

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i:	INF3430/INF4430 Digital systemkonstruksjon
Eksamensdag:	6. desember 2007
Tid for eksamen:	9 - 12
Oppgavesettet er på 8 sider	
Vedlegg:	1
Tillatte hjelpemidler:	Ingen

Kontroller at oppgavesettet er komplett før du begynner å besvare spørsmålene.

Oppgaveteksten består av oppgave 1 – 14 (flervalgsoppgaver) som skal besvares på skjemaet som er vedlagt etter oppgaveteksten og oppgave 15 som besvares på vanlige ark. Oppgave 1 - 14 har til sammen vekt på 40%, mens oppgave 15 har vekt på 60%.

Generelt for oppgave 1- 14:

Hver oppgave består av et tema i venstre kolonne og en del utsagn hver angitt med en stor bokstav. Oppgavene besvares ved å merke tydelige kryss (X) i rett kolonne for riktig svaralternativ (dvs. at et utsagn er sant) i skjemaet i vedlegget. Det er alltid *minst en* riktig avmerking for hver oppgave, men det er ofte *flere* riktige avmerkninger. *For å få best karakter skal man sette flere kryss i en oppgave hvis det er flere riktige utsagn.* Det gis 1 poeng for hver avkrysning der det skal være avkrysning. Det gis -1 poeng for hver avkrysning der det ikke skal være avkrysning. Mangel på kryss der det skal være kryss gir også -1 poeng. Du kan benytte høyre kolonne i oppgaveteksten til kladd. Skjema påført ditt kandidatnummer i vedlegget er din besvarelse.

Oppgave 1

Kretsteknologier	A	En logikkblokk i en FPGA består normalt av en Look-Up Table (LUT) etterfulgt av en vippe (flip-flop)	
	B	En PLA består av OR-porter etterfulgt av AND-porter	
	C	I en PAL er tilkoblingene til AND-portene ikke programmerbare	
	D	I en "full custom" ASIC har designeren full kontroll over hvert maskelag i kretsen	

Oppgave 2

Lagringsteknologi	A	I en CPLD lagres normalt konfigurasjonen i SRAM	
	B	En FPGA basert på antifuse-teknologi er ikke reprogrammerbar	
	C	En FPGA basert på antifuse-teknologi kan ikke slettes med UV-lys	
	D	En EPROM kan slette sitt innhold med en høy spenning	
	E	En SRAM kan kun programmeres et begrenset antall ganger	

Oppgave 3

Konfigurasjon av FPGA	A	En FPGA i master-modus styrer selv nedlastning av konfigurasjonen ved oppstart	
	B	”Daisy-chaining” gjør at flere FPGA-er kan ha et felles konfigurasjonsminne	
	C	En FPGA må alltid konfigureres parallelt hvis den er i slave-modus	
	D	JTAG-porten er egentlig tiltenkt testing men kan også brukes til konfigurasjon	

Oppgave 4

Optimalisert FPGA design	A	Selv om antall input til en funksjon er konstant, øker forbruket av logikk med kompleksiteten til funksjonen	
	B	Antall nivåer med logikk i en FPGA <i>mellom</i> klokkede vipper har betydning for maksimal klokkefrekvensen	
	C	Klokketre i en FPGA bør unngås hvis en skal lage et effektivt synkront design	
	D	Dedikert mentelogikk kobler sammen logikk for hurtig menteforplantning	
	E	Bruk av dedikert mentelogikk gjør at det blir mindre tilgjengelig logikk i FPGA-en og bruken bør derfor begrenses	

Oppgave 5

En 3-input Xilinx LUT (look-up table) med innhold 7F (hex) realiserer en	A	AND funksjon	
	B	OR funksjon	
	C	NAND funksjon	
	D	NOR funksjon	

Oppgave 6

Prosessorkjerner	A	En hard kjerne er implementert fysisk i FPGA-en ved produksjon av kretsen	
	B	Kombinasjon av prosessor og logikk på en FPGA gir liten fleksibilitet i bestemmelsen av hva som blir programvare og hva som blir maskinvare	
	C	Separat buss mellom prosessor og minne gir lite gevinst og bør unngås	
	D	Integrering av et helt system på en krets gir en mer kompakt løsning som også prismessig kan være gunstig	

Oppgave 7

Virtuelle komponenter/ Intellectual Property	A	En IP gitt som ikke-kryptert kildekode er normalt mer effektiv enn en IP gitt som forhåndsrutet IP	
	B	Intellectual Property er betegnelsen på ferdigutviklede blokker	
	C	MicroBlaze er eksempel på en IP	
	D	Det er enkelt å gjenbruke en IP fra en FPGA-produsent på kretser fra andre produsenter	

