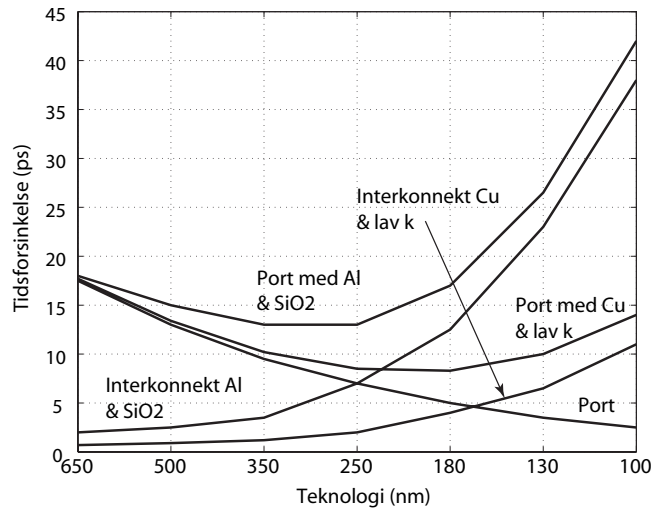


Figure 4.61: Skalering av tidsforsinkelse i porter og ledere. Al $3.0\mu\Omega - cm$, Cu $1.7\mu\Omega - cm$, SiO_2 $k = 4.0$, $Lav k$ $k = 2.0$, Al og Cu tykkelse $0.8\mu m$ og Al og Cu leder $43\mu m$ lang.

Endring i tidsforsinkelse for porter og ledere er vist i Fig. 4.61. Tidsforsinkelsen i porter går ned fra ca. $23ps$ for $650nm$ til ca $3ps$ for $100nm$. For aluminiumledere med lengde $43\mu m$ vil derimot tidsforsinkelsen øke fra ca. $2ps$ til ca. $41ps$. Vi ser at kobber med lav k har mindre tidsforsinkelse, men også her er en betydelig økning fra ca. $1ps$ til ca. $11ps$. Dersom vi ser på en port og en leder vil tidsforsinkelsen tilsammen synke for teknologier ned til ca. $350 - 250nm$ for så å stige for mer moderne teknologier. Nå er det grunn til å bemerke at en leder på $43\mu m$ er relativt lenger i en moderne teknologi enn i en gammel teknologi, men det er ikke alltid lett å skalere lengden på ledere på samme måten som bredden skaleres. Det vil imidlertid legge press på designerne til å konsentrere mer av designarbeidet mot effektiv signalføring i ledere. Korte ledere som brukes til svært lokal ruting av signaler er fortsatt neglisjerbar. I moderne teknologier er antall metallag meget høyt, og de øverste lagene kan brukes til å lage lav-ohmig raske ledere ved å skalere de motsatt av resten av skaleringsparameterne.

4.37.3 Effektforbruk

Effektforbruk, eller presist effekttettheten, i integrerte kretser har økt dramatisk med nye teknologier. Dette kommer av at klokkefrekvensen øker sterkt. Dynamisk effektforbruk er som kjent proporsjonal med frekvensen. Man kunne tro at statisk effektforbruk ville gå ned fordi forsyningsspenningen går ned, men dette blir oppveid av en kraftig stigning i lekkasjestrømmer gjennom transistorer som er skrudd av. Husk at terskelspenningen går ned og at dette vil heve strømmivået for transistorer som

er skrudd av. I moderne teknologier er avstrømmen i nano ampere området.

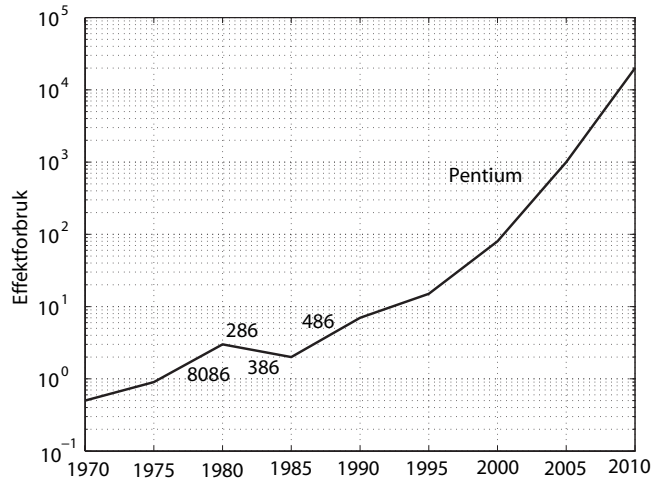


Figure 4.62: Effektforbruk i Intel prosessorer.

Effektforbruket i Intel prosessorer ifra tidlig på syttitallet og frem mot 2010 er vist i Fig. 4.62. Det økte effektforbruket er en kritisk faktor som kan begrense en naturlig utvikling av integrerte kretser framover. Det er grunn til å regne med at økning i klokkefrekvenser ikke kommer til å fortsette i samme takt som før, fordi klokkesignalene vil ligne mer og mer på en sinus. Det er vanskelig å få kretser som er avhengig av klokkeflanker for synkronisering til å fungere optimalt med sinus signaler som klokker. I moderne teknologier vil transistorer med ulike terskel-spenninger være tilgjengelig.

4.37.4 Produktivitet

Antallet transistorer som er tilgjengelig for designere øker langt raskere enn designernes produktivitet. I praksis vil det være mangel på designere når økonomien er god. Det er derfor lagt stor vekt på å utvikle metoder for å øke produktiviteten i designfasen, ofte på bekostning av areal og ytelse. Det er vanlig å designe digitale integrerte systemer ved hjelp av syntese og automatisk plassering og ruting. Dette betyr at det er en forskyvning fra kretsdesign til systemdesign. På den andre siden vil det være en rekke systemer der ytelsen er svært avgjørende og vil kreve en god del detaljdesign. For utvikling av komplekse digitale systemer kreves ofte meget store team.

4.37.5 Fysiske begrensinger

Hvor langt vil CMOS skalere? Det er åpenbart at skaleringen ikke kan fortsette uendelig lenge. Dagens transistorer vil ikke kunne fungere med en oksidtykkelse som er mindre enn et atom, kanaler som er kortere enn et atom og med ladning mindre enn et elektron. CMOS teknologiens endelikt har blitt spådd en rekke ganger. I 1972 trodde man at den nedre grensen var $250nm$ pga. av uheldige elektriske karakteristikk som oppstår. Samtidig ble det anslått at frekvensen ville være $10 - 30MHz$. I 1999 antok man at den nedre grensen på $100nm$ ville bli nådd i 2004.

Det er en rekke faktorer som vil påvirke ytelsen i negativ retning fremover

- Dynamisk effektforbruk.
- Lekkasje i svak inversjon for lav V_{DD} og lav V_t .
- Tunnelling gjennom tynt tynnoxid.
- Lav spenningsforsterkning pga. kortkanaleffekter.
- Optikk for fremstilling av integrerte kretser.
- Elektromigrasjon.
- Tidsforsinkelse i interkonnekt.

Det ser nå ut til at det ikke er noen fundamental barriere før $35nm$ i 2013.

Parameter	Følsomhet	Konstant felt	Lateral
L Lengde		$1/S$	$1/S$
W Bredde		$1/S$	1
t_{ox} Gateoksid		$1/S$	1
V_{DD} Forsyningssp.		$1/S$	1
V_t Terskelspenning		$1/S$	1
N_A Substratdoping		S	
β	$\frac{W}{L} \frac{1}{t_{ox}}$	S	S
I_{ds}	$\beta(V_{DD} - V_t)^2$	$1/S$	S
R (Motstand)	$\frac{V_{DD}}{I_{ds}}$	1	$1/S$
C Gate kapasitans	$\frac{W L}{t_{ox}}$	$1/S$	$1/S$
τ Portforsinkelse	RC	$1/S$	$1/S^2$
f Klokkefrekvens	$1/\tau$	S	S^2
P Dyn effektf.	$CV^2 f$	$1/S^2$	S
A Areal		$1/S^2$	1
Effekttetthet	P/A	1	S
Strømtetthet	I_{ds}/A	S	S

Table 4.7: Skalering av MOS transistorer.

Parameter	Følsomhet	Red.høyde	Konst.høyde
R_w Motstand per l.		$1/S$	1
C_{wf} Fringe kap. per l.	t/s	S^2	S
C_{wp} Parallell pl. kap. per l.	w/h	1	S
C_w tot. kap. per l.	$C_{wf} + C_{wp}$	1	$1...S$
RC per l. t_{wu}	$R_w C_w$	S^2	$S...S^2$
RC per l. t_{wr} ant. konst. felt sk.	$\sqrt{RC R_w C_w}$	\sqrt{S}	$1...\sqrt{S}$
Crosstalk støy	$\frac{t}{s}$	1	S

Table 4.9: Skalering av interkonnekt. Karakteristikk per enhetslengde.

Parameter	Følsomhet	Red.høyde	Konst.høyde
l lengde		$1/S$	$1/S$
RC forsinkelse	$l^2 t_{wu}$	1	$1/S...1$
RC forsinkelse ant. konst. felt sk.	$l t_{wr}$	$\sqrt{1/S}$	$1/S... \sqrt{1/S}$

Table 4.10: Skalering av interkonnekt. Lokale/skalerte interkonnekt karakteristikk.

Parameter	Følsomhet	Red.høyde	Konst.høyde
l lengde		D_c	D_c
RC forsinkelse	$l^2 t_{wu}$	$S^2 D_c^2$	$S D_c^2 ... S^2 D_c^2$
RC forsinkelse ant. konst. felt sk.	$l t_{wr}$	$D_c \sqrt{S}$	$D_c ... D_c \sqrt{S}$

Table 4.11: Skalering av interkonnekt. Globale interkonnekt karakteristikk. D_c står for en skalert chip (die) størrelse, tpsik 1.1

	2001	2004	2007	2010	2013	2016
Størrelse (nm)	130	90	65	45	32	22
V_{DD}	1.1-1.2	1-1.2	0.7-1.1	0.6-1.0	0.5-0.9	0.4-0.9
Transistorer (millioner)	193	385	773	1564	3092	6184
Metallag	8-10	9-13	10-14	10-14	11-15	11-15
Pitch (nm)	450	275	195	135	95	65
Interk. dielectr.	3 3.6	2.6- 3.1	2.3- 2.7	2.1	1.9	1.8
I/O sign.	1024	1024	1024	1280	1408	1472
Klokkef. (MHz)	1684	3990	6739	11511	19348	28751
FO4 forsin/cycl	13.7	8.4	6.8	5.8	4.8	4.7
Max effekt (W)	130	160	190	218	251	288
DRAM (Gbits)	0.5	1	4	8	32	64

Table 4.12: *Forventet utvikling fra ITRS.*

Chapter 5

Kombinatoriske kretser

5.1 Bubble pushing

(Kapittel 6.2.1.1 side 321)

* DeMorgan teorem. (INF1400)

En vanlig problemstilling for en designer er å velge logiske porter for å implementere en boolsk funksjon. De vanligste portene er inverter, NAND og NOR. Enhver boolsk funksjon kan implementeres med ulike kombinasjoner av de enkle portene. Et enkelt hjelpemiddel ved implementasjon av boolske funksjoner er *bubble pushing*. DeMorgans teorem gir oss:

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

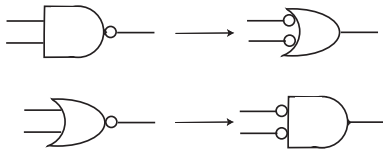


Figure 5.1: *Bubble pushing med DeMorgans teorem. (FIG6.1)*

Bubble (invertering) pushing er vist i Fig. 5.1. Portene på høyre og venstre side er logisk ekvivalente.

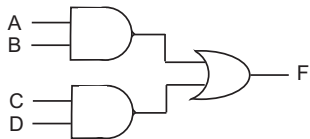


Figure 5.2: $F = AB + CD$ med NAND og NOR porter. (FIG6.3a)

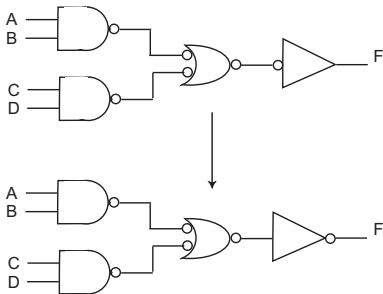


Figure 5.3: $F = AB + CD$ med NAND- og NOR porter og inverter. (FIG6.3c)

I Fig. 5.2 er den enkleste logiske implementasjonen av den boolske funksjonen $F = AB + CD$ vist. Som vi vet er CMOS i seg selv inverterende, dvs. det er naturlig å implementere invertere, NAND og NOR porter og ikke AND og OR porter. Vi kan innføre to “bobler” i en elektrisk node som betyr to inverteringer med resultatet lik ingen invertering. Med 2 inverteringer mellom AND portene og OR porten kan vi erstatte AND portene med NAND og

får inverterte innganger til OR porten som vist i Fig. 5.3. Vi introduserer to inverteringer på utgangen og får da en NOR port istedet for en OR port og legger på et buffer med invertert inngang på utgangen. Et buffer med invertert inngang er ekvivalent med en inverter som vist i den nederste kretsen i figuren.

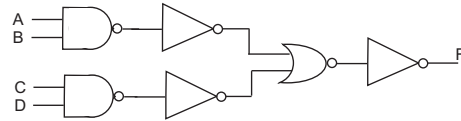


Figure 5.4: $F = AB + CD$ med NAND- og NOR porter og inverter. (FIG6.3b)

I Fig. 5.4 er kretsen vist med inverter mellom NAND portene og NOR porten. En inverter representerer en invertering.

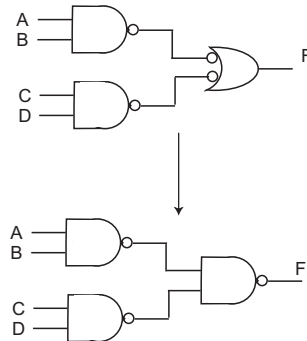


Figure 5.5: $F = AB + CD$ med NAND- og NOR porter. (FIG6.3d)

Vi kan imidlertid la vær å invertere 2 ganger på utgangen og istedet skifte ut OR porten med inverterte innganger til en NAND port som vist i Fig. 5.5.

5.1.1 Mål

Forstå og kunne anvende bubble pushing for design av en boolsk funksjon på ulike måter.

5.2 Komplementær logikk (compound gates)

(Kapittel 6.2.1.2 side 321 - 323)

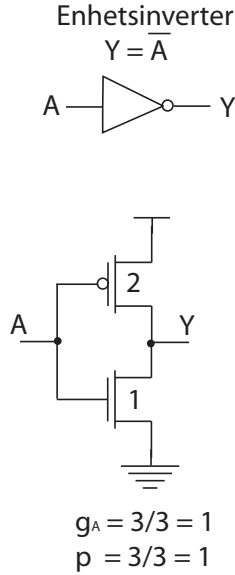


Figure 5.6: Enhetsinverter. (FIG6.4 venstre)

I dette avsnittet skal vi se på forskjeller på hvordan komplementær logikk kan karakteriseres med hensyn på logisk effort og parasittisk tidsforsinkelse. En enhetsinverter, der vi antar at $\mu_n = 2\mu_p$, er vist i Fig. 5.6. Logisk effort og parasittisk tidsforsinkelse er som kjent 1 for enhetsinverteren.

En komplementær implementasjon av den boolske funksjonen $F = \overline{AB + C}$ er vist i Fig. 5.7. Vi antar som før at $\mu_n = 2\mu_p$ og ser at transistorstørrelsene som er vist i figuren gir samme ekvivalent motstand i opp- og nettrekk:

$$\begin{aligned} R_{0 \rightarrow 1} &= R_A + R_C \\ &= 2R_p, \end{aligned} \quad (5.1)$$

der $R_p = R_A = R_C$ er motstand for pMOS transistorene. Worst case opptrekk er via to pMOS transistorer som betyr at det i dette tilfellet ikke er noe poeng å dimensjonere pMOS transistorene forskjellig. Den effektive motstanden vil være lik for alternative nedtrekk. For nedtrekk ser vi to ulike opsjoner:

1. Via nMOS transistor styrt av C. I dette tilfellet vil det bare være en nMOS transistor som betyr at vi velger bredden på denne transistoren som for en inverter. Dette vil si at vi velger bredden lik 1 og får motstanden R_n . For at opptrekket skal ha samme effektive motstand må vi velge transistor bredde på pMOS transistorene slik at $2R_p = R_n$ som i praksis vil si, gitt mobilitetsforskjellene, at bredden på pMOS transistorene må være 4 ganger bredden på en enhets nMOS transistor.

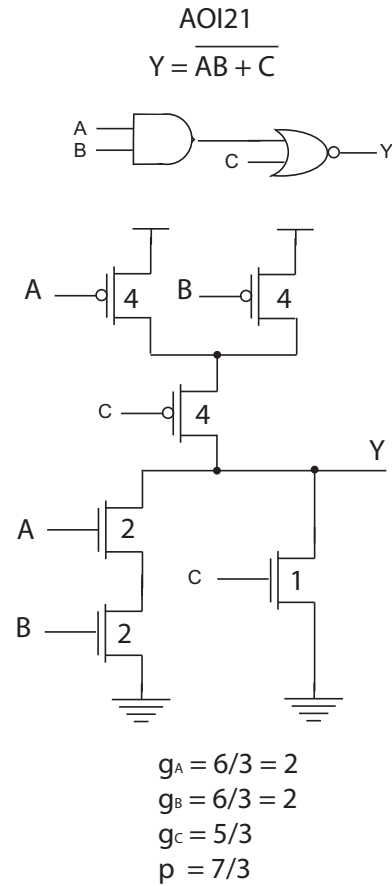


Figure 5.7: Komplementær implementasjon av funksjonen $F = \overline{AB + C}$. (FIG6.4)

2. Via to nMOS transistorer styrt av A og B. Her er det fornuftig å matche den effektive motstanden slik at den blir lik det alternative nedtrekket. To seriekoblede transistorer utgjør dobbel så stor motstand som en tilsvarende transistor. To transistorer med bredde lik 2 ganger enhetsbredde utgjør en ekvivalent motstand $R_n = 2R || 2R = R$.

En komplementær implementasjon av den boolske funksjonen $F = \overline{AB + CD}$ er vist i Fig. 5.8 med transistorstørrelser slik at ekvivalent motstand for opp- og nedtrekk er like.

For denne kretsen blir logisk effort lik for alle inngangene fordi hver inngang går inn på en pMOS- og en nMOS transistor med størrelse $W_p = 4$, og $W_n = 2$. Logisk effort blir derfor $6/3$. Parasittisk tidsforsinkelse blir $(4 + 4 + 2 + 2)/3 = 12/3$.

5.2.1 Eksempel

Vi skal implementere $F = \overline{AB + CD}$ slik at tidsforsinkelsen blir minst mulig. Som vi vet kan denne funksjonen implementeres på mange måter. Vi kan starte med en løsning som er vist i Fig. 5.9. Vi kan anta at inngangene maksimalt kan drive en kapasitiv last tilsvarende en transistor med bredde 20, og vi antar at utgangen skal drive en last tilsvarende en transistor med bredde 100.

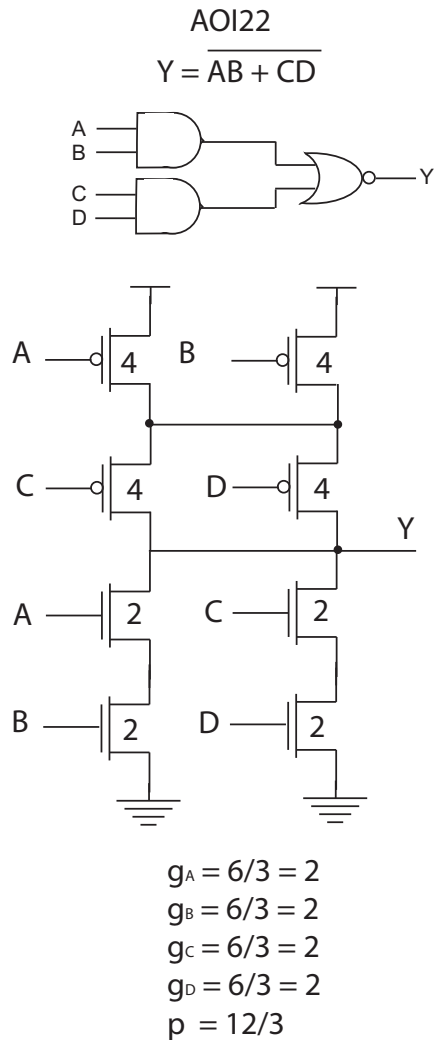


Figure 5.8: Komplementær implementasjon av funksjonen $F = \overline{AB + CD}$. (FIG6.4)

Elektrisk effort for kjedene fra inngang til utgang er gitt av $H = C_{ekstern}/C_{inngang} = 100/20 = 5$. Forgrenings-effort er 1 fordi det ikke er noen last knyttet til elektriske noder i kjedene som ikke inngår i selve kjeden. Vi kan finne kjedens logiske effort $G = (4/3) \cdot (4/3) = 16/9$ og parasittisk tidsforsinkelse $P = 2 + 2 = 4$. Kjeden effort er $F = GBH = (16/9) \cdot 1 \cdot 5 = 8.89 \approx 9$, som gir en optimal port effort for en kjede med like porter $f' = 9^{1/2} = 3$.

Når vi skal bestemme transistorstørrelsene beregner vi først den beste port effort:

$$\begin{aligned}
 C_y &= C_{ekstern} \frac{g_{NAND2}}{f'} \\
 &= 100 \frac{4}{3} \\
 &= 44
 \end{aligned}
 \tag{5.2}$$

For en 2inngangs NAND port skal pMOS- og nMOS transistorene være like store for å få lik ekvivalent motstand i opptrekk og nedtrekk. Vi ender da opp med

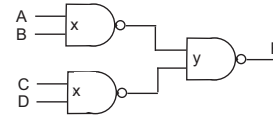


Figure 5.9: Funksjonen $F = AB + CD$ implementert med 3 2inngangs NAND porter.

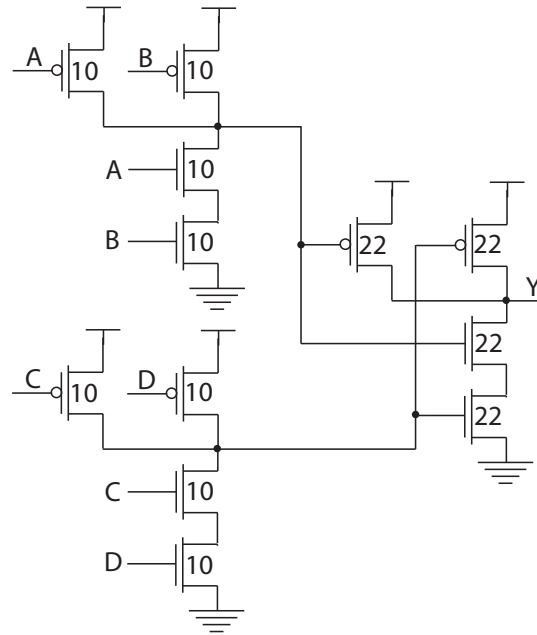


Figure 5.10: Implementasjon av funksjonen $F = AB + CD$ ved hjelp av 3 2inngangs NAND porter. (FIG6.5)

transistorstørrelser på den siste NAND porten lik 22. Vi kan nå bestemme transistorstørrelsene på de to resterende NAND portene. Igjen er det slik at transistorene skal være like store. Vi får da:

$$\begin{aligned}
 C_x &= C_y \frac{g_{NAND2}}{f'} \\
 &= 44 \frac{4}{3} \\
 &\approx 20,
 \end{aligned}
 \tag{5.3}$$

som jo passer med den opprinnelige antagelsen (forutsetningen) at inngangslasten skulle være maksimalt 20.

En implementasjon av funksjonen F med tre 2inngangs NAND porter er vist i Fig. 5.10 og transistorstørrelser slik at ekvivalent motstand for opp- og nedtrekk blir like. Tidsforsinkelse i kjeden blir $D = NF^{1/N} + P = 2 \cdot 3 + 2 + 2 = 10\tau$.

En alternativ implementasjon av funksjonen er vist i Fig. 5.11. Kjeden består her også av to porter som gir en effort for kjeden $F = GHB = (6/3) \cdot 5 \cdot 1 = 10$. Videre får vi optimal port effort $f' = 10^{1/2} \approx 3.2$ og transistorstørrelser for inverteren:

$$3y = C_{ekstern} \frac{g_{INVERTER}}{f'}$$

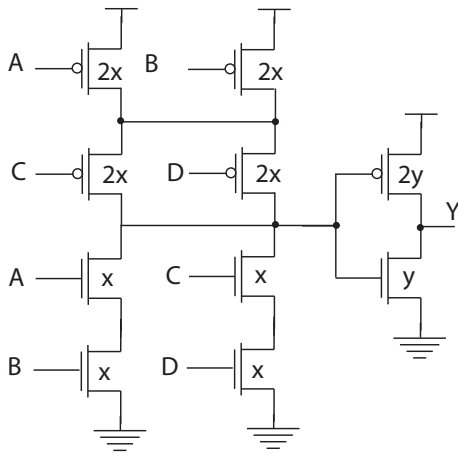


Figure 5.11: Komplementær implementasjon av funksjonen $F = AB + CD$ ved hjelp av en komplementær port ($\overline{AB + CD}$) og en inverter. (FIG6.5)

$$\begin{aligned}
 &= 100 \frac{1}{3.2} \\
 &\approx 31,
 \end{aligned} \tag{5.4}$$

som medfører at $W_p \approx 2W_n = 21$ og $y = W_n = 10$. For den første porten har vi:

$$\begin{aligned}
 3x &= C_{3y} \frac{g_{PORT}}{f'} \\
 &= 31 \frac{6}{3.2} \\
 &\approx 20,
 \end{aligned} \tag{5.5}$$

som gir $W_p \approx 2W_n = 13$ og $x = W_n = 7$. Tidsforsinkelsen i kjeden blir da $D = NF^{1/N} + P = 2 \cdot 3.2 + 4 + 1 = 11.4\tau$. Vi ser at implementasjonen med tre NAND porter representerer minst tidsforsinkelse for funksjonen.

5.2.2 Mål

Kunne implementere en boolsk funksjon ved hjelp av komplementær logikk og kunne beregne logisk effort og parasittisk tidsforsinkelse.

5.3 Hvordan rekkefølgen på inngangssignaler påvirker tidsforsinkelse

(Kapittel 6.2.1.3 side 324)

* Hvordan gate source kapasitans påvirker en inverter som svitsjer i del 5 (INF3400)

I mange tilfeller er logisk effort og parasittisk tidsforsinkelse forskjellig for ulike inngangssignaler til en port. AOI21 porten som er vist i Fig. 5.7 er *asymmetrisk* fordi gatekapasitans som porten representerer er mindre for en av inngangene (C) enn for de to andre inngangene. NAND og NOR porter er symmetriske men kan representere litt forskjellig logisk effort og parasittisk tidsforsinkelse sett fra ulike innganger.

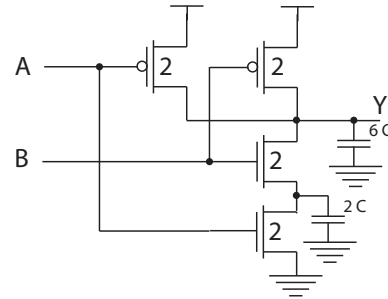


Figure 5.12: 2inngangs NAND gate. (FIG6.6)

Fig. 5.12 viser en 2inngangs NAND port med diffusjons- eller intern kapasitans. Vi skal se nærmere på hva som skjer når vi har en stabil 1er på en inngangene og den andre inngangen stiger fra 0 til 1. Da vil utgangen på porten falle fra 1 til 0. Vi har da to ulike situasjoner:

1. **A er stabil 1 og B stiger fra 0 til 1.** I utgangspunktet er utgangen 1, som betyr at vi får et terskelfall over nMOS transistoren styrt av A. Node x vil i utgangspunktet ha en spenning lik $V_{DD} - V_t$ ¹. Elmore forsinkelsesmodell gir $(R/2)(2C) + R(6C) = 7RC = 2.33\tau$. I tillegg vil vi ha en svak tilbakekobling fra x til A via gate source kapasitans og til B via gate drain kapasitans. Når x trekkes lav fordi A er høy og B skifter fra lav til høy, kan denne tilbakekoblingen påvirke inngangssignalene slik at tidsforsinkelsen øker² noe. Den kapasitive tilbakekoblingen fra x til B er avhengig av at nMOS transistoren styrt av B er i lineært område. Denne tilbakekoblingen er derfor avhengig av spenningen på B og x ved transisjonen og derfor ikke tilstede i hele transisjonen. På den andre siden så har vi Miller effekt for denne kapasitansen slik at vi kan se en liten og tidsbegrenset tilbakekoblingseffekt fra x til B. Vi kaller dette en *negativ tilbakekobling*.

¹Noe lavere på grunn av body effekt.

²Denne effekten er svært liten, men avhengig av de totale last kapasitansene på inngangene A og B.

2. **B** er stabil 1 og **A** stiger fra 0 til 1. Det er nå rimelig å anta at node x i utgangspunktet er 0. Vi trenger derfor ikke å lade ut node x og Elmore forsinkelse blir defor $R(6C) = 2\tau$.

Generelt definerer vi *ytre inngang* som den inngangen som styrer en transistor som ligger nærmest en spenningsforsyning, V_{DD} eller GND . Den *indre inngang* styrer en transistor som ligger nærmest utgangen. Parasittisk tidsforsinkelse er minst når indre inngangen svitsjer senest fordi interne noder i en kjede (seriekobling) da allerede vil være ladet ut (eventuelt opp til V_{DD} i opptrekk. Dersom man kjenner til svitsjetidspunkt for ulike innganger til en port kan man utnytte denne kunnskapen og legge de inngangene som svitsjer senest nærmest utgangen.

5.3.1 Mål

Forstå hvordan rekkefølgen på inngangssignaler påvirker tidsforsinkelsen i en port.

5.4 Asymmetriske porter

(Kapittel 6.2.1.4 side 324 - 325)

I noen tilfeller er det stor forskjell hvor kritiske ulike signaler (innganger) er, dette kan bety at det vil lønne seg å designe porter som er (nominelt) symmetriske, usymmetrisk.

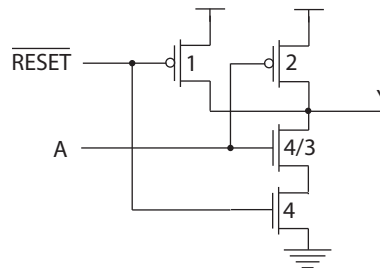


Figure 5.13: 2inngangs NAND gate. (FIG6.7)

Et eksempel på en slik port er vist i Fig. 5.13, der en kritisk signalvei vil være fra A til Y. Det vil i slike tilfeller å være fornuftig å prioritere de kritiske inngangene ved å plassere de som indre innganger og å redusere inngangslasten. I eksemplet i Fig. 5.13 er A en kritisk inngang som er plassert nærmest utgangen og transistoren som A styrer er redusert i størrelse slik at lasten sett fra inngangen blir redusert. I dette tilfellet er inngangen \overline{reset} ikke kritisk og derfor er størrelsen på nMOS transistoren som er koblet mot GND dimensjonert opp fra enhetsstørrelse. Den økte inngangslasten representerer ikke et problem fordi inngangen ikke er kritisk. Motstanden som denne transistoren bidrar med i nedtrekket er $(1/4)R$ der R er motstand i en enhets nMOS transistor (bredde lik 1). nMOS transistoren som styres av den kritiske inngangen A vil bidra med liten last for inngangen, men en større motstand for nedtrekket i porten. Ekvivalent motstand for nedtrekket blir $R_{effektiv} = (1/4)R + (3/4)R = R$ som er ekvivalent med to nMOS transistor i serie med bredde lik 2. Det er fornuftig å velge bredde på pMOS transistoren som styres av A lik 2 som gir en effektiv motstand lik R. Den siste pMOS transistoren er ikke kritisk og vi velger derfor bredden lik minimumsbredde slik at den totale kapasitansen som denne transistoren bidrar med for utgangen av porten blir minst mulig. Den interne lasten knyttet til utgangen er i dette tilfellet $(13/3)C$ i motsetning til $6C$ for en symmetrisk NAND port.

Man kan i motsatte tilfeller gjøre NAND porten enda mer symmetrisk dersom begge inngangene er like kritiske. En fullstendig symmetrisk 2inngangs NAND port er vist i Fig. 5.14.

5.4.1 Mål

Forstå hvorfor og hvordan man designer fullstendig symmetriske porter. Kunne ta hensyn til spesielle føringer i et design som tilsier at man skal designe asymmetriske porter.

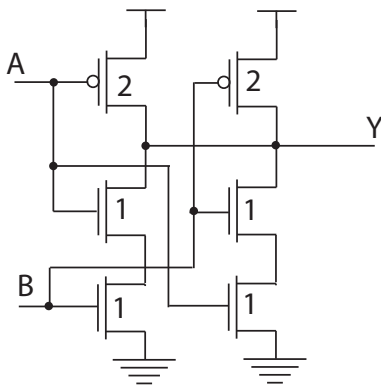


Figure 5.14: Fullstendig symmetrisk 2inngangs NAND gate. (FIG6.8)

5.5 Porter med skew

(Kapittel 6.2.1.5 side 325)

I noen tilfeller kan det være ønskelig å favorisere opptrekk eller nedtrekk i en port. I *HI-skew* porter favoriseres en 0 til 1 transisjon på utgangen og i en *LO-skew* favoriseres en 1 til 0 transisjon.

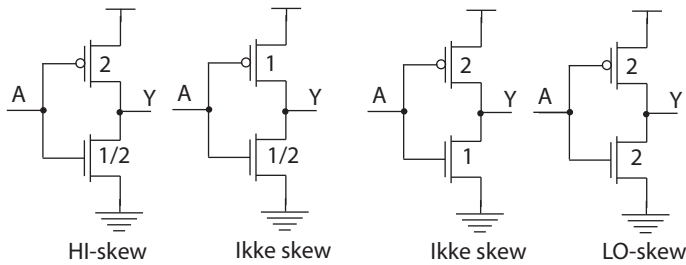


Figure 5.15: Invertere med skew. (FIG6.9)

Invertere med skew er vist i Fig. 5.15. For porter med skew skiller vi logisk effort i *logisk opptrekkseffort* g_u og *logisk nedtrekkseffort* g_d . Poenget med å designe porter med skew er å favorisere kritiske signalveier. Logisk effort i inverter nr. 3 fra venstre i Fig. 5.15 er $g = 1$. I den andre inverteren uten skew vil logisk effort bli $g = 1.5/1.5 = 1$. For HI-skew porten får vi en logisk effort for opptrekket $g_u = 2.5/3 = 5/6$ og logisk effort for nedtrekket $g_d = 2.5/1.5 = 5/3$. For nedtrekket må vi sammenligne med effort for en inverter med lik bredde på nMOS transistoren, som derfor har logisk effort $g = 1.5$. For LO-skew inverteren vil logisk effort for opptrekket være lik $g_u = 4/3$ og logisk effort for nedtrekket blir $g_d = 4/6 = 2/3$.

5.5.1 Mål

Forstå hvordan man designer porter med skew, HI-skew og LO-skew.

5.6 P/N forhold

(Kapittel 6.2.1.6 side 325 - 327)

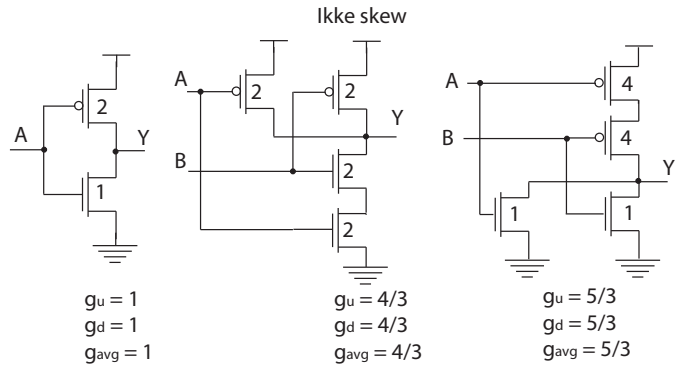


Figure 5.16: Inverter, NAND og NOR porter uten skew. (FIG6.10)

I Fig. 5.16 er det vist inverter, NAND- og NOR porter uten skew.

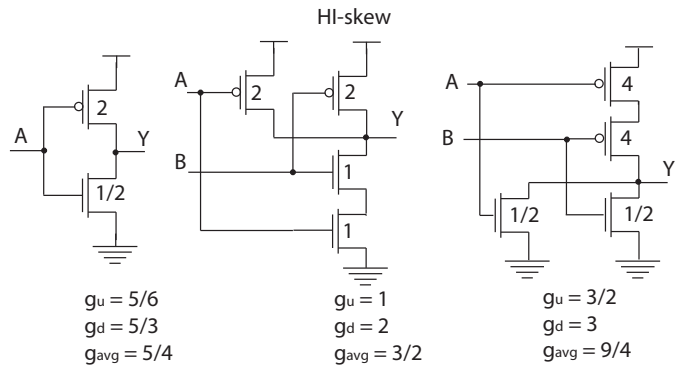


Figure 5.17: HI-skew inverter, NAND og NOR porter. (FIG6.10)

I Fig 5.17 er det vist HI-skew inverter, NAND- og NOR porter.

I Fig. 5.18 er det vist LO-skew inverter, NAND- og NOR porter. Legg merke til at for NOR porten er gjennomsnittelig logisk effort bedre enn for porten uten skew.

I Fig. 5.19 er det vist PN forhold som gir minimal tidsforsinkelse for portene. Merk at for porter med skew får vi typisk forskjellig stige og falltid som vil gi forskjellig tidsforsinkelse for fallende- og stigende transisjoner på utgangen.

5.6.1 Mål

Forstå hvordan bredde på nMOS transistorer i forhold til bredde på pMOS transistorer påvirker logisk effort i en port.

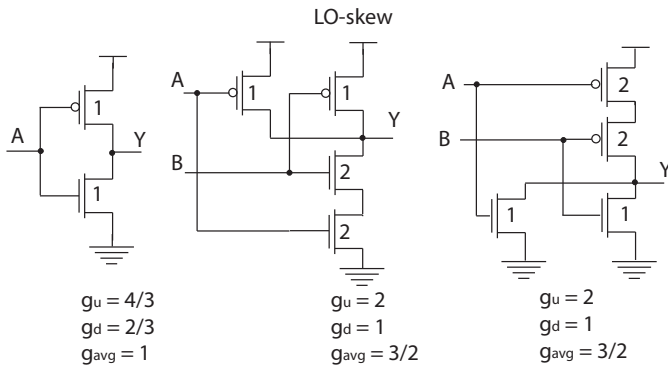


Figure 5.18: LO-skew inverter, NAND og NOR porter. (FIG6.10)

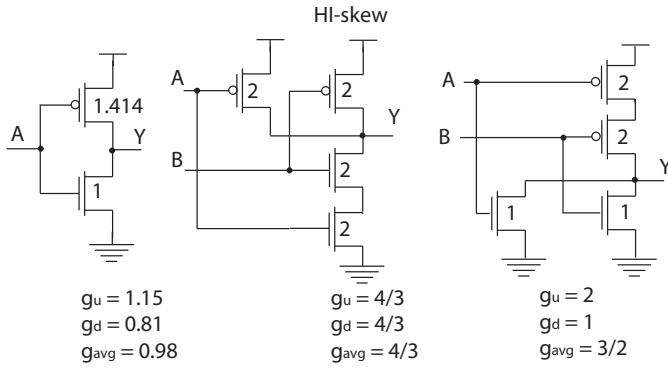


Figure 5.19: Porter med minimum tidsforsinkelse. (FIG6.11)

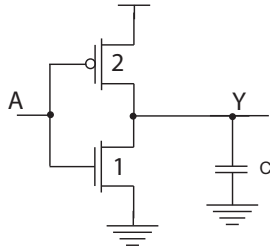


Figure 5.20: CMOS inverter med last.

Den viktigste komponenten i *dynamisk effektforbruk* er opp- og utlading av kapasitanser. Inverteren som er vist i Fig. 5.20 er et eksempel på en komplementær port som skal drive utgangen representert som en last i form av en kapasitans C . Utgangskapasitansen skal kunne trekkes opp til 1 via pMOS transistoren og ned til 0 via nMOS transistoren. Vi kan anta at porten svitsjer mellom 0 og 1 med en gjennomsnittlig frekvens f_{SW} . Over et tidsintervall T vil lasten bli ladet opp og ut $T \cdot f_{SW}$ ganger. Strømmen fra pMOS transistoren vil lade opp lasten og lasten vil lades ut via nMOS transistoren. En lade opp/lade ut sykel kan betraktes som en flytting av ladningen $Q = C \cdot V_{DD}$ fra V_{DD} til GND .

