UKE 11

Plan for i dag

- Gjennomgang av pensum
  - Med eksamensoppgaver
- Ukesoppgaver/oblig

Oblig 3

- Skal leveres 12. november (lørdag)
- Skal leveres i canvas

Sannhetsverditabell

- Hjelper oss med å se alle mulige resultater av alle inngangskombinasjonerfor en logisk funksjon
- En sannhetsverditabell kan ha mange funksjonsbeskrivelser
  - Mange funksjoner kan gi samme sannhetsverditabell
- En funksjon har bare en sannhetsverditabell
  - Et funksjonsuttrykk kan bare ha ett resultat (f.eks. F = A + B)

$$F = A+B$$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Logiske porter

- Består av transistorer
- Utfører en logisk operasjon basert på input

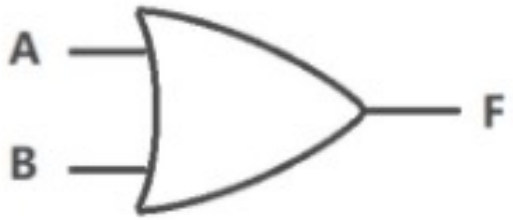
Porter - OR

- Så lenge en av inngangene er 1, er utgangen 1

OR - port

$F = A+B$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



Porter - AND

Gir 1 hvis begge inngangene er 1, ellers gir den 0

AND - port

$F = AB$

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1



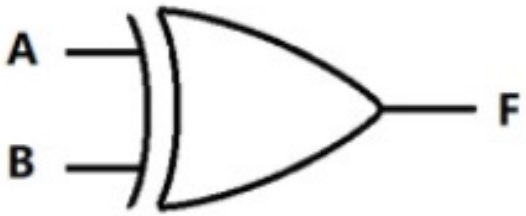
Porter - XOR

- Exclusive OR
- Odde-funksjon
  - Gir 1 så lenge det er et odde antall enere på utgangen

