

**NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET
INSTITUTT FOR DATATEKNIKK OG INFORMASJONSVITENSKAP**



Faglig kontakt under eksamen:

Jon Olav Hauglid, Tlf 93440

Institutt for datateknikk og informasjonsvitenskap, Gløshaugen

EKSAMEN I EMNE TDT4160 DATAMASKINER GRUNNKURS

17. AUGUST 2005

KL. 09.00 – 13.00

Hjelpemidler: D – Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Sensuren faller 7. september 2005.

Resultater gjøres kjent på <http://studweb.ntnu.no/> og sensurtelefon 810 48 014.

Totalt antall sider: 9

Prosentsetter viser hvor mye hver oppgave teller innen settet.

Lykke til!

Oppgave 1 – Flervalgsoppgaver (30%)

Bruk svararket bakerst i oppgaveteksten for å svare på denne oppgaven. Du kan få nytt ark av eksamensvaktene dersom du trenger dette. Poengberegning per spørsmål: Riktig svar 2 poeng, galt svar gir -1 poeng, blankt svar gir 0 poeng. NB! Det er ikke mulig å gardere ved å krysse av flere alternativer. Dette gir i så fall 0 poeng. Kun ett alternativ er korrekt på hvert spørsmål.

- 1) Hvor mange assemblerinstruksjoner er det i en mikroinstruksjon?
 - a) Dette er avhengig av instruksjonssettarkitekturen (ISA)
 - b) Spørsmålet gir ikke mening fordi en assemblerinstruksjon består av flere mikroinstruksjoner
 - c) Tre, en for hver fase (load, execute, store)
 - d) Dette er avhengig av størrelse på hurtiglager (eng. cache)

- 2) Enkelte maskiner har noe som heter segmentregister, hva brukes det til?
 - a) Peker på register som skal leses i en registerfil
 - b) Brukes av hurtiglager for å vite hvor data skal ligge i hovedlager
 - c) Holder øverste bit i en adresse for å få større adresserom
 - d) Holder nederste bit i en adresse for å vite relasjon til hurtiglager

- 3) Hva er riktig om samlebånd (eng. pipelines)?
 - a) Samlebånd øker hastigheten for hver enkelt instruksjon isolert sett
 - b) Samlebånd i prosessorer benyttes ikke lengre, dette er gammel teknologi
 - c) Målet med et samlebånd er å utføre instruksjoner i parallell
 - d) Samlebånd gjør at prosessoren gjør ferdig flere instruksjoner per klokkesykel

- 4) Hva er FEIL om samlebånd?
 - a) Klokkefrekvensen til en prosessor med samlebånd er begrenset av summen av tiden samlebåndet bruker i de ulike trinnene
 - b) Klokkefrekvensen til en prosessor med samlebånd er begrenset av det tregeste trinnet
 - c) Ved å innføre samlebånd kan frekvensen skrus opp fordi hvert enkelt trinn blir enklere
 - d) En prosessor har samme klokkefrekvens for de ulike trinnene

- 5) Dersom du har et vanlig (in-order execution) samlebånd, hvilke dataavhengigheter vil skape problemer i følgende program (Dark-syntaks – se oppgave 5)?


```
ADD $1, $2, $3
ADD $4, $2, $3
```

 - a) Avhengigheten på grunn av register 2
 - b) Avhengigheten på grunn av register 3
 - c) Avhengigheten på grunn av register 1
 - d) Det finnes ingen avhengighet