Gjennomsnittlig dynamisk effektforbruk kan modelleres som:

$$P_{dynamisk} = \frac{1}{T} \int_0^T i_{DD}(t) \cdot V_{DD} dt$$

$$= \frac{V_{DD}}{T} \int_0^T i_{DD}(t) dt. \quad (5.6)$$

Integralet over tidsperioden T gir:

$$\begin{aligned} P_{dynamisk} &= \frac{V_{DD}}{T} [T f_{SW} C V_{DD}] \\ &= C V_{DD}^2 f_{SW}. \end{aligned} \quad (5.7)$$

De fleste porter svitsjer ikke i hver klokkeperiode. Det er derfor vanlig å ta hensyn til aktiviteten til porten med modellen:

$$P_{dynamisk} = \alpha C V_{DD}^2 f, \quad (5.8)$$

der α er *aktivitetsfaktor*. Klokkesignaler svitsjer i hver klokkeperiode og har derfor aktivitetsfaktor $\alpha = 1$. Et vanlig aktivitetsfaktor for statistisk CMOS er 0.1.

5.6.2 Kortslutningseffekt

I det en utgang endrer verdi vil inngangen(e) også ofte være i transisjon. Ser vi nærmere på en inverter har vi når inngangen er mellom V_{tn} og $V_{DD} - |V_{tp}|$ en situasjon der både pMOS- og nMOS transistoren er PÅ. I dette tilfellet vil det gå en strøm direkte fra V_{DD} til GND som vil bidra med et effektforbruk som kalles *kortslutningseffekt*. Dersom inngangssignaler har kort stige/falltid vil kortslutningseffekten bli liten. I tilfeller der utgangslasten (kapasitans) er stor, dvs. vesentlig større enn inngangskapasitansen(e), vil kortslutningseffekten bli redusert.

5.6.3 Mål

Kunne modellere effektforbruk i en krets. Kunne skille mellom statistisk- og dynamisk effektforbruk.

5.7 Pseudo nMOS

(Kapittel 6.2.2.1 side 327 - 330)

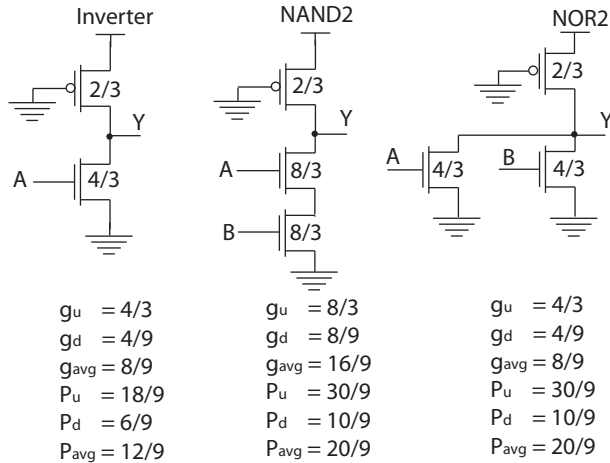


Figure 5.21: Pseudo nMOS porter. (FIG6.12)

Pseudo nMOS inverter, pseudo nMOS 2inngangs NAND og pseudo nMOS 2inngangs NOR porter er vist i Fig. 5.21. Nedtrekket er likt som for statisk eller komplementær logikk, mens opptrekket er erstattet med en pMOS transistor som alltid er PÅ. Det er fornuftig å velge en pMOS transistor som er ca. 1/4 så sterk som nedtrekket. Dette gir et godt kompromiss mellom hastighet og støymargin.

Når vi skal beregne logisk effort tar vi igjen utgangspunkt i en inverter, med samme opp- og nedtrekksmotstand, som kan levere like mye strøm som pseudo nMOS porten. Det er naturlig å skille logisk effort for opptrekk og nedtrekk fordi en pseudo nMOS port er asymmetrisk. I tillegg har pseudo nMOS porter svær ulike elektriske egenskaper i opptrekk og nedtrekk. Ved opptrekk via pMOS transistoren vil pseudo nMOS invertieren tilsvare en vanlig inverter, men med en relativt svak pMOS transistor. I nedtrekket derimot vil pMOS transistoren "holde igjen" slik at nMOS transistoren vil fungere som en svakere transistor enn en tilsvarende transistor i en vanlig inverter. For en vanlig (komplementær) inverter er det antatt at den transistoren som er skrudd av ikke leverer strøm. For pseudo nMOS porter der nedtrekket må kjempe mot opptrekket må vi se på differansen mellom nedtrekk og opptrekk. Det er strømmen i nedtrekket minus strømmen som pMOS transistoren leverer som danner grunnlaget for beregning av logisk effort for nedtrekket. Dette kan vi modellere som en økning i logisk effort, eller økning i effektiv motstand, for nedtrekket.

Med hensyn på effektforbruk representerer pseudo nMOS ofte en dårligere løsning enn en komplementær port. Når utgangen skal trekkes ned til 0 vil pMOS transistoren være PÅ slik at det vil være et statisk effektforbruk lik strømmen gjennom pMOS transistoren multiplisert med forsynningsspenningen. Vi vil derfor dimensjonere pMOS transistoren så liten som mulig for å redusere det statiske effektforbruket, men denne løsningen vil redusere hastigheten når utgangen skal trekkes opp til 1.

Når utgangen er 1 er det ikke statisk effektforbruk i pseudo nMOS porten.

5.7.1 pseudo nMOS inverter

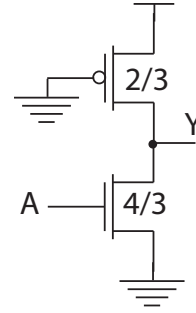


Figure 5.22: Pseudo nMOS inverter. (FIG6.12)

For invertieren i Fig. 5.22 har vi valgt en litt større nMOS transistor enn en enhetstransistor. Vi bestemmer først størrelsen på nMOS transistoren³. Dersom vi igjen antar at $\mu_n = 2\mu_p$ og at opptrekket skal ha 1/4 av styrken til nedtrekkskjeden vil dette gi nMOS- og pMOS transistor med størrelse relativt til enhetstransistor:

$$\begin{aligned}
 W_n &= \frac{4}{3} W_{n_{enhet}} \\
 &= \frac{4}{3} \\
 W_p &= 2 \frac{1}{4} \frac{4}{3} \\
 &= \frac{2}{3}.
 \end{aligned} \tag{5.9}$$

Vi har at $W_n = C_{inngang}$ og $W_p = C_{gate-pMOS}$. Logisk effort for opptrekket blir da:

$$\begin{aligned}
 g_u &= \frac{W_n}{W_p + \frac{1}{2}W_p} \\
 &= \frac{\frac{4}{3}}{\frac{2}{3} + \frac{1}{3}} \\
 &= \frac{4}{3}.
 \end{aligned} \tag{5.10}$$

Vi kan beregne motstanden for transistorene:

$$\begin{aligned}
 R_n &= \frac{1}{W_n} R \\
 &= \frac{3}{4} R \\
 R_p &= \frac{2}{\frac{2}{3}} R \\
 &= 3R.
 \end{aligned} \tag{5.11}$$

³I dette eksemplet velges størrelsen for nMOS transistoren lik 4/3 relativt til en enhetstransistor. Dette vil tilsvare en effektiv motstand i nedtrekket som er lik R .

Vi kan anvende Ohms lov og uttrykke strømmdifferansen mellom nMOS og pMOS transistoren, som vil være lik effektiv, eller netto, strømtrekket for nedtrekk:

$$\begin{aligned} I_n &= \frac{V_{DD}}{R_n} \\ &= \frac{4V_{DD}}{3R} \\ I_p &= \frac{V_{DD}}{R_p} \\ &= \frac{V_{DD}}{3R} \\ I_n - I_p &= \frac{V_{DD}}{R}. \end{aligned} \quad (5.12)$$

pseudo nMOS inverteren vil da ha et nedtrekk som tilsvarer nedtrekket for en inverter med en enhets nMOS transistor.

En annen måte å uttrykke dette på er en effektiv transkonduktans i nedtrekket:

$$\begin{aligned} G_n &= \frac{4}{3}R^{-1} \\ G_p &= \frac{1}{3}R^{-1} \\ G_n - G_p &= R^{-1}. \end{aligned} \quad (5.13)$$

Den effektive motstanden i nedtrekket blir da

$$\begin{aligned} R_{nedtrekk} &= \frac{1}{G_n - G_p} \\ &= R. \end{aligned} \quad (5.14)$$

Vi kan nå uttrykke logisk effort for opptrekk og nedtrekk:

$$\begin{aligned} g_u &= \frac{W_n}{W_p + \frac{1}{2}W_p} \\ &= \frac{\frac{4}{3}}{\frac{2}{3} + \frac{1}{2}\frac{2}{3}} \\ &= \frac{4}{3} \\ g_d &= \frac{W_n}{(W_n - \frac{1}{2}W_p) + 2(W_n - \frac{1}{2}W_p)} \\ &= \frac{\frac{4}{3}}{3(\frac{4}{3} - \frac{1}{3})} \\ &= \frac{4}{9} \\ g_{avg} &= \frac{\frac{4}{3} + \frac{4}{9}}{2} \\ &= \frac{8}{9}. \end{aligned} \quad (5.15)$$

Parasittisk tidsforsinkelse beregnes ut fra effektiv motstand og parasittisk kapasitans (intern kapasitans):

$$\begin{aligned} P_u &= R_{opptrekk}C_{intern} \\ &= R_p \left(\frac{2}{3} + \frac{4}{3} \right) C \\ &= 3R2C \\ &= 2\tau \\ P_d &= R_{nedtrekk}C_{intern} \\ &= 1R \left(\frac{2}{3} + \frac{4}{3} \right) C \\ &= R2C \\ &= \frac{2}{3}\tau \\ p_{avg} &= \frac{4}{3}\tau. \end{aligned} \quad (5.16)$$

5.7.2 2inngangs pseudo nMOS NAND port

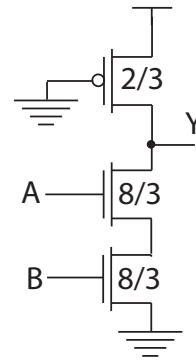


Figure 5.23: Pseudo nMOS NAND port. (FIG6.12)

For 2inngangs pseudo nMOS NAND porten i Fig. 5.23 antar vi at motstanden i opptrekket skal være 4 ganger så stor som den effektive motstanden i de to nMOS transistorene i nedtrekket. Dersom vi velger størrelsen på pMOS transistoren $W_p = 2/3$ vil motstanden i opptrekket være:

$$\begin{aligned} R_p &= 2 \left(\frac{1}{W_p} \right) R \\ &= 3R. \end{aligned} \quad (5.17)$$

Nedtrekket består av to nMOS transistorer som vi antar er like og med motstand lik R_n hver. De to nMOS transistorene vil til sammen utgjøre en motstand lik $2R_n$.

Vi setter ekvivalentmotstanden i de to nMOS transistorene lik $1/4$ av motstanden i opptrekket og finner størrelsen på nMOS transistorene:

$$\begin{aligned} R_n &= W_n^{-1} \\ R_{nMOSkjede} &= (W_n^{-1} + W_n^{-1}) R \end{aligned}$$

$$\begin{aligned}
&= \frac{2}{W_n} R \\
&= \frac{2 \cdot 3}{8} \\
&= \frac{3}{4} R \\
\frac{2}{W_n} R &= \frac{1}{4} R_p \\
2W_n &= \frac{3}{4} \\
W_n &= \frac{8}{3}.
\end{aligned}$$

(5.18)

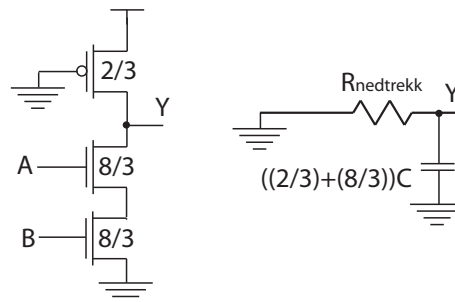


Figure 5.24: Enkel ekvivalent for nedtrekket i 2inngangs pseudo nMOS NAND port.

Motstanden i nedtrekket blir da:

$$\begin{aligned}
R_{nedtrekk} &= \left(R_{nMOSkjede}^{-1} - R_p^{-1} \right)^{-1} \\
&= \left(\frac{W_n}{2} - \frac{1}{3} \right)^{-1} R \\
&= R.
\end{aligned}$$

(5.19)

$$= \frac{10}{3} \tau$$

$$P_d = R_{nedtrekk} \left(\frac{10}{3} \right) C$$

$$= \frac{10}{9} \tau$$

$$P_{avg} = \frac{20}{9} \tau.$$

(5.21)

Logisk effort for opptrekk, og nedtrekk blir:

$$\begin{aligned}
g_u &= \frac{W_n}{W_p + \frac{1}{2}W_p} \\
&= \frac{\frac{8}{3}}{\frac{2}{3} + \frac{1}{3}} \\
&= \frac{8}{3} \\
g_d &= \frac{W_n}{3 \left(\left(\frac{1}{W_n} + \frac{1}{W_n} \right)^{-1} - \frac{1}{2}W_p \right)} \\
&= \frac{W_n}{(1+2) / \text{cdot} \left(\frac{1}{2}W_n - \frac{1}{2}W_p \right)} \\
&= \frac{\frac{8}{3}}{3 \left(\frac{4}{3} - \frac{1}{3} \right)} \\
&= \frac{8}{9} \\
g_{avg} &= \frac{16}{9}.
\end{aligned}$$

(5.20)

I dette tilfellet brukes ikke Elmore forsinkelsesmodell, men en mer optimistisk⁴ modell basert på ekvivalenten i Fig. 5.24. Parasittisk tidsforsinkelse for 2inngangs pseudo nMOS NAND port blir:

$$\begin{aligned}
P_u &= R_{opptrekk} C_{intern} \\
&= 3R \left(\frac{2}{3} + \frac{8}{3} \right) C \\
&= 10RC
\end{aligned}$$

⁴Ved bruk av Elmore modellen ville vi fått en parasittisk tidsforsinkelse på $\frac{13}{9} \tau$.

5.7.3 2inngangs pseudo nMOS NOR port

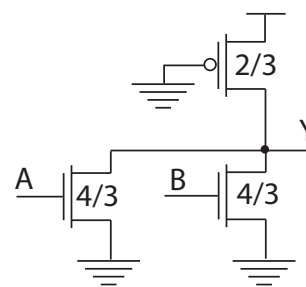


Figure 5.25: Pseudo nMOS NOR port. (FIG6.12)

For 2inngangs pseudo nMOS NOR porten i Fig. 5.25 antar vi at motstanden i opptrekket skal være 4 ganger så stor som den effektive motstanden i en nedtrekkstransistor. Dersom vi velger størrelsen på pMOS transistoren $W_p = 2/3$ vil motstanden i opptrekket være:

$$\begin{aligned}
R_p &= 2 \left(\frac{1}{W_p} \right) R \\
&= 3R.
\end{aligned}$$

(5.22)

Nedtrekket består av to nMOS transistorer i parallell som vi antar er like og med motstand lik R_n hver. Vi antar at bare en av transistorene er PÅ slik at den effektive motstanden i nedtrekkstransistorene blir R_n . Vi antar at motstanden i nedtrekket skal være 1/4 av motstanden i opptrekket:

$$R_n = \frac{1}{4} R_{opptrekk} \quad (5.23)$$

$$= \frac{3}{4}R,$$

som gir transistorstørrelse $W_n = 4/3$, og den effektive motstanden i nedtrekket blir:

$$R_{nedtrekk} = \left(W_n - \frac{1}{2}W_p \right)^{-1} = \left(\frac{4}{3} - \frac{1}{2} \right)^{-1} R \quad (5.24)$$

$$= R. \quad (5.25)$$

Logisk effort for opptrekk, og nedtrekk blir:

$$\begin{aligned} g_u &= \frac{W_n}{W_p + \frac{1}{2}W_p} \\ &= \frac{\frac{4}{3}}{\frac{2}{3} + \frac{1}{3}} \\ &= \frac{4}{3} \\ g_d &= \frac{W_n}{3 \left(W_n - \frac{1}{2}W_p \right)} \\ &= \frac{\frac{4}{3}}{3 \left(\frac{4}{3} - \frac{1}{3} \right)} \\ &= \frac{4}{9} \\ g_{avg} &= \frac{8}{9}. \end{aligned} \quad (5.26)$$

Parasittisk tidsforsinkelse for 2inngangs pseudo nMOS NOR port blir:

$$\begin{aligned} P_u &= R_{opptrekk} C_{intern} \\ &= 3R \left(\frac{2}{3} + \frac{4}{3} + \frac{4}{3} \right) C \\ &= 10RC \\ &= \frac{10}{3}\tau \\ P_d &= R_{nedtrekk} \left(\frac{2}{3} + \frac{4}{3} + \frac{4}{3} \right) C \\ &+ R \frac{10}{3} C \\ &= \frac{10}{9}\tau \\ P_{avg} &= \frac{20}{9}\tau. \end{aligned} \quad (5.27)$$

5.7.4 Pseudo nMOS design

Pseudo nMOS NAND porter er tregere enn komplementære NAND porter. For NOR porter er pseudo nMOS ikke

tilsvarende tregere. Logisk effort for en pseudo nMOS NOR port er uavhengig av antall innganger. Man skal være oppmerksom på støymarginer som vil være dårligere for pseudo nMOS porter enn komplementære porter. For å sikre gode støymarginer svekker man pMOS transistorene, men dette vil gå ut over hastigheten. Parasittisk tidsforsinkelse for opptrekk vil øke når pMOS transistoren svekkes, mens parasittisk tidsforsinkelse for nedtrekket vil reduseres fordi det blir lettere for nedtrekket å trekke utgangen lav. En svekket pMOS transistor vil derfor øke forskjellen i parasittisk tidsforsinkelse i opp- og nedtrekket.

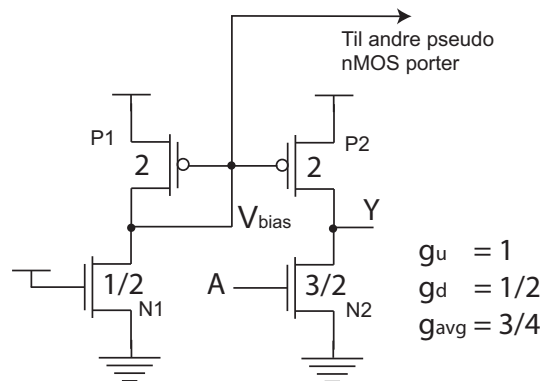


Figure 5.26: Biasering av pseudo nMOS porter. (FIG6.15)

For å redusere følsomhet for en spesifikk prosess kan man benytte en biaseringskrets som vist i Fig. 5.26. Formålet med biaseringskretsen er å generere en biaseringsspenning V_{bias} , som anvendt på en pMOS transistor vil gi en strøm i pMOS transistorer (for eksempel P2) som er lik $1/3$, som er forholdet mellom W_{N1} og W_{N2} , av strømmen i nMOS transistorer (for eksempel N2). Biaseringsspenningen kan brukes som inngang til flere pseudo nMOS porter. Logisk effort for porten som består av transistorene P2 og N2 er gitt av

$$\begin{aligned} g_u &= \frac{W_{N2}}{W_{P2} + \frac{1}{2}W_{P2}} \\ &= \frac{\frac{3}{2}}{2 + 1} \\ &= \frac{1}{2} \\ g_d &= \frac{W_{N2}}{(W_{N2} - \frac{1}{2}W_{N2}) + 2(W_{N2} - \frac{1}{2}W_{N2})} \\ &= \frac{\frac{3}{2}}{3 \left(\frac{3}{2} - \frac{3}{4} \right)} \\ &= \frac{2}{3}. \end{aligned} \quad (5.28)$$

5.7.5 Eksempel

Vi kan implementere en kinngangs AND port ved hjelp av invertere og en kinngangs pseudo nMOS NOR port som vist i Fig. 5.27. Utgangslasten er representert som en

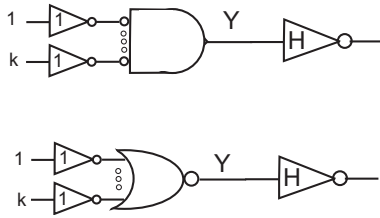


Figure 5.27: AND funksjon implementert med pseudo nMOS NOR port. (FIG6.13)

inverter med størrelse lik H ganger enhetsinverter. Elektrisk effort for kjeden bestående av pseudo nMOS NOR porten og en inverter på inngangen er lik H . Vi antar at NOR porten er dimensjonert som NOR porten i Fig. 5.21 vil gjennomsnittelig logisk effort for NOR porten være $8/9$. Gjennomsnittelig logisk effort for kjeden blir $G = 1 \cdot (8/9) = 8/9$. Forgrenings-effort er lik 1, slik at kjede effort blir $F = GBH = (8/9) \cdot 1 \cdot H = (8H)/9$. Optimal port effort blir da $f' = \sqrt{(8H)/9}$. Vi kan nå bestemme optimal inngangskapasitans for NOR porten:

$$\begin{aligned}
 C_{inngang} &= \frac{C_{ekstern} \cdot g}{f'} \\
 &= \frac{H \frac{8}{9}}{\sqrt{\frac{8H}{9}}} \\
 &= \frac{2\sqrt{2H}}{3}.
 \end{aligned} \tag{5.29}$$

Vi kan dermed sette størrelsen på nMOS transistorene $W_n = C_{inngang} = (2\sqrt{2H})/3$, og størrelsen på pMOS transistoren blir $W_p = W_n/2 = \sqrt{2H}/3$. Vi kan da beregne effektive motstander for opptrekk og nedtrekk:

$$\begin{aligned}
 R_{opptrekk} &= 2W_p^{-1} \\
 &= \frac{6}{\sqrt{2H}} \\
 R_{nedtrekk} &= \left(W_n - \frac{1}{2}W_p\right)^{-1} R \\
 &= \left(\frac{2\sqrt{2H}}{3} - \frac{1}{2} \frac{\sqrt{2H}}{3}\right)^{-1} R \\
 &= \frac{2}{\sqrt{2H}} R.
 \end{aligned} \tag{5.30}$$

Parasittisk forsinkelse i NOR porten blir:

$$\begin{aligned}
 P_u &= \frac{6}{\sqrt{2H}} R \left(\frac{\sqrt{2H}}{3} + k \frac{2\sqrt{2H}}{3} \right) C \\
 &= \frac{6\sqrt{2H}(1+2k)}{3\sqrt{2H}} RC \\
 &= 2(1+2k) RC
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{2(1+2k)}{3} \tau \\
 P_d &= \frac{2}{\sqrt{2H}} R \left(\frac{\sqrt{2H}(1+2k)}{3} \right) C \\
 &= \frac{2(1+2k)}{3} RC \\
 &= \frac{2(1+2k)}{9} \tau \\
 P_{avg} &= \left(\frac{\frac{2(1+2k)}{3} + \frac{2(1+2k)}{9}}{2} \right) \tau \\
 &= \frac{4(1+2k)}{9} \tau.
 \end{aligned} \tag{5.31}$$

Total tidsforsinkelse blir:

$$\begin{aligned}
 D &= Nf' + P \\
 &= 2\sqrt{(8H)/9} + 1 + \frac{4(1+2k)}{9} \\
 &= \frac{4\sqrt{2H}}{3} + \frac{(13+8k)}{9}.
 \end{aligned} \tag{5.32}$$

5.7.6 Mål

Kunne beregne logisk effort og parasittisk tidsforsinkelse for pseudo nMOS porter.

5.8 Ganged CMOS

(Kapittel 6.2.2.2 side 330 - 331)

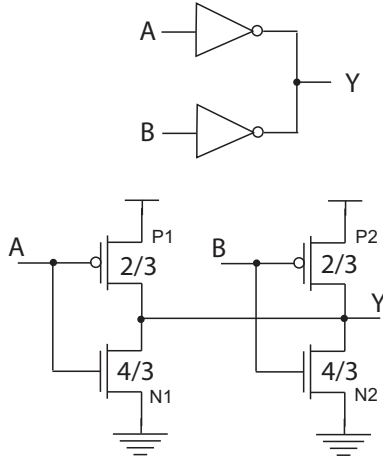


Figure 5.28: Ganged CMOS 2-inngangs symmetrisk NOR port. (FIG6.18)

En ganged CMOS port bestående av to invertere er vist i Fig. 5.28. Når portens transistorer dimensjoneres som vist i porten vil portens funksjon bli 2inngangs NOR (symmetrisk). Dersom pMOS transistorene hver for seg var sterkere⁵ enn en nMOS transistor ville funksjon bli en 2inngangs NAND. Kretsens elektriske egenskaper vil være svært avhengig av logiske verdier på inngangene. Det vil være statisk effektforbruk for visse inngangsmønstre for en ganged CMOS port.

5.8.1 A = B = 0

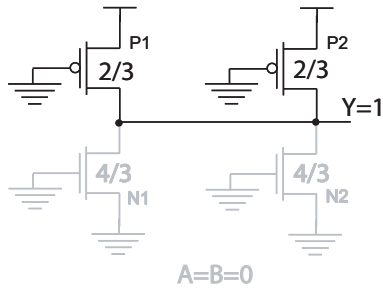


Figure 5.29: Ganged CMOS 2-inngangs symmetrisk NOR port med innganger $A = B = 0$. (FIG6.18)

Ganged NOR port med inngangene $A = B = 0$ er vist i Fig. 5.29. I dette tilfellet vil vi ha to pMOS transistorer i parallell som trekker utgangen opp til logisk 1. Begge nMOS transistorene er skrudd av. Det vil ikke være statisk effektforbruk for denne porten ut over effektforbruk som følge av nMOS transistorene som opererer i svak inversjon (AV). Vi har:

$$g_u = \frac{W_p + W_n}{2W_p + W_p}$$

⁵Vi kunne for eksempel valgt $W_n = 1/3$ og $W_p = 8/3$.

$$\begin{aligned} &= \frac{\frac{2}{3} + \frac{4}{3}}{2\frac{2}{3} + \frac{2}{3}} \\ &= 1 \\ P_u &= R_{\text{oppptrekk}} C_{\text{intern}} \\ &= \left(\frac{R_p \cdot R_p}{R_p + R_p} \right) \cdot \left(\frac{2}{3} + \frac{2}{3} + \frac{4}{3} + \frac{4}{3} \right) \\ &= \frac{1}{2} R_p 4C \\ &= \frac{1}{2} \cdot 2 \cdot \frac{3}{2} R 4C \\ &= 6RC \\ &= 2\tau. \end{aligned} \tag{5.33}$$

5.8.2 A og B er forskjellige

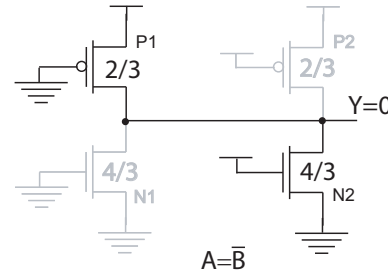


Figure 5.30: Ganged CMOS 2-inngangs symmetrisk NOR port med innganger $A \neq B$. (FIG6.18)

Ganged NOR port med inngangssignalene $A \neq B$ er vist i Fig. 5.30. I dette tilfellet vil vi ha en pMOS transistorer som er PÅ og en nMOS transistor som er PÅ. Dette vil ligne en 2inngangs pseudo nMOS NOR port med en ekstra intern kapasitans for pMOS transistoren som er skrudd av. nMOS transistoren vil være sterkere enn pMOS transistoren slik at utgangen trekkes ned til 0. I dette tilfellet har vi en pMOS transistor som er PÅ og som vil bidra med statisk strøm til nedtrekket via nMOS transistoren som er PÅ. Det statiske effektforbruket er proporsjonalt med strømmen gjennom pMOS transistoren. Vi har:

$$\begin{aligned} g_d &= \frac{W_p + W_n}{(W_n - \frac{1}{2}W_p) + 2(W_n - \frac{1}{2}W_p)} \\ &= \frac{\frac{2}{3} + \frac{4}{3}}{3(\frac{4}{3} - \frac{1}{2}\frac{2}{3})} \\ &= \frac{2}{3} \\ P_d &= R_{\text{nedtrekk}} C_{\text{intern}} \\ &= \left(W_n - \frac{1}{2}W_p \right)^{-1} R \cdot (4C) \\ &= \left(\frac{4}{3} - \frac{1}{2}\frac{2}{3} \right) R 4C \\ &= R 4C \\ &= \frac{4}{3} \tau. \end{aligned} \tag{5.34}$$

5.8.3 A = B = 1

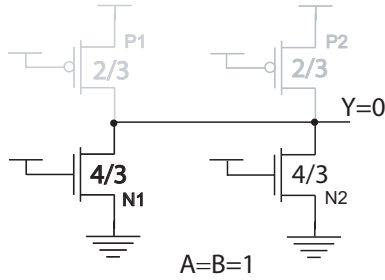


Figure 5.31: Ganged CMOS 2inngangs symmetrisk NOR port med innganger $A = B = 1$. (FIG6.18)

Ganged NOR port med inngangsingene $A = B = 1$ er vist i Fig. 5.31. I dette tilfellet vil vi ha to nMOS transistorer i parallell som trekker utgangen ned til logisk 0. Begge pMOS transistorene er skrudd av. Her er det statiske effektforbruket redusert til et minimum. Vi har:

$$\begin{aligned}
 g_d &= \frac{W_p + W_n}{2W_n + 4W_n} \\
 &= \frac{\frac{2}{3} + \frac{4}{3}}{2 \cdot \frac{4}{3} + 4 \cdot \frac{4}{3}} \\
 &= \frac{1}{4} \\
 P_d &= R_{nedtrekk} C_{intern} \\
 &= \left(\frac{R_n \cdot R_n}{R_n + R_n} \right) \cdot (4C) \\
 &= \frac{1}{2} R_n 4C \\
 &= \frac{1}{2} \cdot \frac{3}{4} R 4C \\
 &= \frac{3}{2} RC \\
 &= \frac{1}{2} \tau. \tag{5.35}
 \end{aligned}$$

5.8.4 Mål

Forstå hvordan en ganged 2inngangs NOR port virker. Kunne beregne logisk effort og parasittiske tidsforsinkelser.

5.9 Source følger opptrekksslogikk

(Kapittel 6.2.2.3 side 331)

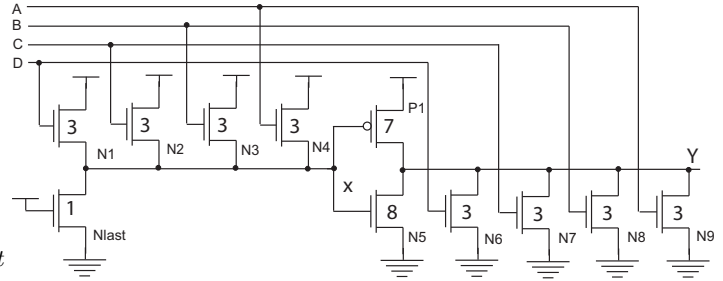


Figure 5.32: Source følger opptrekksslogikk. (FIG6.19)

En port som kan minne om en pseudo nMOS port er vist i Fig. 5.32. Kretsen kalles *source følger opptrekksslogikk* (SFPL Source Follower Pull-up Logic). Her er opptrekket kontrollert av inngangssignalene. Biaseringskretsen består av bare nMOS transistorer, der nMOS transistorene som styres av inngangene bidrar til å dra biaseringsutgangen x nærmere logisk 1 som funksjon av antall innganger som er 1. N_{last} transistoren vil trekke x ned mot 0, og 1ere på inngangssignalene vil bidra til å trekke x opp mot 1 gjennom nMOS transistorer. Husk at nMOS transistorene som er koblet opp mot V_{DD} i biaseringskretsen har source koblet til x , slik at disse transistorene vil være svake. Det vil være et relativt betydelig statisk effektforbruk i SFPL porter, det faktiske effektforbruket er avhengig av inngangsmønstre.

5.9.1 Mål

Etablere en forenklet kjennskap til SFPL logikk.

5.10 Kaskode spenning Svitsj logikk

(Kapittel 6.2.3 side 331 - 332)

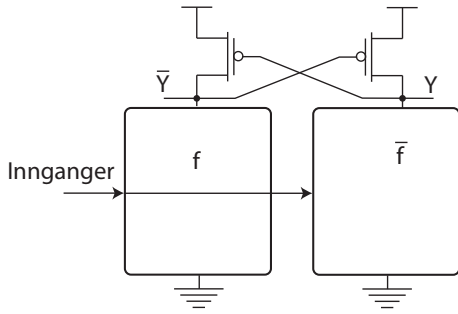


Figure 5.33: Kaskode spenning svitsj logikk .(FIG6.20a)

Ved å bruke to komplementære nedtrekk til å styre hvert sitt opptrekk som vist i Fig. 5.33 vil vi få en port som ikke har statisk effektforbruk fordi opptrekkstransistoren som ikke skal trekke en utgang til 1 blir slått AV. Vi kaller en slik logikk for *kaskode spenning svitsj logikk* (CVSL Cascade Voltage Switch Logic). I CVSL logikk vil en, og bare en, av utgangene Y eller \bar{Y} trekkes ned til 0, og som vil gjennom en pMOS trekke den inverse utgangen til 1. En fordel med CVSL logikk i forhold til pseudo nMOS logikk er at statiske effektforbruk er redusert til et minimum, og at pMOS transistorene kan dimensjoneres opp for å redusere parasittisk tidsforsinkelse i opptrekket uten en økning i statisk effektforbruk. Vi trenger inverterte inngangssignaler i tillegg og mange nMOS transistorer. Dette kan være nyttig logikk dersom vi allerede har tilgang på inverterte innganger og at vi trenger invertert utgang i tillegg til ordinær utgang. CVSL porter vil ha to utganger som alltid svitsjer motsatt vei som kan være gunstig i tilknytning til støyfølsomme kretser⁶.

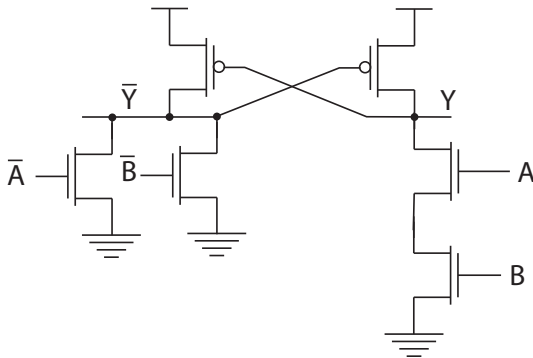


Figure 5.34: Kaskode spenning svitsj logikk for 2inngangs NAND funksjon.(FIG6.20b)

CVSL porten som er vist i Fig. 5.34 er en NAND port, dvs. $Y = \bar{A} \cdot \bar{B}$. Det alternative nedtrekket består av to parallellkoblede nMOS transistorer styrt av \bar{A} og \bar{B} som

⁶Støyfølsomme kretser på samme brikke vil typisk være analoge kretser i et mixed signal system.

vil representere funksjonen $\bar{Y} = \overline{\bar{A} + \bar{B}} = A \cdot B$, som er ekvivalent med $Y = \bar{A} \cdot \bar{B}$.

5.10.1 Eksempel

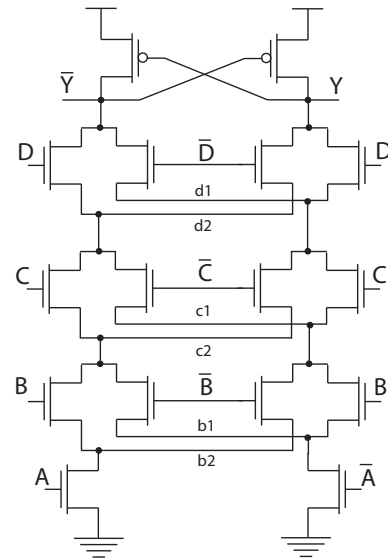


Figure 5.35: Kaskode spenning svitsj logikk for 4inngangs XNOR, dvs. $Y = \bar{A} \oplus \bar{B} \oplus \bar{C} \oplus \bar{D}$ funksjon.(FIG6.20c)

Et eksempel på en variant av CVSL logikk der de to alternative nedtrekkene deler transistorer er vist i Fig. 5.35. Kretsen er en 4inngangs XNOR port. Dersom det er en og bare en inngang som er 1 vil utgangen Y bli lik 1. Legg merke til at for utgangen \bar{Y} er funksjonen en 4inngangs XOR port.

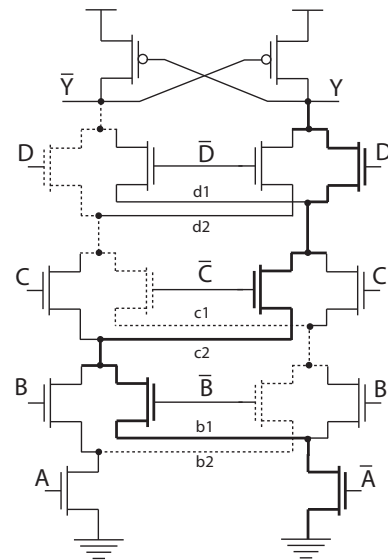


Figure 5.36: Kaskode spenning svitsj logikk for 4inngangs XNOR funksjon.(FIG6.20c)

4inngangs XNOR krets der nedtrekket for inngangsmønstrer D , \bar{C} , \bar{B} og \bar{A} er vist i Fig. 5.36. Dette representerer inngangsekvensen 1000, dvs. en ener.

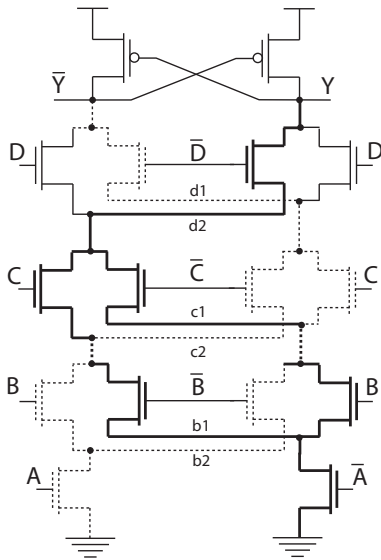


Figure 5.37: Kaskode spenning svitsj logikk for 4-inngangs XNOR funksjon. (FIG6.20c)

Dersom vi antar at $D = 0$ vil de to mitterste transistorene i den øverste rekka av nMOS transistorer være på. Det vil derifra være en rekke kombinasjoner av inngangsmønstre for inngangene A , B og C som gir en og bare en 1er. De alternative nedtrekksmulighetene er vist i Fig. 5.37.

5.10.2 Mål

Kunne implementere logiske funksjoner ved hjelp av CVSL logikk.

5.11 Introduksjon til dynamiske kretser

(Kapittel 6.2.4 side 332 - 334)

Poenget med pseudo nMOS, ganged CMOS, SFPL og CVLS logikk er å redusere inngangskapasitansen ved å unngå å koble inngangene til pMOS transistorer. Ulempe med disse logikkstilene er knyttet delvis til signifikant statisk effektforbruk, spesielt for pseudo nMOS og ganged CMOS, men i første rekke økning i tidsforsinkelse for transisjoner fra 0 til 1. Alle logikkstilene er *statiske*, dvs. det vil til enhver tid være transistorer som er skrudd PÅ og som bidrar til å holde en utgang stabil til en riktig spenningsreferanse, 0 eller 1, uavhengig av tid.

Dynamiske porter eller kretser er grunnleggende forskjellige fra statiske porter ved at de for enkelte inngangsmønstre vil ha tilstander der utgangen ikke drives. Et eksempel kan være at utgangen for en port i utgangspunktet er logisk 1, men der hverken opptrekk eller nedtrekk er PÅ slik at utgangen ikke er sikret å beholde sin logiske verdi gjennom drivegenskaper i porten. For en slik port er det to forhold som er viktige:

1. Andre kretser eller porter som har portens utgangssignal som inngang kan ikke forvente å kunne tolke signalet korrekt utover en liten tidsperiode. Vi kan ikke anta at en dynamisk port er i stand til å sikre en stabil logisk utgang utover en kort tidsperiode som er avhengig av last og lekkasje.
2. Dersom det er alternative porter som driver samme utgang, som for eksempel *tristate porter* er det ikke problemer med dynamiske (ikke drevne) utgangssignaler.

Desom et signal er bestemt av en dynamisk port må vi sørge for at utgangssignalet blir generert ofte nok, dvs. med kort tidsintervall, slik at signalet til enhver tid er gyldig. Det er derfor vanlig å benytte *klokkesignaler* til å synkronisere dynamiske porter.

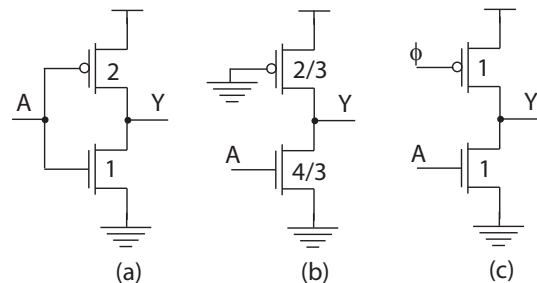


Figure 5.38: (a) Komplementær inverter, (b) pseudo nMOS inverter og (c) dynamisk inverter. (FIG6.21)

Komplementær-, pseudo nMOS- og dynamisk invertere er vist i Fig. 5.38. Den dynamiske inverteren (c) har liten kapasitiv inngangslast for inngangssignalet A . Dette

tilsvarende en pseudo nMOS inverter, men i motsetning til pseudo nMOS invertieren er pMOS transistoren styrt av et klokkesignal som svitsjer mellom 0 og 1 og dermed skruer pMOS transistoren PÅ og AV.