Oppgave 8

Sykelbasert simulering	A	Dette er et alternativ til hendelsesbasert simulering	
	B	En dropper å simulere hver hendelse i en krets men benytter boolske uttrykk på inngangene til registre	
	C	Metoden kan kombineres med hendelsesdrevet simulering for simulering av en krets	
	D	En ulempe, sammenlignet med alternative måter å simulere på, er at tiden for simulering øker betydelig	

Oppgave 9

Syntese	A	Syntese gjøres normalt etter "place-and-route"	
	B	Syntese med informasjon om faktiske tidsforsinkelser i FPGA-en kan gi høyere maksimal klokkefrekvens	
	C	Plassering av registre (vipper) i forhold til logikk har normalt ingen betydning for ytelsen	
	D	Resyntese for optimalisering av kritisk signalvei kan være gunstig	

Oppgave 10

SystemC	A	Språket er definert av en spesifikk verktøyleverandør som selger designverktøy	
	B	Språket er basert på C/C++	
	C	Språket er bedre egnet til verifikasjon enn syntese	
	D	Språket kan spesifisere kode på flere abstraksjonsnivåer enn VHDL	
	E	SystemC brukes i dag like ofte som VHDL for FPGA design	

Oppgave 11

Kodestil for FPGA og ASIC	A	Samlebåndsprossessering (pipelining) kan være med på å øke maksimal klokkefrekvens i et design	
	B	Samlebåndsprossessering (pipelining) vil ofte medføre at en bruker færre vipper i et design	
	C	Tilbakekoblingsløyper der vipper inngår må ikke brukes i en FPGA	
	D	Asynkront design er mulig i en ASIC, men anbefales ikke i en FPGA	

Oppgave 12

Valg mellom ASIC og FPGA	A	FPGA er bedre enn ASIC ved komplekse design	
	B	Det er bedre plass i en ASIC enn i en FPGA når kretsene har omtrent samme fysiske størrelse	
	C	Prototyping av ASIC på FPGA bør unngås på grunn av forskjell i kodestil	
	D	ASIC har lang utviklingstid men de første kretsene er billige å produsere	

Oppgave 13

Høyhastighets serielinker	A	Grunnen til at en overført firkantpuls ved høy datarate kan bli lik et sinussignal er at høyfrekvent frekvensinnhold har blitt kraftig dempet	
	B	Konfigurasjon av parametere i transceiver muliggjør design med forskjellige kommunikasjonsstandarder	
	C	Pre-emphasis motvirker demping i overført signal	
	D	”Comma”-tegn brukes for å dele opp lange bitstrenger	