XOR - port

$F = A \oplus B$

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0



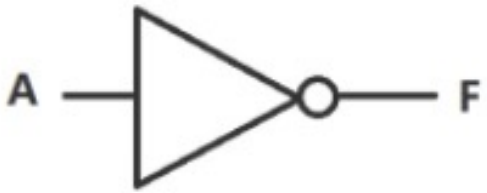
Porter - NOT

- Inverterer signalet som kommer inn
  - 1 blir til 0
  - 0 blir til 1

INV - port

$F = A`$

A	F
0	1
1	0



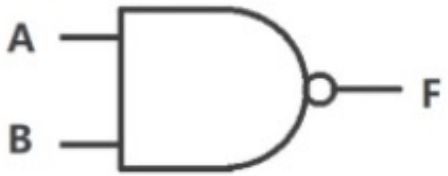
Porter - NAND

- Motsatt av AND

NAND - port

$F = (AB)`$

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0



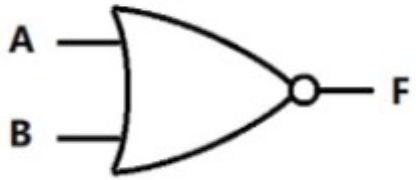
Porter - NOR

- Motsatt av OR

NOR - port

$F = (A+B)`$

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



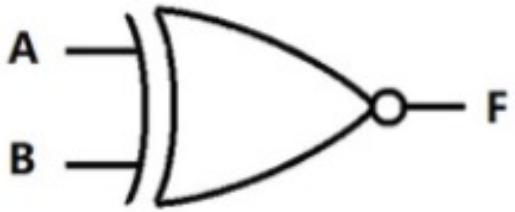
Porter - XNOR

- Motsatt av OR

XNOR - port

$F = (A \oplus B)`$

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1



$$(A + B)' = A'B'$$

Boolean algebra simplification rules

1.	Law of Identity	$A = A$ $\overline{A} = \overline{A}$
2.	Commutative Law	$A \cdot B = B \cdot A$ $A + B = B + A$
3.	Associative Law	$A \cdot (B \cdot C) = A \cdot B \cdot C$ $A + (B + C) = A + B + C$
4.	Idempotent Law	$A \cdot A = A$ $A + A = A$
5.	Double Negative Law	$\overline{\overline{A}} = A$
6.	Complementary Law	$A \cdot \overline{A} = 0$ $A + \overline{A} = 1$
7.	Law of Intersection	$A \cdot 1 = A$ $A \cdot 0 = 0$
8.	Law of Union	$A + 1 = 1$ $A + 0 = A$
9.	DeMorgan's Theorem	$\overline{AB} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A}\,\overline{B}$
10.	Distributive Law	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A + (BC) = (A + B) \cdot (A + C)$
11.	Law of Absorption	$A \cdot (A + B) = A$ $A + (AB) = A$
12.	Law of Common Identities	$A \cdot (\overline{A} + B) = AB$ $A + (\overline{A}B) = A + B$

Boolean algebra simplification rules

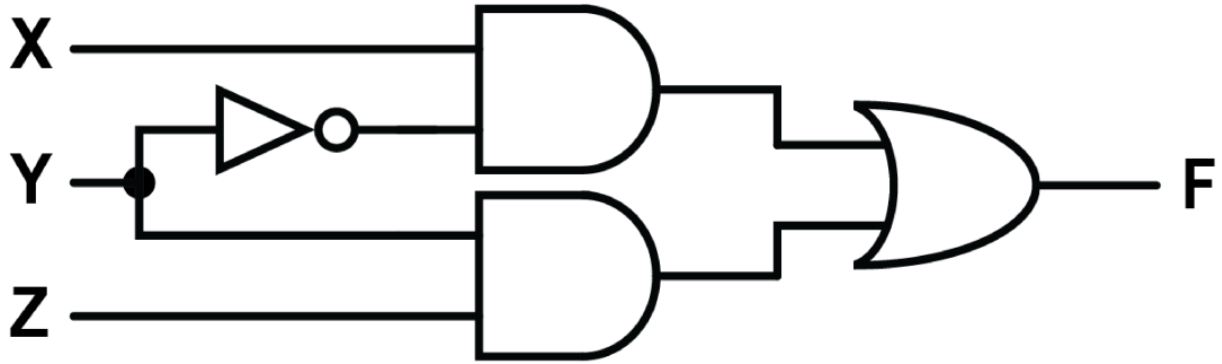
1.	Law of Identity	$A = A$ $\overline{A} = \overline{A}$
2.	Commutative Law	$A \cdot B = B \cdot A$ $A + B = B + A$
3.	Associative Law	$A \cdot (B \cdot C) = A \cdot B \cdot C$ $A + (B + C) = A + B + C$
4.	Idempotent Law	$A \cdot A = A$ $A + A = A$
5.	Double Negative Law	$\overline{\overline{A}} = A$
6.	Complementary Law	$A \cdot \overline{A} = 0$ $A + \overline{A} = 1$
7.	Law of Intersection	$A \cdot 1 = A$ $A \cdot 0 = 0$
8.	Law of Union	$A + 1 = 1$ $A + 0 = A$
9.	DeMorgan's Theorem	$\overline{AB} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A}\,\overline{B}$
10.	Distributive Law	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A + (BC) = (A + B) \cdot (A + C)$
11.	Law of Absorption	$A \cdot (A + B) = A$ $A + (AB) = A$
12.	Law of Common Identities	$A \cdot (\overline{A} + B) = AB$ $A + (\overline{A}B) = A + B$