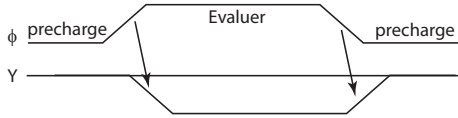


Figure 5.39: Precharge og evaluering av dynamiske porter. (FIG6.22)

Klokkesignalet ϕ og utgangen Y på en dynamisk inverter er vist i Fig. 5.39. Når klokkesignalet er 0 vil utgangen trekkes opp til 1 via pMOS transistoren som er skrudd PÅ. Vi kaller denne perioden for *precharge*. Her er det viktig at inngangssignalet ikke overstyrer klokkesignalet og samtidig trekker utgangen mot 0. Vi må altså forutsette at A er 0 når ϕ er 0. I evalueringsperioden (fasen) når klokkesignalet er 1 kan utgangen trekkes ned til 0 dersom A er 1. I evalueringsperioden *evaluerer* porten inngangssignalet, eller inngangssignalene dersom det er snakk om en mer kompleks port. For at denne porten skal kunne anvendes må det være en nøye sammenheng mellom, eller synkronisering av, ϕ og inngangen. Inngangen kan ikke tillates å bli 1 før klokkesignalet er 1.

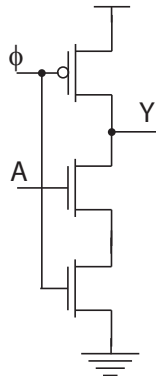


Figure 5.40: Footed dynamisk inverter. (FIG6.23)

En endring av den dynamiske invertieren som er vist i Fig. 5.40 vil resultere i en port som er mer robust og mindre avhengig av synkronisering mellom ϕ og inngangssignal. For denne porten vil utgangen alltid trekkes opp til 1 når ϕ er 0 uavhengig av inngangssignalet A . Vi kaller denne varianten av dynamisk porter for *footed porter*.

En transistor i tillegg i nedtrekket vil påvirke logisk effort og parasittisk tidsforsinkelse for de logiske portene.

5.11.1 Footed inverter

De to implementasjonene av dynamisk inverter som er vist i Fig. 5.41 vil ha ulik logisk effort og parasittisk tidsforsinkelse. For ikke-footed dynamisk inverter har vi:

$$g_d = \frac{W_n}{3W_n}$$

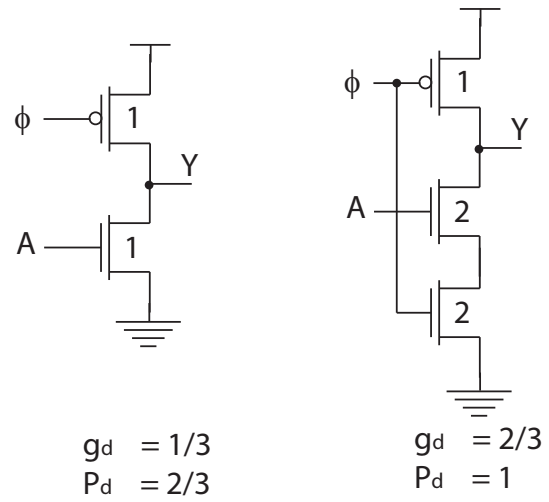


Figure 5.41: Dynamisk invertiere. (FIG6.25)

$$\begin{aligned}
 &= \frac{1}{3} \\
 P_d &= 1R \cdot 2C \\
 &= 2RC \\
 &= \frac{2}{3}\tau, \tag{5.36}
 \end{aligned}$$

der det for ikke-footed dynamisk inverter ved opptrekk er forutsatt at $A = 0$. Det er viktig å være klar over at logisk effort og parasittisk tidsforsinkelse for opptrekk er mindre viktig enn logisk effort og parasittisk tidsforsinkelse for nedtrekket. Dette skyldes at i en vanlig anvendelse av dynamiske porter vil en kjede av logiske porter precharges samtidig (i parallell), mens nedtrekket vil påvirkes av kjedens lengde.

Vi kan beregne den effektive motstanden i nedtrekket:

$$\begin{aligned}
 R_{nedtrekk} &= (R_n + R_n) \\
 &= \left(\frac{1}{W_n} + \frac{1}{W_n} \right) R \\
 &= \left(\frac{1}{2} + \frac{1}{2} \right) R \\
 &= R. \tag{5.37}
 \end{aligned}$$

Hver for seg representerer nMOS transistorene en motstand lik $(1/2)R$ og tilsammen i en kjede vil den effektive motstanden være lik R . Forholdet mellom motstanden for hver av nMOS transistorene i en NAND2 port og motstanden i nMOS transistoren i en enhetsinverter er gitt av:

$$\begin{aligned}
 R_n &= \frac{W_{enhets-nMOS}}{W_n} R \\
 &= \frac{1}{W_n} R. \tag{5.38}
 \end{aligned}$$

I dette aktuelle tilfellet er $1/W_n = 1/2$. Vi ser at det effektive nedtrekket i porten tilsvarer nedtrekket i en enhetsinverter.

For footed inverter har vi:

$$\begin{aligned}
 g_d &= \frac{W_n}{3 \left(\frac{1}{W_n} + \frac{1}{W_n} \right)^{-1}} \\
 &= \frac{W_n}{3 \frac{W_n}{2}} \\
 &= \frac{2}{3} \\
 P_d &= \left(\frac{1}{W_n} + \frac{1}{W_n} \right) R \cdot (W_p + W_n) C \\
 &= \left(\frac{1}{2} + \frac{1}{2} \right) R \cdot (1 + 2) C \\
 &= 3RC \\
 &= 1\tau.
 \end{aligned} \tag{5.39}$$

5.11.2 Footed NAND2 port

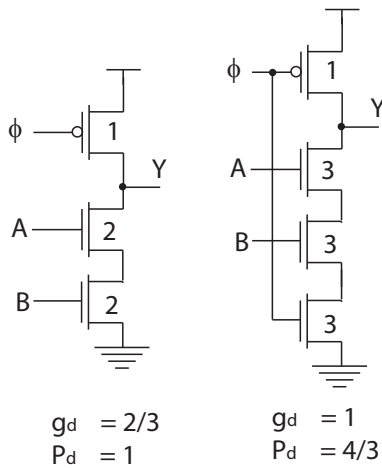


Figure 5.42: *Dynamisk NAND2 porter. (FIG6.25)*

De to implementasjonene av dynamisk 2inngangs NAND porter som er vist i Fig. 5.42 vil ha ulik logisk effort og parasittisk tidsforsinkelse. For ikke-footed 2inngangs dynamisk NAND port har vi:

$$\begin{aligned}
 g_d &= \frac{2}{3} \\
 P_d &= \left(\frac{1}{2} + \frac{1}{2} \right) R \cdot (1 + 2) C \\
 &= 3RC \\
 &= 1\tau,
 \end{aligned} \tag{5.40}$$

der det for ikke-footed dynamisk NAND2 port ved opptrekk er forutsatt at A eller B = 0.

For footed 2inngangs NAND port har vi:

$$\begin{aligned}
 g_d &= \frac{W_n}{3 \left(\frac{1}{W_n} + \frac{1}{W_n} + \frac{1}{W_n} \right)^{-1}} \\
 &= \frac{3}{3 \left(\frac{1}{3} + \frac{1}{3} + \frac{1}{3} \right)^{-1}} \\
 &= 1 \\
 P_d &= \left(\frac{1}{3} + \frac{1}{3} + \frac{1}{3} \right) R \cdot (1 + 3) C \\
 &= 4RC \\
 &= \frac{4}{3}\tau.
 \end{aligned} \tag{5.41}$$

5.11.3 Footed NOR2 port

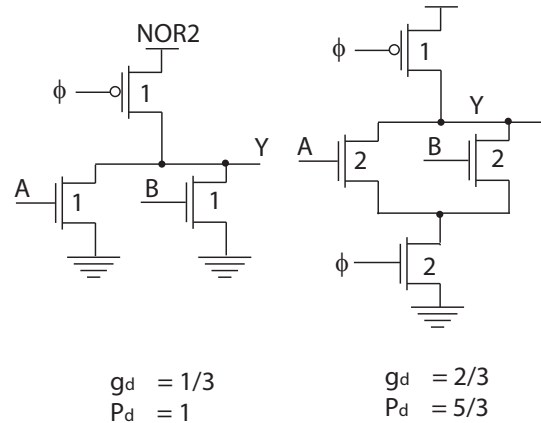


Figure 5.43: *Dynamisk NOR2 porter. (FIG6.25)*

De to implementasjonene av dynamisk 2inngangs NOR porter som er vist i Fig. 5.43 vil ha ulik logisk effort og parasittisk tidsforsinkelse. For ikke-footed 2inngangs dynamisk NOR port har vi:

$$\begin{aligned}
 g_d &= \frac{1}{3} \\
 P_d &= 1R \cdot (1 + 2) C \\
 &= 3RC \\
 &= 1\tau,
 \end{aligned} \tag{5.42}$$

der det for ikke-footed dynamisk NOR2 port ved opptrekk er forutsatt at A og B = 0.

For footed 2inngangs NOR port har vi:

$$\begin{aligned}
 g_d &= \frac{2}{3 \left(\frac{1}{W_n} + \frac{1}{W_n} \right)^{-1}} \\
 &= \frac{2}{3 \left(\frac{1}{2} + \frac{1}{2} \right)^{-1}} \\
 &= \frac{2}{3}
 \end{aligned}$$

$$\begin{aligned}
 P_d &= \left(\frac{1}{2} + \frac{1}{2}\right) R \cdot (1 + 2 + 2) C \\
 &= 5RC \\
 &= \frac{5}{3}\tau.
 \end{aligned}
 \tag{5.43}$$

En footed NOR port vil få en større relativ økning i parasittisk tidsforsinkelse enn en footed NAND port, fordi antall nMOS transistorer i nedtrekkskjeden doubles, mens antallet for NAND porten økes fra $2(n)$ til $3(n+1)$.

5.11.4 Precharge og evalueringsdetaljer

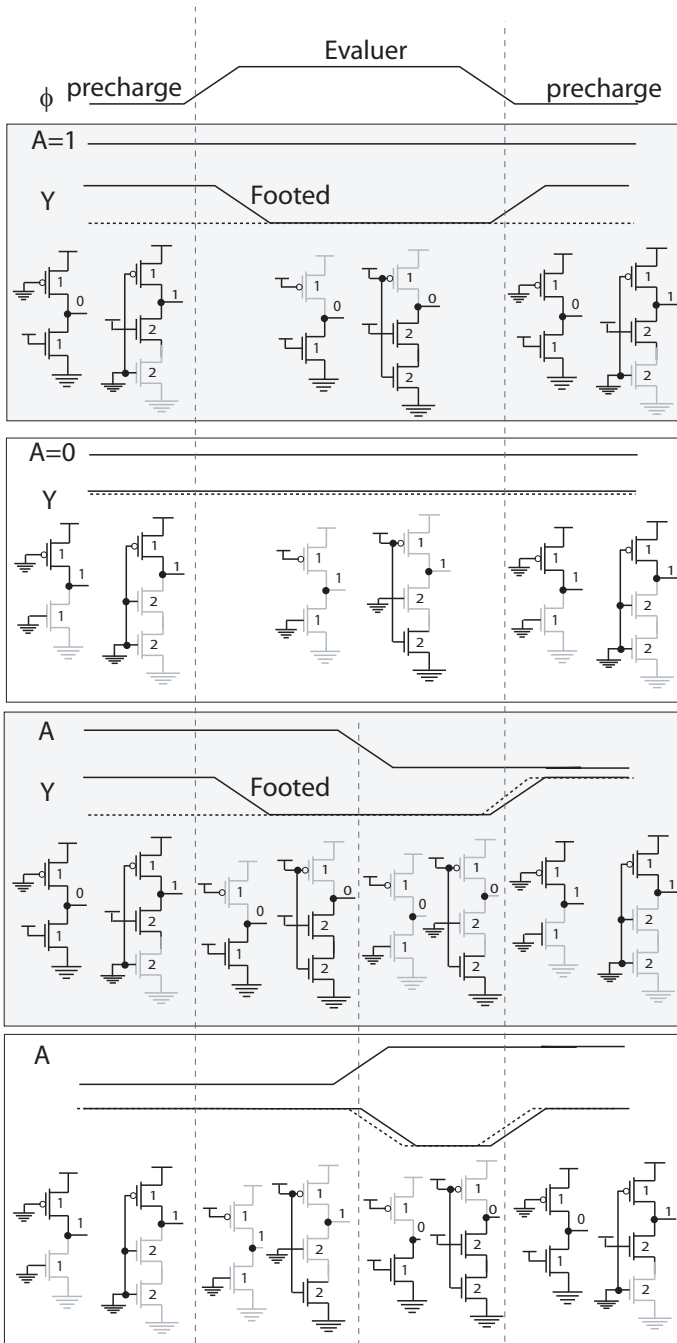


Figure 5.44: Precharge og evaluering av dynamiske invertere.

Detaljert precharge og evaluering av en footed- og ikke footed dynamiske invertere er vist i Fig. 5.44.

I det første tilfellet når $A = 1$ vil footed inverter fungere riktig, dvs. inverteren vil precharges riktig til 1 og utgangen trekkes ned til 0 ($Y = \bar{A}$) i evalueringsfasen. Ikke-footed inverter vil ikke precharges til 1 fordi pMOS transistoren er for svak i forhold til nMOS transistoren.

I tilfelle to når $A = 0$ vil begge inverterne precharges til 1. nMOS transistoren i ikke-footed inverter er skrudd AV. Når klokkesignalet er 1 i evalueringsfasen vil utgangen på inverterne være udrevet (tristate) og vil ikke endre verdi dersom denne perioden (evalueringfasen) er tilstrekkelig kort (dvs. frekvensen på klokkesignalet er tilstrekkelig høy).

I tilfelle tre antar vi at A er 1 i første precharge periode for å så å svitsje til 0 i løpet av evalueringsperioden. Ikke-footed inverter vil ikke kunne precharges til 1 på grunn av nMOS transistoren som drar utgangen til 0. I første del av evalueringsperioden vil utgangen på begge inverterne dras ned til 0 fordi pMOS transistorer stenges og nedtrekkene skrur PÅ. Begge inverterne har nå riktig verdi ($Y = \bar{A}$). I siste del av evalueringsperioden når A endres fra 1 til 0 vil utgangene på begge inverterne forbli lave fordi begge pMOS transistorene er skrudd AV. I dette tilfellet er både opprekk og nedtrekk for begge inverterne AV og utgangen drives ikke og vil ligge på samme verdi som i første del av evalueringsperioden. Dette vil representere en feil verdi fordi utgangen er i denne perioden lik inngangen ($Y = A$).

I det siste tilfellet antar vi at A er 0 i første precharge periode for å så å svitsje til 1 i løpet av evalueringsperioden. Begge inverterne precharges riktig i første precharge periode og vil beholde verdien logisk 1 i første halvdel av evalueringsperioden fordi $A = 0$. I siste halvdel av evalueringsperioden vil utgangen på begge inverterne endres til logisk 0 fordi $A = 1$, som er riktig verdi.

Hvis vi bare ser på evalueringsperioden vil det ikke være forskjell på footed- og ikke-footed inverter. Vi kan dermed slå fast at det å introdusere footed inverter ikke vil resultere at dynamiske porter vil evaluere (virke) riktig.

Evalueringsproblemet for en footed inverter er vist i Fig. 5.45. Under precharge vil både X og Y være logisk 1. Helt i starten av evalueringsperioden vil X trekkes ned, dersom vi antar at A er 1. Samtidig vil Y trekkes ned fordi X er 1 idet ϕ svitsjer fra 0 til 1. X vil ikke bli trukket ned til 0 før Y er blitt trukket ned mot 0. Den riktige verdien for Y skal være 1 fordi X er 0, med det er ikke mulig å trekke Y opp mot 1 når utgangen er lav, før neste precharge periode. I dette tilfellet var det kritisk at Y rakk å respondere på X før X fikk sin riktige verdi. Ved sammenkobling av flere dynamiske porter i en kjede vil problemet forsterkes utover i kjeden. Problemet er at vi precharger inngangene til nMOS transistorene i nedtrekkskjedene til 1 og dermed er transistorene skrudd PÅ.

For å kunne garantere riktig utgang må inngangene til transistorer ved precharge sørge for at transistorene er skrudd AV.

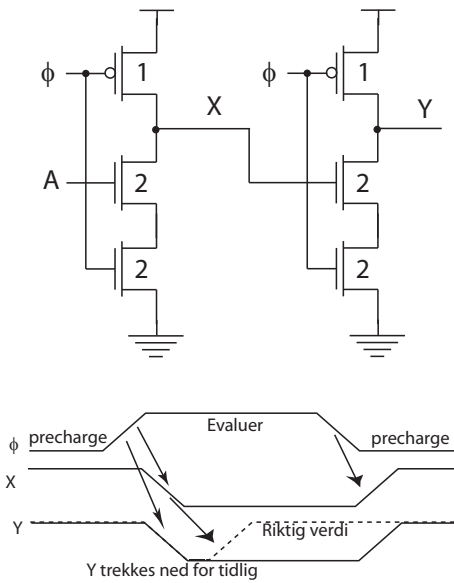


Figure 5.45: Precharge og evaluering av footed dynamiske invertere. (FIG6.27)

5.11.5 Mål

Forstå fundamentale problemer med footed og ikke-footed dynamiske porter.

5.12 Domino logikk

(Kapittel 6.2.4.1 side 334 - 336)

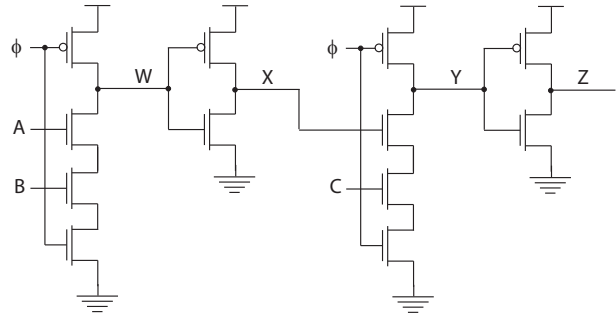


Figure 5.46: Domino logikk. (FIG6.28a)

Vi kan unngå det grunnleggende problemet med footed dynamisk logikk ved å sette inn en inverter, eller et oddet antall inverterende porter, mellom hver dynamisk port som vist i Fig. 5.46. Vi kaller denne logikkstilen *domino logikk*. For precharge porten med utgang Y vil inngangen fra en tilsvarende port via en inverter X alltid være 0 under precharge.

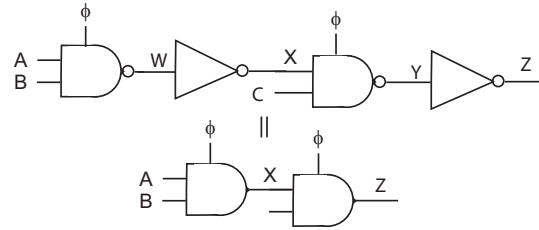


Figure 5.47: Domino logikk. (FIG6.28bc)

Detaljer for domino logikk er vist i Fig. 5.47. Vi antar at $A = 0$ ved precharge⁷. Legg merke til at det ikke er nødvendig å forutsette at inngangen B og C er 0 ved precharge fordi nedtrekket er skrudd av ved hjelp av A og X. Legg merke til at dersom vi kan forutsette at ett av inngangssignalene i nedtrekkskjeden i domino logikk er 0 vil det ikke være nødvendig med den nederste transistoren i nedtrekket. Vi kan med andre ord bruke en ikke-footed dynamisk port.

En domino implementasjon av en 8inngangs multiplerer er vist i Fig. 5.48. Vi antar at alle kontrollsignalene $S_0 - S_7$ er 0 ved precharge.

⁷Vi kan anta at A er en invertert utgang fra en domino port.

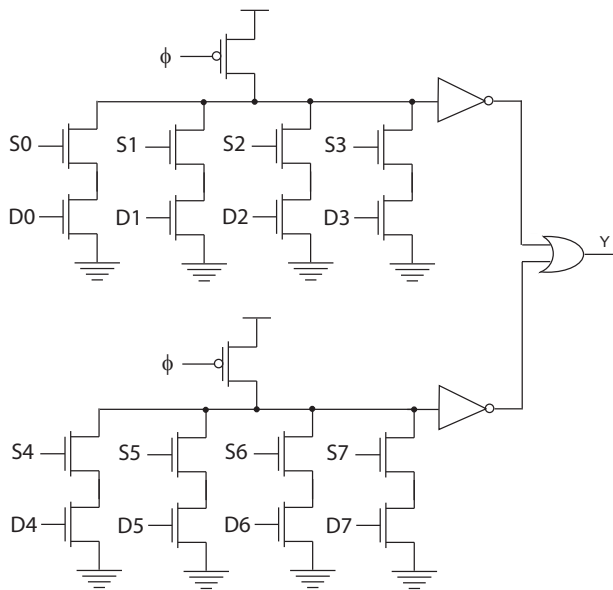


Figure 5.48: Domino logikk multiplekser. (FIG6.29)

5.12.1 Mål

Kunne implementere ulike logiske funksjoner ved hjelp av domino logikk.

5.13 Dual-rail domino logikk

(Kapittel 6.2.4.2 side 336 - 337)

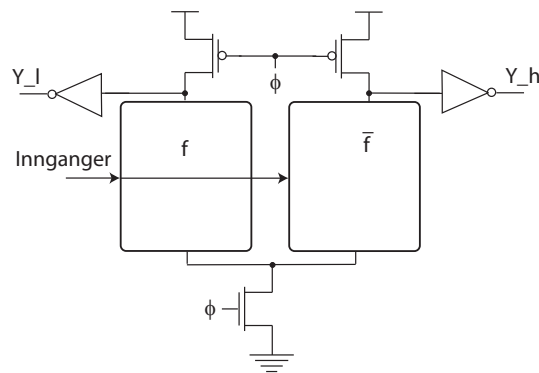


Figure 5.49: Dual-rail domino logikk. (FIG6.30a)

En domino logikk variant er vist i Fig. 5.49. Vi ser at porten ligner på en CVLS port, med komplementære nedtrekkskjeder og komplementære utganger. I stedet for krysskoblede pMOStransistorer som vi har i CVLS porter er de her erstattet med precharge pMOS transistorer. I tillegg er det en klokke nMOS transistor til GND som for footed dynamisk logikk. Vi kaller denne varianten for *dual-rail domino logikk*. Begge utgangene Y_l og Y_h blir precharget til 1. En av utgangene blir trukket ned til 0 i evalueringfasen, mens den andre utgangen forblir høy.

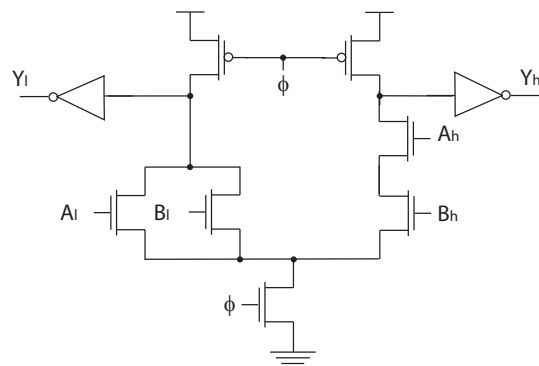


Figure 5.50: Dual-rail domino logikk. (FIG6.30b)

Et eksempel på dual-rail domino logikk er vist i Fig. 5.50. Utgangene vil være henholdsvis $Y_l = \overline{A_l + B_l} = \overline{A_l} \cdot \overline{B_l} = \overline{A_h} \cdot \overline{B_h} = \overline{A \cdot B}$ og $Y_h = \overline{Y_l} = A \cdot B$. Dual-rail domino logikk kan brukes der det er behov for inverterte signaler.

Ved å koble en 2inngangs NAND port direkte til inngangen til inverterne som vist i Fig. 5.51 får vi et signal som varsler om at evalueringen i den dynamiske porten er ferdig. Når porten precharges vil begge inngangene til NAND porten være 1 og utgangen *Ferdig* være 0. Når den dynamiske porten evaluerer vil en av inngangene til NAND porten bli trukket ned til 0 og utgangen på NAND porten vil dermed trekkes opp til 1.

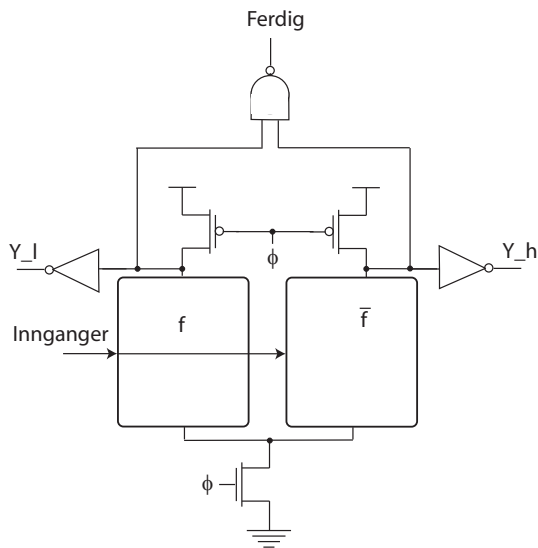


Figure 5.51: Dual-rail domino logikk med ferdig deteksjon. (FIG6.31)

5.13.1 Mål

Kunne implementere ulike funksjoner ved hjelp av dual-rail domino logikk.

5.14 Ladningsdeling

(side 340)

Et viktig problem knyttet til dynamisk logikk er *ladningsdeling*.

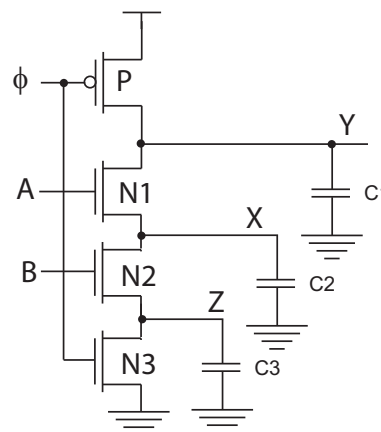


Figure 5.52: Ladningsdeling i en dynamisk port.)

Den dynamiske porten i Fig. 5.52 vil bli precharget til 1 når $\phi = 0$. Vi kan anta følgende situasjon: **Precharge:** $\phi = 0$ og $A = B = 0$ etterfulgt av **evaluate:** $\phi = 1$, $A = 0 \rightarrow 1$ og $B = 0$. Vi antar at noden $X = 0$ i slutten av prechargeperioden. I starten av evalueringsperioden vil inngangen A stige og skru på transistoren $N1$. En strøm i $N1$ vil ikke bare sørge for at spenningen på noden X stiger, men ikke minst at utgangen Y faller fordi pMOS transistoren vil være skrudd av under evaluering. Vi har i denne situasjonen en logisk høy utgang, men med en begrenset ladning knyttet til utgangskapasitansen C_1 . Noe av denne ladningen vil flyttes via transistoren $N1$ til noden X . Effekten av dette er at spenningen på X stiger og utgangsspenningen Y faller. En forenklet⁸ analyse gir

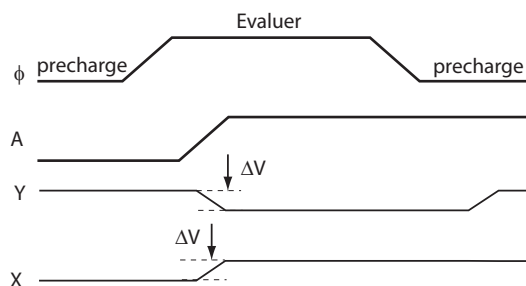


Figure 5.53: Endring i spenninger ved ladningsdeling i en dynamisk port.

$$\Delta V = \left(1 - \frac{C_1}{C_1 + C_2}\right) \cdot V_{DD}. \quad (5.44)$$

$$V_Y = V_{DD} - \Delta V$$

⁸Vi tar ikke hensyn til body effekt og terskelspenning.

$$V_X \approx V_Y. \quad (5.45)$$

Av denne analysen er det lett å se at den kritiske faktoren er endring i utgangssignalet Y . Dersom endringen er liten slik at etterfølgende porter ikke har problemer med å tolke Y som logisk 1 vil effekten av ladningsdelingen ikke være signifikant. Dersom utgangskapasitansen (lasten), i dette tilfellet C_1 , er stor i forhold til interne kapasitanser, i dette tilfellet C_2 , vil det ikke bli en signifikant endring av utgangen.

Transistoren $N1$ vil stenge før X trekkes helt opp til Y fordi det ikke vil være en effektiv gate source spenning $V_{gs} = V_{DD} - V_X$ som er tilstrekkelig høy for at transistoren er skrudd på. I dette tilfellet vil body effekt bidra til å redusere effekten av ladningsdeling. Dette betyr at $V_X < V_Y$ og at V_Y faller mindre enn antatt med forenklet analyse.

5.14.1 Mål

Kunne modellere ladningsdeling i dynamiske porter ved hjelp av forenklet analyse.

5.15 Precharge av interne noder i nedtrekket

(Kapittel 6.2.4.4 side 340 - 341)

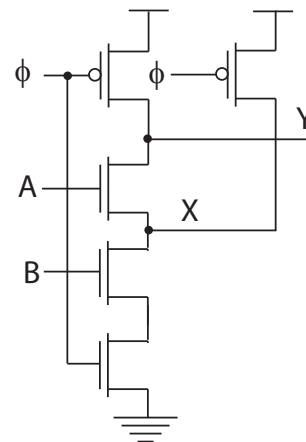


Figure 5.54: Precharge av interne noder i nedtrekket. (FIG6.40)

Vi kan redusere problemer med ladningsdeling i dynamiske porter ved å bruke precharge transistorer i tillegg til å precharge utgangen. Ved å precharge interne noder i nedtrekket som kan tenkes å utgjøre et problem ved ladningsdeling reduseres ladningsdelingen og portens robusthet vil øke. Ulempen i tillegg til større areal er økt parasittisk kapasitans i nedtrekk (noden X) som vil bidra til å øke parasittisk tidsforsinkelse ved nedtrekk. Ved precharge vil de interne nodene som precharges til 1 også bli tyngre å dra ned til 0 fordi det vil være en høy spenning (mye ladning) som må fjernes. Dette vil bidra til ytterligere å øke parasittisk tidsforsinkelse for nedtrekket. En annen ulempe er at lasten knyttet til klokkesignalet ϕ øker.

5.15.1 Mål

Kunne redusere problemer med ladningsdeling i logiske porter ved å bruke precharge transistorer for precharge av interne noder i nedtrekk.

5.16 Blødere (keepers)

(Kapittel 6.2.4.3 side 338 - 340)

En viktig begrensning for dynamiske porter er *ladningslekkasje*. Dersom frekvensen på klokkesignalet ϕ er relativt lav og/eller porten har betydelig lekkasje kan utgangsspenningen falle signifikant i evalueringsfasen selv om nedtrekket er skrudd AV. Dette problemet kommer i tillegg til laddingsdeling.

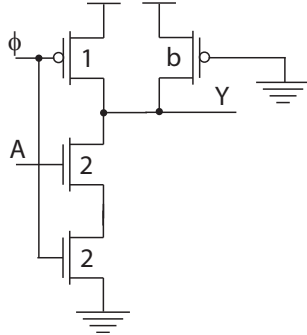


Figure 5.55: Statisk bløder.

For å redusere et uønsket fall i utgangsspenning på en dynamisk port er det vanlig å bruke en relativt svak transistor som er PÅ når utgangen er 1. En slik transistor kalles en *bløder* (keeper). Blødertransistoren må være tilstrekkelig stor dvs. bredde/lengde, slik at transistoren klarer å holde utgangen høy når nedtrekket er skrudd AV. En statisk blødertransistor er vist i Fig. 5.55. Blødertransistoren vil øke parasittisk tidsforsinkelse på grunn av en økning i den effektive nedtrekksmotstanden tilsvarende som for *pseudo nMOS* og på grunn av en økt kapasitans (last) på utgangen.

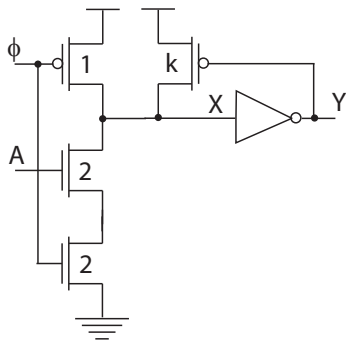


Figure 5.56: Dynamisk bløder. (FIG6.33)

En dynamisk blødertransistor er vist i Fig. 5.56. Denne transistoren vil ikke bare bidra til å holde en 1er på utgangen, men også øke den effektive motstanden i starten av et nedtrekk og dermed øke parasittisk tidsforsinkelse for nedtrekket både gjennom økningen i effektiv motstand og ved økt kapasitans på utgangen X. Når nedtrekket er PÅ vil X trekkes ned mot 0 og Y mot 1 og dermed skru av bløderen. I tillegg til en økt kapasitans for noden X, vil utgangen Y få vi en økt kapasitans på grunn av tilkobling til pMOS bløderen. Det er vanlig å dimensjonere bløderen

$W_p = \text{minimum}$ og $L_p = \text{stor}$ som vil gi en liten tilleggskapasitans for X, men en relativt stor tilleggskapasitans for Y.

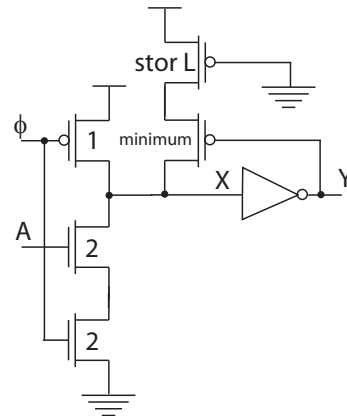


Figure 5.57: Dynamisk port med statisk og dynamisk bløder. (FIG6.34)

Økningen i tidsforsinkelse for Y kan begrenses ved å redusere lengden på blødertransistoren. Det er vanlig å begrense strømmen i bløderen ved å koble til en statisk bløder mellom V_{DD} og den dynamiske bløderen der lengden på den statiske bløderen kan være stor. Den statiske blødertransistoren vil fungere som en strømbegrenser. En dynamisk port med både statisk og dynamisk bløder er vist i Fig. 5.57.

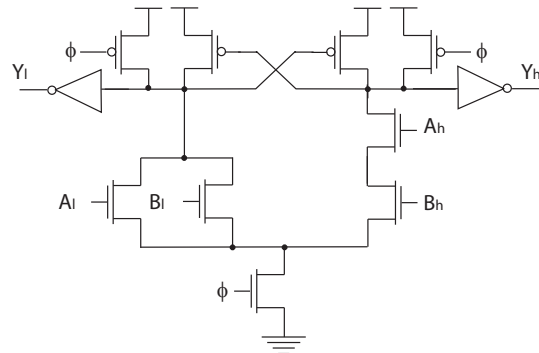


Figure 5.58: Differensiell dynamisk port med dynamisk bløder. (FIG6.35)

I Fig. 5.58 er det vist en differensiell dynamisk port med dynamisk bløder. Denne porten minner om CVSL logikk og kretsen er rask fordi pMOS transistorene ikke vil redusere hastigheten for nedtrekket.

Et alternativ til dynamisk bløder er vist i Fig. 5.59. Funksjonen er 2inngangs NAND og vil trekke aktivt høy når A eller B er 0. Dette aktive opptrekket medfører at kretsen blir støyrobust i tillegg til at det ikke vil være signifikant lekkasje når utgangen skal holdes høy eller det er fare for laddingsdeling. Som krets ligner denne porten mer på en statisk- enn dynamisk port. Ulempen med porten er økt kapasitans på utgangen Y som vil gi relativt stor tidsforsinkelse. Vi trenger imidlertid ikke forutsette noe om inngangssignalene under precharge og utgangen kan dermed kobles direkte til en tilsvarende krets.

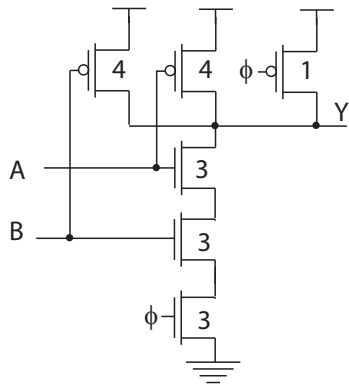


Figure 5.59: Støyrobust precharge. (FIG6.36)

5.16.1 Mål

Kunne benytte ulike blødertransistorer (keepers) ved implementasjon av robuste dynamiske porter.

5.17 Logisk effort i dynamiske kjeder

(Kapittel 6.2.4.5 side 341 - 343)

*INF3400 Del 6: Optimalt antall porter i en kjede[?], (Kapittel 4.3.2 side 178 - 181)

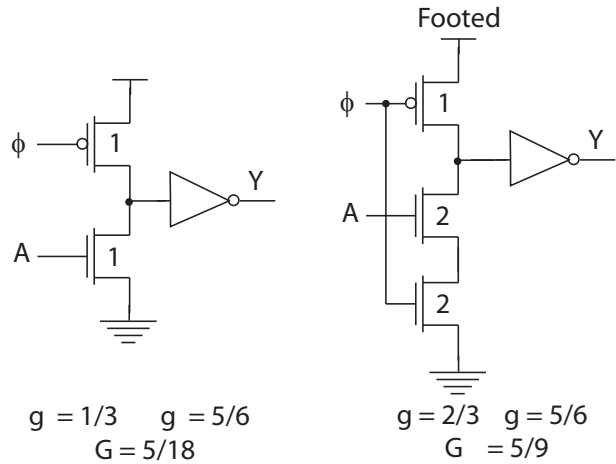


Figure 5.60: Logisk effort i dynamiske kjede. (FIG6.41)

Fotted- og ikke-fotted dynamiske invertorer er vist i Fig. 5.60. For en dynamisk inverter vil det være større parasittisk tidsforsinkelse når porten skal precharges enn tidsforsinkelse ved evaluering (eventuelt nedtrekk). Det er derfor vanlig å bruke high-skew statiske porter i tilknytning til dynamiske porter. En high-skew inverter, med $W_p = 2$ og $W_n = 1/2$, har logisk effort for opptrekk $g_u = 5/6$. Vi kan beregne logisk effort i kjeden bestående av en ikke-fotted dynamisk inverter og en statisk high-skew inverter som:

$$\begin{aligned}
 G_u &= \frac{1}{3} \cdot \frac{5}{6} \\
 &= \frac{5}{18} \\
 G_d &= \frac{2}{3} \cdot \frac{5}{3} \\
 &= \frac{10}{9}.
 \end{aligned} \tag{5.46}$$

For en kjede med en footed dynamiske invertorer og en statisk inverter får vi:

$$\begin{aligned}
 G_u &= \frac{2}{3} \cdot \frac{5}{6} \\
 &= \frac{5}{9} \\
 G_d &= \frac{4}{3} \cdot \frac{5}{3} \\
 &= \frac{20}{9}.
 \end{aligned} \tag{5.47}$$

For en kjede av statiske invertere som skal drive en last er den optimale port effort lik 4^9 [?]. Når vi benytter dynamiske porter i domino kjeder er det fornuftig å redusere port effort til mellom 2 og 3^{10} .

5.17.1 Eksempel

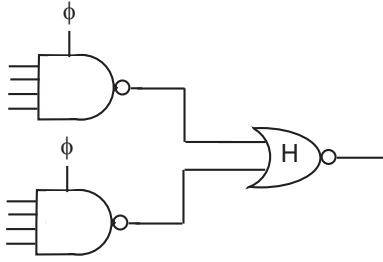


Figure 5.61: Logisk effort i dynamiske kjede. (FIG6.42b)

Gitt kretsen i Fig. 5.61 med en high-skew statistisk NOR port, med logisk effort $g_u = 3/2$. Anta at 4inngangs NAND portene er footed, med $W_n = 5$ som vil en effektiv nedtrekksmotstand lik R og logisk effort for nedtrekket $g_d = 5/3$. Vi kan beregne logisk effort og tidsforsinkelse i kjeden ved at de dynamiske NAND portene trekker begge inngangssignalene til NOR porten lave og dermed trekkes utgangen av NOR porten opp:

$$\begin{aligned}
 G_u &= \frac{5}{3} \cdot \frac{3}{2} \\
 &= \frac{5}{2} \\
 P_u &= (5C + C) \cdot R + \left(4C + 2 \cdot \frac{1}{2}C\right) R \\
 &= 11RC \\
 &= \frac{11}{3}\tau.
 \end{aligned} \tag{5.48}$$

Vi har $N = 2$ og $f' = F^{\frac{1}{2}} = (G_u \cdot 1 \cdot H)^{\frac{1}{2}} = \sqrt{\frac{5H}{2}}$, der H er elektrisk effort i kjeden. Tidsforsinkelse i kjeden er gitt av:

$$\begin{aligned}
 D &= NF^{\frac{1}{N}} + P_u \\
 &= 2 \left(\frac{5H}{2}\right)^{\frac{1}{2}} + \frac{11}{3}.
 \end{aligned} \tag{5.49}$$

Gitt kretsen i Fig. 5.62 med high-skew statiske invertere, med logisk effort $g_u = 5/6$. Anta at 4inngangs NAND portene er footed med $W_n = 5$. Vi kan beregne logisk effort og tidsforsinkelse i kjeden når utgangen er høy:

$$G_u = \frac{5}{3} \cdot \frac{5}{6} \cdot 1 \cdot \frac{5}{6}$$

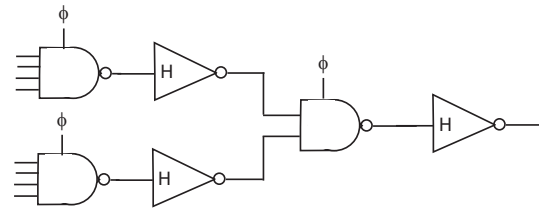


Figure 5.62: Logisk effort i dynamiske kjede. (FIG6.42a)

$$\begin{aligned}
 &= \frac{125}{108} \\
 P_u &= (5C + C) \cdot R + \left(2C + \frac{1}{2}C\right) R \\
 &+ (3C + C) \cdot R + \left(2C + \frac{1}{2}C\right) R \\
 &= 15RC \\
 &= 5\tau.
 \end{aligned} \tag{5.50}$$

Vi har $N = 4$ og $f' = F^{\frac{1}{4}} = (G_u \cdot 1 \cdot H)^{\frac{1}{4}} = (125/108H)^{1/4}$, der H er elektrisk effort i kjeden. Tidsforsinkelse i kjeden er gitt av:

$$D = 4 \left(\frac{125H}{108}\right)^{\frac{1}{4}} + 5. \tag{5.51}$$

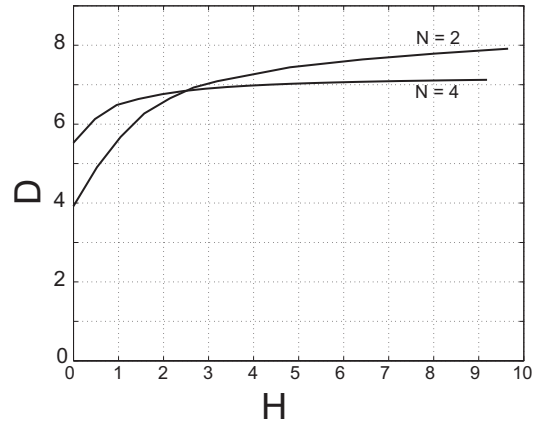


Figure 5.63: Tidsforsinkelse i dynamiske kjeder som funksjon av elektrisk effort i kjeden. (FIG6.43c)

I Fig. 5.63 er tidsforsinkelse i de to kjedene vist som funksjon av kjedens elektriske effort H . Når elektrisk effort er liten, dvs. mindre enn 3, lønner det seg å velge varianten med to porter ($N = 2$). Varianten med 4 porter ($N = 4$) har tidsforsinkelse som i svært liten grad er påvirket av kjedens elektriske effort.

