



Norwegian University of Science and Technology
Engineering
The Department of Computer and Information Science

TDT4160

DATAMASKINER GRUNNKURS

EKSAMEN

4. DESEMBER, 2014, 09:00–13:00

Kontakt under eksamen:

Gunnar Tufte 73590356/97402478

Tillatte hjelpemidler:

D.

Ingen trykte eller håndskrivne hjelpemiddel er tillete.

Enkel godkjent kalkulator er tillete.

Målform:

Nynorsk

OPPGÅVE 1: OPPSTART, LITT AV KVART (25 %)

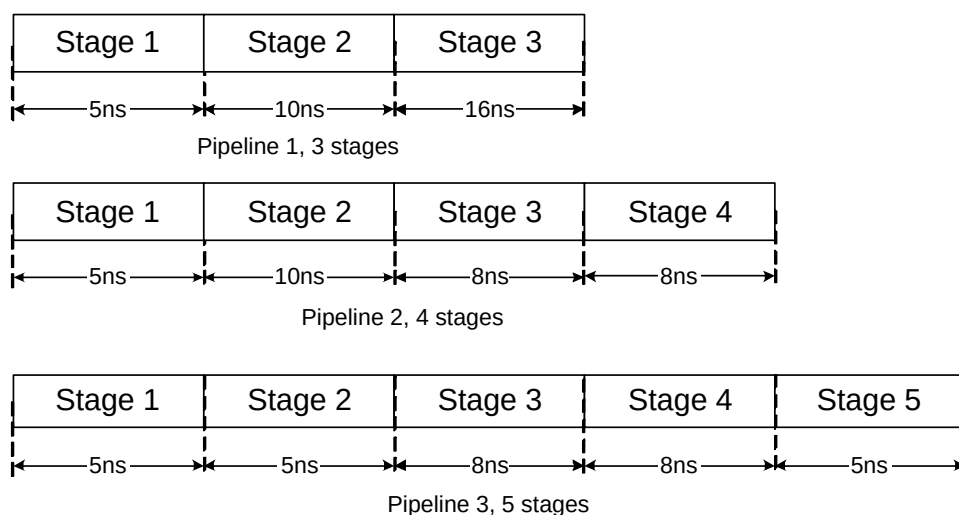
- a. Eit minnesystem med eit nivå hurtigbuffer (cache) der hovudminnet har en aksesstid på $1\mu s$ og hurtigbuffer har ein aksesstid på $0,05\mu s$ med et trefforholdstal (hit ratio) på 80 %.

Du har to valg for å auke ytelsen:

- 1 Endre designet på hurtigbufferet til å ha eit trefforholdstal på 95 %.
- 2 Bytte til eit hovudminne med aksesstid på $0,5\mu s$

Kva alternativ vil gi det raskaste minnesystemet? Kva er ytelseforbedringen?

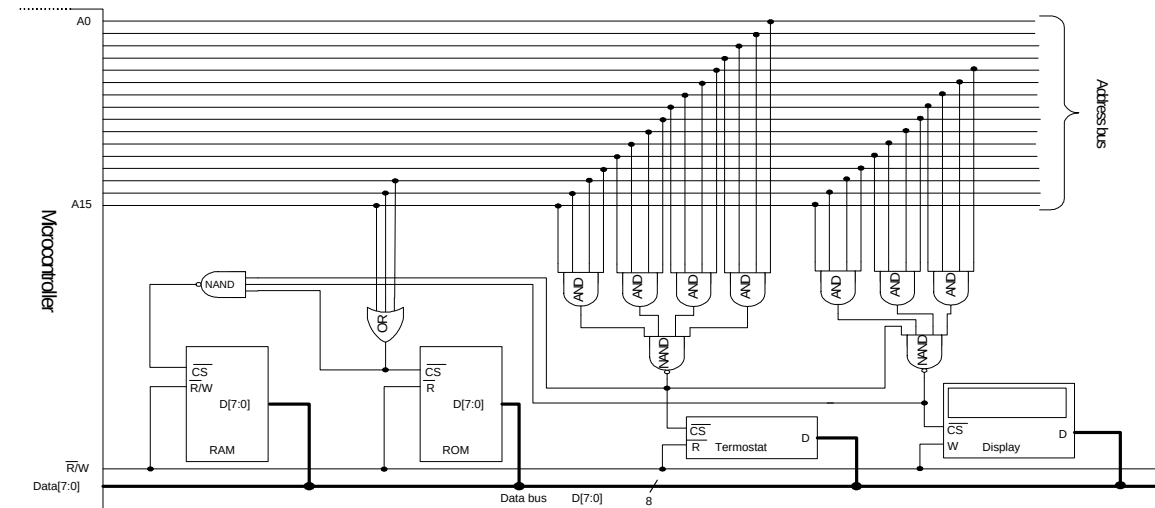
- b. Kvifor blir dynamisk minne som oftast brukt som teknologi i hovudminne?
- c. Forklar kort "Moore's law" (Moore sin lov).
- d. Er felles ISA nødvendig i eit "Heterogeneous" Chip Multiprocessor (CMP) system?
- e. Figur 1 viser tidsforbruket i tre ulike samleband (pipelines). Alle er ulike implementasjonar av den samme ISA. Estimer maksimal klokkefrekvens for dei tre samlebanda. Kva samleband har høgast ILP?