- 6) Hva skiller indirekte adresseringsmodus fra direkte adresseringsmodus (velg det alternativet som er mest riktig)?
- a) Indirekte er enklere enn direkte adressering fordi indirekte adressering krever et ekstra oppslag
 - b) Indirekte adressering adresserer relativt til stakkregisteret, mens direkte adressering adresserer relativt til instruksjonsregisteret
 - c) Direkte adressering gir lengre kildekode, og moderne prosessorer benytter kun indirekte adressering
 - d) Ved indirekte adressering beregnes adressen som skal aksesseres ved kjøretid, mens ved direkte adressering er adressen gitt av assemblerkoden
- 7) Hvorfor snakker man om 1ers komplement og 2ers komplement?
- a) De to måtene å kode tall på benyttes i forskjellige regneoperasjoner
 - b) Det er lettere for en datamaskin å addere to tall som er på 2ers komplement enn 1ers komplement
 - c) 1ers komplement har større tallområde enn 2ers komplement
 - d) 1ers komplement er tenkt til 1-tallssystemet, mens 2ers komplement er tenkt til 2-tallssystemet
- 8) Hvilken prosessortype har som oftest kortest instruksjonsord?
- a) Akkumulatormaskin
 - b) Load/store-maskin
 - c) Stakkmaskin
- 9) Hva er et instruksjonsregister?
- a) Mikroprosessen leser instruksjonen som skal utføres til dette registret
 - b) Instruksjonen benytter dette registret til å lagre heltall
 - c) Instruksjonen benytter dette registret til å lagre både heltall og flyttall
 - d) Instruksjonsregisteret angir adressen til neste instruksjon som skal utføres.
- 10) Hvordan lagres data i hurtigminnet (SRAM) på en PC?
- a) Hver bit ligger lagret som strøm i en transistor
 - b) Hver bit ligger lagret som ladning i en kondensator
 - c) Hver bit ligger lagret i to invertere som er koblet mot hverandre for å låse data
 - d) Data ligger lagret som strøm i magnetpoler
- 11) Hvis man har to ulike datamaskiner som utfører like mange instruksjoner per sekund, hva kan man si om de to maskinene?
- a) Antall MIPS (millions of instructions per second) for de to maskinene er like
 - b) Ytelsen er lik
 - c) Maskinene er like
 - d) Den maskinen som er nyest er sannsynligvis raskest
- 12) Hvordan programmerer man en mikroprosessor med samlebånd?
- a) Man må passe seg for "farer" (eng. hazards) i enkelte arkitekturer
 - b) Man må spesifisere rekkefølgen på de ulike trinnene for hver instruksjon
 - c) Til dette benyttes mikrokode
 - d) Det skjer alltid i et høynivå språk som Java eller C (++).

- 13) Hvilket av følgende utsagn er mest FEIL om harddisker?
- Harddisker er mekaniske og tåler dårlig støt
 - Harddisker er basert på optiske plater som roterer relativt raskt (ca 7200 RPM).
 - Harddisker inneholder tosidige plater der data lagres
 - Harddisker har roterende plater, og man må vente til den har rotert til riktig plass før data kan leses
- 14) Hvordan henter en datamaskin vanligvis informasjon fra omgivelsene?
- Dette skjer ved at DRAM brikken er koblet til eksterne linjer
 - Dette skjer ved at ALU enheten er koblet til eksterne linjer
 - Dette skjer ved at porter er koblet på adresse og data bussen
 - Statusregisteret blir oppdatert med eksterne verdier
- 15) Hva er ulempen til USB?
- Man kan ikke ta utstyr fra en maskin og forvente at det fungerer på en annen maskin
 - Protokollen er ikke laget for overføring av store datamengder, her er *firewire* raskere
 - Det er lite datautstyr som støtter denne protokollen
 - USB kablene blir dyre fordi det er mange ledere i selve kabelen

Oppgave 2 – Direct memory access (DMA) (18%)

- Tegn en logisk skisse av en datamaskin (mikroprosessor, busser og dataminne) med en DMA-kontroller.
- Hvilke oppgave har DMA-kontrolleren?
- Beskriv de ulike trinnene i en typisk DMA-overføring. Hvilke signaler sendes mellom mikroprosessor og DMA-kontroller? Hvilke registre benyttes?