5.17.2 Mål

Kunne beregne logisk effort og tidsforsinkelse i kjeder med dynamiske porter.

⁹Eller mer presist lik 3.59.

¹⁰2 dersom det bare er dynamiske porter og 3 dersom det er et stort innslag av statiske porter i kjeden.

5.18 NP og Zipper domino

(Kapittel 6.2.4.7 side 343 - 345)

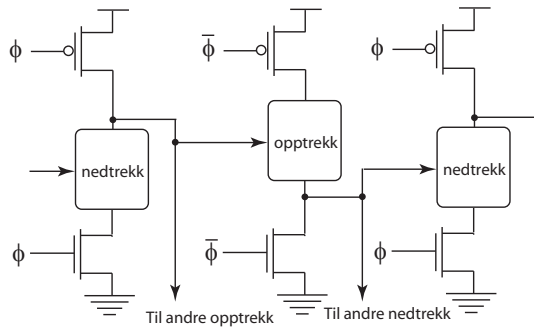


Figure 5.64: NP domino. (FIG6.46a)

En annen variant av dynamisk porter er vist i Fig. 5.64. Her har man alternerende precharge med ϕ og $\bar{\phi}$ for nedtrekk og opptrekk. Denne varianten kalles *NP domino* eller *NORA domino*. Her vil precharged verdi for den første porten være 1 som kan brukes direkte inn på en pMOS transistor (AV). Neste port precharges i samme periode, men da precharges utgangen til 0 som kan kobles direkte inn på en nMOS transistor.

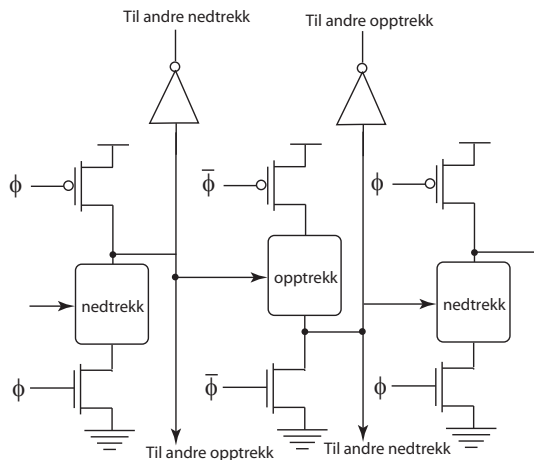


Figure 5.65: NP domino. (FIG6.46c)

NP domino logikk kan utvides ved å benytte statiske invertere slik at utgangene på en precharged port kan brukes i både opptrekk og nedtrekk som vist i Fig. 5.65.

Zipper domino er tilsvarende NP domino logikk der precharge klokkesignaler har redusert sving, dvs. ϕ svinger mellom 0 og $V_{DD} - |V_{tp}|$ og $\bar{\phi}$ svinger mellom V_{tn} og V_{DD} . Precharge transistorene vil da fungere som blødere.

5.18.1 Mål

Kunne anvende NP domino logikk.

5.19 CMOS med transmisjonsporter

(Kapittel 6.2.5.1 side 347 - 348)

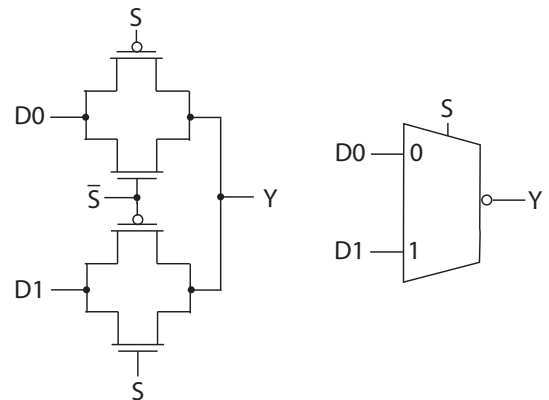


Figure 5.66: Transmisjons port multiplekser. (FIG1.27)

Kretser slik som tristate, latcher og multipleksere er ofte tegnet med transmisjonsporter i tillegg til statisk CMOS logikk. Et eksempel er transmisjonsgate multiplekser som vist i Fig. 5.66.

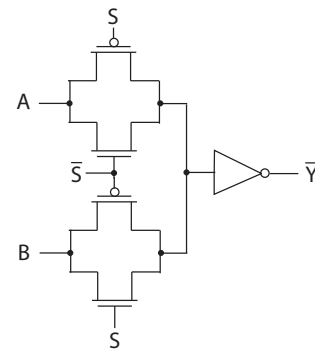


Figure 5.67: CMOS transmisjonsport med utgangsbuffer. (FIG6.47)

En transmisjonsport kan utvides med et utgangsbuffer som vist i Fig. 5.67. Vi kaller en slik krets for *CMOSTG*. Utgangsbufferet er en inverter som vil sørge for at utgangen holder gode logiske verdier. I utgangspunktet kan man få inntrykk av at transmisjonsporter er fundamentalt forskjellige fra vanlig statisk CMOS. I realiteten er det ikke stor forskjell.

To ulike realiseringer av CMOSTG multipleksere er vist i Fig. 5.68. I kretsen til venstre er det brukt invertere og transmisjonsporter, mens kretsen til høyre er en statisk CMOS port. Dersom vi ser på de fire transistorene til venstre på kretsen til høyre i figuren ser vi at dette er en tristate¹¹ inverter med A som inngang. De fire transistorene til høyre utgjør en tristate inverter med B som inngang. Vi ser at dersom vi kobler sammen nodene mellom pMOS transistorene og nMOS transistorene i de to tristate inverterne, N1 og N2, vil vi få en helt identisk krets som til

¹¹Dette tilsvarer en kloknet CMOS port C^2MOS .

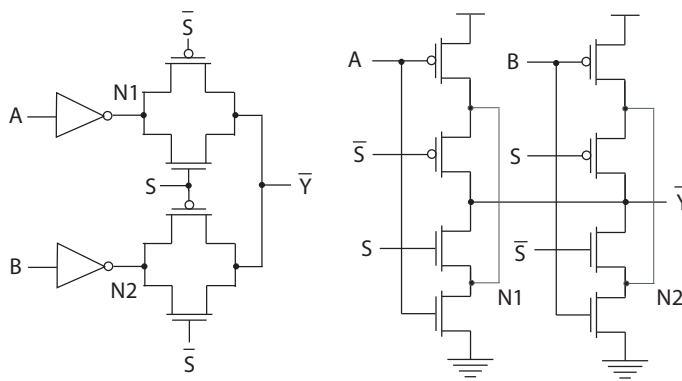


Figure 5.68: Alternative implementasjoner av CMOS 2-to-1 multiplekserer. (FIG6.48)

venstre i figuren. Kretsene blir da identiske, både logisk og elektrisk. Denne sammenkoblinger er unødvendig og vil bare bidra med økt motstand og kapasitans og derfor bidra med økt parasittisk tidsforsinkelse.

5.19.1 Mål

Kunne implemetere CMOS med transmisjonsporter.

5.20 Komplementær pass transistor logikk (CPL)

(Kapittel 6.2.5.2 side 348 - 349)

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[?])

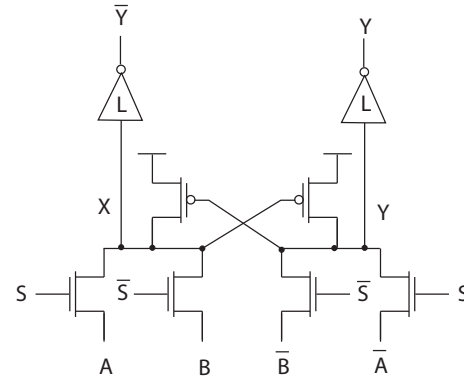


Figure 5.69: Komplementær pass transistor logikk (CPL). (FIG6.51a)

Komplementær pass transistor logikk (CPL) er vist i Fig. 5.69. Vi ser at kretsen minner om CMOS. Inverterne på utgangen er lav skew der nedtrekket er prioritert.

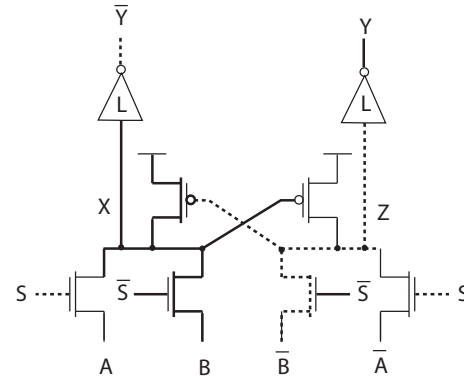


Figure 5.70: Komplementær pass transistor logikk (CPL), $S = 0$ og $B = 1$. (FIG6.51a)

CPL multiplekseren med $S = 0$ og $B = 1$ er vist i Fig. 5.70. Her er det to signalveier fra B og \overline{B} til henholdsvis \overline{Y} og Y . Vi ser først på signalveien fra \overline{B} til Y via Z . Signalet går via en nMOS transistor til Z som vil bli trukket helt ned til 0. X derimot kan ikke trekkes helt opp til 1 fordi nMOS transistoren som forbinder B med X vil ha et terskelfall. pMOS transistoren som er koblet til X og som har Z som gate signal vil bidra til å trekke X helt opp til 1.

CPL multiplekseren med $S = 1$ og $A = 1$ er vist i Fig. 5.71. pMOS transistorene koblet til nodene X og Z bidrar til å trekke enten X eller Z til logisk 1, dvs. X i dette tilfellet. CPL kan implementeres uten de to pMOS transistorene. En av nodene X eller Z vil da bli trukket helt ned til 0, mens den andre vil bli trukket opp til $V_{DD} - V_t$. Denne reduserte logiske 1 vil bli invertert til en skikkelig

5.21 Lean Integration med pass transistorer (LEAP)

(Kapittel 6.2.5.3 side 349 - 350)

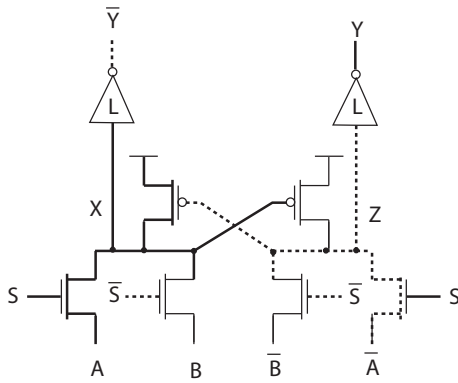


Figure 5.71: Komplementær pass transistor logikk (CPL), $S = 1$ og $B = 1$. (FIG6.51a)

logisk 0 etter utgangsinverteren. Utgangsinverteren som skal drive \bar{Y} til 0 vil da ikke bli skrudd helt på. Dette bidrar ikke bare til forsinkelse, men også til økt effektforbruk fordi inngangstransisjonen for utgangsinverteren blir treg.

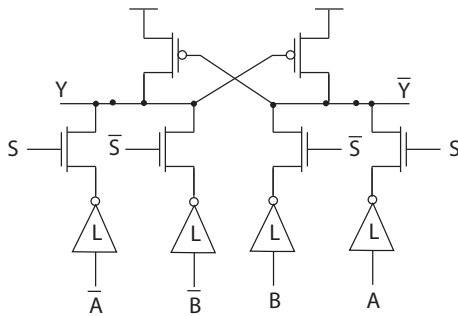


Figure 5.72: Komplementær pass transistor logikk (CPL.) (FIG6.51b)

I Fig. 5.72 er utgangsinvertere til multipleksere på inngangen vist. I dette tilfellet er utgangsinverterne dropet.

5.20.1 Mål

Kunne implementere multipleksere ved hjelp av komplementær pass transistor logikk (CPL).

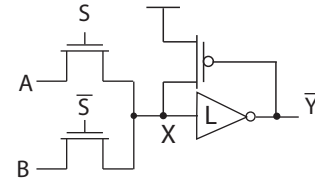


Figure 5.73: Lean Integration med pass transistorer (LEAP) (FIG6.47)

En lean integration krets med pass transistorer (LEAP) er vist i Fig. 5.73. Transmisjonsporene er erstattet med passtransistorer. Node X blir presset opp til 1 når utgangen \bar{Y} er 0. På denne måten kan vi erstatte to pMOS transistorer ved inngangene med en transistor som er styrt fra utgangen.

5.21.1 Mål

Forstå hvordan lean integration krets med pass transistorer (LEAP) implementeres.

5.22 Andre pass transistor familier

(Kapittel 6.2.5.4 side 350)

5.22.1 DPL

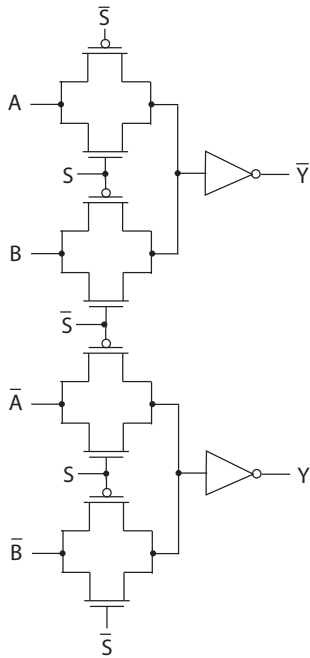


Figure 5.74: *Dobbel pass transistor logikk (DPL) (FIG6.47)*

Dobbel pass transistor logic (DPL) er vist i Fig. 5.74. DPL ligner på CMOS, men består av dobbelt sett med transmisjonsporter når vi trenger inverterte utganger i tillegg til ordinære utganger. Kretsen trenger ikke tilbakekobling for å sikre gode logiske verdier.

5.22.2 EEPL

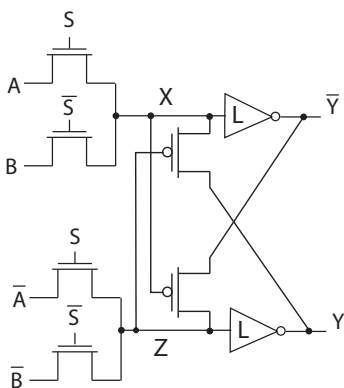


Figure 5.75: *Energi effektiv pass transistor logikk (EEPL) (FIG6.47)*

Energi effektiv pass transistor logikk (EEPL) er vist i Fig. 5.75. pMOS transistorene som skal dra nodene X eller Z er ikke koblet til V_{DD} , men istedet koblet til ut-

gangen. Dette betyr økt tidsforsinkelse, men redusert effektforbruk.

5.22.3 PPL

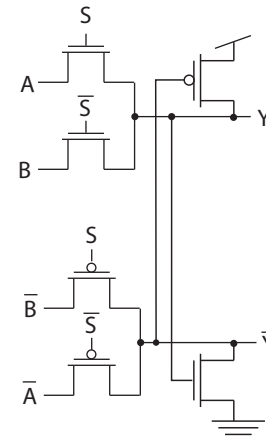


Figure 5.76: *Push-pull pass transistor logikk (PPL) (FIG6.47)*

Push-pull pass transistor logikk (PPL) er vist i Fig. 5.76. I dette tilfellet er det to nMOS pass transistorer og to pMOS pass transistorer, der de to komplementære utgangene brukes til å trekke den andre utgangen til riktig verdi. Vi får et terskelfall over de to nMOS passtransistorene slik at Y bare kan trekkes opp til $V_{DD} - V_t$. Vi ser at når utgangen Y skal være logisk 1 så vil den andre utgangen \bar{Y} være logisk 0. I dette tilfellet vil \bar{Y} nær null skru på en pMOS transistor koblet med drain til Y slik at denne utgangen trekkes helt opp til 1. Tilsvarende vil en utgang Y nær 1 skru på nMOS transistoren med drain koblet til utgangen \bar{Y} trekke \bar{Y} helt ned til 0.

5.22.4 SRPL

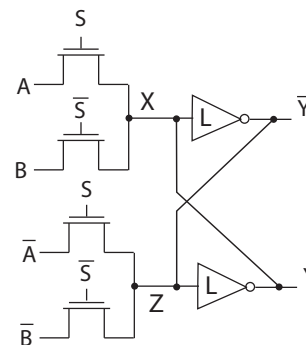


Figure 5.77: *Sving-restored pass transistor logikk (SRPL) (FIG6.47)*

Sving-restored pass transistor logikk (SRPL) er vist i Fig. 5.77. Logikken ligner på EEPL, men er enklere. Her er pMOS transistorene som bidrar til å trekke nodene X og Z opp til 1 droppet. I stedet bidrar de to krysskoblede

inverterne på utgangen til å trekke begge utgangene til skikkelige logiske verdier.

5.22.5 DCVSPG

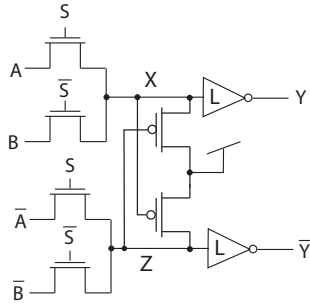


Figure 5.78: *Differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG). (FIG6.47)*

Differensiell kaskode spenning svitsj med pass transistor logikk (DCVSPG) er vist i Fig. 5.78. Logikken er forholdsvis lik kaskode voltage svitsj logikk (CVSL). Inngangene er koblet til drain/source på transistorer slik at det blir pass transistor logikk i stedet for komplementære nedtrekk som CVSL har.

5.22.6 Mål

Få et overblikk over ulike varianter av logikk med transisjonsporter og pass transistorer.

5.23 Differensielle kretser

(Kapittel 6.4.1 side 359 - 360)

*Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[?])

Den vanligste CMOS logikken er statisk logikk. Dynamisk logikk som anvendes relativt ofte er domino-, pass-transistor- og pseudo nMOS logikk. I tillegg er det ikke uvanlig å anvende ulike typer *differensiell logikk* særlig i anvendelser der det er meget viktig å begrense digital støy. Typiske systemer er *mixed-mode* eller *mixed-signal* som er integrerte systemer der analoge og digitale delkretser deler samme substrat. Mange differensielle kretser er utviklet fra kaskode spenning svitsj logikk (CVSL).

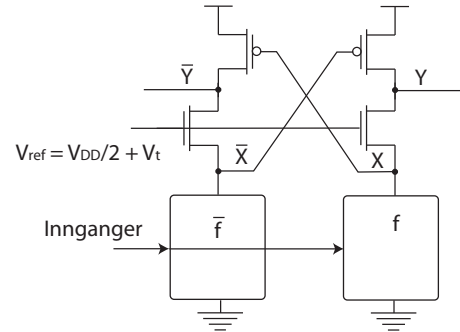


Figure 5.79: *Differensiell split-level (DSL). (FIG6.61)*

Differensiell split-level (DSL) er vist i Fig. 5.79. Nedtrekket består av komplementære nedtrekk lik CVSL, men i tillegg er det koblet nMOS transistorer i serie opp til utgangene. Disse nMOS transistorene har gate terminalen koblet til en spenningsreferanse $V_{ref} = V_{DD}/2 - V_t$, slik at utgangen av de komplementære nedtrekkene (X og \bar{X}) får redusert sving til $0 \rightarrow V_{DD}/2$. Dette reduserer parasittisk tidsforsinkelse ved nedtrekk som blir utnyttet til å redusere lengden på transistorene. I moderne design er imidlertid lengden ofte minimum slik at dette ikke er utnyttbart i praksis.

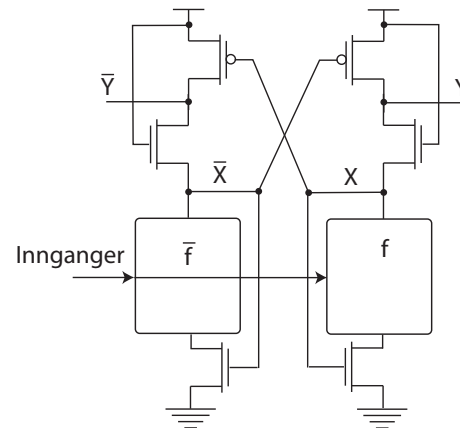


Figure 5.80: *Kaskode nonthreshold logikk (CNTL). (FIG6.61)*

Kaskode nonthreshold logikk (CNTL) er vist i Fig. 5.80. Logikken bygger på DSL og *nonthreshold logikk (NTL)* som

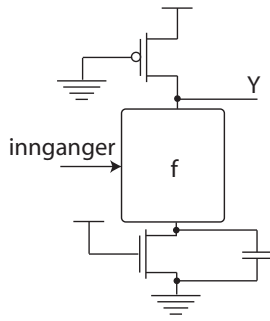


Figure 5.81: *Nonthreshold logikk (NTL). (FIG6.62)*

er vist i Fig. 5.81. NTL logikk tilsvarer en pseudo nMOS logikk med tillegg en nMOS transistor og en kapasitans i parallell til GND . NTL er tregere enn pseudo nMOS logikk og bruker mer statisk effekt, og er derfor ikke vanlig å bruke. nMOS transistorene mot GND i CNTL logikk vil bidra til å redusere strømtrekket fordi transistorene vil stenge nedtrekket når utgangen på en av de komplementære blokkene trekkes ned mot GND .

Både DSL og CNTL er forholdsvis uvanlig å bruke i moderne CMOS.

5.23.1 Mål

Forstå hvordan DSL og CNTL logikk virker.

5.24 Oversikt over kretsproblemer

(Kapittel 6.3 side 350-351)

Designere har en tendens til å bruke enkle kretser fordi de er robuste. Statisk CMOS er mest robust og bør brukes når det er mulig. Det er en rekke forhold som kan få kretser til å feile.

- Terskelfall.
- Nivåfeil.
- Lekasje.
- Laddningsdeling.
- Støy på spenningsforsyningen.
- Cross talk.
- Injeksjon av miniritetsbærere.
- Back-gate variasjon.
- Følsomhet for støy i diffusjonsområder.
- Race tilstander.
- Matching av tidsforsinkelser.
- Metastabilitet.
- Hot spots.
- soft errors.
- Prosess følsomhet.

5.25 Terskelfall

(Kapittel 6.3.1 side 351)

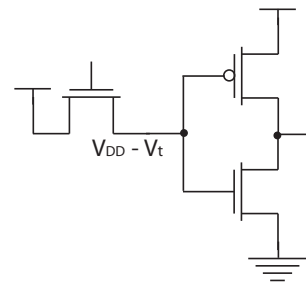


Figure 5.82: *Passtransistor med terskelfall.*

Passtransistorer er gode til å trekke noder i en bestemt retning, nMOS ned og pMOS opp. I motsatt retning vil passtransistoren bli skrudd av når source spenningen nærmer seg gatespenningen. En nMOS passtransistor som skal drive en port er vist i Fig.5.82. Utgangen av passtransistoren kan bare trekkes opp til $V_{DD} - V_t$. Vi får derfor et

terskelfall over en passtransistor. Dette er egentlig for optimistisk fordi body effekten vil begrense spenningssvinget på utgangen på passtransistoren ytterligere. Spenningen blir for lav til at pMOS transistoren i porten, dvs. inverteren, blir skrudd helt av. Dette resulterer i statisk effektforbruk i porten. Dersom V_{DD} er svært lav kan til og med nivåfeil på utgangen forekomme. I eldre prosesser med høy forsyningsspenning kunne et terskelfall tolereres.

5.26 Nivåfeil

(Kapittel 6.3.2 side 352)

* Passtransistor- og differensiell CMOS logikk. (INF3400 Del 12: Kapittel IV side 3) * Passtransistor- og differensiell CMOS logikk. (INF3400 Del 12: Kapittel V side 4)

Pseudo nMOS logikk illustrerer nivåbegrensinger som oppstår når en node trekkes opp og ned samtidig. Svake transistorer må være så svake at utgangsnivået faller under V_{IL} for neste port på grunn av støymarginer. Nivålogikk bør simuleres i SF og FS hjørner.

Et annet eksempel på nivåfeil er knyttet til kretser med tilbakekobling, for eksempel dynamiske keepers, SRPL[2] og LEAP[3] med nivå restaurering.

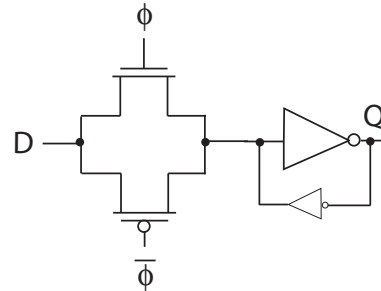


Figure 5.83: Statisk latch med svak tilbakekoblingsinverter.

For kretser med input i diffusjon (D) er det spesielt viktig å være oppmerksom på mulige nivåfeil. I FIG. 5.83 er det vist en statisk latch med en svak tilbakekoblingsinverter. Det er viktig at tilbakekoblingsinverteren er så svak at den ikke overstyrer signalet som kommer fra inngangen D via transmisjonsporten.

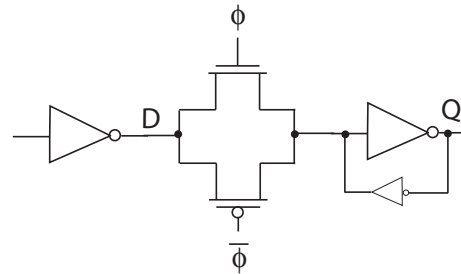


Figure 5.84: Statisk latch med styrket inngangsdriever.

I FIG. 5.84 er det vist en statisk latch der inngangen D er styrket for å motstå tilbakekoblingsinverteren.

I FIG. 5.84 er det vist en statisk latch der inngangen D er styrket for å motstå tilbakekoblingsinverteren og modell for motstand og kapasitans i lederen er inkludert.

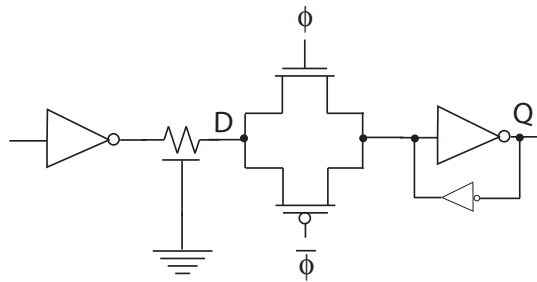


Figure 5.85: Statisk latch med modellering av leder.

5.27 Lekkasje

(Kapittel 6.3.3 side 352-353)

* Svak inversjon. (INF3400 Del 8: Kapittel II side 1)

Lekkasjeproblemer blir et stadig et større og større problem. Lekkasjestrømmer er knyttet til strøm gjennom transistorer som er skrudd av (av-strøm), gate tunnelering og reversstrøm gjennom dioder. Av-strømmene blir stadig større fordi terskelspenningene reduseres og fordi antall transistorer øker. Gate lekkasje vil bli mer signifikant når tykkelsen på tynnoksid reduseres. I tillegg til at lekkasje bidrar til statisk effektforbruk kan nivåfeil oppstå. Dette gjelder spesielt for elektriske noder som er svakt drevet. Tiden som trengs for at en dynamisk node skal endres spenningsmessig med ΔV som følge av lekkasjestrøm $I_{lekkasje}$ kan uttrykkes som

$$t = \frac{C_{node} \Delta V}{I_{lekkasje}}. \quad (5.52)$$

Av-strømmer vil gradvis bidra til at nodespenninger endres gjennom transistorer som er av. Fullstendig dynamiske kretser er ofte ikke mulige å bruke i de fleste moderne prosesser. Problemet med lekkasjestrømmer og logisk nivå blir større når det er flere transistorer i serie, som for eksempel i NOR porter. Av-strømmer blir mye mindre når det er to transistorer i serie som er skrudd av fordi den transistoren som er koblet mot spenningsreferansen vil ha en mye mindre drain-source spenning og dermed mindre DIBL[4] effekt. Det blir etterhvert vanlig med teknologier der man har et utvalg av transistorer som har forskjellig terskelspenning.

5.28 Støy som skyldes ladningsdeling

(Kapittel 6.3.4 side 353)

* Ladningsdeling. (INF3400 Del 9: Kapittel V side 7)

Fig. 5.86 viser en dynamisk port som skal drive en transmisjonsport. Anta at den dynamiske porten har blitt precharget og utgangen X er høy, men udrevet eller flytende. Dersom transmisjonsporten er av kan vi anta at $Y = 0$. Dersom transmisjonsporten blir skrudd på kan vi anta at både X og Y er udrevet slik at spenningsnivået på

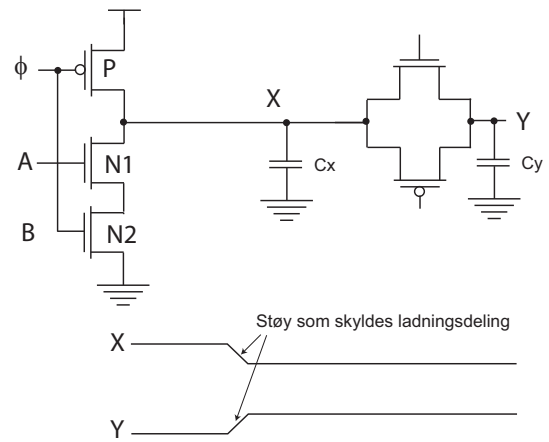


Figure 5.86: .

dise nodene vil nærme hverandre på grunn av ladningsdeling.

5.29 Støy i forsyningsspenningen

(Kapittel 6.3.4 side 353)

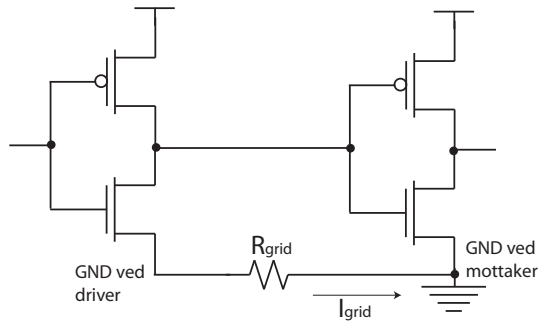


Figure 5.87: .

V_{DD} og GND er ikke konstant over en hel integrert krets. Begge forsyningslinjene vil bli påvirket av støy som følge av fall i IR og di/dt støy. IR fall skjer fordi det er motstand i forsyningsledere mellom eksterne innganger (selve forsyningen) og ulike blokker som trekker strøm i den integrerte kretsen. Dette er vist i Fig. 5.87. di/dt støy oppstår i forsyningsledere på grunn av induktans og strømmer i rask endring.

Støy i forsyningsledere vil påvirke ytelse i en krets negativt. Typisk vil støy på forsyningsledere resultere i 5% – 10% avik for V_{DD} .

5.30 Hot spots

(Kapittel 6.3.5 side 354-355)

*Temperaturavhengighet for MOS transistorer. (INF3400 Del 14: Kapittel XI side 11)

Transistorenes ytelse, dvs. som strømkilde, avtar med økende temperatur[6]. Ujevn temperatur og såkalte *hot spots* i en integrert krets kan forekomme fordi det vil være ujevnt strømmtrekk eller effektforbruk i ulike deler av en krets. Dette fører til at tidsforsinkelse vil variere avhengig av lokale temperaturer. Det vil være fordelaktig å designe kretser der effektforbruket blir jevnest mulig over hele kretsarealet, eller i det minste at man simulerer effekten av lokal temperaturvariasjoner på kretsens ytelse.

5.31 Injeksjon av minoritetsbærere

(Kapittel 6.3.7 side 355-356)

*Temperaturavhengighet for MOS transistorer. (INF3400 Del 14: Kapittel XI side 11)

*Latchup. (INF3400 Del 14: Kapittel IXX side 16 - 16)

Det vil kunne forekomme spenninger som ikke ligger innenfor begrensningene gitt av forsyningsspenningene, dvs. intervallet GND til V_{DD} . Slike spenninger kan genereres via kapasitive koblinger og induksjon i I/O drivere. Disse spenningsverdiene kan i noen tilfeller, som regel tidsbegrenset, resultere i at pn overganger (dioder) blir foroverforspent typisk ved uheldig spenningsforskjeller mellom drain på MOS transistorer og substrat. Resultatet av dette blir at det vil gå en lekkasjestrøm til substrat. Denne effekten kalles *injeksjon av minoritetsbærere*. Resultatet kan også bli latchup[7].

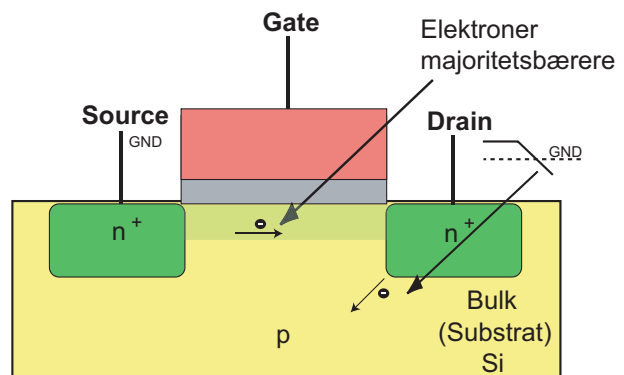


Figure 5.88: Injeksjon av minoritetsbærere..

I Fig. 5.88 er injeksjon av minoritetsbærere i en nMOS transistor vist. Elektroner er majoritetsbærere fra source til drain og minoritetsbærere fra drain til substrat (eller source).

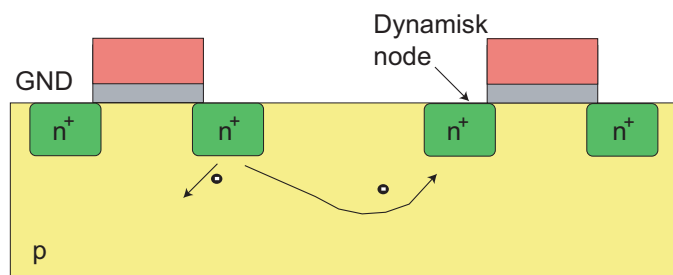


Figure 5.89: Injeksjon av minoritetsbærere og problemer med dynamiske noder.

I Fig. 5.89 er det vist et eksempel på injeksjon av minoritetsbærere til substrat og til en dynamisk elektrisk node i form av et drain/source område på en nærliggende nMOS transistorer. Dette er et problem som er spesielt viktig for dynamiske porter og følsomme analoge kretser.

I Fig. 5.90 er det vist hvordan dynamiske noder kan beskyttes mot injeksjon av minoritetsbærere. Et $p+$

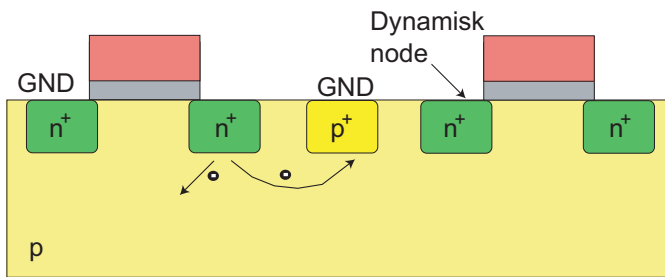


Figure 5.90: Beskyttelse av dynamiske noder.

områder koblet til *GND* vil effektivt beskytte en dynamisk node mot minoritetsbærere.

5.32 Back-gate

(Kapittel 6.3.8 side 356)

Dynamiske porter som skal drive statiske porter med flere innganger er følsomme for en effekt som kalles *back-gate*¹².

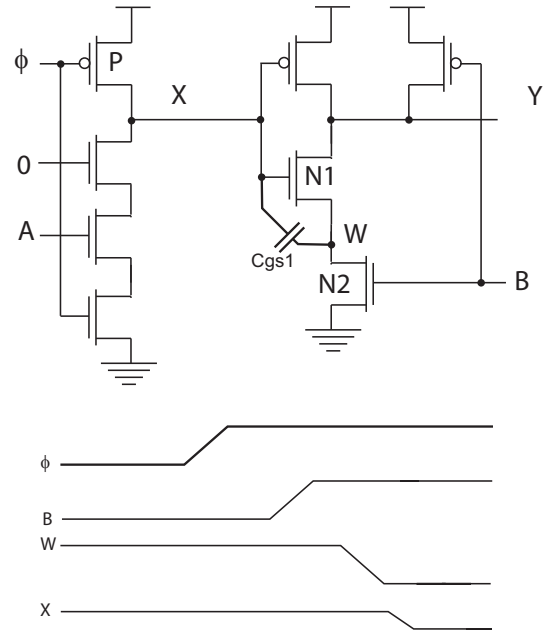


Figure 5.91: Back-gate kobling.

Et eksempel på back-gate kobling er vist i Fig. 5.91 der en dynamisk NAND port skal drive en statisk NAND port. Gate til source kapasitans C_{gs1} for transistor $N1$ er vist eksplisitt i figuren. Dersom den dynamiske NAND porten precarges til 1 og en av eller begge inngangene er 0 skal utgangen X på porten forbli 1. Dersom utgangen på den statiske porten Y er 1 og inngang B er 0 vil den elektriske noden W være nesten logisk 1, begrenset av bodyeffekt i transistor $N1$. Dersom B endres fra 0 til 1 vil W trekkes ned til 0 slik X endres, dvs. trekkes noe ned, på grunn av en kapasitiv kobling mellom X og W .

¹²Back-gate brukes også om substrattilkobling på MOS transistorer.

5.33 Følsomhet for støy i diffusjon

(Kapittel 6.3.9 side 357)

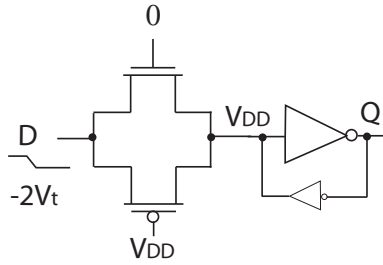


Figure 5.92: Støy i diffusjon på inngangen til en latch.

Fig. 5.92 viser en latch med et diffusjonsområde på inngangen som er eksponert for støy. Når verdien i latchen skal være stabilt er transmisjonsporten skrudd av. Q vil da ligge stabilt på 0. Dersom inngangen D faller under 0, dvs. ca. 2 ganger terskelspenningen, på grunn av lokal støy vil nMOS transistoren i transmisjonsporten skrus på og den lagrede verdien $V_{DD} = \bar{Q}$ påvirkes og trekkes ned mot $-2V_t$. Dette kan videre føre til at Q endres fra 0 til V_{DD} . Dermed er latchens verdi eller tilstand endret dramatisk med en bitfeil.

5.34 Prosessfølsomhet

(Kapittel 6.3.10 side 357)

Kretser som er lite robuste kan fungere under normale prosessforhold, dvs. med typiske parameterverdier, men feile i ulike prosesshjørner. Alle kretser bør simuleres i alle hjørner for å redusere sannsynligheten for at kretsene feiler. Vi kaller slike design for lite følsomme for prosessvariasjoner. Det kan også være fornuftig å ta hensyn til prosessmigrasjon ved design. Dette dreier seg også om robusthet i forhold til prosessvariasjoner. Det er en rekke faktorer som er predikterbare ved prosessmigrasjon, for eksempel økt lekkasje, lavere terskelspenning, lavere forsyningsspenning, mindre forsterkning osv.