Forsøk å forenkle følgende uttrykk: $A + AB$

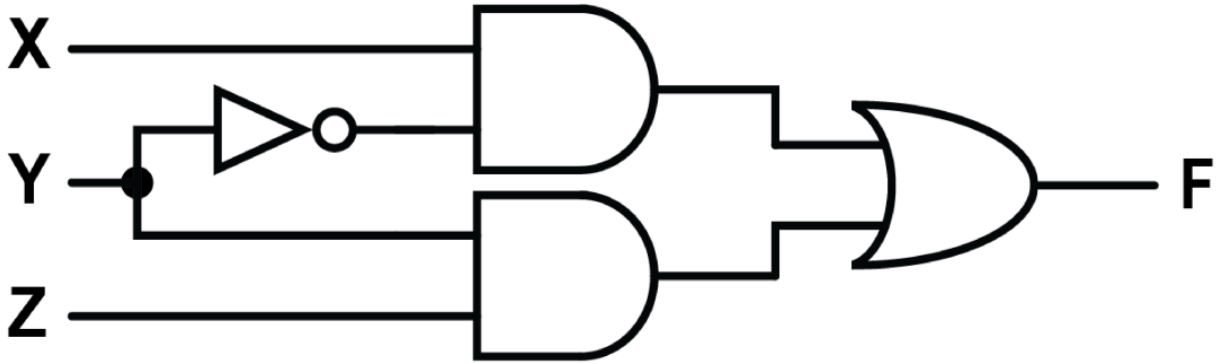
Boolean algebra simplification rules

1.	Law of Identity	$A = A$ $\overline{A} = \overline{A}$
2.	Commutative Law	$A \cdot B = B \cdot A$ $A + B = B + A$
3.	Associative Law	$A \cdot (B \cdot C) = A \cdot B \cdot C$ $A + (B + C) = A + B + C$
4.	Idempotent Law	$A \cdot A = A$ $A + A = A$
5.	Double Negative Law	$\overline{\overline{A}} = A$
6.	Complementary Law	$A \cdot \overline{A} = 0$ $A + \overline{A} = 1$
7.	Law of Intersection	$A \cdot 1 = A$ $A \cdot 0 = 0$
8.	Law of Union	$A + 1 = 1$ $A + 0 = A$
9.	DeMorgan's Theorem	$\overline{AB} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A}\,\overline{B}$
10.	Distributive Law	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A + (BC) = (A + B) \cdot (A + C)$
11.	Law of Absorption	$A \cdot (A + B) = A$ $A + (AB) = A$
12.	Law of Common Identities	$A \cdot (\overline{A} + B) = AB$ $A + (\overline{A}B) = A + B$

Kretsanalyse – hva er funksjonsuttrykket til kretsen?



Kretsanalyse



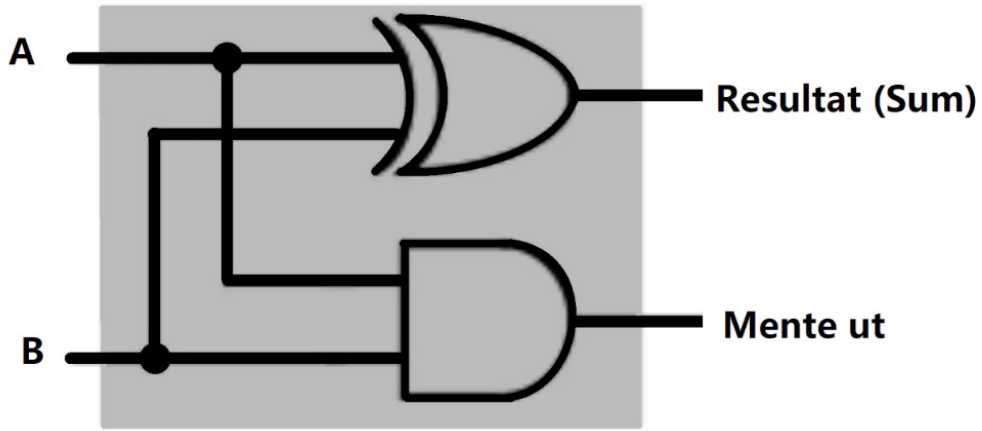
Commutative law: ab = ba
xy′ + yz = xy′ + zy

