

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for informasjonsteknologi,
matematikk og elektroteknikk

Institutt for datateknikk
og informasjonsvitenskap

BOKMÅL



AVSLUTTENDE EKSAMEN I

TDT4160
Datamaskiner Grunnkurs
Løsningsforslag

Torsdag 29. November 2007
Kl. 09.00 – 13.00

Faglig kontakt under eksamen:

Marius Grannæs

Hjelpemidler:

Kalkulator tillatt. Ingen trykte eller håndskrevne hjelpemidler tillatt.

Sensurdato:

20. Desember 2007. Resultater gjøres kjent på <http://studweb.ntnu.no/> og sensurtelefon 81548014.

Det er angitt i poeng hvor mye hver deloppgave teller ved sensur. Gjør nødvendige antagelser der dette er nødvendig. Husk: *korte og konsise* svar er ofte de beste.

Lykke til!

Oppgave 1 (30%) Multiple choice

Rett svar gir 2 poeng, feil svar vil gi -0.5 poeng. Flere avkryssinger på en oppgave gir 0 poeng. Bruk eget svarark på slutten av oppgavesettet.

1) Hvilken av disse påstandene om RISC er *ikke* sann?

- a) RISC-maskiner har som regel flere registre enn CISC-maskiner.
- b) RISC har som regel fast instruksjonslengde.
- c) RISC har som regel flere adresseringsmodi enn CISC.
- d) RISC har egne LOAD/STORE instruksjoner.

Svar: c)

2) Hvilket av disse typene er *ikke* en form for avbildning i hurtigbuffer (eng: cache)?

- a) Sett-assosiativt avbildning
- b) Direkte avbildning
- c) Kummulativ avbildning
- d) Fullt assosiativ avbildning

Svar: c)

3) Hva er hensikten med hurtigbuffer (eng: cache)?

- a) Å øke minnekapasiteten til maskinen.
- b) Å senke den gjennomsnittlige aksesstiden til minnet.
- c) Å skille mellom instruksjoner og data
- d) Å sørge for at alle prosessorer i en superdatamaskin har like data.

Svar: b)

4) Hvilket element er *ikke* en egenskap ved en seriell buss?

- a) En seriell buss krever at enhetene har en felles klokke.
- b) Overfører et bit om gangen.
- c) Kan være raskere enn en parallell buss.
- d) Benyttes av f.eks USB og Firewire.

Svar: a)

5) Hvilken av disse metodene er *ikke* en metode for å kontrollere I/O – enheter?

- a) Avbruddstyrt I/O
- b) Programstyrt I/O
- c) Direct Memory Access (DMA)
- d) Serielle registre

Svar: d)

6) Organisasjonen av en prosessor deles ofte i:

- a) Styreenhet og utførende enhet
- b) Register og aritmetisk-logisk enhet (eng: ALU)
- c) Minne og styreenhet
- d) Register og minne

Svar: a)

7) Hvilket av disse utsagnene om minne er *ikke* sant?

- a) Dynamisk RAM (DRAM) bruker kondensatorer for å lagre informasjon, og krever derfor oppfriskning med jevne mellomrom
- b) Statisk RAM (SRAM) bruker transistorer for å lagre informasjon, og krever derfor ikke oppfriskning.
- c) Statisk RAM bruker mindre areal enn dynamisk RAM per bit.
- d) Dynamisk RAM er som oftest organisert i en matrise med kolonner og rader.

Svar: c)

8) Hvilken påstand om three-state buffer er *ikke* sann?

- a) Three-state buffer brukes ofte i busser for å unngå at flere enheter driver bussen samtidig.
- b) Når et three-state buffer kobler en enhet av bussen kalles denne tilstanden for 'Z'.
- c) Three-state buffer kan bare brukes på serielle busser.
- d) Utgangen til et three-state buffer er enten 0,1 eller Z.

Svar: c)

9) Hvilken av disse påstandene er *ikke* sann om arbitrering?

- a) Arbitrering styrer hvem som kan bruke bussen til enhver tid.
- b) Arbitrering kan være sentral eller desentralisert.
- c) "Daisy chaining" er en form for arbitrering.
- d) Arbitrering er ikke nødvendig dersom bussen er asynkron.