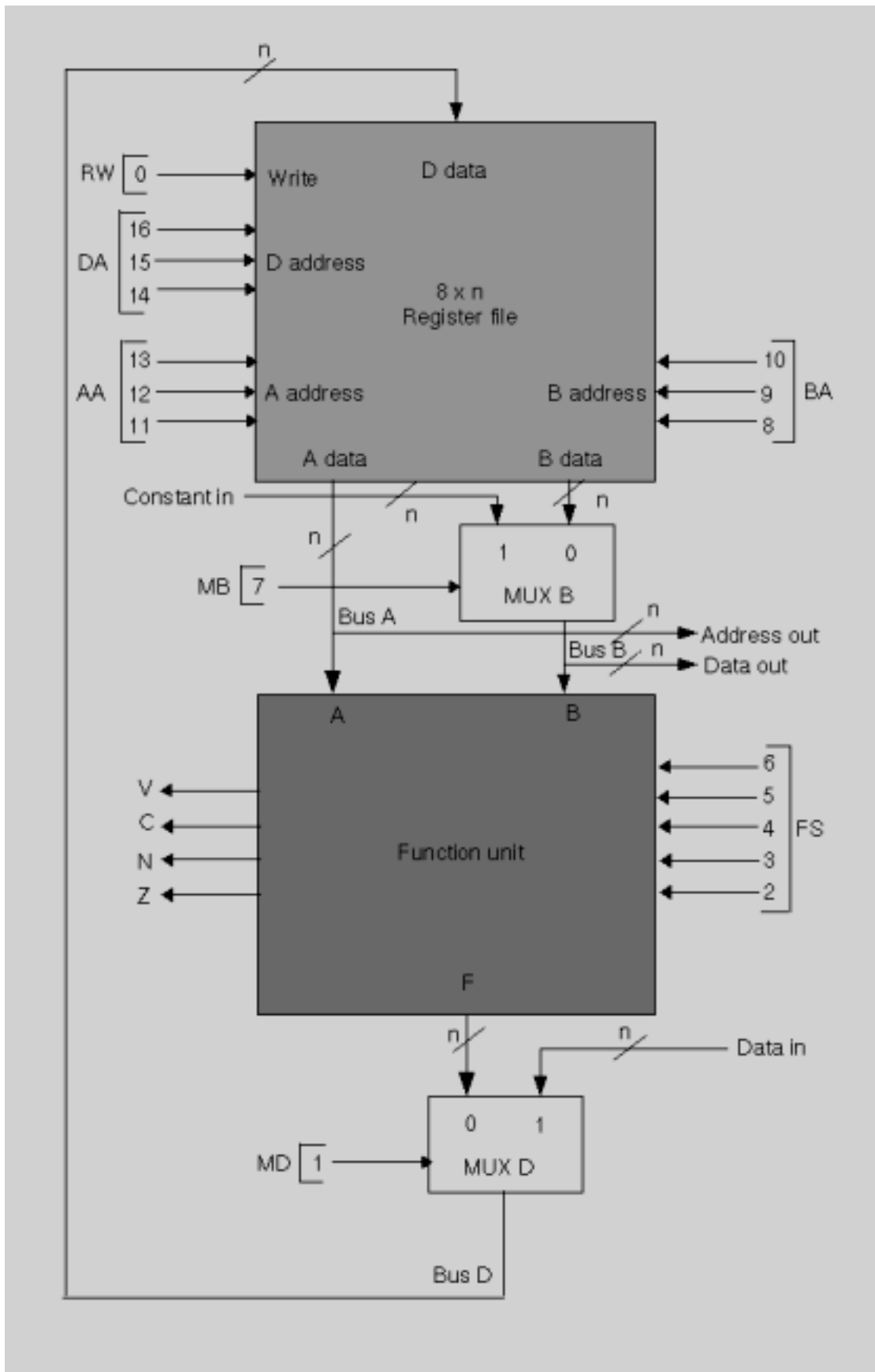
Oppgave 3 – Seriell overføring (12%)

- Beskriv begrepene startbit, databit, paritetsbit og stoppbit (se figur under).



- Når benyttes ofte seriell dataoverføring framfor parallell dataoverføring?

Oppgave 4 – Mikroarkitektur og assembler (20%)



DA, AA, BA		MB		FS		MD		RW	
Function	Code	Function	Code	Function	Code	Function	Code	Function	Code
<i>R0</i>	000	Register	0	$F = A$	00000	Function	0	No Write	0
<i>R1</i>	001	Constant	1	$F = A + 1$	00001	Data In	1	Write	1
<i>R2</i>	010			$F = A + B$	00010				
<i>R3</i>	011			$F = A + B + 1$	00011				
<i>R4</i>	100			$F = A + \sim B$	00100				
<i>R5</i>	101			$F = A + \sim B + 1$	00101				
<i>R6</i>	110			$F = A - 1$	00110				
<i>R7</i>	111			$F = \sim A$	00111				
				$F = A \wedge B$	01000				
				$F = A \vee B$	01010				
				$F = A \text{ XOR } B$	01100				
				$F = A$	01110				
				$F = \text{sr } A$	10000				
				$F = \text{sl } A$	10001				

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DA		AA		BA	M	FS						M	R	W		

Bruk tabellene gitt ovenfor, som beskriver styreordet til en prosessor, og figuren på forrige side, som skisserer en utførende enhet i denne prosessoren, til å besvare oppgavene under.

1. Hva blir styreordet for operasjonen $R3 \leftarrow R6 + R5 + 1$ der x skal beskrive ubrukte bits?
2. Hva blir konsekvensene av mikrooperasjonen med styreordet $111010xxxx0000100$ der x beskriver ubrukte bits?