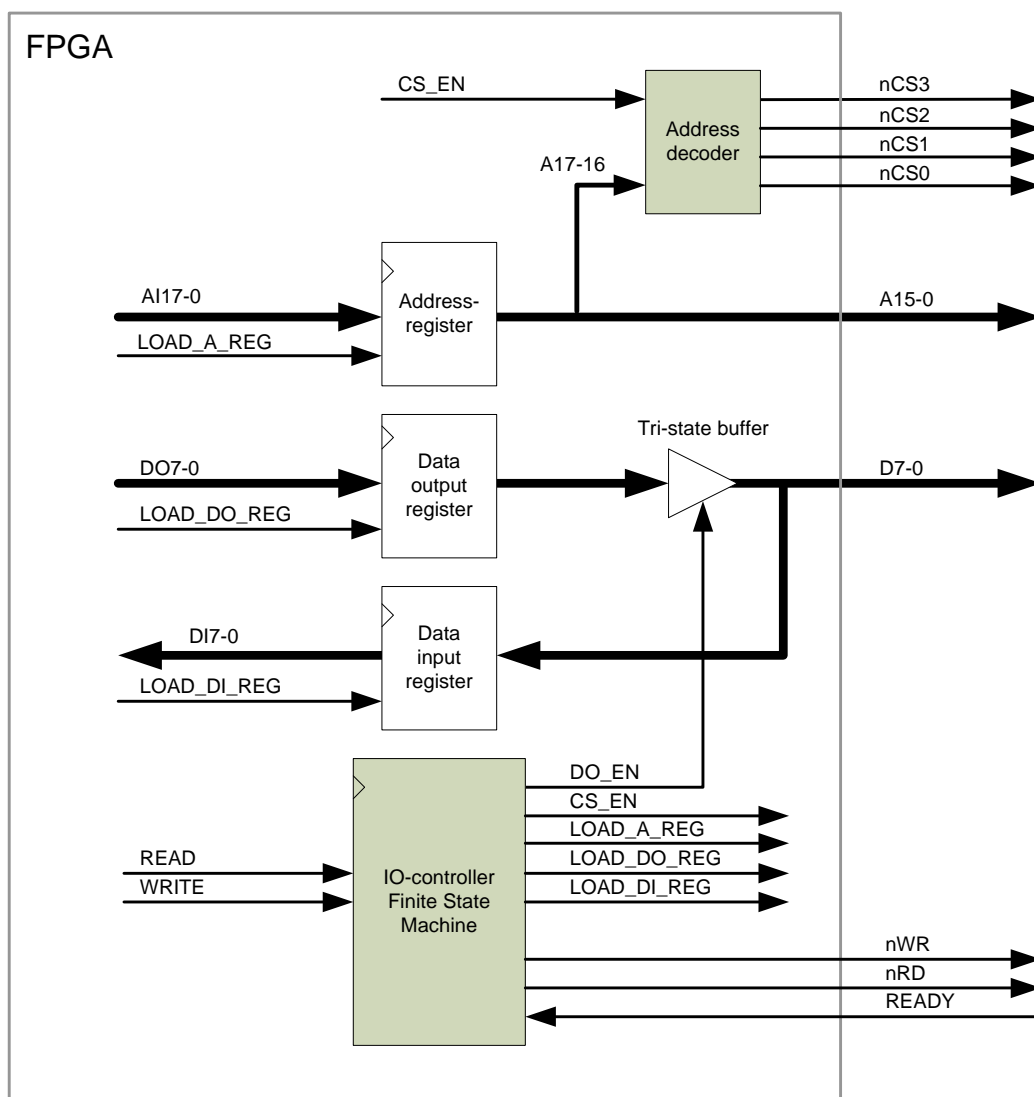
Oppgave 14

Rekonfigurering av aktiv FPGA	A	Virtuell maskinvare er en betegnelse som brukes om denne teknikken	
	B	Teknikken muliggjør å kunne utføre en større oppgave enn det kretsen tilsynelatende har logikk til	
	C	Effektforbruket kan ofte øke ved denne metoden	
	D	Lang rekonfigureringstid er en av hovedutfordringene	
	E	Det vil være ønskelig med denne metoden å rekonfigurere hele kretsen og ikke kun en begrenset del av den	

Oppgave 15

Vi skal i denne oppgaven konstruere deler av et grensesnitt som skal styre eksternt minne og/eller input/output kretser som er koblet til en FPGA. Vi vil referere til dette som I/O-grensesnittet. I/O-grensesnittet skal være del av en mikrokontroller i FPGAen.

Figur 1 viser en oversikt over I/O-grensesnittet. De gråskraverte boksene i figuren viser de delene av I/O-grensesnittet vi skal konsentrere oss om i de påfølgende oppgavene.