5.35 Domino støyregnskap

(Kapittel 6.3.11 side 357-358)

* Domino logikk. (INF3400 Del 9: Kapittel III side 5)

Domino logikk er en viktig logikkstil i moderne mikroprosessorer med krav til høy ytelse. Denne logikkstilen er følsom for støy både med hensyn på variasjon i tidsforsinkelse og logiske nivåer. Kretser med dynamiske utganger er spesielt følsomme for støy når utgangen er høy og ikke drives. Ved bruk av domino logikk bør følgende forhold tenkes nøye gjennom:

1. **Lekkasje av ladninger.** For tiden er lekkasje gjennom transistorere som er skrudd av og som er koblet

til udrevne noder mest signifikant. Denne form for lekkasjer er vanligst i NOR kretser under høy temperatur. For å holde riktig nivå bør det benyttes keepers som vil bidra med en effektiv tilbakekobling som motvirker effekten av lekkasjer.

2. **Ladningsdeling.** Ladning kan flyttes fra utgangen på en dynamisk port til interne noder i porten. Fordi det ikke er tilførsel av ladning når porten er udrevet vil dette kunne medføre at utgangen ikke vil ha riktig verdi eller logisk nivå. Man bør unngå å koble dynamiske porter til transmisjonsporter (source/drain) fordi dette lett kan føre til at inngangen til porter som drives gjennom transmisjonsporten får feil logisk nivå som følge av ladningsdeling.

3. **Kapasitiv kobling.** Kapasitive koblinger kan forekomme på både innganger og utganger på porter. Innganger til dynamiske porter har liten støymargin og er derfor spesielt utsatt for støy som følge av kapasitive koblinger. Man kan redusere kapasitive koblinger ved å bruke korte ledere og ha god avstand mellom ledere.

4. **Back-gate.** Støy, feks gjennom lekkasjestrømmer, i substrat kan påvirke utgangen på dynamiske porter som skal drive statiske porter.

5. **Injeksjon av minoritsbærere.** Dynamiske noder bør beskyttes mot noder som kan bidra med injeksjon av mioritetsbærere. I praksis vil dette si at dynamiske porter ikke må legges nær I/O padder. Det er også fornuftig å bruke gusrd rings.

6. **Støy i spenningsforsyninger.** Statiske porter som skal drive dynamiske porter bør plasseres så nær de dynamiske portene som mulig.

5.36 Sense-amplifier kretser

(Kapittel 6.4.2 side 360 - 362)

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[?])

* Dual-rail domino logikk. (INF3400 Del 9: Dynamisk CMOS[?])

Sense-amplifier kretser er kretser som forsterker små differensielle signaler til store spenningsving på utgangen. Sense-amplifier kretser brukes ofte i hukommelser der bitlinjer ofte har stor kapasitiv last¹³ og vil derfor ha stor forsinkelse og trege transisjoner. Kretser som reagerer på små spenningsendringer (transisjoner) vil da kunne reagere raskt når en bitlinje er i ferd med å få endret sin verdi. Sense-amplifier kretser er utsatt for ladningsdeling som kan påvirke kretsens utgang fordi kretsen reagerer på små forandringer. Dette påvirker robusthet negativt.

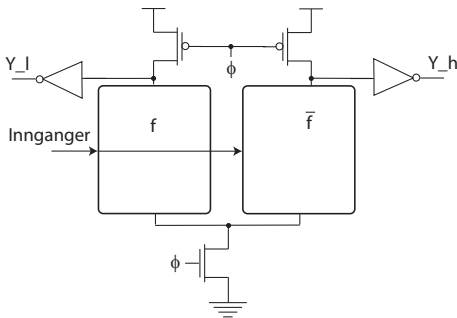


Figure 5.93: Dual-rail domino logikk. (FIG6.63)

Sense-amplifier kretser er basert på CVSL kretser eller mer presist *dual-rail domino logikk* som er vist i Fig. 5.93. Vi kan betrakte sense-amplifier kretser som dual-rail domino logikk med en sense-amplifier for deteksjon og forsterkning av små differensielle spenninger.

En generisk sense-amplifier krets er vist i Fig. 5.94 øverst. Den er mest nyttig for komplekse nedtrekk som representerer betydelig parasittisk forsinkelse. Inngangene er differensielle som eksempelet nederst i Fig. 5.94 viser. Det vil alltid være slik at bare et av nedtrekkene er PÅ.

5.36.1 Mål

Forstå generisk sense-amplifier krets.

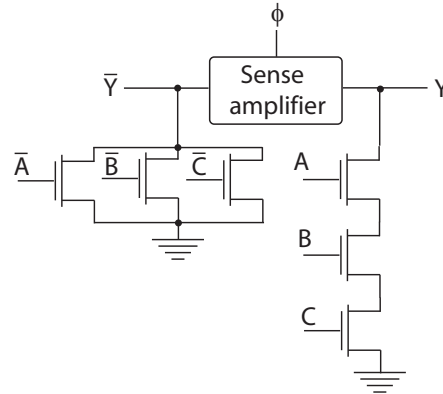
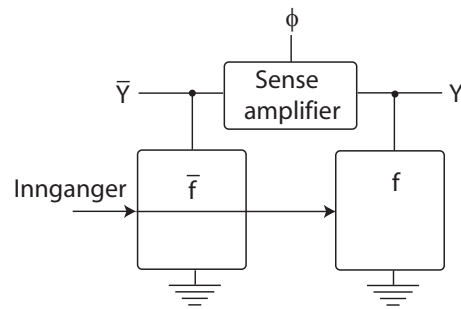


Figure 5.94: Generisk sense-amplifier krets. (FIG6.64)

5.37 Sample set differensiell logikk (SSDL)

(Kapittel 6.4.2.1 side 362 - 363)

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[?])

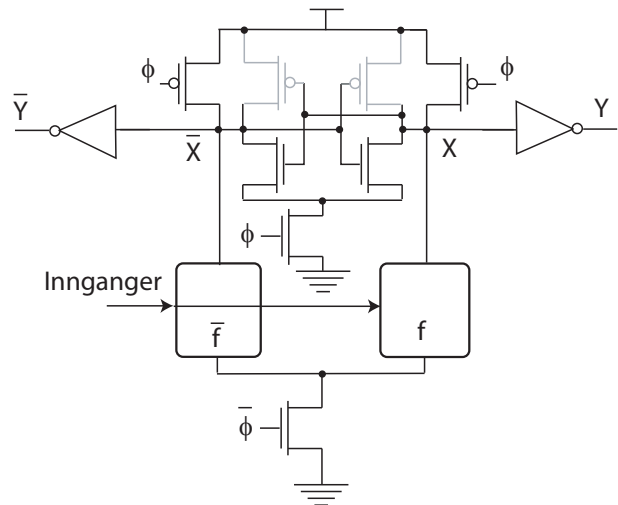


Figure 5.95: Sample set differensiell logikk (SSDL). (FIG6.63)

Sample set differensiell logikk (SSDL) er vist i Fig. 5.95. Vi ser at SSDL skiller seg fra dual-rail domino logikk ved at SSDL ikke har en ren precharge fase. Legg merke til at nMOS transistoren som er brukt for å koble de komplementære nedtrekkene til GND er styrt av $\bar{\phi}$ og ikke ϕ som for dual-rail domino logikk. Dette betyr at porten ikke

¹³Bitlinjene skal typisk drive svært mange transistorer.

precharger, men *sampler*. Vi må forutsette at nedtrekkene ikke kan overstyre pMOS transistorene som skal trekke både X og \bar{X} opp mot V_{DD} eller logisk 1.

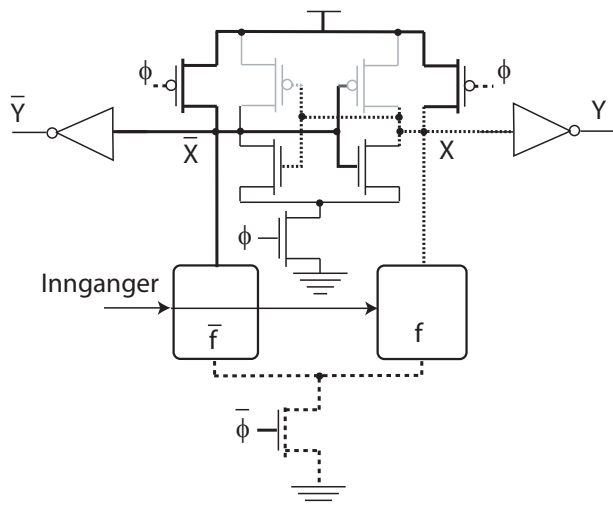


Figure 5.96: *Sample set differensiell logikk (SSDL) ved sample fase. (FIG6.63)*

Sample set differensiell logikk (SSDL) ved sampling er vist i Fig. 5.96. Vi antar at inngangene er stabile slik at ett av nedtrekkene er PÅ (f i dette eksemplet). Nedtrekket som er PÅ vil forsøke å trekke noden X ned til GND eller logisk 0. Vi kan anta at transistorene er dimensjonert slik at noden X vil få en spenning som er litt lavere enn V_{DD} , illustrert som stiplet linjer i figuren. Sense-amplifier kretsen som har en felles nMOS transistor i nedtrekket ned mot GND som er styrt av ϕ vil skru av nedtrekkene i sense-amplifieren. Poenget med sense-amplifierer er å lage et alternativt nedtrekk for nodene X og \bar{X} som er mye mer effektivt enn de komplekse nedtrekkene (f og \bar{f}). Vi ser at porten sampler inngangene slik at en av nodene X og \bar{X} blir trukket noe ned fra V_{DD} istedet for å precharges til V_{DD} . I denne samplingsfasen er det statisk effektforbruk fordi en av utgangene X eller \bar{X} vil ha opptrekk og nedtrekk som er på samtidig.

Når ϕ skifter fra 0 til 1 vil porten gå over i en *set* fase (motsvarende evaluering for dual-rail domino logikk) som vist i Fig. 5.97. nMOS transistoren i det effektive nedtrekket i sense-amplifieren vil trekke \bar{X} raskt ned til GND fordi X og ϕ er begge 1. Poenget er at kretsen slipper å trekke en av utgangene X eller \bar{X} ned til 0 via de komplekse nedtrekkene. De grå transistorene antas å være svake og brukes til å motvirke reduksjon i den av utgangene X og \bar{X} som fortsatt skal være logisk 1.

5.37.1 Mål

Forstå hvordan SSDL logikk virker.

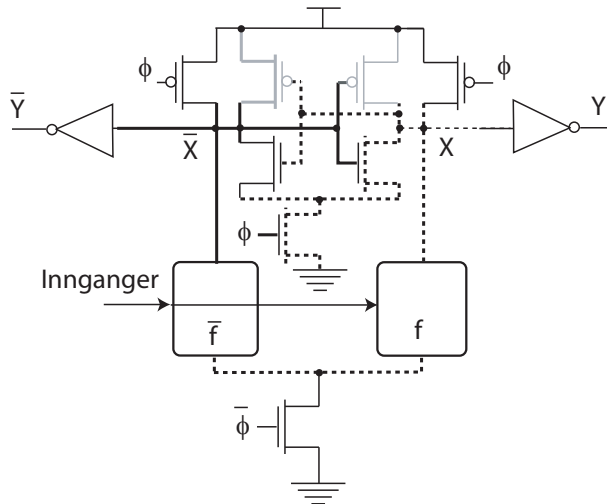


Figure 5.97: *Sample set differensiell logikk (SSDL) ved set fase. (FIG6.63)*

5.38 Enable/Disable CMOS differensiell logikk (ECDL)

(Kapittel 6.4.2.2 side 363 - 364)

*Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statistisk CMOS[?])

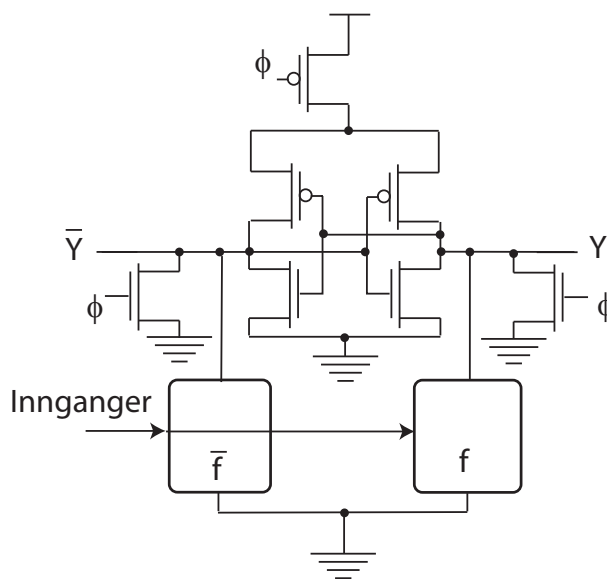


Figure 5.98: *Enable/disable CMOS differensiell logikk (ECDL). (FIG6.63)*

Enable/disable CMOS differensiell logikk (ECDL) som er vist i Fig. 5.98 representerer en forbedring av SSDL logikk ved at statisk effektforbruk blir redusert.

ECDL logikk ved disable er vist i Fig. 5.99. Porten disables når $\phi = 1$, slik at utgangene blir trukket ned til 0 som vist i Fig. 5.99. Her er “precharge” transistorene nMOS transistorer slik at de ikke vil danne en strømvei mellom spenningsreferansene V_{DD} og GND og dermed bidra med statisk effekt. pMOS transistoren i toppen sørger for at

5.38.1 Mål

Forstå hvordan ECDL virker.

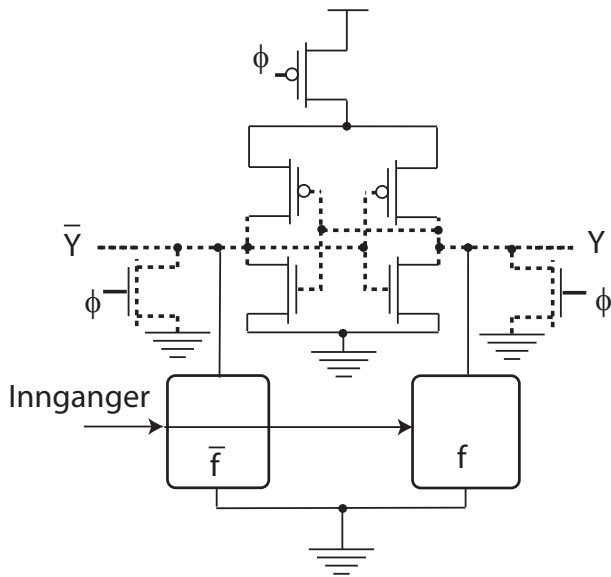


Figure 5.99: Enable/disable CMOS differensiell logikk (ECDL) ved disable. (FIG6.63)

det ikke ikke er opptrekk som er PÅ. Legg merke til at de komplementære nedtrekkene kun eventuelt bidrar til å hjelpe med nedtrekket når porten skal disables.

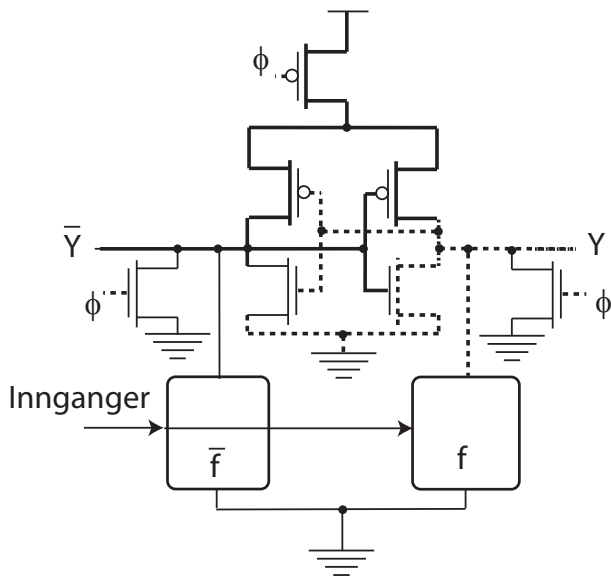


Figure 5.100: Enable/disable CMOS differensiell logikk (ECDL) ved enable. (FIG6.63)

ECDL logikk ved disable er vist i Fig. 5.100. Porten disables når $\phi = 0$, slik at pMOS transistoren i toppen som er styrt av ϕ skrur av og "precharge" transistorene skrur av som vist i Fig. 5.100. De to pMOS transistorene i det som logisk er to krysskoblete invertere er i starten PÅ og vil forsøke å trekke begge utgangene til logisk 1. I figuren har vi antatt at nedtrekket f er PÅ og det vil derfor bidra til å holde utgangen Y lav. De to krysskoblete inverterne vil holde utgangene til riktig logiske verdier.

5.39 Latched CMOS differensiell logikk (LCDL)

(Kapittel 6.4.2.3 side 364)

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[?])

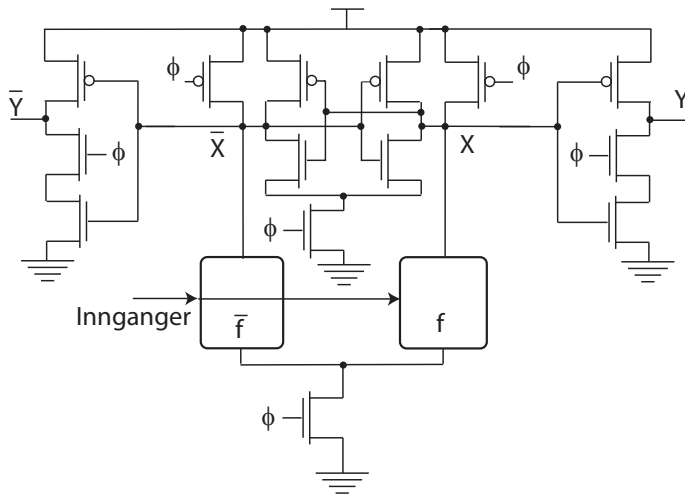


Figure 5.101: Latched CMOS differensiell logikk (LCDL). (FIG6.63)

Latched CMOS differensiell logikk (LCDL), som er vist i Fig. 5.101, ligner på SSDL. nMOS transistoren under nedtrekkene er styrt av ϕ og ikke $\bar{\phi}$. På utgangene er det en latch som er likt utgangstrinnet på en ekte-en fase (TSPC) latch.

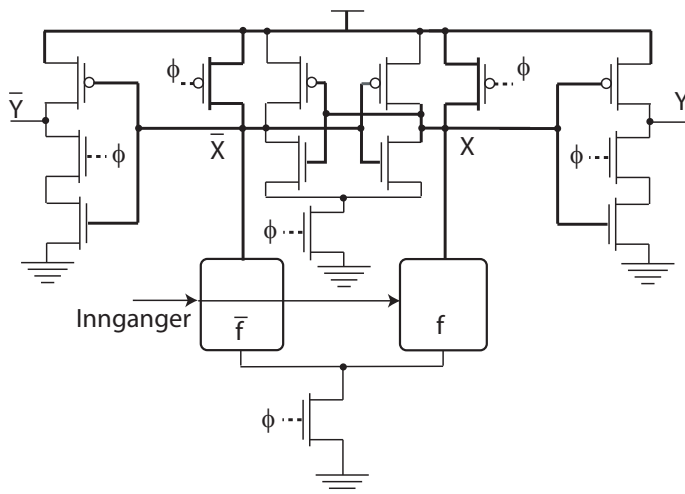


Figure 5.102: Latched CMOS differensiell logikk (LCDL) ved precharge. (FIG6.63)

LCDL i precharge fasen er vist i Fig. 5.102 der $\phi = 0$. Nodene X og \bar{X} precharges til logisk 1. Begge nedtrekkene, f og \bar{f} , er skrudd av ved hjelp av ϕ som styrer en nMOS transistor. Legg merke til at utgangene ikke påvirkes av eventuelle endringer på X og \bar{X} ved precharge fordi utgangstrinnet er likt et utgangstrinn for en TSPC latch.

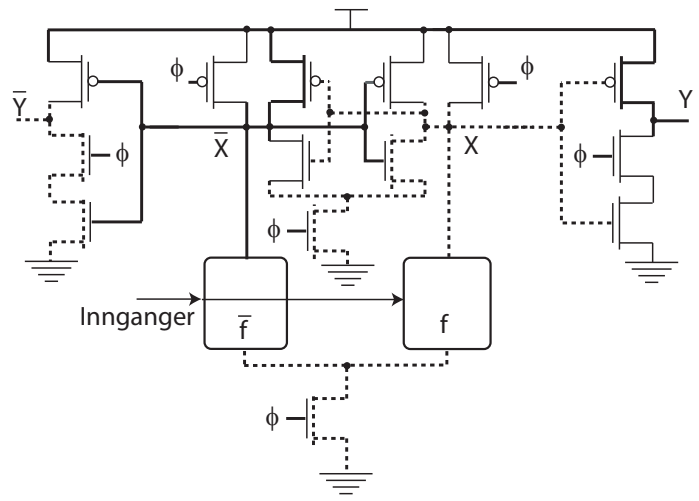


Figure 5.103: Latched CMOS differensiell logikk (LCDL) ved evaluering. (FIG6.63)

LCD i evalueringsfasen er vist i Fig. 5.102 der $\phi = 1$. Et av nedtrekkene vil nå trekke en av nodene X eller \bar{X} ned mot 0 og dermed trigge sense-amplifier kretsen som raskt bidrar med å trekke den aktuelle noden raskt helt ned til 0. Utgangstrinnene virker nå bare logisk som invertere.

5.39.1 Mål

Forstå hvordan LCDL virker.

5.40 Differensiell strøm svitsj logikk (DCSL)

(Kapittel 6.4.2.4 side 364)

* Kaskode spenning svitsj logikk (CVSL). (INF3400 Del 8: Effektforbruk og statisk CMOS[?])

Differensielle kretser kan bruke mye effekt fordi en av utgangene vil ha en transisjon i hver klokkesykel.

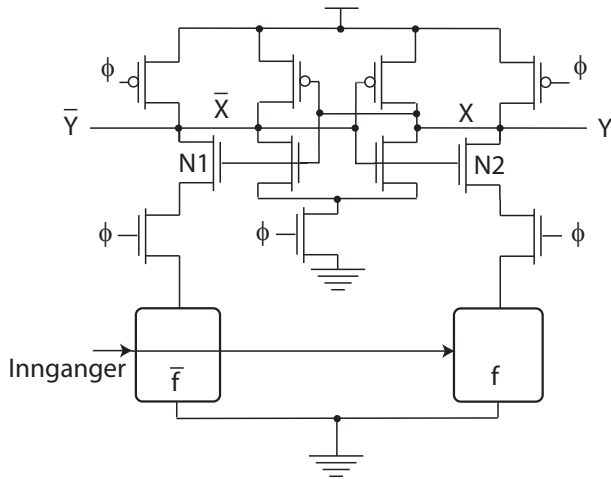


Figure 5.104: Differensiell strøm svitsj logikk (DCSL). (FIG6.63)

Med differensiell strøm svitsj logikk (DCSL), som vist i Fig. 5.104 forsøker man å redusere effektforbruket relatert til interne noder og øke hastigheten ved å redusere spenningen som nedtrekket skal lade ut.

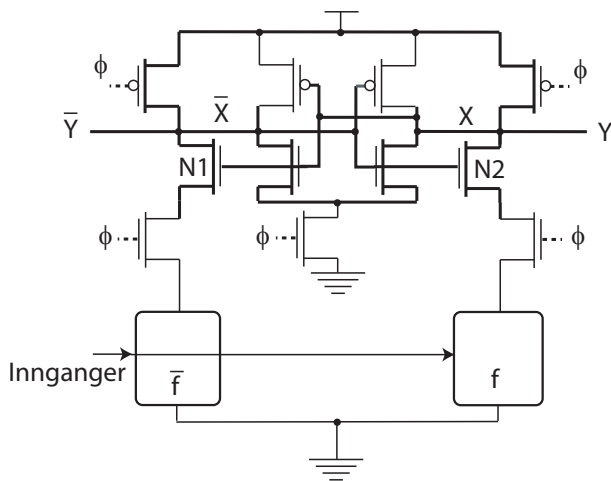


Figure 5.105: Differensiell strøm svitsj logikk (DCSL) ved precharge. (FIG6.63)

DCSL ved precharge er vist i Fig. 5.105, der vi antar at $\phi = 0$. Legg merke til at nMOS transistorene N1 og N2 er PÅ mens nMOS transistorene mellom disse transistorene og de komplekse¹⁴ nedtrekkene er AV. Interne noder i nedtrekkene vil ikke lades opp under precharge.

¹⁴Antar at nedtrekkene har mange interne noder.

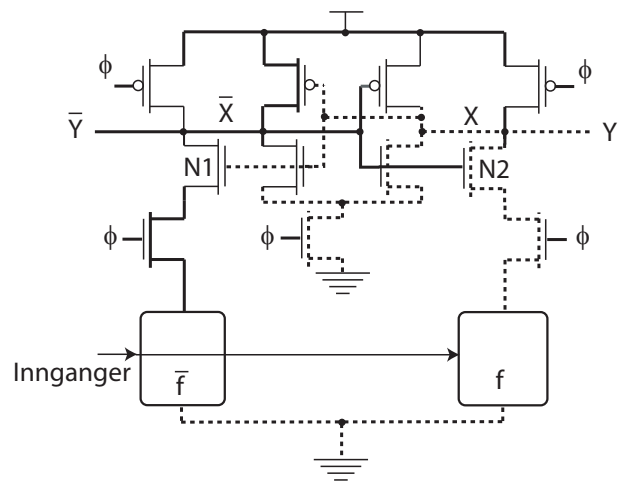


Figure 5.106: Differensiell strøm svitsj logikk (DCSL) ved evaluering. (FIG6.63)

DCSL ved evaluering er vist i Fig. 5.106, der vi antar at $\phi = 1$. Ved evaluering vil en av transistorene N1 eller N2 alltid stenge og sørge for at den av utgangene som ikke skal trekkes til 0 ikke vil ha strømveier ned i nedtrekket (\bar{f}). Det nedtrekket som er PÅ vil sørge for å dra en av utgangene ned til 0 (Y) og dermed interne (de fleste) nodene i dette nedtrekket også ned til 0.

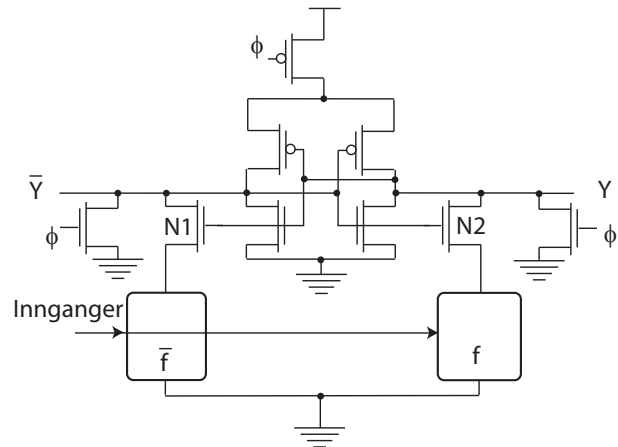


Figure 5.107: Differensiell strøm svitsj logikk (DCSL2). (FIG6.63)

DCSL2 som er vist i Fig. 5.107 precharger utgangene til 0 tilsvarende ECDL. Transistorene N1 og N2 virker på tilsvarende måte som for DCSL1.

I DCSL3, som er vist i Fig. 5.108, er de to nMOS precharge transistorene ertattet med en pass transistor som kortslutter de to utgangene slik at begge utgangene blir precharged til 0 fordi pMOS transistorene i de to krysskoblede inverterne ikke kan levere strøm på grunn av pMOS transistorene styrt av ϕ som er koblet til logisk spenningsreferansen V_{DD} .

5.41 BiCMOS kretser

(Kapittel 6.4.3 side 365)

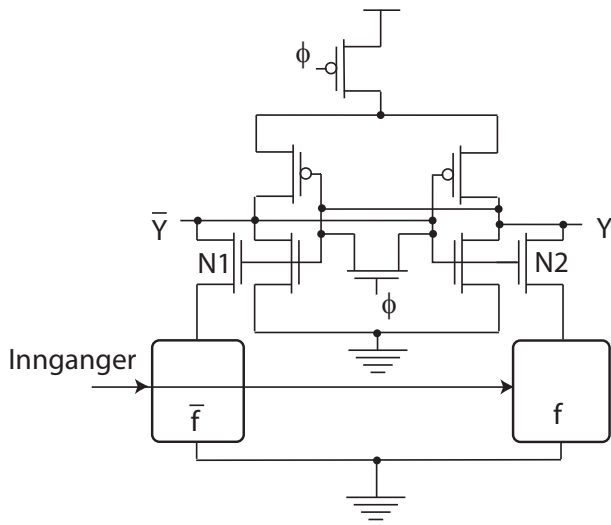


Figure 5.108: *Differensiell strøm svitsj logikk (DCSL3).* (FIG6.63)

5.40.1 Mål

Forstå hvordan ulike former for differensiell strøm svitsj logikk (DCSL) virker.

Bipolare transistorer kan levere mye mer strøm enn CMOS transistorer og brukes i noen kretser der det er spesielt stort behov for sterke utgangsdrivere. Noen CMOS prosesser tillater implementasjon av bipolare transistorer. Slike prosesser kalles BiCMOS.

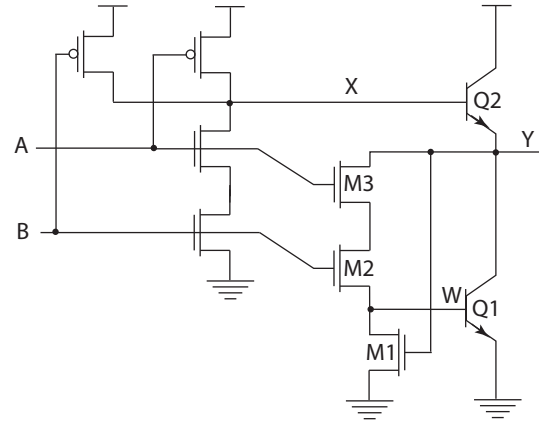


Figure 5.109: *BiCMOS NAND port.* (FIG6.67)

En BiCMOS NAND port er vist i Fig. 5.109. Transistorene M2 og M3 tilsvarer nedtrekket for en CMOS NAND port. Transistoren M1 brukes for å trekke basen W på npn transistoren Q1 til 0 slik at nedtrekket for den bipolare utgangen blir mest mulig effektivt.

5.41.1 Mål

Forstå hvordan enkle logiske porter kan implementeres i en BiCMOS prosess med bipolare transistor som utgangstrinn.

Logikk	nMOS	pMOS	Differens.	Stat. effekt	Sving	Dynamisk	Restore	Ratioed	Kaskade	Robust
Statisk	k	k	<i>NEI</i>	NEI	JA	NEI	JA	NEI	JA	Høy
pseudo nMOS	k	1	<i>NEI</i>	<i>JA</i>	<i>NEI</i>	NEI	JA	<i>JA</i>	JA	Middels
SFPL	2k+2	1	<i>NEI</i>	<i>JA</i>	<i>NEI</i>	NEI	JA	<i>JA</i>	JA	Middels
CVSL	2k	2	JA	NEI	JA	NEI	JA	NEI	JA	Høy
Dynamisk	k+1	1	<i>NEI</i>	NEI	JA	<i>JA</i>	JA	NEI	<i>NEI</i>	<i>Lav</i>
Domino	k+2	2	<i>NEI</i>	NEI	JA	<i>JA</i>	JA	NEI	JA	<i>Lav</i>
Dual-rail D.	2k+3	4	JA	NEI	JA	<i>JA</i>	JA	NEI	JA	<i>Lav</i>
CMOSTG	k	k	<i>NEI</i>	NEI	JA	NEI	JA	NEI	JA	Høy
LEAP	k	2	<i>NEI</i>	NEI	JA	NEI	JA	<i>JA</i>	JA	Middels
DPL	2k	2k	JA	NEI	JA	NEI	JA	NEI	JA	Høy
CPL	2k	4	JA	NEI	JA	NEI	JA	NEI	JA	Middels
EEPL	2k	4	JA	NEI	JA	NEI	JA	NEI	JA	Middels
SRPL	2k	2	JA	NEI	JA	NEI	JA	<i>JA</i>	JA	<i>Lav</i>
DCVSPG	2k-2	2	JA	NEI	JA	NEI	<i>NEI</i>	NEI	JA	Middels
PPL	k	k	JA	NEI	JA	NEI	<i>NEI</i>	NEI	JA	<i>Lav</i>
DSL	2k+2	2	JA	<i>JA</i>	<i>NEI</i>	NEI	JA	NEI	JA	Middels
CNTL	2k+4	2	JA	<i>JA</i>	<i>NEI</i>	NEI	JA	NEI	JA	Middels
NTL	k+1	2	<i>NEI</i>	<i>JA</i>	<i>NEI</i>	NEI	JA	<i>JA</i>	JA	Middels
SSDL	2k+6	6	JA	<i>JA</i>	JA	NEI	JA	NEI	<i>NEI</i>	<i>LAV</i>
ECDL	2k+4	3	JA	NEI	JA	NEI	JA	NEI	<i>NEI</i>	<i>LAV</i>
LCDL	2k+8	6	JA	NEI	JA	NEI	JA	NEI	<i>NEI</i>	<i>LAV</i>
DCSL1	2k+7	4	JA	NEI	JA	NEI	JA	NEI	<i>NEI</i>	<i>LAV</i>
BiCMOS	2k+1	k	<i>NEI</i>	<i>JA</i>	<i>NEI</i>	NEI	JA	NEI	JA	Middels

Table 5.1: *CMOS logikk.*

5.42 Sammenligning av CMOS logikk typer

(Kapittel 6.6 side 367 - 369)

Sammenligning av ulike typer CMOS logikk er gjort i tabell 5.1, der en rekke viktige faktorer som er knyttet til logikk typer; kompleksitet eller antall transistorer, om logikken er differensiell, statisk effektforbruk, utgangsving, dynamiske interne noder, signal restore, "ratioed" (transistor dimensjonering i opptrekk/nedtrekk), mulighet for å koble i kaskade og robusthet, er tatt med.

5.42.1 Mål

Forstå grunnleggende forskjeller i ulike CMOS logikker.

Chapter 6

Sekvensielle kretser

6.1 Introduksjon til sekvensielle kretser

(Kapittel 7.1 side 383 - 384)

Dynamiske kretser kan karakteriseres ved at utgangene på kretsene, eller portene, er en funksjon av inngangene i samme tidsperiode. *Sekvensielle kretser* derimot kan karakteriseres ved at utgangene på kretsene er en funksjon av inngangene i forrige tidsperiode og samme tidsperiode. Vi sier at en sekvensiell krets har en *tilstand*. Tilstandsmaskiner og pipeline systemer er viktige eksempler på sekvensielle kretser.

Sekvensielle kretser lages oftest ved hjelp av *latcher* eller *vipper* og kalles ofte *hukommelse*. Formålet med latcher og vipper er ikke primært hukommelse, men å skille en tilstand fra forrige tilstand i en sekvens. Vi kaller derfor slike latcher og vipper sekvensielle kretser.

Det er vanlig å skille statiske- og dynamiske sekvensielle kretser. Det er viktig å være klar over at med *statiske sekvensielle kretser* mener vi ikke kretser uten klokkeinnganger, men latcher eller vipper som vil holde en verdi uten signifikant lekkasje. *Dynamiske sekvensielle kretser* vil bare holde en verdi en begrenset tid og er derfor avhengig av hyppig oppfriskning av signalet. Statiske sekvensielle kretser vil ha en eller annen form for tilbakekobling som bidrar til å holde et signal uavhengig av klokkefrekvens i et system.

6.1.1 Mål

Kunne skille mellom statisk- og dynamisk sekvensering.

6.2 Introduksjon til sekvensering av statiske kretser

(Kapittel 7.2 side 384)

* Latcher og vipper[?]. (INF3400 Del 1: Kapittel XI side 10 - 11 og Kapittel XII side 11 - 12)

* RC forsinkelsesmodeller[?]. (INF3400 Del 4: Kapittel VII side 7 - 8)

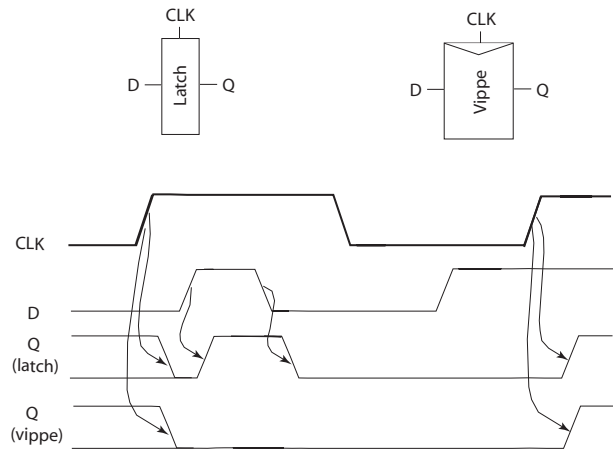


Figure 6.1: Latcher og vipper. (FIG7.1)

De vanligste sekvenseringselementene er latcher og vipper som vist i Fig. 6.1. Latcher og vipper har inngang (D) og styres av klokkesignal (CLK), og utgang (Q).

En *latch* er transparent når $CLK = 1$ som betyr at utgangen er følsom for endringer på inngangen. Når $CLK = 0$ vil ikke utgangen på en latch påvirkes av inngangen. En latch er *nivåfølsom*, dvs. utgangen følger inngangen når $CLK = 1$, og utgangen er stabil (buffer) når $CLK = 0$.

En *vippe* er en *kantfølsom* krets, dvs. ved en positiv klokkeovergang vil verdien på inngangen D kopieres til utgangen Q, og utgangen vil ikke påvirkes av endringer på inngangen D i resten av klokkeperioden.

6.2.1 Mål

Forstå den grunnleggende virkemåten til latcher og vipper, og få et inntrykk av forskjellen på latcher og vipper.

6.3 Sekvenseringsmetoder

(Kapittel 7.2.1 side 385 - 387)

* Latcher og vipper[?]. (INF3400 Del 1: Kapittel XI side 10 - 11 og Kapittel XII side 11 - 12)

Dersom vi har delkrets i form av *kombinatorisk logikk*, dvs. porter eller delkretser uten synkroniseringssignaler (klokker) som i praksis vil si en transparent delkrets, vil det som regel være behov for å synkronisere den kombinatoriske delkretsen med andre signaler fra andre delkretser. Det er ulike metoder for å synkronisere signaler eller ulike sekvenseringsmetoder som kan anvendes. Synkroniseringselementene eller sekvenseringselementene som benyttes er typisk latcher eller vipper.

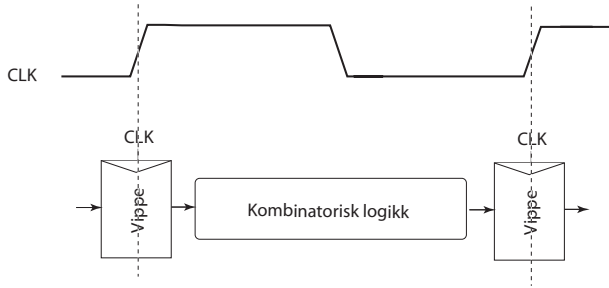


Figure 6.2: *Sekvensering (synkronisering) med vipper.* (FIG7.2)

I Fig. 6.2 er det vist kombinatorisk logikk synkronisert ved hjelp av *kantfølsomme vipper*. Synkroniseringspunktet er gitt av en positiv klokkeflanke. Ved en positiv klokkeflanke vil vippene kopiere inngangssignalet til utgangen og holde verdien inntil neste positive klokkeflanke. Ved å invertere synkroniseringssignalene vil man få synkroniseringspunkt ved negative klokkeflanker.

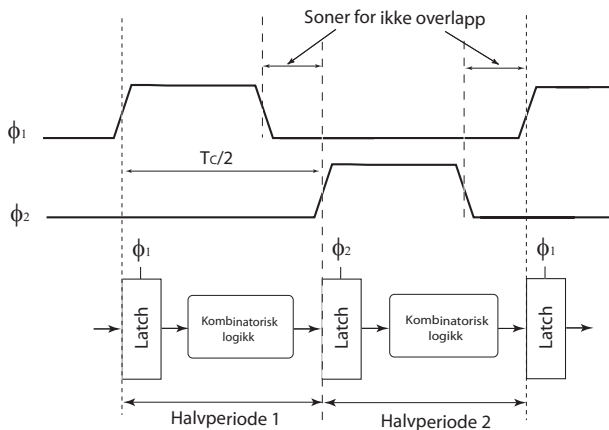


Figure 6.3: *Sekvensering (synkronisering) med latcher.* (FIG7.2)

Synkronisering ved hjelp av latcher er vist i Fig. 6.3. I et to-fase system er det vanlig å bruke to-fase ikke-overlappende klokker, der klokkefasene (klokkesignalene ϕ_1 og ϕ_2) ikke er høye samtidig. Hele klokkeperioden er T_c . En halvperiode vil bestå av en tid der en av klokkes-

ignalene er høye etterfulgt av en tid der ingen av klokkesignalene er høye (ikke-overlappende).

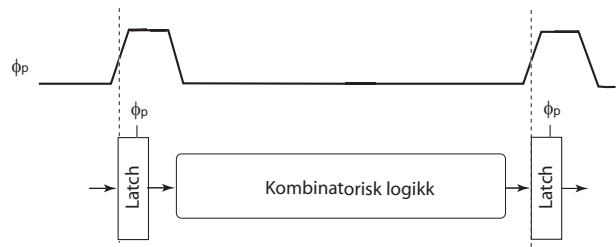


Figure 6.4: *Sekvensering (synkronisering) med latcher som styres av pulser.* (FIG7.2)

Synkronisering ved hjelp av latcher som styres av pulser er vist i Fig. 6.4. Som vi ser er klokkesignalet annerledes enn for vanlige latcher eller vipper. Klokkesignalet kan karakteriseres som korte pulser, dvs. med en duty cycle som er vesentlig mindre enn 50%. Latchene styres med korte klokkepuls som trigger en innlesning (sampling) av inngangen. Latchen vil selv etter kort tid sørge for å holde den samplede verdien når klokkesignalet er lavt. Samplingstiden er meget kort sammenlignet med tidsforsinkelsen i kombinatorisk logikk mellom latchene. På denne måten vil kombinatorisk logikk virke som en lagring av en tilstand.