Binær addisjon

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 0$$

$$\begin{array}{r} {\scriptstyle 1} \\ 0\;1\;0\;1 \\ +\;0\;1\;1\;0 \\ \hline =\;1\;0\;1\;1 \end{array} \qquad \begin{array}{r} {\scriptstyle 1} \\ 05 \\ +\,06 \\ \hline =11 \end{array}$$

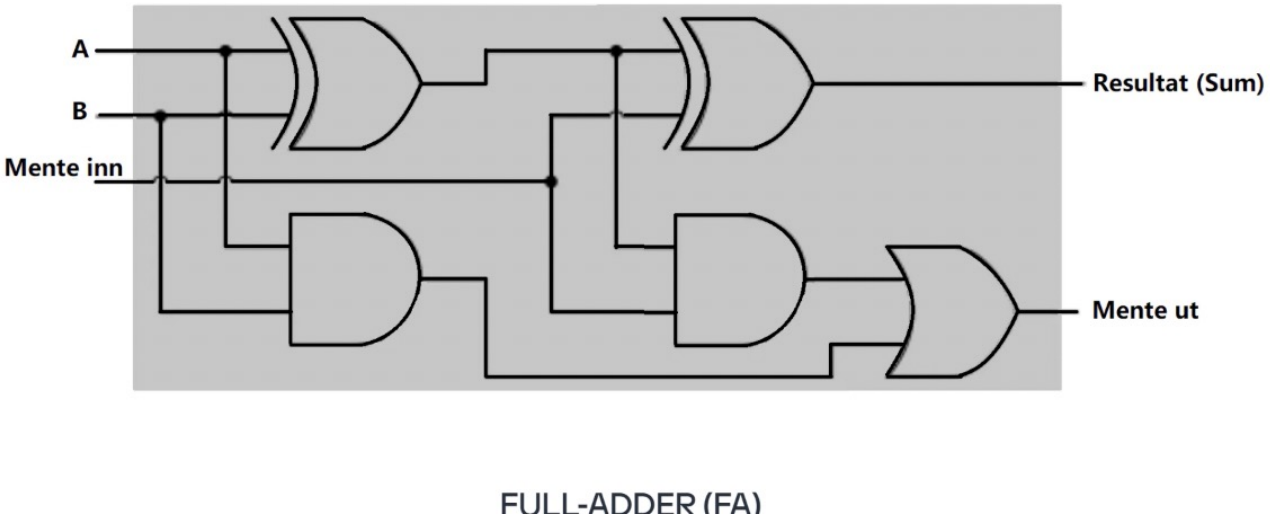
Halv-adder



HALV-ADDER (HA)