Figur 1: Samlebandtidsforbruk.

OPPGÅVE 2: DIGITALT LOGISK NIVÅ (25 % (15 % PÅ A, 5 % PÅ B OG C))

I figur 2 er det eksterne bussgrensesnittet for ein mikrokontroller vist. Det er brukt ein RAM-brikke for data og ein ROM-brikke for program. Ein termostad overvakar temperaturen. Eit display angir temperatur. Alle einingane nyttar eit aktivt lågt (logisk "0") CS (Chip Select)-signal.



Figur 2: Address decoding.

- a.
 - i) Tegn minnekart for systemet.
 - ii) Kva er det største programmet (i byte) som får plass i ROM.
- b.
 - i) Angi antal addresser som er tilgjengelig for diplayeiningen.
 - ii) Er det overlapp i minnesystemet? I så fall, korleis påverkar det funksjonaliteten?
- c. Ut i frå den informasjonen du har tilgjengeleg:
 - i) Er det mogleg å utvide systemet med fleire einingar? Forklar.
 - ii) Er påstanden om at systemet er programert til å bruke programstyrt I/O (polling/bussy wait) sansynlig? Forklar.

OPPGÅVE 3: MIKROARKITEKTUR OG MIKROINSTRUKSJONER (25 % (5 % PÅ A, 10 % PÅ B OG C))

Bruk vedlagte diagram i figur 4, figur 5, figur 6 og figur 7 for IJVM til å løyse oppgåvene.

- a. MBR-registeret er på 8 bit, men MPC-registeret er på 9 bit. Kva type instruksjonar nyttar dette niende bitet? Forklar kort.
- b. Lag mikroinstruksjon(ar) for følgjande IJVM-operasjon: $TOS = TOS + LV + OPC$.
Du trenger ikkje ta omsyn til Addr- og J-feltene. Oppgi korrekt bit-verdi for ALU-, C-, Mem- og B-feltene. – Sjå figur 5.
- c. IJVM-registerna i figur 4 er sett til følgjande verdier:

"SP": hex(AAAA),
"LV": hex(0505),
"CPP": hex(5555),
"TOS": hex(0004),
"OPC": hex(0005),
"H": hex(FF0A).