Svar: d)

10) Vektorisert avbrudd vil si at:

- a) At man har flere avbruddsrutiner, en for hvert avbrudd som kan komme.
- b) At man bruker vektor-registre for å behandle avbruddet.
- c) At avbruddshåndteringsrutinen får inn en vektor som første parameter.
- d) Flere avbrudd bruker samme avbruddsrutine.

Svar: a)

11) Hvilket utsagn om forgreningspredikering (eng: branch prediction) er *ikke* riktig?

- a) Forgreningspredikering er bare viktig for mikrokontrollere.
- b) Forgreningspredikering prøver å forutse om et hopp blir utført eller ikke.
- c) Forgreningspredikering benytter seg ofte av historiebits.
- d) Moderne forgreningspredikering er svært nøyaktig.

Svar: a)

12) Hvilken av disse påstandene er *ikke* sann om samleband?

- a) Man kan benytte høyere klokkefrekvens med samleband enn uten.
- b) Samleband krever at man tar spesielle hensyn til avhengigheter mellom instruksjoner.
- c) Samlebandet er bare så raskt som det tregeste steget i samlebandet.
- d) At en maskin bruker samleband må spesifiseres på ISA-nivå.

Svar: d)

13) Hva er det ISA *ikke* spesifiserer?

- a) Hvilke mikroinstruksjoner som finnes
- b) Hvilke registre som er tilgjengelig for programmereren..
- c) Hvilke datatyper som skal støttes av maskinvaren.
- d) Hvilke instruksjoner/opkoder som finnes.

Svar: a)

14) Hvilket av disse er *ikke* en adresseringsmodi?

- a) Direkte adressering
- b) Register adressering
- c) Register-indirekte adressering
- d) Hurtigbuffer (eng: cache) adressering

Svar: d)

15) Når man kaller en prosedyre kan man lagre returadressen på flere forskjellige måter. Hvilken av disse metodene gir størst fleksibilitet?

- a) Å lagre returadressen på stakken.
- b) Å lagre returadressen i et spesielt register.
- c) Å lagre returadressen i et spesielt minneområde.
- d) Å lagre returadressen i et spesielt minneområde, et område for hver funksjon.

Svar: a)

Oppgave 2 – Chip Multiprocessor (10% - 5% på a) og 5% på b)

a) Nevn fire grunner til at flerkjerne prosessorer har blitt mer vanlig de siste årene

Svar:

1. Økende strømforbruk
2. Økende prosessor/minne gap
3. Økende designkompleksitet
4. Vanskelig å utnytte mer ILP (instruksjonsnivå parallellitet)

b) Hva vil det si at en chip MultiProcessor har heterogene kjerner?

Svar:

Heterogene kjerner vil si at kjernene er ulike. Dvs de har forskjellig instruksjonsett og/eller ytelse.

Oppgave 3 – IJVM (30% - a, b og d teller 5% hver, c teller 15%)

Nødvendige detaljer til IJVM finnes bakerst i eksamenssettet. Registrene til IJVM er som følger (Alle tall er angitt på heksadesimal form):

SP = 0x0100, H = 0x03FF, TOS = 0x0001, OPC = 0xFFFF

a) Hva er den symbolske ekvivalenten til følgende mikroprogram?

Instruksjon 1: 00 110101 10000000 000 0100

Instruksjon 2: 00 111100 00100000 000 0111

Svar:

Instruksjon 1: $H = SP + 1$

Instruksjon 2: $TOS = TOS + H$

b) Hvilke verdier inneholder registrene over (SP,H, TOS og OPC) etter at de to mikroinstruksjonen har kjørt?

Svar:

Etter instruksjon 1:

$H = SP + 1 = 0x0100 + 1 = 0x0101$

Etter instruksjon 2:

$TOS = TOS + H = 0x0001 + 0x0101 = 0x102$

Totalt:

$SP = 0x0100, H = 0x0101, TOS = 0x102, OPC = 0xFFFF$

c) En ung student ønsker å legge til en ny instruksjon til IJVM. Instruksjonen heter IADDINC. IADDINC popper de to øverste elementene av stakken og summerer dem. Deretter legger den til 1 til svaret og legger svaret på toppen av stakken. De to opprinnlige elementene blir borte.