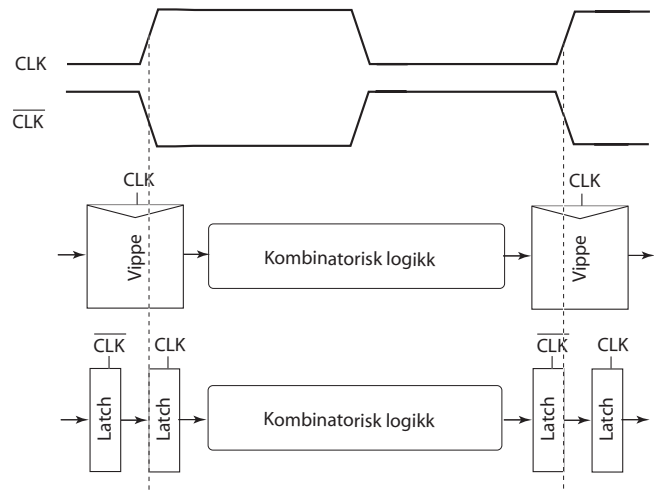


Figure 6.5: *Sekvensering (synkronisering) med vipper realisert som latcher med klokke og invertert klokke.* (FIG7.3)

Sekvensering ved hjelp av vipper styrt av klokke (CLK) og invertert klokke (\overline{CLK}) er vist i Fig. 6.5. Her er to latcher plassert inntil hverandre. Vi må da forutsette at de to latchene ikke er transparente samtidig, dvs. vi kan ikke tillate at CLK og \overline{CLK} er høye samtidig. I praksis må vi sikre oss ved å ha en ikke-overlappende periode der CLK og \overline{CLK} er lave samtidig. Et slikt system setter strenge krav til forholdet mellom CLK og \overline{CLK} .

I tabell 6.1 er de ulike uttrykkene (termene) for forsinkelse som legger begrensninger for timing i sekvensielle kretser.

Term	Kommentar
t_{pd}	Logisk propagering forsink.
t_{cd}	Logisk contamination forsink.
t_{pcq}	Latch/vippe klokke til Q propagering forsink.
t_{ccq}	Latch/vippe klokke til Q contamination forsink.
t_{pdq}	Latch D til Q propagering forsink.
t_{cdq}	Latch D til Q contamination forsink.
t_{setup}	Latch/vippe setup tid
t_{hold}	Latch/vippe hold tid

Table 6.1: Notasjon for sekvenseringselementer.

6.3.1 Timing for kombinatorisk logikk

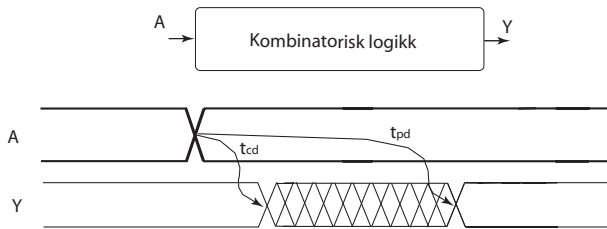


Figure 6.6: Timing for kombinatorisk logikk. (FIG7.4a)

Tidsforsinkelse i kombinatorisk logikk er gitt av *contamination forsinkelse* og *propagering forsinkelse* og er vist i Fig. 6.6. Utgangen kan begynne å endre seg etter tiden gitt av contamination forsinkelse t_{cd} og utgangen vil bli stabil etter maksimal tidsforsinkelse fra inngang til utgang som er gitt av propageringsforsinkelse t_{pd} . Contamination forsinkelse og propageringsforsinkelse kan være forskjellige på grunn av forskjellige kjeder eller signalveier i logikken.

6.3.2 Timing for vippe

Dersom vi har et sekvensielt system bestående av vipper som klokkes med CLK og \overline{CLK} må vi forutsette at disse signalene er presist i motfase.

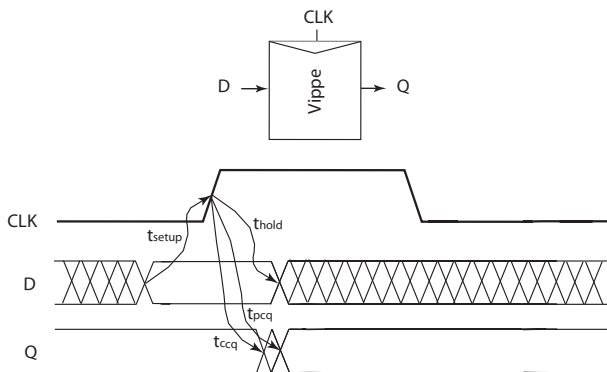


Figure 6.7: Timing for vippe. (FIG7.4b)

Tidsforsinkelse i en vippe er vist i Fig. 6.7. Vippen vil holde en tilstand i hele perioden med unntak av lagring av ny verdi en kort stund etter at klokkesignalet skifter fra 0 til 1. Vippen er avhengig av *setup-* og *hold tid*. Dette betyr

at inngangen D må være stabil en stund før klokkesignalet CLK endres fra 0 til 1 og forbli stabil tilstrekkelig lenge etter at klokkesignalet har blitt endret. Setup- og hold tid er sikkerhetsmarginer for å sikre riktig vippe funksjon.

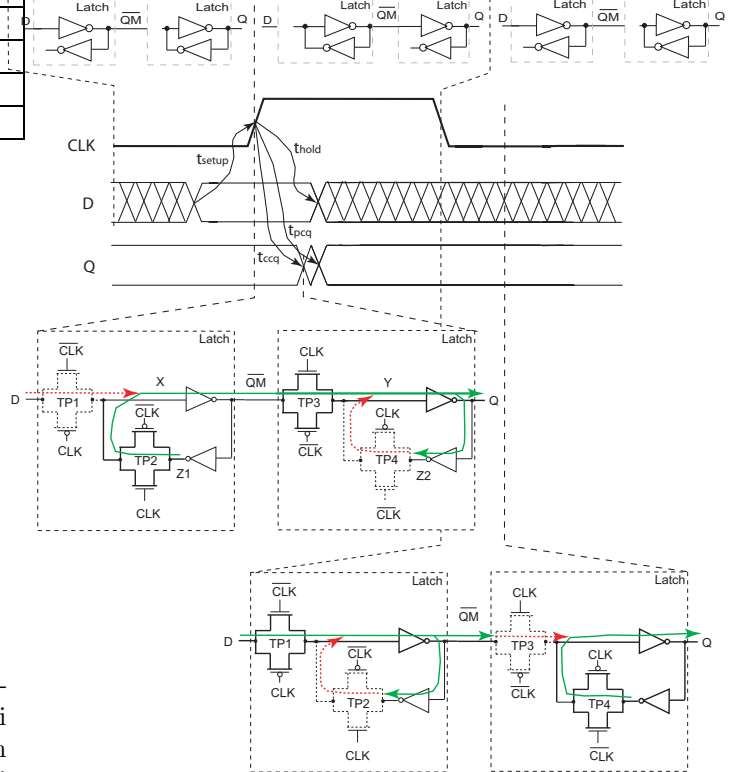


Figure 6.8: Timing for vippe. (FIG7.4b)

Timing detaljer for vipper er vist i Fig. 6.8, der vippene er realisert som to latches i motfase. Når klokkesignalet CLK er lavt vil inngangslatches i vippene være transparente og kontinuerlig sample inn $\overline{QM} = D$, samtidig med at utgangen Q oppfriskes ved hjelp av to inverterte i tilbakekobling. Vippens funksjon er å sample inn D i slutten av tidsperioden når $CLK = 0$, dvs. vi ønsker å laste inn verdien D akkurat når CLK endres fra 0 til 1. Transmisjonsport 1 (TP1) stenger og TP2 åpner slik at signalet som ligger på $Z1 = \overline{QM}$ vil påvirke X via TP2 og deretter \overline{QM} . Vi har en situasjon der $Z1 = X (= D)$. Dersom D forandres og påvirker X før TP2 åpner helt kan vi få en endring i X og deretter i \overline{QM} som er uønsket. En hold tid på D vil sikre at \overline{QM} ikke endres uønsket. I tillegg vil det være gunstig at TP1 stenger før TP2 åpner slik at X ikke kan drives via TP1 når tilbakekoblingen skal være aktiv. For latchen som kontrollerer utgangen på vippene (Q) har vi nå en situasjon der TP3 skal overstyre TP4. Det kan være gunstig at TP4 stenger før TP3 åpner for å redusere støy på utgangen. Vi ser at contamination forsinkelse og propageringsforsinkelse er tilnærmet like og er tidsforsinkelsen fra \overline{QM} til Q når TP3 åpner, dvs fra stigende klokkeflanke. Ved fallende klokkeflanke vil det kunne oppstå en transparent situasjon dersom TP3 og TP1 er åpne samtidig. Det er viktig at TP3 stenger før TP1 åpner for å sikre at ikke inngangssignalet D påvirker Q . Vi ønsker i denne situasjonen når $CLK = 0$ at vippene

skal holde sin lagrede verdi.

6.3.3 Timing for Latch

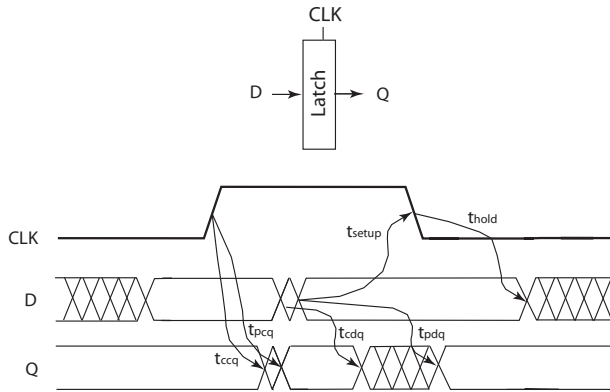


Figure 6.9: *Timing for latch. (FIG7.4c)*

Timing for en latch er vist i Fig. 6.9. Samplingtidspunktet for latchen er ved fallende klokkeflanke. Vi må forutsette at inngangssignalet D kommer fra en latch styrt av et annet klokkesignal, for eksempel invertert klokke som gir klokkesignaler i motfase. Mellom latchene er det typisk kombinatorisk logikk.

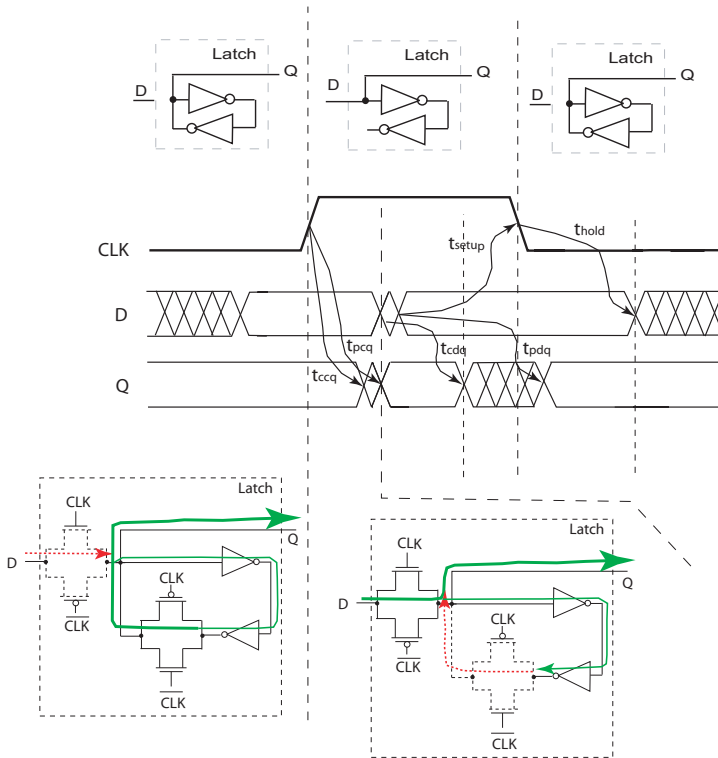


Figure 6.10: *Timing for latch. (FIG7.4c)*

Timing detaljer for en latch er vist i Fig. 6.10. Når klokkesignalet CLK er lavt vil latchen fungere som et minneelement med tilbakekobling. Dersom vi ser på latchen og kombinatorisk logikk som prosesserer signalet fra latchen og neste latch får vi et system som vist i Fig. 6.3 med $CLK = \phi_1$ og $\overline{CLK} = \phi_2$. Fra stigende klokkeflanke får

vi klokke til Q forsinkelse t_{ccq} og t_{pcq} som vist i Fig. 6.10. Latchen er transparent når klokkesignalet er høyt og enhver forandring på D vil påvirke Q . Vi må forutsette at D er stabil en liten stund før fallende klokkeflanke slik at latchen rekker å sample riktig verdi. Vi kaller denne tiden setup tid. For å være sikker på riktig sampling må D være stabil en stund etter at klokkesignalet har blitt 0. Vi kaller dette for hold tid.

6.3.4 Mål

Forstå detaljer ved timing av sekvensielle kretser.

6.4 Begrensninger for maks forsinkelse

(Kapittel 7.2.2 side 388 - 392)

Ideelt vil vi ønske hele klokkeperioden tilgjengelig for prosessering av signaler. Sekvenseringen vil medføre en viss overhead som er knyttet til sekvenseringselementene. Dersom tidsforsinkelsen i kombinatorisk logikk er for stor vil en vippe som skal lagre resultatet få signalet for sent slik at situasjonen ikke tilfredstiller setup tid. Vi kaller en slik feil for set feil eller maks forsinkelse feil.

6.4.1 Vipper

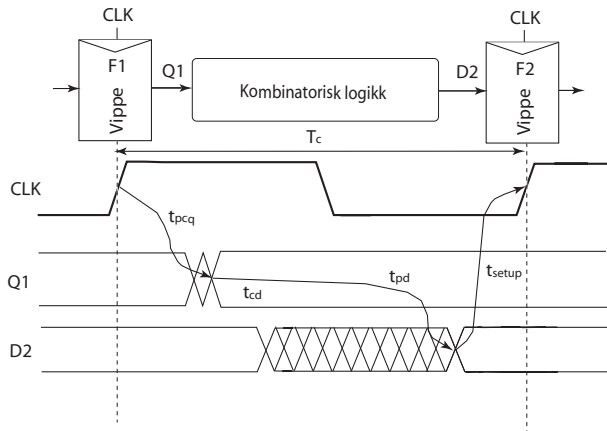


Figure 6.11: Begrensninger for maks forsinkelse. (FIG7.5)

Begrensninger for maks forsinkelse er vist i Fig. 6.11. Dersom vi antar at vippene $F1$ og $F2$ styres av identiske klokkesignaler, dvs. at klokkesignalene har transisjoner på nøyaktig samme tidspunkt, er maksimal tid fra en vippe til neste vippe gitt av T_c :

$$T_c \geq t_{pcq} + t_{pd} + t_{setup}, \quad (6.1)$$

der t_{pcq} er propageringsforsinkelse for klokke til utgang (Q) for vippe, t_{pd} er propageringsforsinkelse i kombinatorisk logikk og t_{setup} er setup tid for vippe.

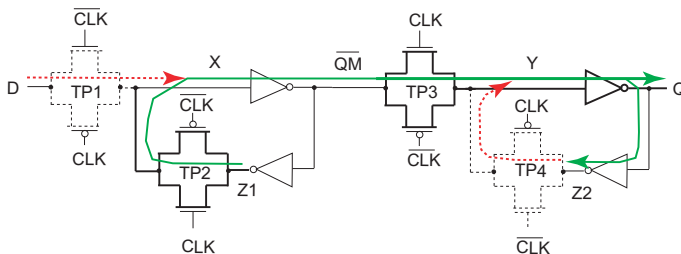


Figure 6.12: Propageringsforsinkelse for klokke til utgang (Q) for vippe.

Propageringsforsinkelse for klokke til utgang (Q) for vippe er vist i Fig. 6.12. Når klokkesignalet er lavt vil

inngangslatchen sample inn D til \overline{QM} kontinuerlig. Når klokkesignalet endres fra 0 til 1 lukker transmisjonsportene $TP1$ og $TP4$ mens $TP2$ og $TP3$ åpner. Verdien som er ligger på \overline{QM} (lagret) vil transmitteres via $TP3$ til Q . Propageringstidsforsinkelsen i dette tilfellet er gitt av en transmisjonsport og en inverter i kjede og last på Y og utgangen Q .

Propageringsforsinkelse i kombinatorisk nettverk mellom de to vippene i Fig. 6.11 er gitt av en den signalveien mellom $Q1$ og $D2$ som har størst tidsforsinkelse. Vi kaller den aktuelle signalveien en *kritisk signalvei*.

Setuptiden for vippene er en sikkerhetsmargin for å sample riktig verdi i inngangslatchen i vippene. Setuptid varierer med prosess og implementasjon av vippene.

Vi kan uttrykke begrensning for maks tidsforsinkelse på en annen form:

$$t_{pd} \leq T_c - (t_{setup} + t_{pcq}), \quad (6.2)$$

der $(t_{setup} + t_{pcq})$ er overhead ved sekvenseringen. Vi ser at ved å redusere t_{setup} og t_{pcq} til et minimum får vi mer av tiden som er tilgjengelig til å prosessere signaler i kombinatorisk logikk (gjøre beregninger).

Tiden tilgjengelig i en klokkeperiode T_c er direkte avhengig av t_{setup} og t_{pcq} for vippene og tidsforsinkelse i kritisk signalvei i kombinatorisk logikk mellom vippene. Klokkefrekvens, eller maksimal klokkefrekvens, er direkte knyttet til klokkeperioden:

$$f_{max} = \frac{1}{T_c}. \quad (6.3)$$

6.4.2 Latcher

Timingdetaljer i et sekvenseringssystem med transparente latcher som styres av tofase klokker er vist i Fig. 6.13. Vi antar at inngangen $D1$ ankommer latch 1 når $\phi_1 = 1$ og propagerer i kombinatorisk logikk $KL1$ fordi latch 1 er transparent når $\phi_1 = 1$. Kritisk signalvei i $KL1$ vil bestemme maksimal tidsforsinkelse t_{pd1} for $KL1$. Vi må forutsette at $D2$ (etter $KL1$) er stabil i god tid før ϕ_2 svinger fra 1 til 0 slik at vi får riktig verdi samlet (latched) i latch 2. Tilsvarende argumentasjon gjelder for latch 3 osv. Vi kan uttrykke en klokkeperiode T_c som:

$$T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2}. \quad (6.4)$$

Dersom vi løser med hensyn på total propageringsforsinkelse i hele klokkeperioden får vi:

$$\begin{aligned} t_{pd} &= t_{pd1} + t_{pd2} \\ &\leq T_c - (2t_{pdq}), \end{aligned} \quad (6.5)$$

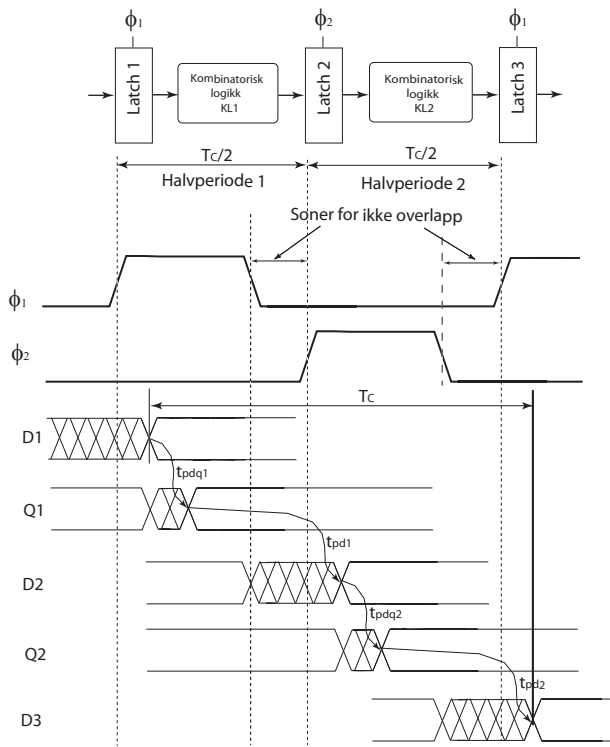


Figure 6.13: Begrensninger for maks forsinkelse i et sekvenseringssystem med latcher styrt av tofase klokker (FIG7.7).

der ($2t_{pdq2}$) er overhead gitt av propageringsforsinkelse i lathene, som vi antar er lik for de aktuelle lathene.

Begrensninger for maks forsinkelse i et sekvenseringssystem med latcher styrt av pulser er vist i Fig. 6.14. Dersom pulsbredden t_{pw} er større enn setuptiden t_{setup} har vi situasjonen som vist øverst i figuren. Vi trenger i dette tilfellet ikke å forutsette at utgangen av kombinatorisk logikk er stabil før klokkepulsene kommer. Dersom klokkepulsene er mindre enn setup tiden har vi situasjonen som vist nederst, og vi må da forutsette at inngangen til lathene er stabil før klokkepulsene kommer. Vi kan beskrive dette som:

$$T_c \geq \max(t_{pdq} + t_{pd}, t_{pcq} + t_{pd} + t_{setup} - t_{pw}) \quad (6.6)$$

som gir:

$$t_{pd} \leq T_c - \max(t_{pdq}, t_{pcq} + t_{setup} - t_{pw}), \quad (6.7)$$

der $\max(t_{pdq}, t_{pcq} + t_{setup} - t_{pw})$ er overhead inkludert forsinkelse i en latch.

6.4.3 Mål

Forstå hva som begrenser utnyttelse av en klokkeperiode ved sekvensering.

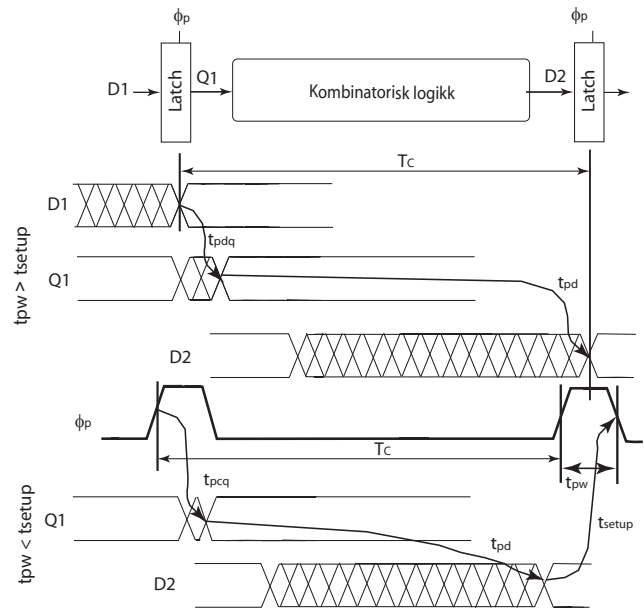


Figure 6.14: Begrensninger for maks forsinkelse i et sekvenseringssystem med latcher styrt av pulser. (FIG7.8).

6.5 Begrensninger for minimum forsinkelse

(Kapittel 7.2.3 side 392 - 396)

Sekvenseringselementer bør kunne plasseres inntil hverandre uten vesentlig kombinatorisk logikk mellom elementene. Eksempel på sekvensielle systemer med minimal kombinatorisk logikk mellom sekvenseringselementene er *pipeline systemer*.

Dersom hold tid er stor og contamination forsinkelsen er liten kan data propagere gjennom to sekvenseringselementer ved en klokkeflanke. En slik feil kalles *race feil*, *hold tid feil* eller *minimum-forsinkelse feil*.

6.5.1 Vipper

Begrensninger for minimum tidsforsinkelse for vipper er vist i Fig. 6.15 der vi antar at klokkesignalene til de to vippene er helt i fase (like).

Detaljer for et system med to vipper som er koblet sammen uten (minimalt) kombinatorisk logikk for $CLK = 0$ er vist i Fig. 6.16. Vi ser at så lenge $CLK = 0$ vil latch 1-1 i den første vippene følge inngangen D , dvs. vi latcher inn D i $\overline{Q1M}$. Utgangen på den første vippene $Q1$ holdes stabil ved hjelp av tilbakekobling i latch 1-2, og $Q1$ føres via eventuell kombinatorisk logikk til den neste vippene, nærmere bestemt latch 2-1 som latcher (sampler) inn $Q1$ til $\overline{Q2M}$. Vi ser at i slutten av perioden hvor $CLK = 0$ vil TP1, TP4, TP5 og TP8 være helt åpne og TP2, TP3, TP6 og TP7 være helt lukket. Vi må forutsette at inngangen D er stabil en stund før stigende klokkeflanke (t_{setup}). En kritisk situasjon som medfører feil er dersom TP3 og TP5 er åpne samtidig slik at $\overline{Q1M}$ blir transmittert til $\overline{Q2M}$ via $Q1$. Vippenes funksjon er å lagre verdier i etterfølgende klokkeperioder.

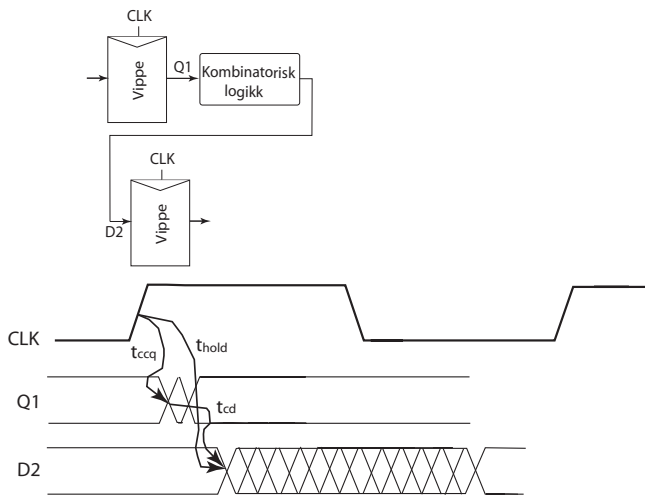


Figure 6.15: Begrensinger på minimumsforsinkelse for vipper. (FIG7.9).

Dersom vi ser på detaljene rundt stigende klokkeflanke, som er vist i Fig. 6.17, ser vi at $\overline{Q1M}$ skal endre $Q1$ med klokke til Q contamination forsinkelse t_{ccq} . Det vil si at vi får en endring, men ikke nødvendigvis stabil verdi, på $Q1$ ved tidspunktet t_{ccq} etter stigende klokkeovergang. Ideelt sett har nå den neste vippen lukket TP5 og kan ikke påvirkes av endringer på $D2 = Q1$. For å sikre at en vippe ikke endres feilaktig er det påkrevd at det defineres en hold tid for inngangen. I dette tilfellet betyr det at den siste vippen forutsetter at inngangen $D2$ er stabil en liten stund etter stigende klokkeflanke. Det er avgjørende at ikke $D2$ endres som følge av endring på $Q1$ idet ved stigende klokkeflanke før vippens setup tid er over. Vi kan uttrykke dette som

$$t_{cd} \geq t_{hold} - t_{ccq}, \quad (6.8)$$

der t_{cd} er contamination forsinkelse i kombinatorisk logikk¹ mellom vippene. Med andre ord, det er viktig at tidsforsinkelsen mellom vippene er så stor at inngangen til vippe nummer 2 ikke har fått ny verdi fra latch 1-1 før setup tiden til vippe 2 er over. Dersom $D2$ endres før setuptiden er over vil latch 1-2 og latch 2-1 være transparente samtidig slik at $\overline{Q2M}$ blir lik $\overline{Q1M}$, som vil medføre at $Q2$ blir lik $Q1$ i neste omgang. Dersom contamination forsinkelse, dvs. klokke til Q forsinkelse, for vippene er større enn hold tid kan vippene plasseres helt inntil hverandre. I dette tilfellet vil ikke vippe nummer 2 rekke å reagere på endringer på inngangen for tidlig.

6.5.2 Latcher

I Fig. 6.18 er begrensninger på minimumsforsinkelse for latcher som er styrt av to fase lokker vist. Latchene styres

¹I eksemplet er det ikke kombinatorisk logikk mellom vippene slik at contamination forsinkelse blir minimum forutsatt at ikke avstanden mellom vippene er stor.

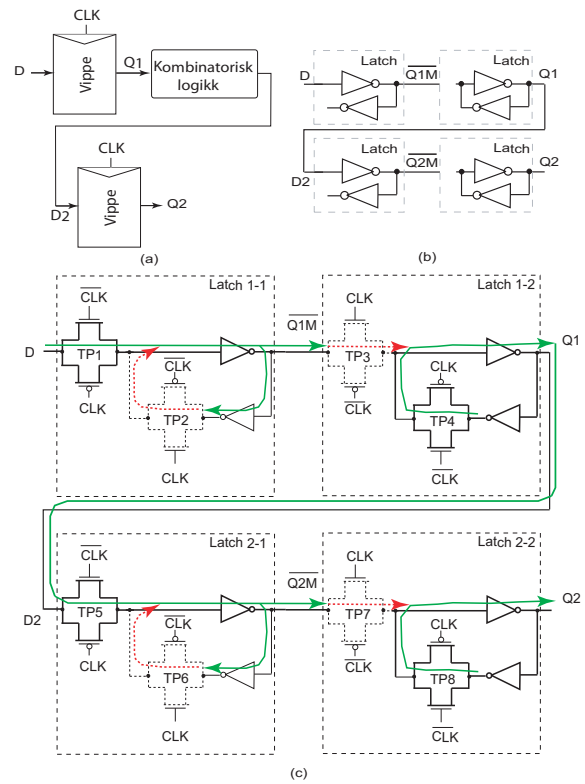


Figure 6.16: Begrensinger på minimumsforsinkelse for vipper. Detaljer i timing når $CLK = 0$. (FIG7.9).

av to fase ikkeoverlappende klokker som skal garantere at to latcher som styres av hver sin klokkefase ikke er åpne samtidig. Når begge klokkefasene ϕ_1 og ϕ_2 er lave samtidig skal begge latchene være lukket slik at utgangene ikke skal kunne påvirkes av inngangene. Ved stigende klokkeflanke på ϕ_1 åpner latchene som er styrt av ϕ_1 slik at $D1$ latches inn til $Q1$. Merk at latchene er *nivåfølsomme*, dvs. utgangen på latchene vil påvirkes av inngangen så lenge latchen er åpen, i motsetning til en vippe som er *kantfølsom*. Vi forutsetter derfor at latchen som styres av ϕ_2 har en hold tid som går utover tiden når $\phi_2 = 1$. Vi kan anta at denne hold tiden t_{hold} er så lang at den kan påvirke utgangen $Q2$ etter at $Q1$ og $D2$ er endret som følge av latching ved tidspunktet når ϕ_1 svinger fra 0 til 1.

Timing detaljer for latcher som er styrt av to fase lokker ved stigende transisjon på ϕ_1 er vist i Fig. 6.19. Dersom hold tiden for latch styrt av ϕ_2 er for lang i forhold til tidsforsinkelse mellom latchene kan vi latche inn feil verdi. Vi har en situasjon der TP1 er PÅ slik at latch styrt av ϕ_1 er åpen og TP3 ikke er helt AV slik at latch styrt av ϕ_2 er delvis åpen. I denne situasjonen er den ene latchen åpen og den andre delvis åpen slik at de to latchene satt sammen blir delvis transparent. Vi kan uttrykke betingelser for korrekt latching ved å sette en nedre grense for contamination forsinkelse for kombinatorisk logikk mellom latchene:

$$t_{cd1}, t_{cd2} \geq t_{hold} - t_{ccq} - t_{ikkeoverlapp}, \quad (6.9)$$

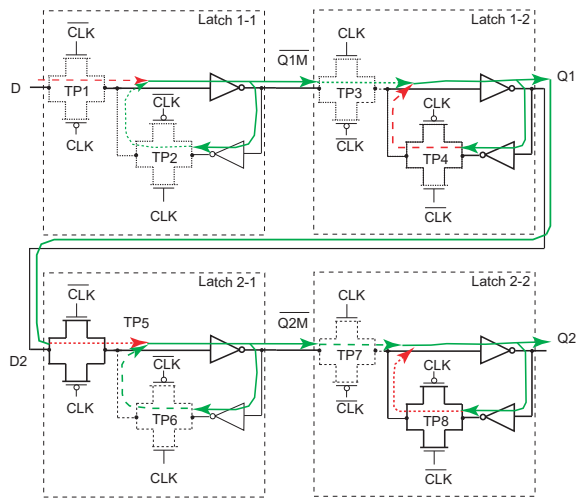


Figure 6.17: Begrensninger på minimumsforsinkelse for vipper. Detaljer i timing når $CLK = 0 \rightarrow 1$. (FIG7.9).

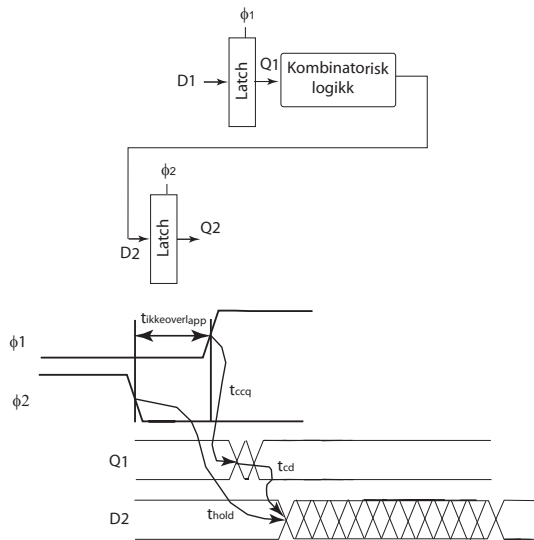


Figure 6.18: Begrensninger på minimumsforsinkelse for latcher som er styrt av to fase klokker. (FIG7.10).

der t_{hold} er hold tid for latchene, t_{ccq} er klokke til Q contamination forsinkelse for latchene og $t_{ikkeoverlapp}$ er tiden der begge klokkefasene er lave. Alternativt kan vi uttrykke dette som en begrensning på hold tiden:

$$t_{hold} \leq t_{ikkeoverlapp} + t_{ccq} + t_{cd}. \quad (6.10)$$

Dersom tiden der begge klokkefasene er lave og $t_{ikkeoverlapp}$ er tilstrekkelig lang vil vi ikke få problem med for liten tidsforsinkelse i kombinatorisk logikk mellom latchene.

Latcher som styres av klokkepulser er vist i Fig. 6.20. Det er tilsvarende begrensninger for minimum tidsforsinkelse i kombinatorisk logikk mellom latchene som for tofase latcher. Vi kan uttrykke dette som:

$$t_{cd} \geq t_{hold} - t_{ccq} + t_{pw}. \quad (6.11)$$

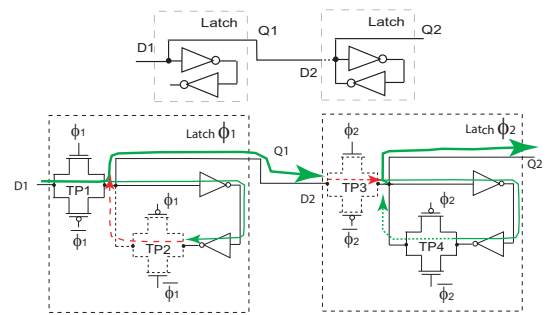


Figure 6.19: Begrensninger på minimumsforsinkelse for latcher som er styrt av to fase lokker. Detaljer ved stigende transisjon på ϕ_1 . (FIG7.10).

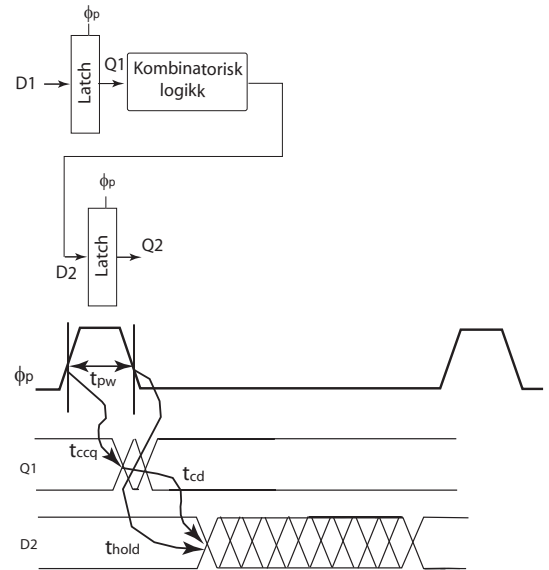


Figure 6.20: Begrensninger på minimumsforsinkelse for latcher som er styrt av klokkepulser. (FIG7.11).

6.5.3 Mål

Forstå hva som begrenser utnyttelse av en klokkeperiode ved sekvensering.

6.6 Fordeling av tid mellom klokkefasene

(Kapittel 7.2.4 side 396 - 399)

I et system med vipper vil data være tilgjengelig på utgangen ved stigende klokkeflanke. Den neste vippen (etterfølgende) skal være klar til å lache ved neste stigende klokkeflanke. Dersom data ankommer tidligere vil vippen blokkere for inngangen ved at inngangslatchen samler inn alle forandringer på inngangen, men slipper ikke gjennom signalet til utgangen før neste stigende klokkeflanke.

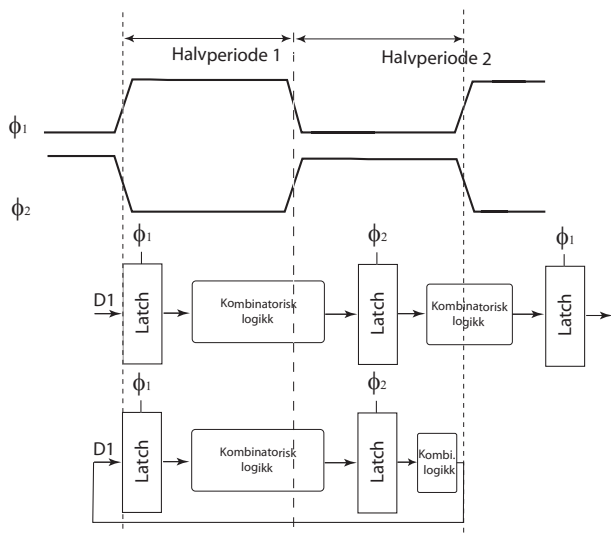


Figure 6.21: Fordeling av tid mellom klokkefasene. (FIG7.12).

I et system med latches derimot er ikke latchedidspunktet knyttet til klokkeflanke, men til klokkenivåer. Med andre ord er latches transparente i en tidsperiode slik at presist tidspunkt for latchingen ikke er viktig. Vi kan med andre ord lage systemer der tidsforsinkelse i kombinatorisk logikk er forskjellig mellom ulike latches uten at dette reduserer klokkefrekvensen. Dette kalles fordeling av tid mellom klokkefasene (time borrowing) og er vist i Fig. 6.21.

Maksimal fordeling av tid mellom to ikkeoverlappende klokkefasene er vist i Fig. 6.22. På grunn av at inngangsdata ikke må være stabil før ved fallende klokkesignal for mottager latch kan en fase låne opptil en halvpart av en halvperiode for en klokke frekvens:

$$t_{laan} \leq \frac{T_c}{2} - (t_{setup} + t_{ikkeoverlapp}). \quad (6.12)$$

I praksis vil fordeling av tid mellom klokkesyklusene i et system med latches styrt av klokkepulser være svært begrenset og gitt av:

$$t_{laan} \leq t_{pw} + t_{setup}, \quad (6.13)$$

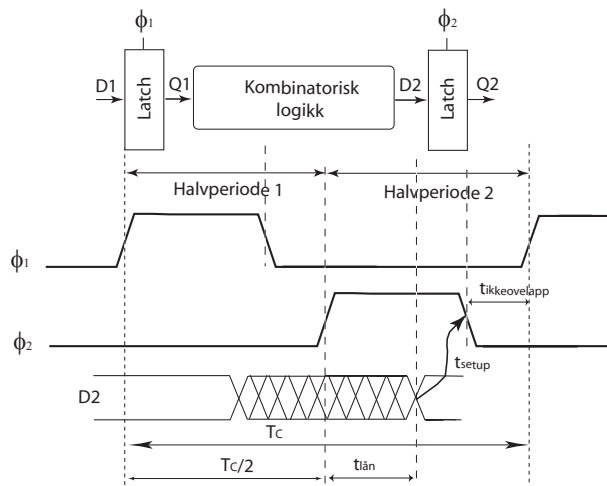


Figure 6.22: Fordeling av tid mellom klokkefasene. (FIG7.13).

der t_{pw} er bredden på klokkepulsen.

I praksis vil ofte fordeling av tid mellom klokkefasene ikke være aktuelt ved design, men vil kunne fungere som en ekstra sikkerhet for at et system vil virke på grunn av ulik tidsforsinkelse i kombinatorisk logikk mellom latches i et system.

6.6.1 Mål

Forstå hvordan man kan fordele tiden i en klokkeperiode ujevnt mellom to faser eller halvperioder.