For en gitt CISC-prosessor kan MOV-instruksjonen i assembler være definert med flere adresseringsmodi slik:

```
MOV Rn, #Imm      ; Immediate
MOV Rn, Addr      ; Direkte
MOV Rn, [Addr]    ; Indirekte
MOV Rn, Rm        ; Register
MOV Rn, [Rm]      ; Register indirekte
MOV Rn, Addr(Rm)  ; Displacement
PUSH Rn           ; Stakk push
POP Rn            ; Stakk pop
```

Rn/Rm indikerer register nummer n/m. Addr og Imm kan være tallverdier eller variabler. MOV fungerer slik: MOV til-operand, fra-operand

Gitt et lite utdrag av en datamaskins lager på et gitt stadium som skal benyttes for å besvare spørsmålene under:

Hovedlager		Registre		Definerte variabler	
Adresse	Verdi	Registernr	Verdi	Navn	Verdi
0	71	R0	5	teller	64
1	54	R1	15	i	3
2	12	R2	8	antall	4
3	86	R3	7	tab	3
4	28	R4	8	x	7
5	56	R5	55	retur	11
6	3	R6	9		
7	8	R7	35		
8	13				
9	7				
10	45				
11	82				

- Hva vil verdien til R1 være etter følgende instruksjon: MOV R1, #45?
- Følgende instruksjoner skal utføres etter hverandre. Skriv ned hvilke registre hver enkelt instruksjon endrer og hva sluttverdiene til R4, R5, R6 og R7 vil være:

```
PUSH R4
MOV R5, #antall
MOV R7, [R5]
POP R5
MOV R6, 1(R5)
MOV R7, 10
```

Oppgave 5 – Assemblerprogrammering (20%)

Skriv et assemblerprogram for Dark load/store maskin (se under) som gjør følgende jobb:

```
for(i=0; i≤n; i++) a[i]=b[i]+i
```

Anta at:

§3 inneholder adressen for a[0], første elementet av en tabell med n+1 ord

§4 inneholder verdien n.

§5 inneholder adressen for b[0], første elementet av en tabell med n+1 ord

Manual DARK load/store-arkitektur

Syntax	Semantik
add REG,REG,{REG NUM VAR}	$reg \leftarrow reg + \{reg NUM VAR\}$
and REG,REG,{REG NUM VAR}	$reg \leftarrow reg \wedge \{reg NUM VAR\}$
band REG,REG,{REG NUM VAR}	$reg \leftarrow reg \wedge \{reg NUM VAR\}$
bnot REG,REG	$reg \leftarrow \neg reg$
bor REG,REG,{REG NUM VAR}	$reg \leftarrow reg \vee \{reg NUM VAR\}$
bxor REG,REG,{REG NUM VAR}	$reg \leftarrow reg \oplus \{reg NUM VAR\}$
call ETIKETT	returstack $\leftarrow pc$ $pc \leftarrow ETIKETT$
dec REG	$reg \leftarrow reg - 1$
div REG,REG,{REG NUM VAR}	$reg \leftarrow reg / \{reg NUM VAR\}$
inc REG	$reg \leftarrow reg + 1$
jeq REG,REG,ETIKETT	{ $pc \leftarrow ETIKETT$ }
jge REG,REG,ETIKETT	{ $pc \leftarrow ETIKETT$ }
jgt REG,REG,ETIKETT	{ $pc \leftarrow ETIKETT$ }
jle REG,REG,ETIKETT	{ $pc \leftarrow ETIKETT$ }
jlt REG,REG,ETIKETT	{ $pc \leftarrow ETIKETT$ }
jmp ETIKETT	$pc \leftarrow ETIKETT$
jne REG,REG,ETIKETT	{ $pc \leftarrow ETIKETT$ }
load REG,{REG,NUM NUM VAR}	$reg \leftarrow \{mem[reg+NUM] NUM VAR\}$
mod REG,REG,{REG NUM VAR}	$reg \leftarrow reg \setminus \{reg NUM VAR\}$
mul REG,REG,{REG NUM VAR}	$reg \leftarrow reg * \{reg NUM VAR\}$
{mv mov} REG,REG	$reg \leftarrow reg$
not REG,REG	$reg \leftarrow \neg reg$
or REG,REG,{REG NUM VAR}	$reg \leftarrow reg \vee \{reg NUM VAR\}$
ret	$pc \leftarrow returstack$
store REG,{REG,NUM VAR}	$\{mem[reg+NUM] VAR\} \leftarrow reg$
sub REG,REG,{REG NUM VAR}	$reg \leftarrow reg - \{reg NUM VAR\}$
xor REG,REG,{REG NUM VAR}	$reg \leftarrow reg \oplus \{reg NUM VAR\}$

Svarark for oppgave 1

Studentnr: _____

Fagnummer: _____

Eksamensdato: _____

Side _____ av _____

	a	b	c	d
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

NB! Ikke glem å levere dette arket! Bla igjennom den papirbunken du gir fra deg til slutt, for å sjekke at svar på avkrysningsoppgaven er med.