Han skriver følgende mikroprogram:

iaddinc1 MAR = SP-1;rd

iaddinc2 H = TOS

iaddinc3 MDR = MDR + H + 1; wr: goto main1

Programmet inneholder imidlertid to feil. Hva er de, og hva vil du gjøre for å rette på dem?

Svar:

1) SP peker på elementet over toppen av stakken, og 2) TOS blir ikke oppdatert. En mulig løsning vil være å endre programmet slik:

iaddinc1 MAR = SP = SP-1;rd
 iaddinc2 H = TOS
 iaddinc3 MDR = TOS = MDR + H + 1; wr: goto main1

d) Denne typen maskin er relativt treg, nevntre ting man kan gjøre for å øke ytelsen til maskinen.

Svar:

Her er det mange muligheter, og flere riktige svar. Poeng gis for alle gode forslag (superskalaritet osv). De tre mest nærliggende (og som blir vist i boken) er:

1. Instruction Fetch Unit – En enhet som henter instruksjoner og bufrer dem opp slik at man selv slipper å gjøre det eksplisitt i mikrokoden.
2. A – buss. En ekstra buss som gjør at man kan legge et vilkårlig register på A-inngangen og ikke bare H.
3. Pipelining. Sett inn latcher foran ALU og etter ALU slik at datapath blir delt opp i flere deler.

Oppgave 4 – Avhengigheter (15% - 5% på a) og 10% på b))

a) Hvilke typer avhengigheter har man mellom instruksjoner ?

Svar:

1. Sanne datavhengigheter (RAW). En foregående instruksjon skriver til et register som en senere instruksjon leser.
2. Utavhengigheter (WAW). En foregående instruksjon skriver til samme register som en senere instruksjon også skriver til.
3. Antiavhengigheter (WAR). En foregående instruksjon leser fra et register som en senere instruksjon skriver til.

b) Gitt følgende kodesnutt:

```
I1:  Add R2, R1, R0
I2:  Mul R1, R4, R5
I3:  Sub R6, R2, R1
I4:  Div R6, R0, R4
I5:  Add R6, R6, R1
```

Identifiser alle avhengigheter i kodesnutten, og angi typen avhengighet.

Svar:

Instruksjon 2: WAR til I1 (R1).

Instruksjon 3: RAW til I1 (R2), RAW til I2 (R1)

Instruksjon 4: WAW til I3 (R6)

Instruksjon 5: WAW til I4 (R6), WAW til I3 (R6), RAW til I4 (R6), RAW til I2 (R1)

Oppgave 5 (15% - 7,5 % på hver)

a) Hva er forskjellen på 3-adresse instruksjoner, 2-adresse instruksjoner, 1-adresse instruksjoner og 0-adresse instruksjoner? Gi kodeeksempler

Svar:

Forskjellen er i antall operander som er angitt eksplisitt. Eksempler:

3-adresse:

Add r1, r2, r3 ; adderer r2 og r3 og legger svaret i r1

2-adresse

Cmp r1, r2 ; sammenlikner r1 og r2

1-adresse

Jmp 0x100 ; hopp til adresse 100

0-adresse

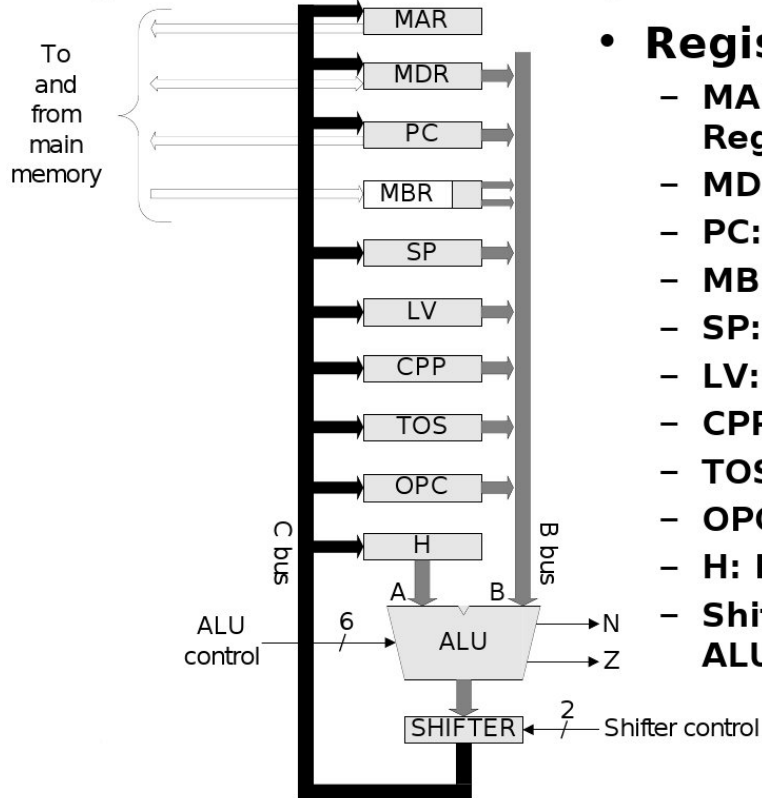
Return ; Returner fra prosedyre kall

b) Nevn fire forskjellige adresseringsmodi, og forklar hvordan de fungerer.

Svar:

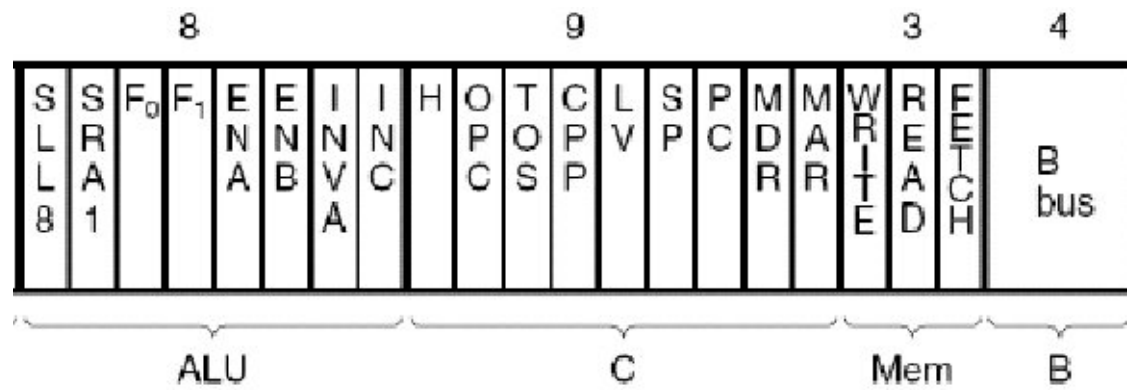
Adresseringsmodi er beskrevet på side 360 i Tannenbaum. Kort liste:

- Immediate Addressing
- Direct Addressing
- Register Addressing
- Register Indirect Addressing
- Indexed Addressing
- Base-Indexed Addressing

VEDLEGG – IJVM arkitektur

- **Register:**

- **MAR: Memory Address Register**
- **MDR: Memory Data Register**
- **PC: Program Counter**
- **MBR: Memory Buffer Register**
- **SP: Stack Pointer**
- **LV: Local variable**
- **CPP: Constant Pool Pointer**
- **TOS: Top of Stack**
- **OPC: OpCode register**
- **H: Holding register**
- **Shifter: shift register og ALU utverdi**



B bus registers

0 = MDR	5 = LV
1 = PC	6 = CPP
2 = MBR	7 = TOS
3 = MBRU	8 = OPC
4 = SP	9-15 none

F_0	F_1	ENA	ENB	INVA	INC	Function
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	\bar{A}
1	0	1	1	0	0	\bar{B}
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B - A
1	1	0	1	1	0	B - 1
1	1	1	0	1	1	-A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
1	1	0	0	0	1	1
1	1	0	0	1	0	-1

SVARARK – MULTIPLE CHOICE (Sett kryss)

Riv ut dette arket og lever det sammen med besvarelsen.

Oppgave	A)	B)	C)	D)
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				