Figur 1. I/O-grensesnittet

Tabell 1. Signaler i I/O-grensesnittet

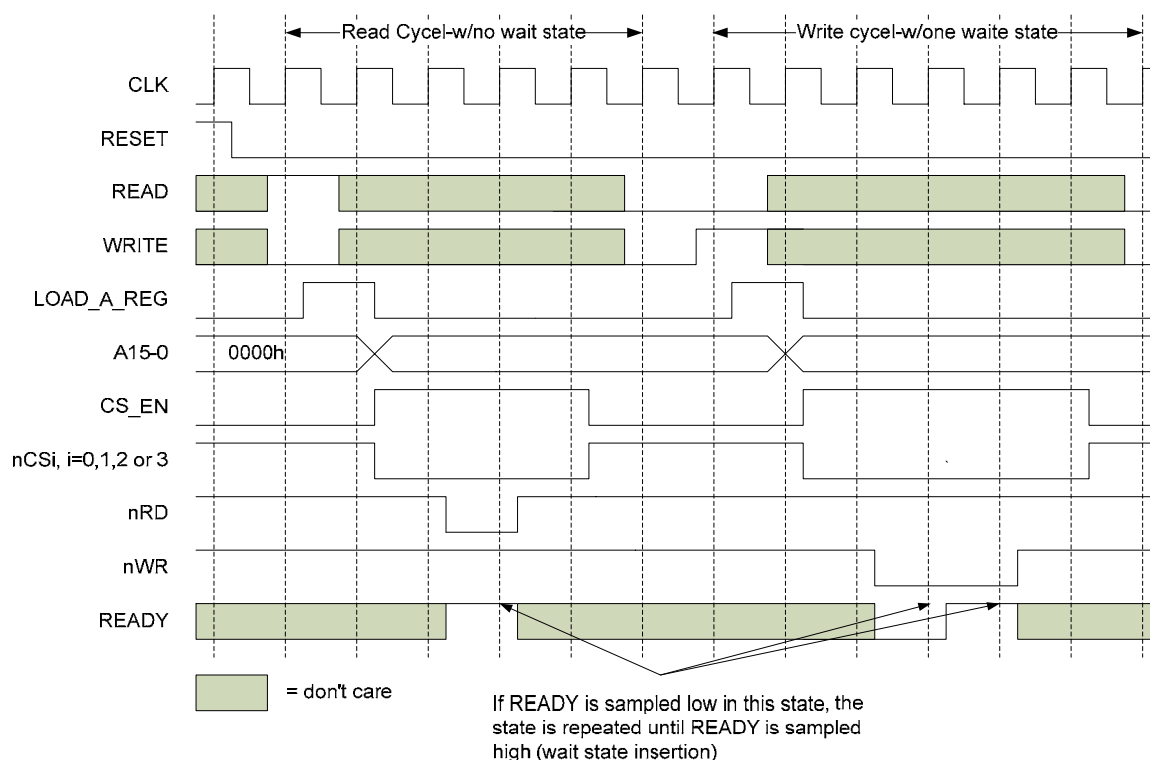
Signalnavn	Beskrivelse	Retning
Klokke og reset er ikke vist i figur 1		
CLK	50MHz systemklokke	Input til alle registre
RESET	Asynkron reset. Aktivt høyt	Input til alle registre
Eksterne signaler:		
A15-0	Adresse signaler	Output
D7-0	Data signaler	Input/Output/Tri-state
nCS _i , i=0,1,2,3	Chip select signaler. Benyttes for å adressere eksternt minne eller I/O. Aktivt lave	Output fra adressedekoder
nWR	Write strobe. Aktivt lavt	Output fra tilstandsmaskin
nRD	Read strobe. Aktivt lavt.	Output fra tilstandsmaskin
READY	Viser om en I/O krets har data klare eller er klar til å ta i mot data. Aktivt høyt. Benyttes for å forlenge en les eller skriv I/O operasjon ved å sette inn ventetilstander (Wait states).	Input til tilstandsmaskin.
Interne signaler:		
READ	Starter en leseoperasjon fra I/O grensesnittet. Aktivt høyt	Input til tilstandsmaskin
WRITE	Starter en skriveoperasjon til I/O grensesnittet. Aktivt høyt	Input til tilstandsmaskin
AI17-0	Interne adressesignaler.	Input til adresseregisteret og adressedekoderen
DO7-0	Data output signaler	Input til data output registeret
DI7-0	Data input signaler	Output fra data input registeret
CS_EN	Enabler nCS _i , i=0,1,2,3	Output fra tilstandsmaskinen
LOAD_A_REG	Lagrer adressene AI i adresseregistret. Aktivt høyt	Output fra tilstandsmaskinen
LOAD_DO_REG	Lagrer output data i data out registeret. Aktivt høyt	Output fra tilstandsmaskinen
DO_EN	Styrer output tri-state buffer	Output fra tilstandsmaskinen
LOAD_DI_REG	Lagrer input data i data input register. Aktivt høyt.	Output fra tilstandsmaskinen

Adressedekoderen "Address decoder" skal virke i henhold til sannhetstabellen under.

Sannhetstabell 1. Adressedekoderen

Inputs		Outputs			
CS_EN	A17-16	nCS0	nCS1	nCS2	nCS3
0	X	1	1	1	1
1	00	0	1	1	1
1	01	1	0	1	1
1	10	1	1	0	1
1	11	1	1	1	0

En les eller skriv I/O-operasjon er bygd opp av flere tilstander og starter med at READ- eller WRITE-signalet går aktivt. READ og WRITE kommer fra en tilstandsmaskin som eksekverer programmer og er ikke aktive samtidig. Etter at READ/WRITE har vært aktiv skal de interne adressesignalene, A17-0, lagres i adresseregisteret styrt av signalet LOAD_A_REG. A15-0 føres ut på pinner, mens A17-16 sammen med CS_EN er input til adressedekoderen som gir output i henhold til sannhetstabell 1. Resten av les eller skriv operasjonen skal følge timingdiagram 1 under. Legg merke til at en les eller skriv operasjon forlenges dersom READY-signalene er lavt når nRD eller nWR er aktivt. Benytt signalnavn som angitt over når du løser de etterfølgende oppgavene.



Timingdiagram 1

15a). Vekt 10%

Implementer sannhetstabell 1 ved å benytte en process i VHDL. Du trenger ikke å ta med entiteten.

Vi skal nå designe en tilstandsmaskin "IO-controller" for å lage kontrollsignalene til I/O grensesnittet. Vi skal begrense oss til kontrollsignalene: LOAD_A_REG, CS_EN, nRD og nWR i I/O-grensesnittet.

15b). Vekt 20%

Tegn et ASM flytdiagram som beskriver tilstandsmaskinen gitt av tekst og timingdiagram over.

15c). Vekt 20%

Implementer tilstandsmaskinen beskrevet i ASM flytdiagrammet i 15b) i VHDL. Du trenger ikke å ta med entiteten.

15d). Vekt 10%

Skissør en testbenk for å verifisere tilstandsmaskinen (du skal ikke lage en komplett testbenk).

INF3430/INF4430 Oppgavesvar for kandidat nr: _____

Oppgave	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					