6.7 Klokke skew

(Kapittel 7.2.5 side 399 - 402)

Vi har til nå forutsatt ideelle klokkesignaler uten *skew* eller forskyvning. I praksis vil klokkeflankene komme til litt forskjellige tider for ulike latcher og vipper. Dette skyldes at sekvenseringselementene vil være plassert ulike steder på en krets og dermed representere ulik last² for klokkesignalet.

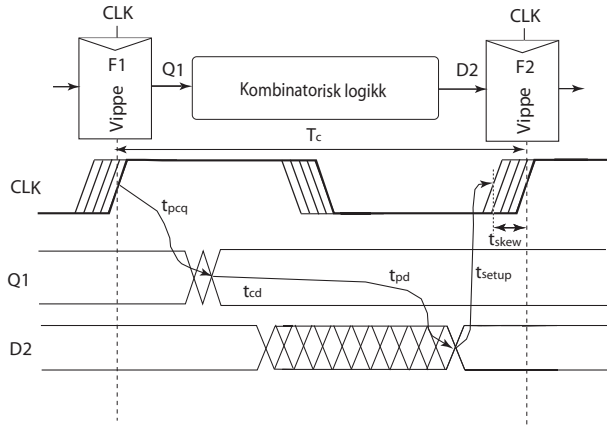


Figure 6.23: Klokkeskew og vipper. (FIG7.15a).

I Fig. 6.23 er det vist et system med vipper som styres av et klokkesignal hvor tykk linje for *CLK* markerer det seneste tidspunktet for klokkeovergangen. Klokkeovergangene kan komme tidligere som vist i figuren. Den kritiske situasjonen for maksimal forsinkelse i et system med vipper er om vippene som sender et signal får klokkeovergangen sent og mottager vippene får klokkeovergangen tidlig. I dette tilfellet må klokkeskew trekkes fra den tiden systemet har tilgjengelig for å prosessere signaler i kombinatorisk logikk mellom vippene. Man regner da klokkeskew som en del av overheaden ved sekvenseringen.

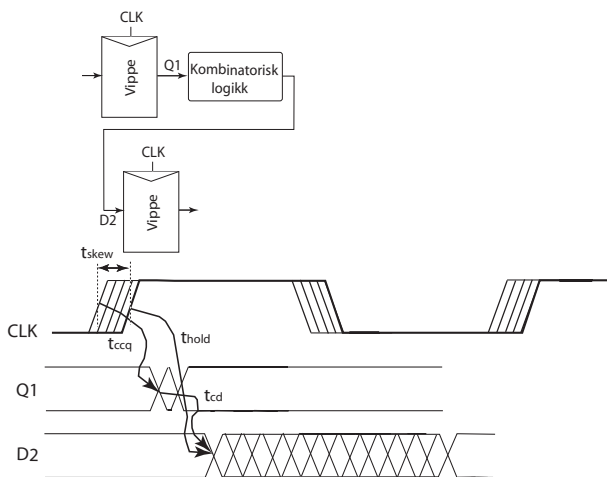


Figure 6.24: Klokkeskew og vipper. (FIG7.15b).

Den kritiske situasjonen for minimum tidsforsinkelse har

²Spesielt ulik kapasitans og motstand på grunn av interkonnekt.

vi når sendervippen får klokkeovergangen tidlig og mottager vippene får klokkeovergangen sent som vist i Fig. 6.24. I dette tilfellet vil den effektive holdtiden øke og vi får begrensningene:

$$t_{pd} \leq T_c - (t_{pcq} + t_{setup} + t_{skew}), \quad (6.14)$$

der $(t_{pcq} + t_{setup} + t_{skew})$ er overhead i sekvenseringen. Vi har da:

$$t_{cd} \geq t_{hold} - t_{ccq} + t_{skew}. \quad (6.15)$$

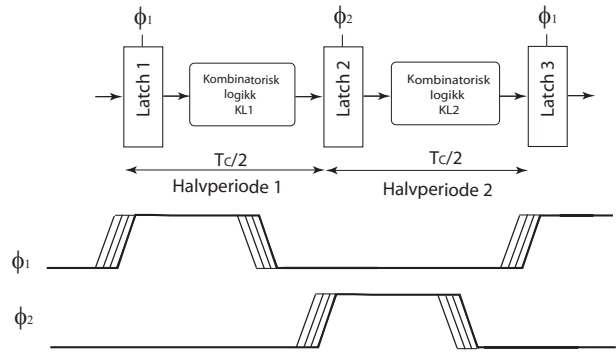


Figure 6.25: Klokkeskew og transparente latcher. (FIG7.16).

I et tofase system med transparente latcher vil ikke klokkeskew redusere ytelsen, som vist i Fig. 6.25, så lenge de to klokkefasene er ikke-overlappende. Vi kaller et system med transparente latcher med tofase ikke-overlappende klokker for *skew tolerant*. Muligheten for å fordele tid mellom klokkefasene vil imidlertid bli mer begrenset:

$$t_{pd} \leq T_c - 2t_{pdq}$$

$$t_{cd1}, t_{cd2} \geq t_{hold} - t_{ccq} - t_{ikkeoverlapp} + t_{skew} \quad (6.16)$$

der t_{cd} er contamination forsinkelse for kombinatorisk logikk mellom latcher.

For latcher som styres av klokkepulser vil fordeling av tid i kombinatorisk nettverk mellom latcher bli svært begrenset fordi klokkeskew kan føre til at klokkepulserne blir svært smale.

6.7.1 Mål

Forstå hvordan klokkeskew påvirker et sekvenseringssystem

6.8 Konvensjonelle CMOS latcher

(Kapittel 7.3.1 side 402 - 405)

*Terskelfall [?]. Pass transistor DC karakteristikk, Kapittel 2.5.6 side 101 - 102.

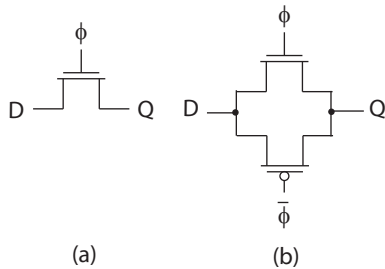


Figure 6.26: *Dynamisk en-transistor- og transmisjonsport latch.* (FIG7.17a og b)

I prinsippet kan man lage en *latch* ved hjelp av en transistor som vist i Fig. 6.26 (a). Ideelt vil utgangen Q følge inngangen D når ϕ er høy og holde verdien når ϕ er lav. I utgangspunktet er ofte få transistorer å fortrekke framfor flere transistorer når man implementere en krets. For pass transistoren som latch vil det imidlertid være fire viktige begrensninger:

- Utgangen vil ikke svinge mellom GND og V_{DD} . Spesielt vil ikke en nMOS transistor kunne brukes til å drive en logisk høy verdi (V_{DD}) på grunn av terskelfall.
- Utgangen er *dynamisk*, dvs. utgangen vil være udrevet og flyte når kontrollsignalet (ϕ) er lavt. Dette kan medføre at utgangen endres slik at latchen ikke holder riktig verdi.
- Inngangen D driver direkte til en source/drain terminal på en transistor og ikke en gate terminal. Dette kan resultere i merkbar støy og gjør det vanskelig å prediktere forsinkelse i en krets.
- Tilstanden på den lagrede noden (Q) kan påvirkes av støy på utgangen (også Q), slik at lagringsnoden er utsatt for endringer fra utgangen.

I Fig. 6.26 (b) er det vist en transmisjonsport som en latch. Den vil fungere bedre enn pass transistoren, men vil også ha begrensninger som latch. Ved å utvide fra pass transistor til transmisjonsport har vi bare fjernet den første begrensningen. Utvidelsen medfører også et behov for et invertert kontrollsignal $\bar{\phi}$.

I Fig. 6.27 har vi lagt til en inverter på utgangen slik at utgangen blir invertert \bar{Q} og dermed isolert latchens lagrede verdi fra utgangen. Latchen er da ikke utsatt for

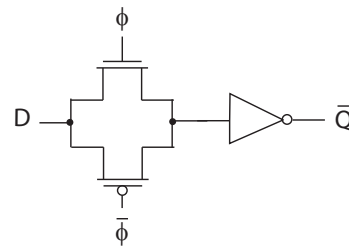


Figure 6.27: *Dynamisk latch med transmisjonsport og inverter.* (FIG7.17c)

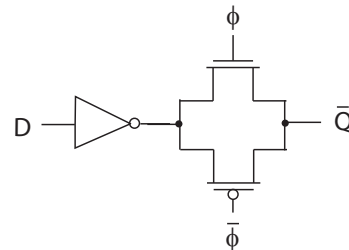


Figure 6.28: *Dynamisk latch med inverter og transmisjonsport.* (FIG7.17d)

støy fra utgangen, men vil fortsatt være dynamisk og med inngang koblet til source/drain terminal på transistor.

I Fig. 6.28 er det vist en latch med inverter og transmisjonsport. Vi har nå inngangen koblet til gate, men utgangen vil være utsatt for støy. En logisk ekvivalent krets som vist i Fig. 6.28, men med mindre arealbehov er en *klokke CMOS inverter* (C^2MOS) som er vist i Fig. 6.29. C^2MOS er noe tregere enn en inverter og en transmisjonsport fordi transistorene som styres av klokkesignalene aldri vil bidra i parallell. Det er derfor ikke vanlig å bruke klokke CMOS på inngangen til latchen.

Ved å kombinere latchene i Fig. 6.27 og 6.28 og klokke transmisjonsportene i motfase får vi en *statisk latch* som vist i Fig. 6.30. Det som nå mangler er gate terminal innngang.

I Fig. 6.31 har latchen fått en inverter på inngangen og utgangen blir dermed Q på grunn av to inverteringer. Utgangen lastes av C^2MOS inverteren i tillegg til eksterne kretser. En raskere latch der lasten på utgangen er redusert er vist i Fig. 6.32. Dette er en latch som ikke har noen av de begrensninger som ble beskrevet for pass transistor latchen. Vi ser imidlertid at latchen har blitt relativt kompleks, som medfører økt tidsforsinkelse, effektforbruk og økt areal (utlegg). Enklere latcher med gode elektriske egenskaper baseres på latchen i Fig. 6.32 med forenklinger som øker latchens ytelse og reduserer arealbehovet.

En forenkling av latchen i Fig. 6.32 er vist i Fig. 6.33, der C^2MOS inverteren er erstattet med en svak³ inverter. Når latchen sampler (latcher inn) vil inngangssignalet via inngangsinverteren og transmisjonsporten overstyre tilbakekoblingen slik at utgangen Q får ny verdi. Når transmisjonsporten er skrudd AV vil tilbakekoblingen

³Med svak inverter menes en inverter som leverer lite strøm på grunn av lite W/L forhold for transistorene.

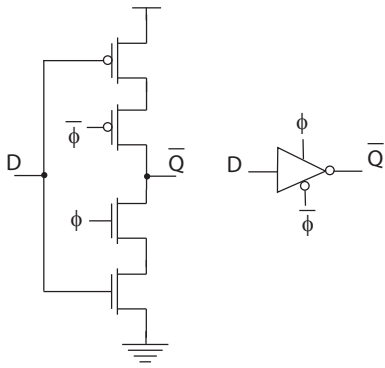


Figure 6.29: Klokke CMOS latch. (FIG7.18)

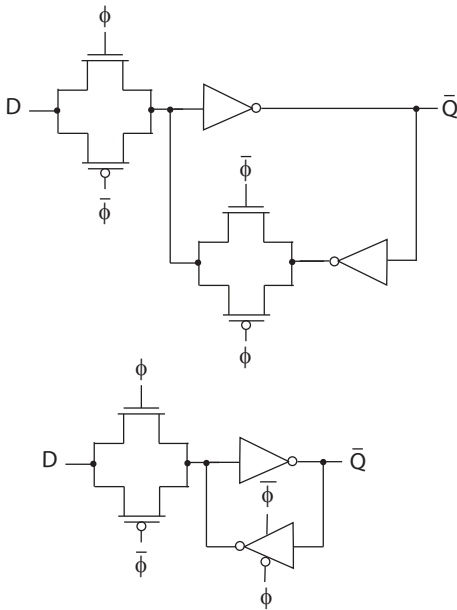


Figure 6.30: Statisk latch med transmisjonsport og inverter og tilbakekobling i motfase øverst. Tilbakekobling med C²MOS nederst. (FIG7.17e)

være tilstrekkelig sterk til å holde verdien på Q. Denne latchen er derfor statisk.

6.8.1 Varianter av transparente latcher

Latchen som er vist i Fig. 6.34 brukes typisk i registre eller FGPA (Field Programmable Gate Array) kretser. Latchen styres av kontrollsignalene WR (WRite) og RD (ReaD). Legg merke til at inngangen er koblet til en pass transistor. Dette betyr at noden X i utgangspunktet ved latching ikke kan trekkes helt opp til logisk 1 (V_{DD}), men X vil eventuelt bli trukket helt opp til logisk 1 ved hjelp av inverteren i tilbakekoblingen. Latchene har vanligvis sammenkoblet utgang som forutsetter at bare en av mange latcher med felles utgang kan selekteres ved et gitt tidspunkt.

En annen variant av statisk latch er vist i Fig. 6.35. Inverteren til venstre brukes for å generere klokke invertert lokalt. Inngangen er koblet til source/drain terminaler på en transmisjonsport slik at man må være oppmerk-

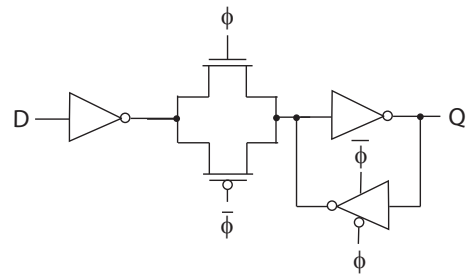


Figure 6.31: Statisk latch med inverter inngang og utgang Q. (FIG7.17f)

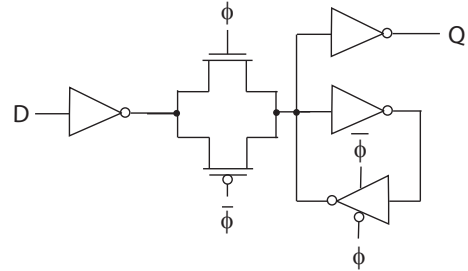


Figure 6.32: Statisk latch med utgang Q. (FIG7.17g)

som på at latchen kan være “tung” å drive for inngangen. Latchen kan utvides med en inverter før transmisjonsporten og dermed vil utgangen bli Q. Tilbakekoblingen er nå delvis klokke, dvs. X blir precharge til logisk 1 dersom $\bar{X} = 0$ (som betyr at $X = 1$) og trukket ned til 0 når $\bar{\phi} = 1$ og $\bar{X} = 1$ (som betyr at $X = 0$). Når latchen skal sample, dvs. når $\phi = 1$ må transmisjonsporten overstyre precharge i tilbakekoblingen, og derfor er pMOS transistoren i tilbakekoblingen svak.

6.8.2 Mål

Forstå hvordan konvensjonelle latcher i CMOS kan implementeres.

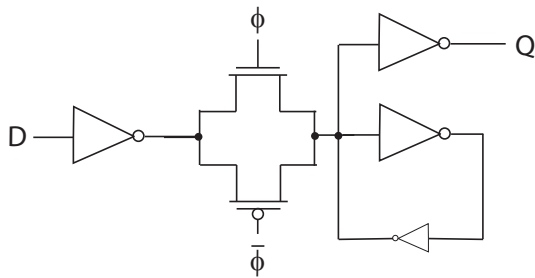


Figure 6.33: Statisk latch med utgang Q og svak uklokke tilbak kobling. (FIG7.17i)

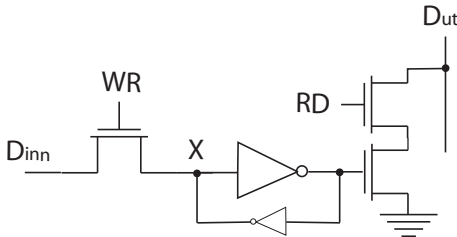


Figure 6.34: Statisk latch for FPGA (Field Programmable Gate Array). (FIG7.17j)

6.9 Konvensjonelle CMOS vipper

(Kapittel 7.3.2 side 405 - 407)

En dynamisk vippe er vist i Fig. 6.36. Denne vippen er satt sammen av to dynamiske latches som klokkes i motfase.

En statisk vippe med to statiske latches som er klokke i motfase er vist i Fig. 6.37. Denne vippen har både Q og \bar{Q} utganger. Det er vanlig at vipper bare har en klokkeinnang ϕ og genererer invertert klokkesignal lokalt.

Ved lokal generering av invertert klokkesignal kan man få en liten forsinkelse for det inverterte klokkesignalet som vist i Fig. 6.38. Ved negativ klokkeflanke, dvs. ϕ skifter fra 1 til 0 vil det ta en viss tid t_{00} der begge klokkesignalene er lave. Signalveier som er markert med tykke linjer er da PÅ. Vi ser at den første transmisjonsporten $TP1$ er PÅ slik at inngangslatchen samler inngangen. Tilbak koblingen i inngangslatchen burde vært skrudd av,

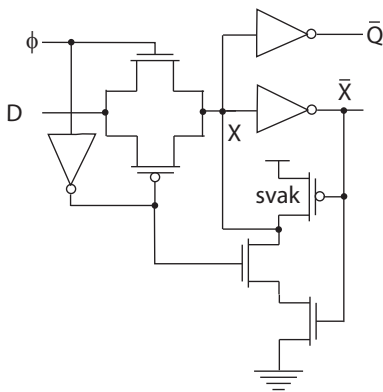


Figure 6.35: Statisk latch variant. (FIG7.17k)

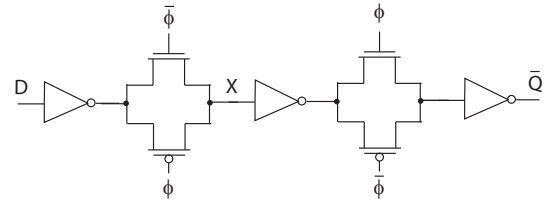


Figure 6.36: Dynamisk vippe. (FIG7.19a)

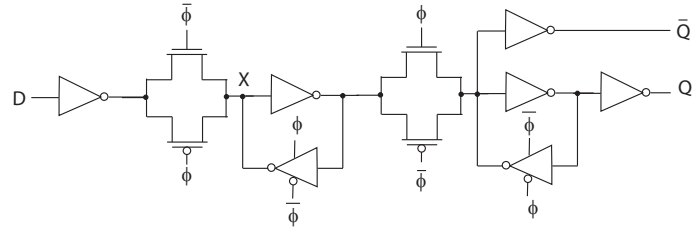


Figure 6.37: Statisk vippe. (FIG7.19b)

men vil i perioden t_{00} ha ett opptrekk som er PÅ. Dette opptrekket er egentlig bare PÅ når $Y = 0$ som betyr at $X = 1$. I en situasjon der inngangen $D = 1$ får vi en konflikt i noden X fordi inngangen via en inverter og $TP1$ vil drive X til 0 mens tilbak koblingen vil drive X til 1. Dette er bare et temporært problem fordi vi må forutsette at $\bar{\phi}$ endres til 1 før positiv klokkeflanke kommer. Etter t_{00} vil tilbak koblingen skrus AV og nodene X og Y vil få riktig verdi drevet fra inngangen D . Problemet er mer betydelig enn man kan få inntrykk av ved bare å studere inngangslatchen i perioden t_{00} . Husk at utgangslatchen har samme klokkesignaler slik at i perioden t_{00} vil transmisjonsporten for utgangslatchen $TP2$ også være feilaktig PÅ. Dette medfører at noden Z vil påvirkes av Y (direkte fra D) og via tilbak koblingen i utgangslatchen C^2MOS 2. Vi ser at bare opptrekket i tilbak koblingen er PÅ slik at tilbak koblingen vil forsøke å precharge Z til 1. Den korrekte funksjonen til utgangslatchen er at $TP2$ er AV og tilbak koblingen er PÅ. Vi ser at i perioden t_{00} er hele vippen transparent slik at D kan påvirke Q og \bar{Q} direkte. En kritisk situasjon er når $Z = 0$ og $U = 1$ rett før t_{00} og $D = 1$ som betyr at Z drives mot 1 via Y og X fra D . I denne situasjonen er ikke tilbak koblingen i utgangslatchen aktiv og Z kan derfor drives til 1, som igjen endrer U til 0 og bidrar til å holde $Z = 1$ feilaktig. Når perioden t_{00} er over vil $TP2$ stenge, men dette er for sent til å unngå en feilaktig endring av utgangene.

Vi får et tilsvarende problem ved positiv klokkeflanke som vist i Fig. 6.39. I perioden t_{11} vil begge klokkesignalene være høye slik at vippen blir temporært transparent. Riktig vippe funksjon er at inngangslatchen ikke samler inngangen, men har en aktiv tilbak kobling. Dette betyr at $TP1$ skal være AV og C^2MOS 1 skal være PÅ. For utgangslatchen skal $TP2$ være PÅ og tilbak koblingen C^2MOS 2 være AV. I perioden t_{11} kan vi få en alvorlig situasjon dersom nodene X og Y endres på grunn av D og tilbak koblingen C^2MOS 1 i inngangslatchen ikke kan overstyre $TP1$. I denne situasjonen blir vippen transparent.

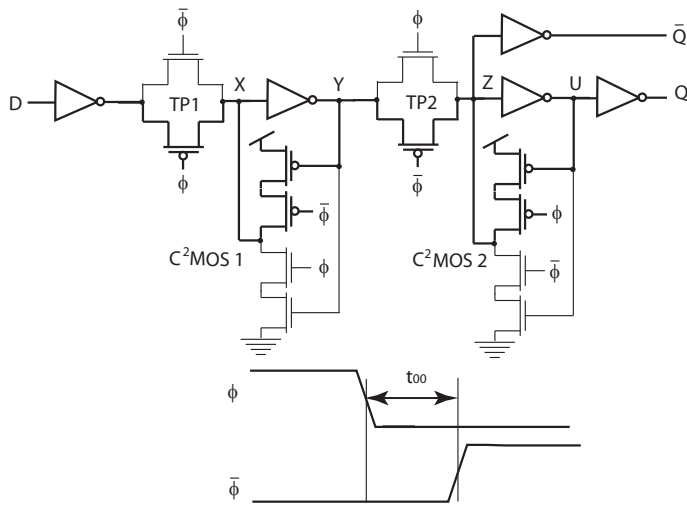


Figure 6.38: Statisk vippe ved negativ klokkeflanke og lokal generering av invertert klokke. (FIG7.19b)

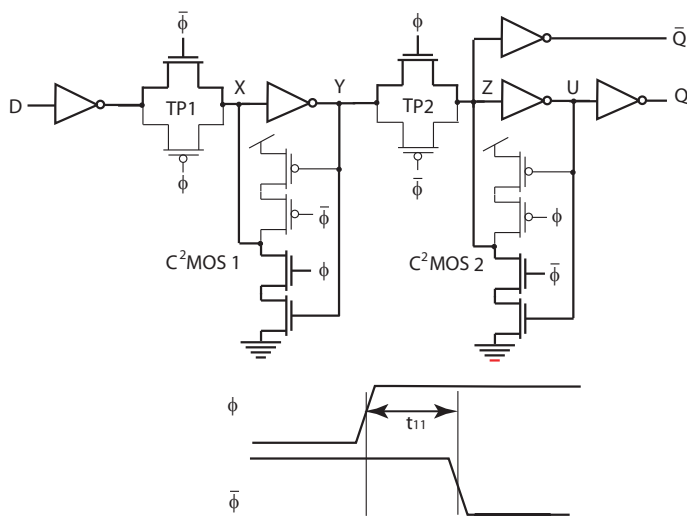


Figure 6.39: Statisk vippe ved positiv klokkeflanke og lokal generering av invertert klokke. (FIG7.19b)

En vanlig løsning på problemet med delvis transparente vipper er å bruke tofase ikke-overlappende klokker som vist i Fig. 6.40.

6.9.1 Mål

Forstå hvordan konvensjonelle vipper i CMOS kan implementeres.

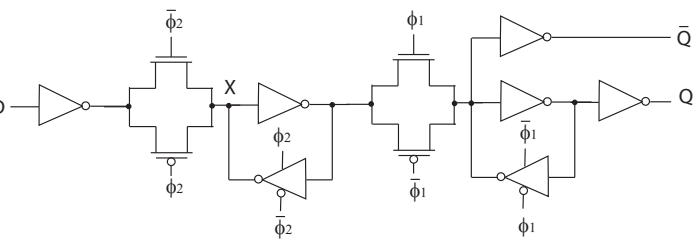


Figure 6.40: Statisk vippe med tofase ikke-overlappende klokker. (FIG7.21)

6.10 Latcher som styres av klokkepuls

(Kapittel 7.3.3 side 407 - 408)

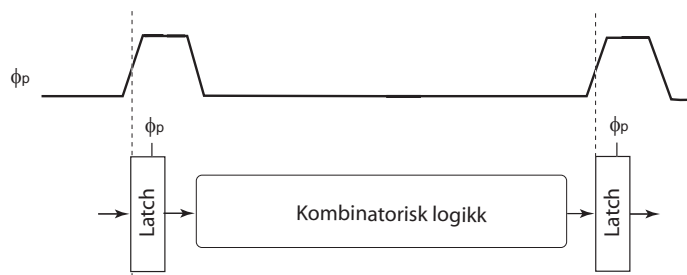


Figure 6.41: Latch som styres av klokkepuls.

En latch som styres av klokkepuls minner om en konvensjonell transparent latch. Et slikt system er avhengig av forholdsvis stor tidsforsinkelse i kombinatorisk logikk mellom latchene som vist i Fig. 6.41. For at to latch med kombinatorisk logikk mellom latchene ikke skal være transparent må det settes krav til tidsforsinkelse i den kombinatoriske logikken:

$$t_{cd} \geq t_{hold} - t_{ccq} + t_{pw}, \quad (6.17)$$

der t_{cd} er contamination (minimum) forsinkelse i kombinatorisk logikk, t_{hold} er hold tid for inngang fra negativ klokkeflanke, t_{ccq} er klokke til utgang contamination forsinkelse for latchen og t_{pw} er pulsbredden på klokkesignalet.

6.10.1 Puls generatorer

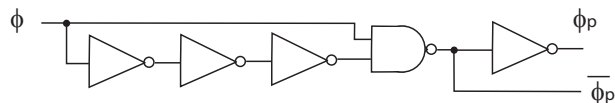


Figure 6.42: Enkel puls generator. (FIG7.22a)

Med utgangspunkt i et klokke signal ϕ med dutycycle lik 50% kan vi generere en klokke med pulser $\leq 50\%$ som vist i Fig. 6.42. I dette tilfellet blir invertert klokkesignal også generert.

6.10.3 Mål

Forstå hvordan latcher som styres av klokkepulser kan implementeres.

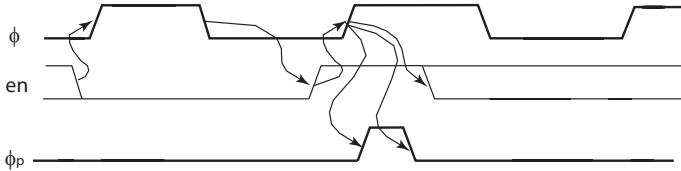
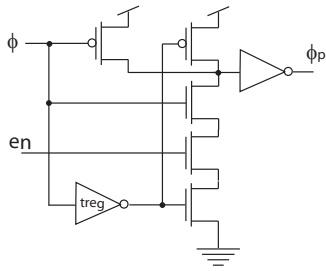


Figure 6.43: Puls generator. (FIG7.22b)

En annen puls generator er vist i Fig. 6.43. Denne puls generatoren genererer pulser med meget kort bredde.

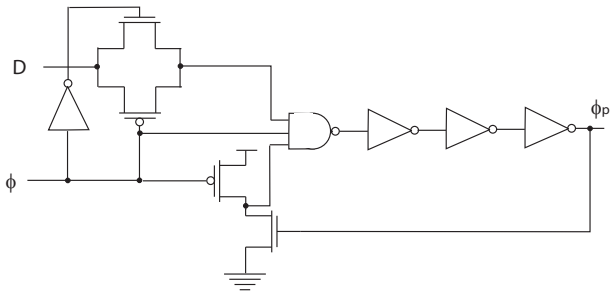


Figure 6.44: Puls generator. (FIG7.22d)

En tredje pulsgenerator med betydelig bredere pulser er vist i Fig. 6.44.

Ulike pulsgenerators med forskjellig pulsbredden passer til ulike spesielle latcher.

6.10.2 Puls latch

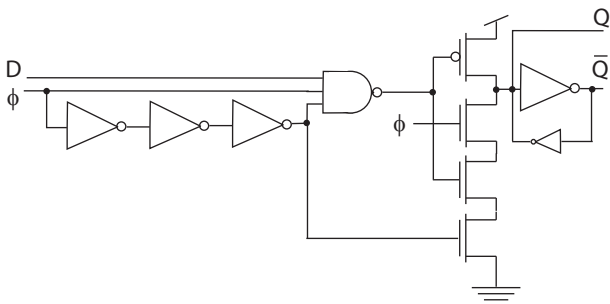


Figure 6.45: Partovi puls latch. (FIG7.23)

Et eksempel på en latch som er styrt av klokkepulser er vist i Fig. 6.45. Dette er en såkalt Partovi puls latch som har puls generatoren innebygd i selve latchen.

6.11 Latcher og vipper som kan resettes

(Kapittel 7.3.4 side 408 - 409)

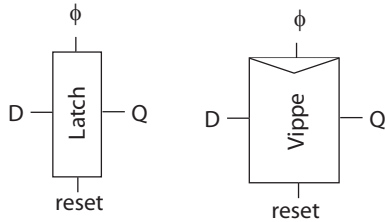


Figure 6.46: Symboler for latch og vippe med reset signal. (FIG7.24)

Det er praktisk å kunne benytte et reset signal slik at tilstanden til et sekvenseringselement er kjent ved oppstart. Symboler for latch og vippe med reset signal er vist i Fig. 6.46. Det er to typer av reset:

- *Synkron reset.* Synkrone reset signaler må være stabile før setup- og hold tid ved klokkeflanker.
- *Asynkron reset.* Asynkrone reset signaler resetter et element uavhengig av klokkesignaler.

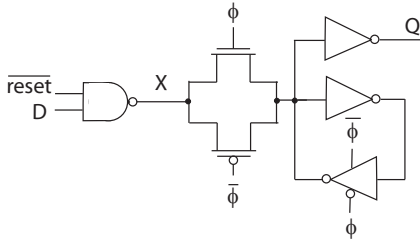


Figure 6.47: Synkron latch med reset signal. (FIG7.24)

En latch med synkron reset er vist i Fig. 6.47. Som kjent er ikke latches følsom for inngangen når $\phi = 0$. NAND porten på inngangen av latches vil slippe gjennom D når \overline{reset} er 1, vi har for en 2inngangs NAND port (NAND2) $X = \overline{D \cdot \overline{reset}}$ som gitt at $\overline{reset} = 1$ kan forenkles til $X = \overline{D}$. Når $\overline{reset} = 0$ kan uttrykket for NAND porten forenkles til $X = 1$. Når transmisjonsporten åpner for $\phi = 1$ vil latches sample inn enten \overline{D} eller 1. I det siste tilfellet skal latches resettes slik at utgangen $Q = 0$ uavhengig av D . Vi legger merke til at latches ikke resettes før $\phi = 1$.

En vippe med synkron reset er vist i Fig. 6.48. For inngangslatches i vippen gjelder samme argumentasjon som for synkron reset av latch, men der inngangslatches er klokke i motfase⁴.

En latch med asynkron reset er vist i Fig. 6.49. NAND porten på inngangen fungerer som beskrevet for latch med synkron reset, dvs. Q via Y og X settes til 0 når $reset = 1$

⁴Inngangslatches kan resettes når $\phi = 0$.

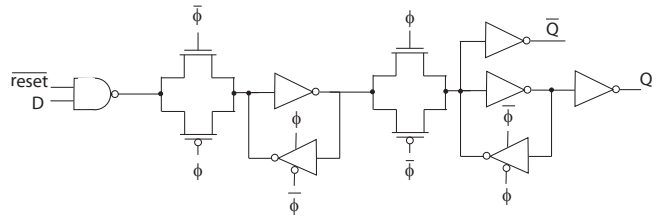


Figure 6.48: Synkron vippe med reset signal. (FIG7.24)

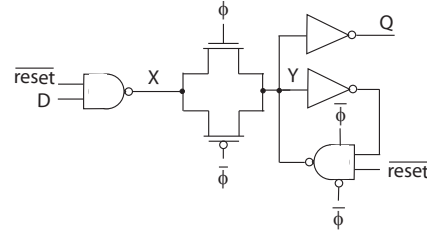


Figure 6.49: Asynkron latch med reset signal. (FIG7.24)

og $\phi = 1$. Latches blir da resatt via transmisjonsporten på inngangen. Det er i tillegg plassert en dynamisk NAND port i tilbakekoblingen slik at noden Y kan settes til 1, og dermed utgangen Q settes til 0 når $reset = 1$ og $\phi = 0$. Dette betyr at utgangen Q settes til 0 når $reset = 1$ uavhengig av D og ϕ .

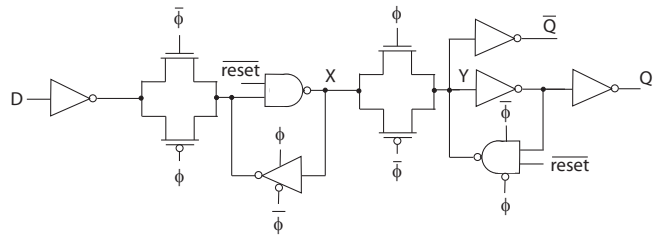


Figure 6.50: Asynkron vippe med reset signal. (FIG7.24)

En vippe med asynkron reset er vist i Fig. 6.50. Inngangslatches vil presse noden X til 1 når $reset = 1$ uavhengig av D , ϕ og (tidligere) verdi på X . Når $\phi = 0$ vil noden Y få verdien 1. Vi har da en situasjon der noden Y blir satt til 1 fra X via transmisjonsporten når $\phi = 1$ eller fra den dynamiske NAND porten når $\phi = 0$. Dette betyr at Y blir resatt til 1 uavhengig av ϕ og Q blir resatt til 1. Legg merke til at denne vippes resettes til 1 når $reset = 1$.

Vi kunne ha byttet ut NAND portene med NOR porter og \overline{reset} med $reset$, som vist i Fig. 6.51, slik at noden Y ble resatt til 0 for å få resatt utgangen Q til 0 når $reset = 1$.

6.11.1 Vippe med asynkron set og reset

I Fig. 6.52 er en vippe med asynkron set og reset vist. Kretsen benytter to signaler set og $reset$ til å sette vippes i to ulike tilstander. Inngangslatches har set signal som styrer NAND porten som setter noden X til 1 når $set = 1$. For utgangslatches vil set signalet sette noden Y lik 1.

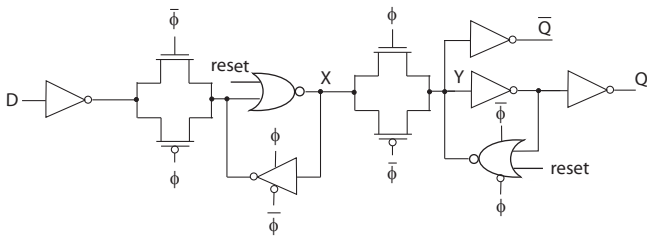


Figure 6.51: *Asynkron vippe med reset signal.*

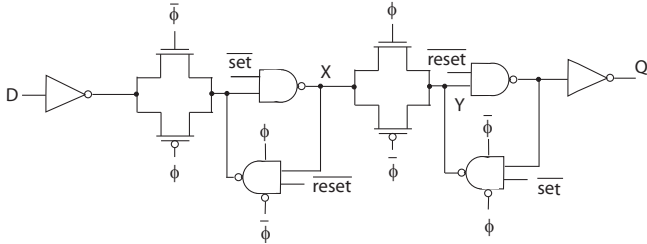


Figure 6.52: *Vippe med asynkron reset og set signal.*

\overline{set} signalet setter utgangen på C^2MOS NAND porten i tilbakekoblingen i inngangslatchen til 1 når $reset = 1$ samtidig som NAND porten i utgangslatchen setter inngangen til inverteren til 1 og dermed utgangen Q til 0.

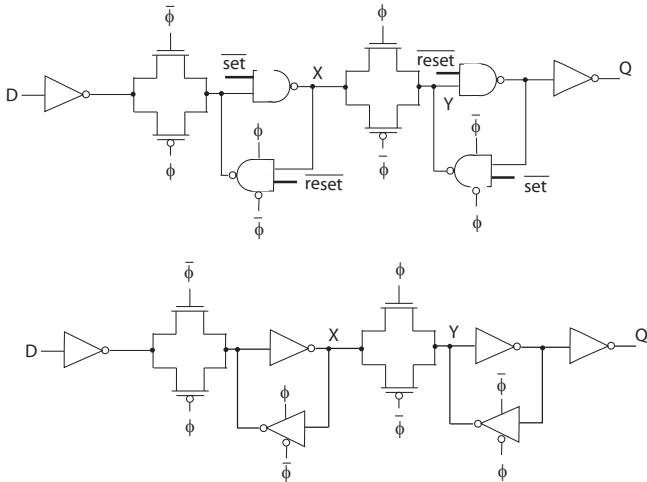


Figure 6.53: *Vippe med asynkron reset og set signal. Set = reset = 0*

Vippen med $set = reset = 0$ er vist øverst i Fig. 6.53. For alle NAND portene vil det være en av inngangene som er 1, dvs. $\overline{set} = \overline{reset} = 1$. Forenklet port og logisk ekvivalent, men ikke elektrisk, ekvivalent, er en inverter som vist i den nederste kretsen i Fig. 6.53.

Vippen i reset funksjon er vist i Fig. 6.54 øverst. I dette tilfellet forutsetter vi at det andre kontrollsignalet $set = 0$. For inngangslatchen vil da utgangen på C^2MOS NAND porten i tilbakekoblingen bli satt til 1 slik at den andre NAND porten i inngangslatchen vil sette X til 0. Dette betyr at inngangslatchen vil bli resatt til 0 som er tilsvarende som om vi samplet inn 0 fra inngangen. For utgangslatchen vil NAND porten med \overline{reset} som inngang

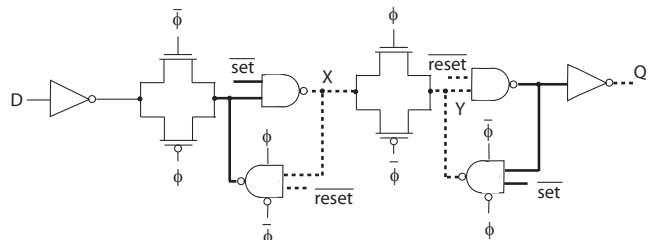


Figure 6.54: *Vippe med asynkron reset og set signal. Reset = 1*

sette inngangen til inverteren til 1 og dermed utgangen Q til 0. C^2MOS NAND porten i tilbakekoblingen i utgangslatchen vil sørge for at Y blir lik 0 (som er samme verdi som X). Dersom kontrollsignalet $reset$ settes til 0 etter at kretsen er korrekt resatt vil vippen være i tilstanden vist i Fig. 6.54 nederst⁵ inntil vippen eventuelt sampler inn en ny verdi $D = 1$ når $\phi = 0$, eller vippen settes til 1 ved hjelp av kontrollsignalet set .

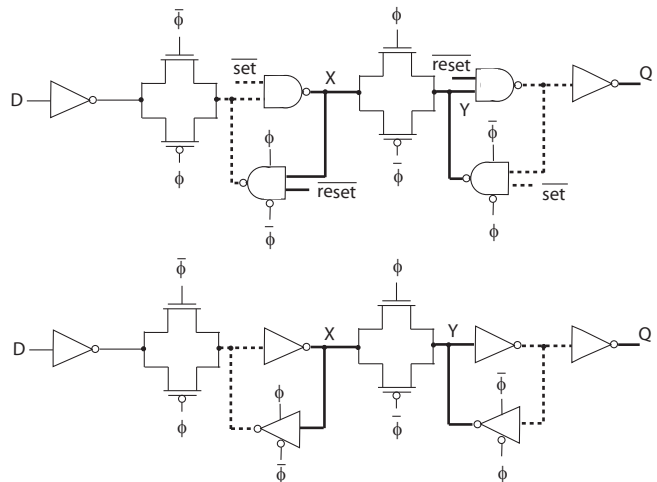


Figure 6.55: *Vippe med asynkron reset og set signal. Set = 1*

Vippen i set funksjon er vist i Fig. 6.55 øverst. I dette tilfellet forutsetter vi at det andre kontrollsignalet $reset = 0$. Inngangslatchen vil sette noden X til 1 som igjen vil sette utgangen til C^2MOS NAND porten i tilbakekoblingen til 0. Denne verdien vil holde seg lik 0 gjennom forenklet logisk ekvivalent vist for inngangslatchen i Fig. 6.55 nederst. For utgangslatchen vil C^2MOS NAND porten i tilbakekoblingen sørge for at $Y = 1$ som vil sette inngangen til utgangsinverteren til 0 og dermed blir utgangen Q lik 1. Vippen nederst i figuren er logisk ekvivalent inntil

⁵Kretsen nederst er logisk ekvivalent når $set = reset = 0$, men ikke elektrisk ekvivalent.

inngangslatchen samler inn ny verdi $D = 0$ når $\phi = 0$, eller vippen resettes ved hjelp av kontrollsignalet *reset*.