- Tar inn to tall
- Returnerer summen
- Returnerer mente

Full-adder

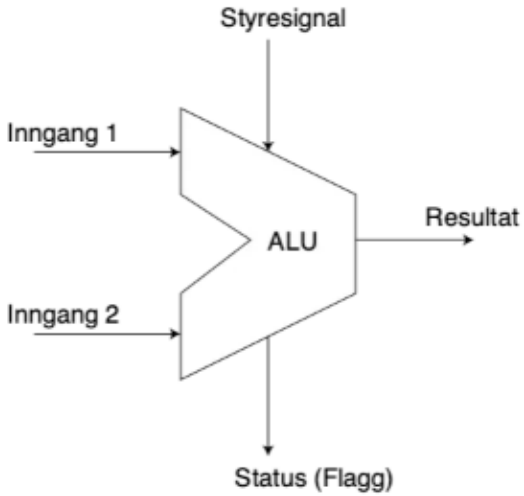


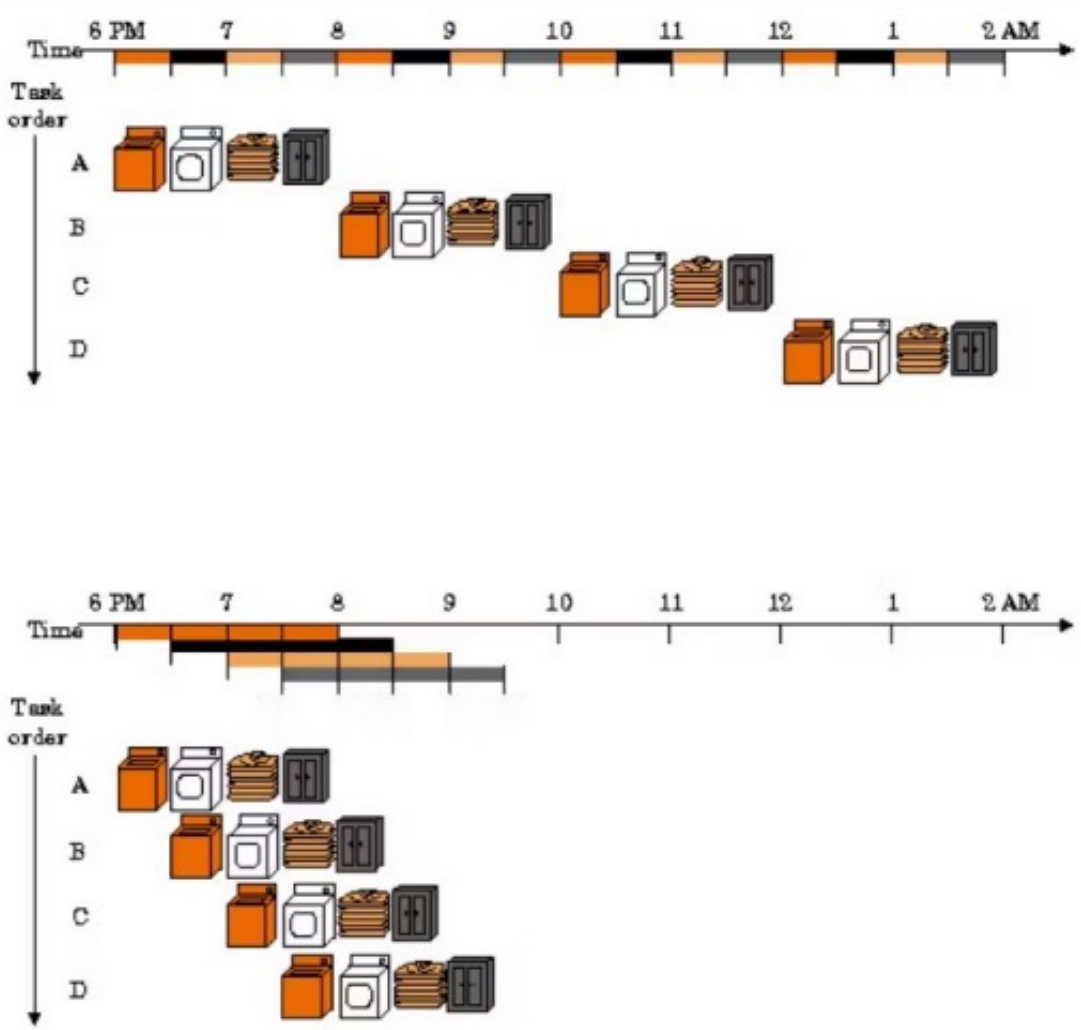
FULL-ADDER (FA)

- Tar mente inn fra forrige full-adder/halv-adder

ALU

- Utfører alle logiske og matematiske beregninger
- Del av CPU'en
- Har en addisjonskrets som utfører binær addisjon
- Styresignalet bestemmer hvilken operasjon som skal utføres av ALU

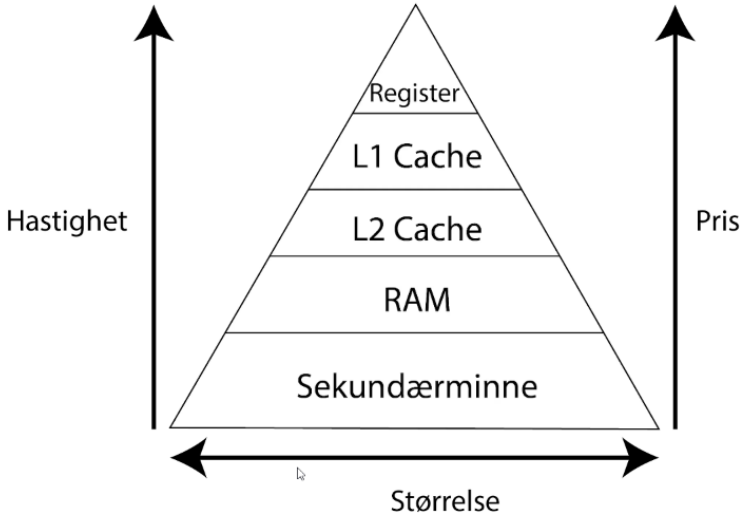




Pipeline

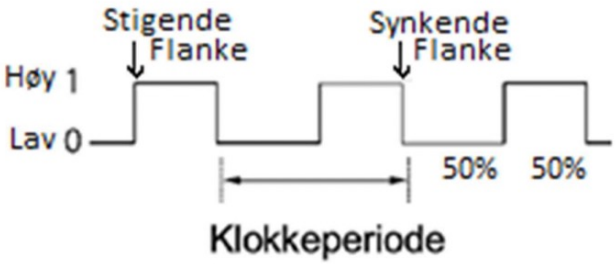
- Konsept som går ut på å dele opp instruksjoner i uavhengige sub-instruksjoner
- Ved å gjøre dette kan man starte neste instruksjon før forrige er ferdig utført, slik at man bruker tiden mer effektivt

Minnehierarki



- Minneelementene som er raskest å aksessere er de som koster mest, men også de som har minst lagringskapasitet.
- Når CPU'en vil ha tilgang til data/instruksjoner, så 'spør' den først registeret om den har dataen. Om registeret ikke har den ønskede dataen, må CPU'en 'spørre' cache.
  - **Cache miss**: når informasjonen CPUen trenger ikke ligger tilgjengelig i cache.
  - **Cache hit**: når informasjonen CPUen trenger ligger i cache og kan brukes

Klokker og synkronisering



- Klokkesignal: signal som går fra 0 til 1 i fast takt
- Klokkesignaler hjelper CPU'en for å vite når instruksjoner skal kjøres. (Klokkesignaler gjør synkronisering mulig)
- Når klokken er høy, utføres instruksjoner
- Klokkesykel/klokkeperiode: fra stigende flanke til stigende flanke

Beregning av cache miss

(2) Beregning av Cache-miss:
Anta at en prosessor vil prosessere en instruksjon per klokkesykel og skal totalt prosessere 1.000.000 instruksjoner som typisk vil ha behov for å hente data fra minne for 60% av instruksjonene, hvor det uheldigvis vil være 50% cache-miss. Anta videre at det typisk vil være en penalty (økt forsinkelse) for cache-miss på 10 klokkesykler. Hva lang mange klokkesykler vil det hele ta?

Beregning av cache miss

Anta at det er igjen 3000 instruksjoner og at en instruksjon tar 1 klokkesykel, bortsett fra eventuelle cache-miss. Videre kan du anta at det vil være 50% cache-miss hvor det vil ta totalt 4 klokkesykel for hver instruksjon i cache-miss.

Hvor lang til vil det totalt ta?

Beregning av cache miss

Anta at prosessoren har 100 instruksjoner igjen å utføre. Det tar én klokkesykel pr instruksjon, og man regner med å ha en minneaksessering på 80% og med cache-miss på 50%. En cache-miss fører til en total tidsbruk på 20 klokkesykler. Cache-hit bruker total én klykkesykel. Hva er total antall gjenstående klokkesykler?