6.11.2 Detaljer

Vi har nå forutsatt at vippen kan ha tre ulike modi:

- **Vippe.** $set = reset = 0$. Vippen fungerer som en vanlig vippe som vist i Fig. 6.53 nederst.
- **Resett til 0.** $reset = 1$ og $set = 0$. Vippen resettes til 0, dvs. både utgangen og noden X resettes til 0. Når reset signalet endres til 0 vil kretsen operere som kretsenekvivalenten vist nederst i Fig. 6.54.
- **Sett til 1.** $set = 1$ og $reset = 0$. Vippen resettes til 1, dvs. både utgangen og noden X resettes til 1. Når reset signalet endres til 0 vil kretsen operere som kretsekvivalenten vist nederst i Fig. 6.55.

Det er en kombinasjon av kontrollsignalene som vi ikke har vurdert. Dersom vi antar at vippen har kontrollsignalene $reset = set = 1$ har vi en situasjon som ikke kan tillates. Vippen skal i denne situasjonen både settes til 1 og 0 som er selvmotsigende og meningsløst. For ordens skyld kan det være fornuftig å analysere vippen for å se hva som skjer dersom vi ved en feil påtrykker denne uvelogelige kombinasjonen av kontrollsignaler.

I Fig. 6.56 øverst er det vist hvordan vippen virker når $set = reset = 1$. Vi ser at utgangen Q blir resatt til 0 som i utgangspunktet ligner en vanlig reset. Legg merke til at nodene X og Y blir satt til 1 samtidig. Dette samsvarer ikke med en vanlig reset. Vippens tilstand før eventuell ny sampling av inngangen er avhengig av hvilke av de to kontrollsignalene som skrues av først. Dersom $reset$ blir satt til 0 mens $set = 1$, som vist nest øverst i figuren, vil kretsen oppføre seg som om den ble satt til 1 slik at utgangen Q blir satt til 1. Dersom set blir satt til 0 mens $reset = 1$, som vist nest nederst i figuren, vil kretsen oppføre seg som om den ble satt til 0 slik at utgangen Q forblir 0. I den nederste vippen er det antatt at set og $reset$ endres fra 1 til 0 samtidig. Situasjonen vil da være ukjent, dvs. vi kan ikke forutsi om vippen blir satt til 0 eller 1.

6.11.3 Mål

Kunne implementere latcher og vipper med synkron eller asynkron reset.

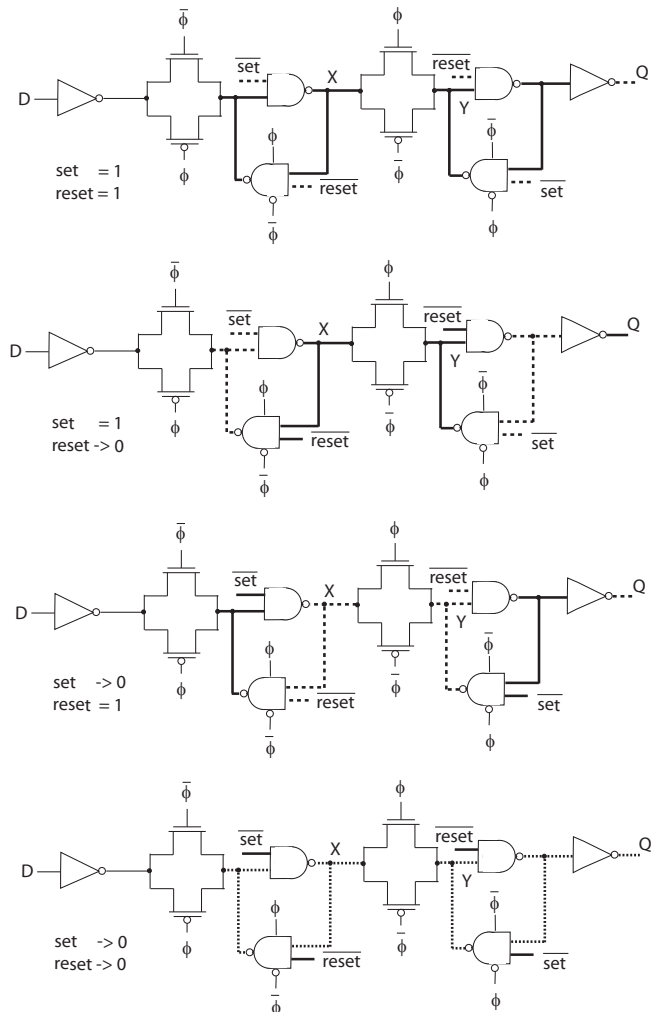


Figure 6.56: Vippe med asynkron reset og set signal. $Set = Reset = 1$

6.12 Latcher og vipper som kan enables

(Kapittel 7.3.5 side 410)

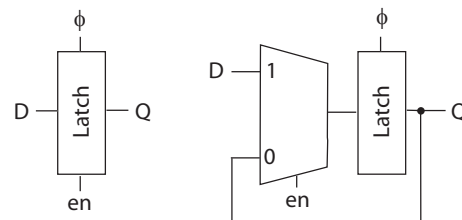


Figure 6.57: Latch med enable realisert med multiplekser. (FIG7.26)

I mange tilfeller kan det være hensiktsmessig å kombinere en enable funksjon i latcher og vipper. En latch med *enable* signal er vist i Fig. 6.57 der en latch er kombinert med en multiplekser. Kretsen har en forholdsvis lang signalvei i tilbakekoblingen via multiplekser som vist i Fig. 6.58 når $en = 0$ og $\phi = 1$. I tillegg til å bidra

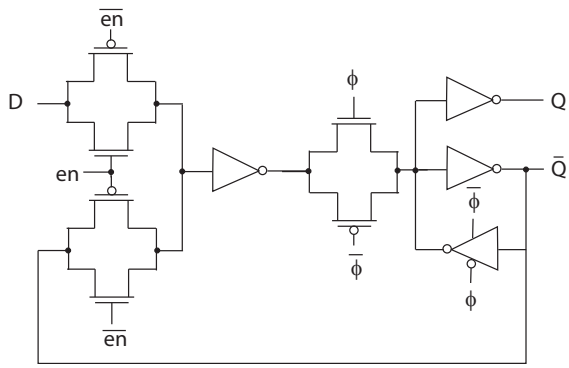


Figure 6.58: Latch med enable realisert med multiplekserer.

med forsinkelse vil multiplekseren bidra med en betydelig arealøkning.

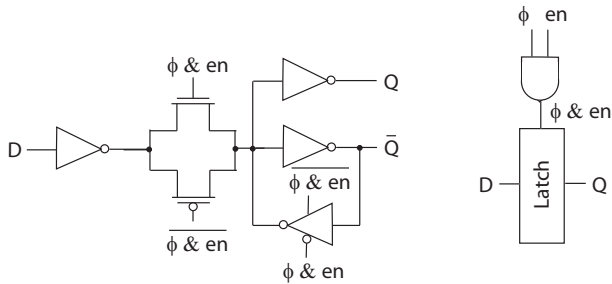


Figure 6.59: Latch med enable funksjon realisert med "clock gating design". (FIG7.26)

I Fig. 6.59 er det vist en logisk ekvivalent latch med enable der vi har beholdt den opprinnelige latchen og endret de lokale styresignalene. Dette kalles *clock gating design*. Ved å ANDE *en* og ϕ til $en \& \phi$ vil kretsen bare sample når både ϕ og *en* er logisk 1, ellers vil den lokale tilbakekobling i latchen sørge for å holde den lagrede verdien.

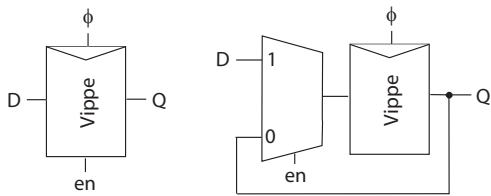


Figure 6.60: Vippe med enable realisert med multiplekserer. (FIG7.26)

En vippe med enable signal er vist i Fig. 6.60 der en vippe er kombinert med en multiplekser.

En tilsvarende forenkling som for latchen med enable i Fig. 6.59 er vist for vippen i Fig. 6.61.

AND porten som er benyttet for å endre styresignaler til latchen og vippen kan deles av mange sekvenseringselementer og vil derfor ikke bidra med betydelig areal.

6.12.1 Mål

Forstå hvordan man kan implementere latcher og vipper med enable signal.

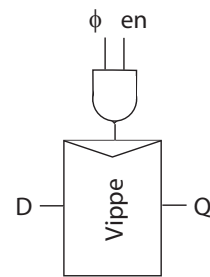


Figure 6.61: Vippe med enable funksjon realisert med "clock gating design". (FIG7.26)

6.13 Latcher med logikk

(Kapittel 7.3.6 side 410 - 411)

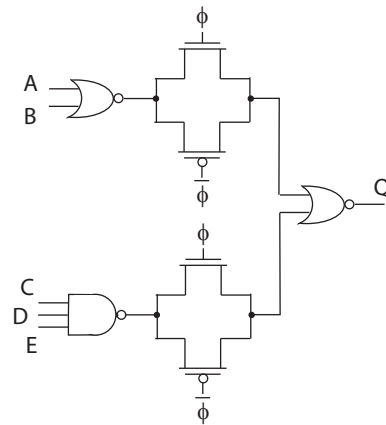


Figure 6.62: Latch med logikk. (FIG7.27)

Latchene kan lett bygges ut til å prosessere signaler. Et eksempel på en latch med logikk er vist i Fig. 6.62 der $Q = (A + B) \cdot C \cdot D \cdot E$. Latchen mangler tilbakekobling og er derfor dynamisk.

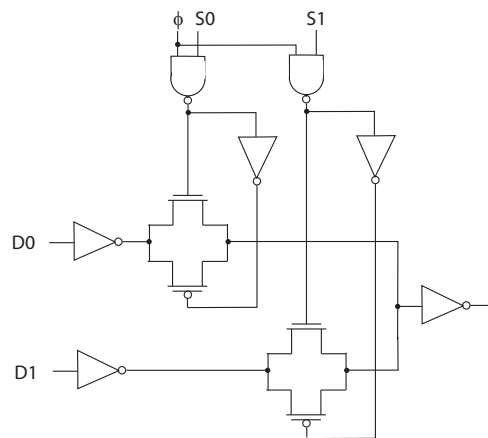


Figure 6.63: Latch med logikk og clock gating. (FIG7.27)

En multiplekser latch er vist i Fig. 6.63 der lokale kontrollsignaler blir styrt av $S0$ og $S1$. Denne latchen er også dynamisk.

Statiske latcher og vipper kan også utvides med logikk på samme måte.

6.13.1 Mål

Forstå hvordan logikk kan inkluderes i statiske- og dynamiske latcher og vipper.

6.14 Klass semidynamisk vippe (SDFF)

(Kapittel 7.3.7 side 411 - 412)

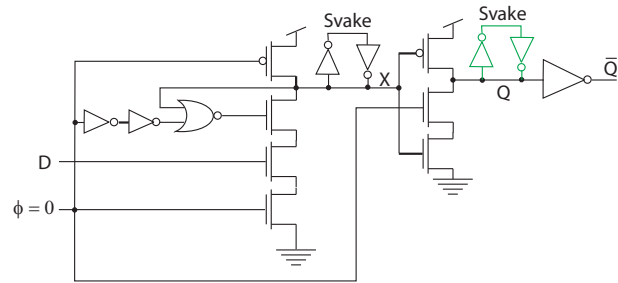


Figure 6.64: *Klass semidynamisk vippe. (FIG7.28)*

En *Klass semidynamisk vippe* er vist i Fig. 6.64. Latchen er en krysning mellom latch styrt av klokkepuls og en vippe.

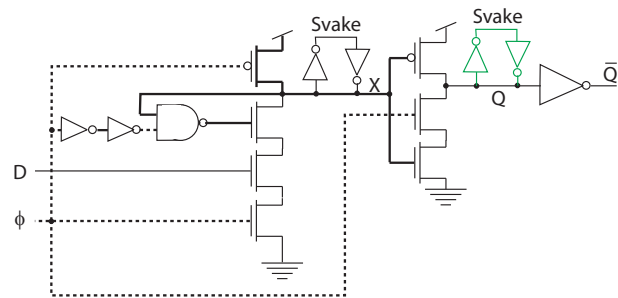


Figure 6.65: *Klass semidynamisk vippe der $\phi = 0$. (FIG7.28)*

I Fig. 6.65 er vippen vist når $\phi = 0$. Noden X vil da precharges til 1 og ikke kunne påvirke utgangene Q og \bar{Q} . Ved noden Q er det koblet to svake invertere som sørger for å holde verdien på Q og dermed motvirke lekkasje og ladningsdeling.

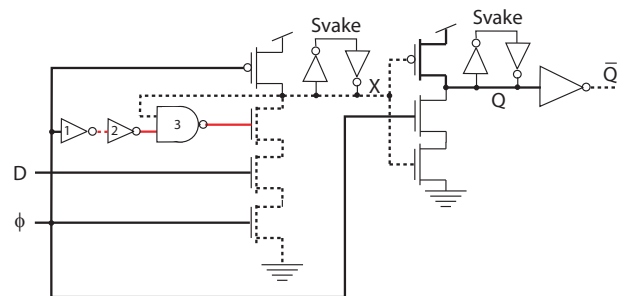


Figure 6.66: *Klass semidynamisk vippe der $\phi = 1$. (FIG7.28)*

I Fig. 6.66 er vippen vist når $\phi = 1$. Vi antar nå at kretsen skal latche (sample) inn en ny verdi fra inngangen D . Vi ser at precharge transistoren koblet til X er slik at noden X enten vil holde sin verdi eller trekkes ned til 0. I eksemplet vist i Fig. 6.66 er $D = 1$. Vi antar nå at utgangen på NAND porten er 1 som vist i Fig. 6.66 rett før ϕ skifter fra 0 til 1. Det som skjer rett etter denne transisjonen er at noden X trekkes ned til 0 før utgangen

6.15 Differensielle vipper

(Kapittel 7.3.8 side 412 - 413)

på NAND porten rekker å reagere på endring av ϕ . Når X er trukket ned til 0 vil tilbakekoblingen via NAND porten sørge for å holde utgangen på NAND porten til 1. Den andre inngangen til NAND porten kommer fra ϕ via to invertere, 1 og 2. Vi må forutsette at ikke dette signalet rekker å skifte fra 0 til før X er trukket ned til 0. Noden Q vil nå bli trukket opp til 1 via en precharge transistor koblet til noden. Utgangen \bar{Q} blir da satt til 0. Vi har nå en situasjon der $\bar{Q} = \overline{D}$ som jo er vippens korrekte funksjon.

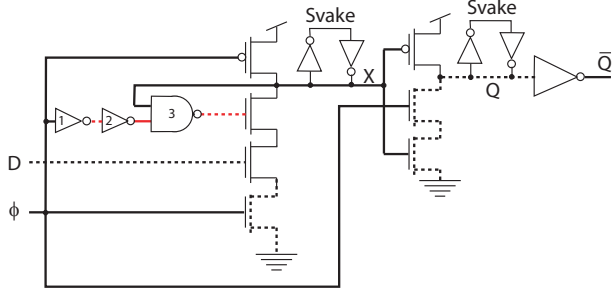


Figure 6.67: *Klass semidynamisk vippe. (FIG7.28)*

I Fig. 6.67 er vippen vist når $\phi = 1$ og $D = 0$. I denne situasjonen er nedtrekkskjeden koblet til X skrudd av slik at X forblir 1. Noden Q vil da trekkes ned via to nMOS transistorer styrt av X og ϕ slik at utgangen $\bar{Q} = 1$. Dette medfører også en situasjon der $\bar{Q} = \overline{D}$ som jo er vippens korrekte funksjon.

6.14.1 Mål

Kunne implementere en Klass semidynamisk vippe.

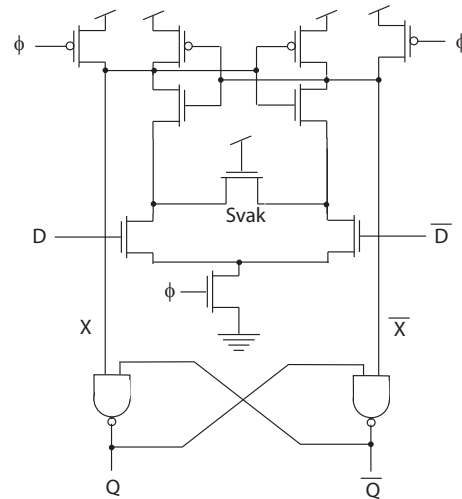


Figure 6.68: *Differensiell sense-amplifier vippe. (FIG7.29a)*

En *differensiell* vippe er vist i Fig. 6.68. Vippen er basert på en såkalt *sense amplifier* som består at et inngangstrinn med D og \bar{D} med en felles transistor med ϕ inngang ned mot GND. De to NAND portene brukes til å holde utgangene stabile.

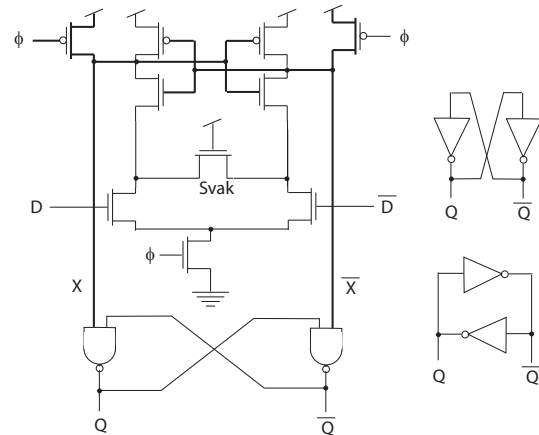


Figure 6.69: *Differensiell sense-amplifier vippe med $\phi = 0$. (FIG7.29a)*

Den differensielle vippen med $\phi = 0$ er vist i Fig. 6.69. Når $\phi = 0$ vil nodene X og \bar{X} precharges til 1 slik at de to NAND portene kan forenkles logisk som vist til høyre for vippen. De to kretsene med krysskoblede invertere er identiske og tilsvarer utgangslatchen på en vanlig vippe.

Den differensielle vippen med $\phi = 1$ er vist i Fig. 6.70. Vippen skal nå sample inn ny verdi. Som vi ser er vippen fullstendig symmetrisk, vi ser derfor på eksemplet der $D = 1$ som vist i figuren. Noden X blir trukket ned til 0 og dermed \bar{X} opp til 1. Utgangen $Q = D$ som vil holdes til sampling ved neste positive klokkeklanke. Det er en viss tidsforsinkelse gjennom de to NAND portene på utgangene.

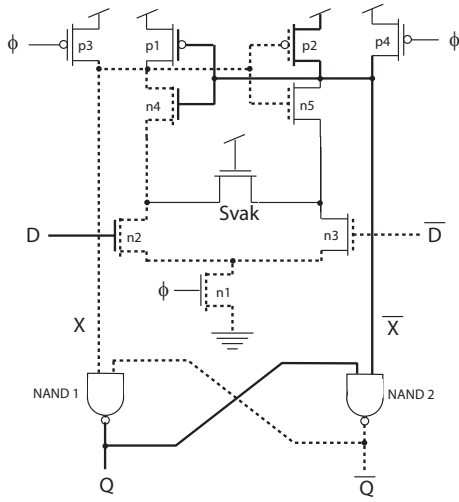


Figure 6.70: *Differensiell sense-amplifier vippe $\phi = 1$. (FIG7.29a)*

Den differensielle vippen i Fig. 6.71 har en raskere respons enn vippen med krysskoblede NAND porter. Utgangen blir trukket opp til 1 via en pMOS transistor direkte fra X . De to krysskoblede portene bidrar til å holde verdien i vippen.

6.15.1 Mål

Kunne implementere differensielle vipper.

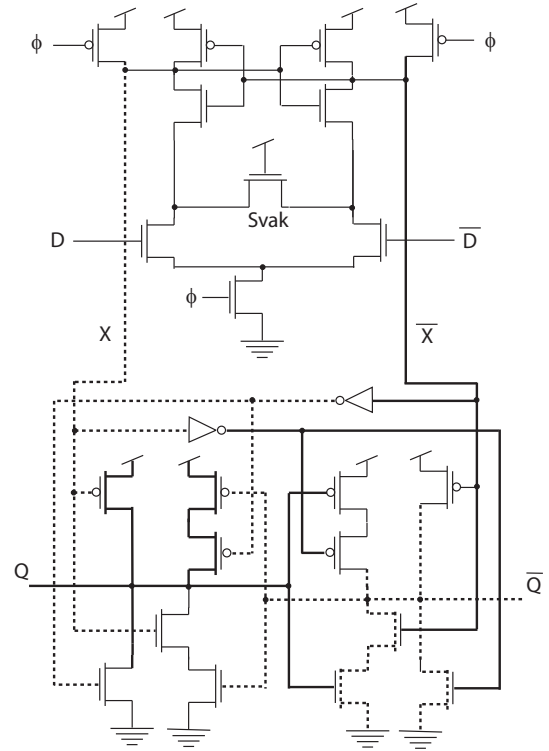


Figure 6.71: *Differensiell sense-amplifier vippe. (FIG7.29b)*

6.16 Ekte en-fase (TSPC) latcher og vipper

(Kapittel 7.3.9 side 414)

Vanlige latcher og vipper benytter i tillegg til klokkesignal også invertert klokkesignal. I moderne CMOS blir typisk invertert klokkesignal generert lokalt ved latchene eller vippene.

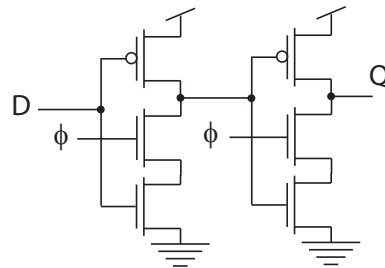


Figure 6.72: *Ekte en-fase latch. (FIG7.30a)*

En latch som kun benytter ett klokkesignal er vist i Fig. 6.72. Vi kaller dette for en *ekte en-fase latch*.

En ekte en-fase latch med $\phi = 0$ er vist i Fig. 6.73. Latchen skal holde utgangen stabil så lenge $\phi = 0$. Vi ser at nedtrekkene er skrudd AV ved hjelp av ϕ . I utgangspunktet har vi to mulige tilstander; Q var 1 før ϕ skiftet fra 1 til 0 (som vist på venstre side) og Q var 0 opprinnelig (som vist på høyre side). En forutsetning for at $Q = 1$ er at $X = 0$ som vist til venstre. Når nedtrekket koblet til utgangen ikke kan trekkes ned pga. ϕ vil ikke kretsen kunne endre utgangen så lenge $\phi = 0$. Legg merke

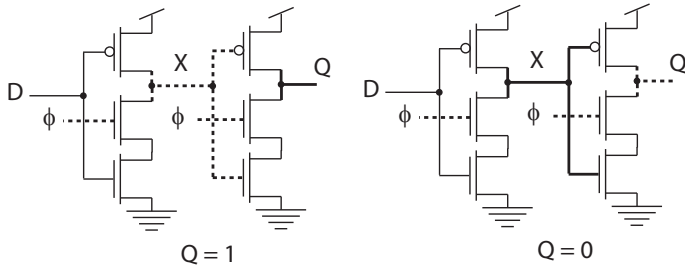


Figure 6.73: *Ekte en-fase latch med $\phi = 0$. (FIG7.30a)*

til at latchen er dynamisk slik at lekkasje kan påvirke utgangssignalet etter en viss tid. Utgangen holdes høy ved hjelp av en pMOS transistor som er skrudd på forsi $X = 0$. Noden X er ikke drevet og kan endres som følge av lekkasje og dermed påvirke utgangen Q .

Til høyre er en tilstand der $Q = 0$ og $X = 1$. I dette tilfellet er hverken Q eller X drevet og derfor utsatt for lekkasje.

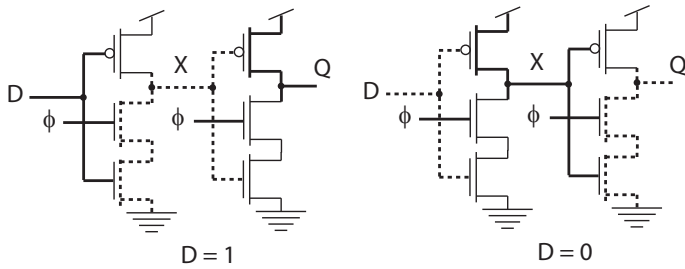


Figure 6.74: *Ekte en-fase latch med $\phi = 1$. (FIG7.30a)*

Ved latching av ny verdi (sampling) er $\phi = 1$ som vist i Fig. 6.74. I denne situasjonen kan vi (logisk) se bort i fra transistorene styrt av klokke signalet ϕ . Kretsen vil da logisk være to invertere i serie slik at vi alltid får $Q = D$.

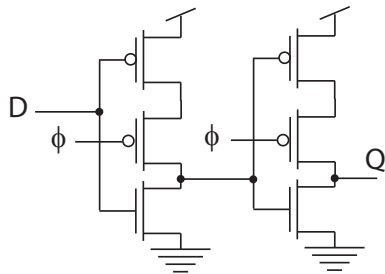


Figure 6.75: *Ekte en-fase latch. (FIG7.30b)*

En latch som er følsom for motsatt klokkenivå er vist i Fig. 6.75. Isteden for å bruke samme latch med invertert klokke signal erstatter vi de klokkestyrte nMOS transistorene med pMOS transistorene og flytter utgangene mellom pMOS og nMOS transistorer.

En ekte en-fase vippe er vist i Fig. 6.76. Denne vippen er enda enklere en to en-fase latcher klokket i motfase.

En-fase vippen nåt $\phi = 0$ og $\phi = 1$ er vist i henholdsvis Fig. 6.77 og 6.78.

En-fase latcher og vippe som er beskrevet i dette avsnittet er dynamiske.

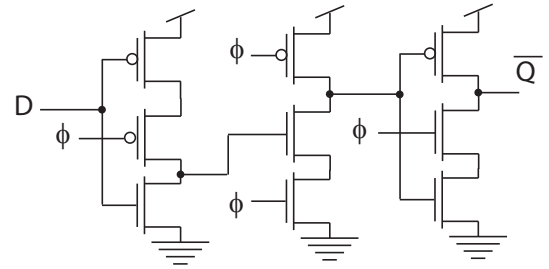


Figure 6.76: *Ekte en-fase vippe. (FIG7.30c)*

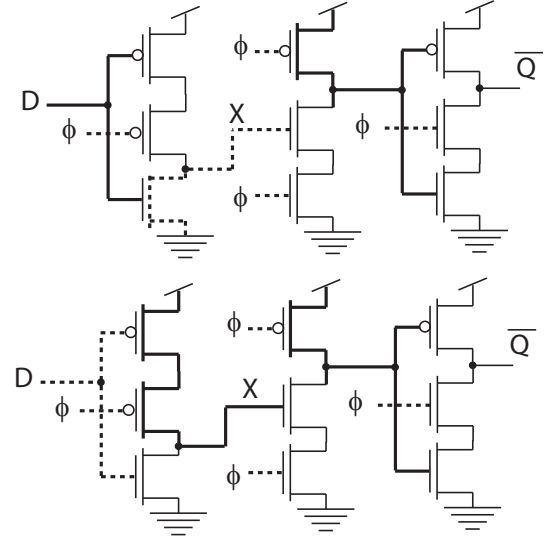


Figure 6.77: *Ekte en-fase vippe når $\phi = 0$. (FIG7.30c)*

6.16.1 Mål

Kunne implementere ekte en-fase latcher og vippe.

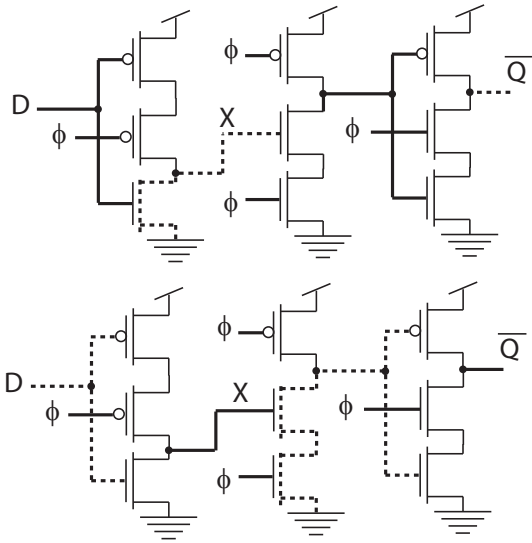


Figure 6.78: Ekte en-fase vippe når $\phi = 1$. (FIG7.30c)

Chapter 7

Indeks

- ϵ_{ox} 27
- γ (body effekt parameter) 31
- Φ_s (overflate potensial) 31
- Ψ_0 (Innebygd potensial) 60
- π modell 88
- Si_3N_4 55
- SiO_2 47, 52
- τ 76
- Akkumulasjon 24
- Aktiv 56
- Aktivitetsfaktor 111
- Asymmetrisk port108
- Asynkron reset 165
- Avsetning 52
- Avsetning av atom lag (ALD) 52
- Avstand mellom ledere 90
- B (forgreningseffort i kjede) 77
- b (forgreningseffort) 77
- Back-gate 140
- Bløder 128
- Body effekt (substrateffekt) 31
- Body effekt parameter (γ) 31
- Boltzmanns konstant k 60
- Bootstrapping 75
- Bredde på ledere 90
- Bubble pushing 105
- Bulk 23
- C_0 58
- C_{Jsw} 60
- $C_{ekstern}$ 68
- C_g 27, 57
- C_{gb} 58
- C_{gd} 58
- C_{gd0} 58
- C_{gs} 58
- C_{gs0} 58
- C_{intern} 68
- C_j 60
- C_{jbs} 60
- C_{jbsw} 60
- C_{last} 63
- C_{ox} 28
- C_{sb} 60
- C^2MOS 160
- CMOS 23
- CMOS inverter 24, 47
- CMOSTG 131
- CNTL 135
- Contamination forsinkelse 152
- CPL 132
- Crosstalk 89
- Crosstalk støy 89
- Cutoff 27
- CVSL 119, 128
- D (forsinkelse i kjede) 77
- d (normalisert tidsforsinkelse) 69
- D_F 77
- D-vippe 20
- DCSL 146
- DCVSPG 135
- Deplesjon 25
- Deplesjonskoeffisient M_j 60
- Designhjørner 93
- Dielektrikum 51, 86
- Dielektrisk konstant k 86
- Differensiell kaskode spenning svitsj logikk med pass transistor logikk (DCVSPG) 135
- Differensiell logikk 135
- Differensiell split-level (DSL) 135
- Differensiell strøm svitsj logikk (DCSL) 146
- Differensiell vippe 170
- Diffusjon 23
- Diffusjonskapasitans 60
- Diffusjonskapasitans på source C_{sb} 60
- Diode 23
- Dobbel pass transistor logic (DPL) 134
- Domino logikk 124
- Dopakonsentrasjon 60
- Dopakonsentrasjon N_A 32
- Dopakonsentrasjon N_D 60
- Doping 23
- DPL 134
- Drain 12
- Drain-induced barrier lowering (DIBL) 32
- DSL 135
- Dual-rail domino logikk 125, 142
- Dynamisk effektforbruk 82, 111
- Dynamisk latch 160
- Dynamiske kretser 150
- Dynamiske porter 120
- Dynamiske sekvensielle kretser 150
- ECDL 143

EEPL 134
 Effektforbruk 82
 Effektiv motstand $R_{effektiv}$ 62
 Effort tidsforsinkelse 68, 69
 Egenmotstand 84
 Ekstern kapasitans C_{extern} 68
 Ekstern last $C_{ekstern}$ 68
 Ekte en-fase latch 171
 Ekvivalent motstand 63
 Elektrisk effort (h) 68
 Elektronmigrasjon 95
 Elmore forsinkelsesmodell 67
 Enable/disable CMOS differensiell logikk (ECDL) 143
 Energi effektiv pass transistor logikk (EEPL) 134
 Epitaksi 52
 Evaluering 121
 F (kjede effort) 77
 f (port effort) 69
 Fanout 68
 Falltid t_f 62
 Felt oksid 53
 FGPA 161
 Field programmable gate array 161
 Flankefølsom vippe 20
 Footed porter 121
 Forgreningseffort b 77
 Forsterkning 35
 Fotolitografi 47
 Fotoresist 47
 G (logisk effort i kjede) 76
 g (logisk effort) 69, 69
 g_d 110
 g_u 110
 Ganged CMOS 117
 Gate 12
 Gate bulk kapasitans C_{gb} 58
 Gate drain kapasitans C_{gd} 58
 Gate kapasitans C_g 27, 57
 Gate lekkasje 33
 Gate source kapasitans C_{gs} 58
 Gate strøm 33
 Gjennomsnittelig tid mellom feil 94
 Gjennomsnittstid t_{rf} 62
 H (elektrisk effort i kjede) 76
 h (elektrisk effort) 68
 Halvleder 23
 Harde feil 94
 Hastighetsmetning 64
 HI-skew 110
 High-skew 129
 Hold tid 152
 Hold tid feil 155
 Horisontalt utleggspor 49
 Hot spots 139
 Hukommelse 150
 Hull 23
 Høy-k dielektrikum 86
 Implantering 52
 Indre inngang 109
 Injeksjon av minoritetsbærere 139
 Innebygd potensial Ψ_0 60
 Inngangsterskel 37
 Interkonnekt 84
 Intern kapasitans C_{intern} 68
 Intern last C_{intern} 68
 Intrisikk ubiaserte gatekapasitans C_0 58
 Inversjon 25
 Inverter 13
 I-V karakteristikk 27
 k 86
 Kanal 25
 Kanal stop 53
 Kanalforkortning 30
 Kanallengdemodulasjon 30
 Kanalmotstand 28
 Kantfølsom 150, 156
 Kantfølsom vippe 151
 Kapasitans i ledere 86 Kaskode nonthreshold logikk (CNTL) 135
 Kaskode spenning svitsj logikk 119
 Keeper 128
 Kjede effort F 77
 Kjede effort tidsforsinkelse D_F 77
 Kjede elektrisk effort H 76
 Kjede forgreningseffort B 77
 Kjede logisk effort G 76
 Kjede parasittisk tidsforsinkelse P 77
 Kjedeforsinkelse $D77$
 Klass semidynamisk vippe 169
 Klokkesignaler 120
 Klokket CMOS (C^2MOS) 160
 Kombinatorisk logikk 14, 151
 Komplementær logikk 16
 Komplementær pass transistor logikk (CPL) 132
 Konduktans 43
 Konstant felt skalering 98
 Kortslutningseffekt 111
 Kritisk signalvei 154
 L modell 88
 Ladningsbærere 23
 Ladningsdeling 126
 Ladningslekkasje 128
 Lag i en krets 47
 Lastkapasitans C_{last} 63
 Lastmotstand 39
 Laststrøm 39
 Latch 19, 160
 Latch med asynkron reset 165
 Latch med enable 167
 Latch med synkron reset
 Latched CMOS differensiell logikk (LCDL) 145
 Latcher 150, 150
 Latchup 96
 Lateral diffusjon 58, 34
 Lateral Skalering 98
 Lav-k dielektrikum 86
 LCDL 145
 Lean integration med pass transistorer (LEAP) 133

LEAP 133
 Lett dopet drain 55
 Lineært område (transistor) 25
 Logisk effort 68, 69, 69
 Logisk nedtrekkseffort g_d 110
 Logisk opptrekkseffort g_u 110
 Lokal interkonnekt 55
 LOCOS 53
 LO-skew 110
 M_{JSW} 60
 M_j 60
 Majoritetsbærere 24
 Maske for n-brønn 47
 MCF 89
 Metning (transistor) 26
 Metningsspenningen for hastighetsmetning V_{sat} 64
 Miller effekt 74, 89
 Miller koblingsfaktor (MCF) 89
 Minimum-forsinkelse feil 155
 Minoritetsbærere 24
 Mismatch 94
 Mixed-mode 135
 Mixed-signal 135
 Moderat inversjon 32
 Moores lov 98
 MOS transistor 23
 MOSFET 23
 Multiplekser 18
 N_A 32, 60
 N_D 60
 n_i 32, 60
 n-brønn prosess 50
 n-diffusjon 47
 n-select 56
 n-type 23
 NAND 14
 Nedtrekk 14
 Negativ bias temperatur instabilitet (NBTI) 96
 Negativ tilbakekobling 108
 Nivåfølsom 150, 156
 Nivåfølsom latch 19
 nMOS transistor 12, 23
 Nonthreshold logikk (NTL) 135
 NOR 16
 Nora domino 131
 Normalisert forsinkelse 69
 Normalisert tidsforsinkelse 69
 NP domino 131
 NTL 135
 Oksidkapasitans 28
 Opptrekk 14
 Overflate potensial Φ_s 31
 Overglass 56
 Overlappskapasitans 58
 Overlappskapasitans gate drain C_{gd0} 58
 Overlappskapasitans gate source C_{gs0} 58
 P (parasittisk tidsforsinkelse i kjede) 77
 p (parasittisk tidsforsinkelse) 67, 68, 70
 p-brønn prosess 50
 p-select 56
 p-type 23
 Parasittisk kapasitans 68
 Parasittisk tidsforsinkelse t_{pd} , p 67, 68, 70
 Parallellkobling 15
 Pass transistor 17
 Passivering 56
 Permittivitetet 27
 Pipeline systemer 155
 Pitch 90
 pMOS transistor 12, 23
 pn overgang 23
 Poly 54
 Port effort f 69
 PPL 134
 Precharge 121
 Propageringsforsinkelse t_{Pd} 71, 152
 Pseudo nMOS 40, 112, 128
 Push-pull pass transistor logikk (PPL) 134
 $R_{effektiv}$ 62
 Race feil 155
 RC ekvivalent 61
 RC modeller 61
 Referanse tidsforsinkelse τ 76
 Ruting 84
 Sekvensielle kretser 150
 Sense amplifiser 170
 Sense amplifiser kretser 142
 Seriekobling 15
 Setup tid 152
 SFPL 118
 Shallow trench isolation
 Shielding 92
 Silisiumdioksid SiO_2 27, 47, 52
 Silisium nitrid (Si_3N_4) 55
 SiO_2 27
 Skalering med konstant spenning 98
 Skew 159
 Skew tolerant 159
 Slope faktor 32
 Source 12
 Source følger opptrekklogikk 118
 SRPL 134
 Standard celle 48
 Statisk effektforbruk 82, 83
 Statisk latch 160
 Statistiske porter 120
 Statistiske sekvensielle kretser 150
 Sterk inversjon 32
 Stick diagram 49
 Stigetid t_r 62
 Strømretning 35
 Støymargin 38
 Substrat 23
 Substrateffekt (body effekt) 31
 Subterskel 32
 Svak inversjon 32
 Sving-restored pass transistor logikk (SRPL) 134
 Synkron reset 165 t_b 71

t_{cd} (tidsforsinkelse) 62
 t_f (falltid) 62
 t_{ox} (tynnoksid) 27
 t_{Pd} (propageringsforsinkelse) 71
 t_{pd} (parasittisk tidsforsinkelse) 67, 68, 70
 t_{pdf} 71
 t_{pdr} 72
 t_r (stigetid) 62
 t_{rf} 62
T modell 88
Termisk spenning V_T 32, 60
Terskefall 41
Terskefeil 137
Terskelspenning V_t 25
Tidsforsinkelse 61, 62
Tilstand 150
Transfer karakteristikk 37
Transistor karakteristikk 27
Transistorlengde 28
Transmisjonsport 17, 42
Triode 28
Triple-brønn prosess 50
Tristate 18
Tristate buffer 18
Tristate inverter 27, 44
Tristate porter 120
Tunneleringsstrøm 33
Twinn-brønn prosess 50
Tynnoksid 27
Tørroksidering 52
Ulik svitsjetidspunkt for innganger t_b 71
Utleggsregler 46
 V_T (termisk spenning) 32, 60
 V_t (terskelspenning) 25
V-V (DC) karakteristikk 35
Varmeutvikling (self-heating) 95
Vippe 20
Vippe med asynkron reset 165
Vippe med enable 168
Vippe med synkron reset 165
Vipper 150, 150
Våtoksidering 52
Wafer 47, 51
Wear out 94
Yield 56
Ytre inngang 109
Zipper domino 131

Bibliography

- [1] Neil H.E. Weste og David Harris “CMOS VLSI DESIGN, A circuit and system perspective” tredje utgave 2005, ISBN: 0-321-26977-2, *Addison Wesley*,
- [2] Yngvar Berg “Andre pass transistor familier, INF3400 Del:12 kapittel V”
- [3] Yngvar Berg “Lean integration med pass transistorer (LEAP), INF3400 Del:12 kapittel IV”
- [4] Yngvar Berg “Svak inversjon, INF3400 Del:8 kapittel II”
- [5] Yngvar Berg “Ladningsdeling, INF3400 Del:9 kapittel V”
- [6] Yngvar Berg “Temperaturavhengighet for MOS transistorer, INF3400 Del:14 kapittel XI”
- [7] Yngvar Berg “Latchup, INF3400 Del:14 kapittel IXX”
- [8] Yngvar Berg “Domino logikk, INF3400 Del:9 kapittel III”