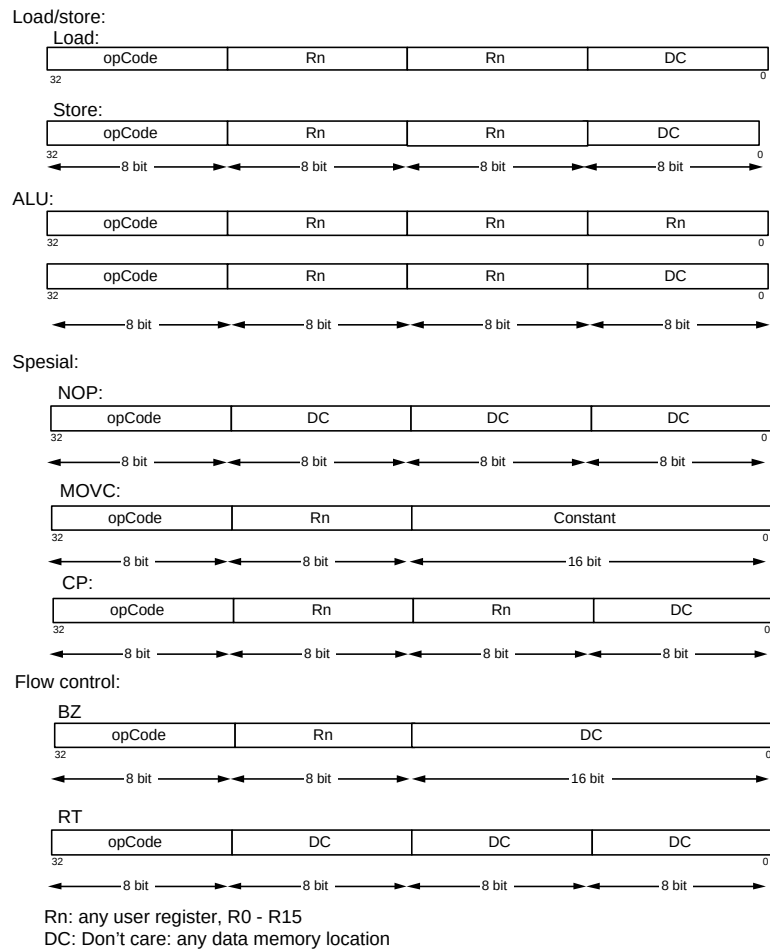
Kva er innhaldet i TOS-registeret etter at dei to følgjande mikroinstruksjonane har blitt utført? Oppgi svaret i hex-format.

- 1: ALU: 010100, C: 100000000, Mem: 000 og B: 0110
- 2: ALU: 001100, C: 001000000, Mem: 000 og B: 0100

Kva vil N- og Z-bita være etter utføringa av mikroinstruksjonane?

OPPGÅVE 4: INSTRUKSJONSSETT ARKITEKTUR (ISA)(25 % (5 % PÅ A, B, C OG 10 % PÅ D))

Ein sær enkel prosessor har ein load, ein store, 8 ALU-instruksjonar og nokre spesialinstruksjonar, inkludert NOP-instruksjonen og to flytkontrollinstruksjonar (flow control instructions). Prosessoren har 16 generelle register som er tilgjengeleg. Instruksjonsformatet for instruksjonane er vist i figur 3. Alle register og bussar er 32-bit. Prosessoren har ein Harvard arkitektur.



Figur 3: Instruction formats.

Instructions set:

LOAD: Load data from memory.

load Rn, Rn Load register Rn from memory location in Rn.

STORE: Store data in memory.

store Rn, Rn Store register Rn in memory location in Rn.

ALU: Data manipulation, register–register operations.

ADD Rn, Rn, Rn ADD, $Rn = Rn + Rn$. Set Z-flag if result =0.

NAND Rn, Rn, Rn Bitwise NAND, $Rn = \overline{Rn \cdot Rn}$. Set Z-flag if result =0.

OR Rn, Rn, Rn Bitwise OR, $Rn = Rn \text{ or } Rn$. Set Z-flag if result =0.

INV Rn, Rn Bitwise invert, $Rn = \overline{Rn}$. Set Z-flag if result =0.

INC Rn, Rn Increment, $Rn = Rn + 1$. Set Z-flag if result =0.

DEC Rn, Rn Decrement, $Rn = Rn - 1$. Set Z-flag if result =0.

MUL Rn, Rn, Rn Multiplication, $Rn = Rn * Rn$. Set Z-flag if result =0.

CMP, Rn, Rn Compare, Set Z-flag if $Rn = Rn$

Special: Misc.

CP Rn, Rn Copy, $Rn \leftarrow Rn$

NOP Waste of time, 1 clk cycle.

MOVC Rn, constant Put a constant in register $Rn = C$.

Flow control: Branch.

BZ, Rn Conditional branch on zero, $PC = Rn$.

RT Return, return from branch.

Rn: Any user register.

DC: Don't care.

- a. Dette er ein RISC-prosessor. Kvifor?
- b.
 - i) Kva type (instruction format) instruksjon er MOVC?
 - ii) Kva type adresseringsmodi (addressing modes) brukar MOVC?
 - iii) Kva type adresseringsmodi (addressing modes) brukar CP?
- c.
 - i) Er det mogleg å utvide med fleire register enn 32 utan å endre instruksjonslengda? Forklar kort.
 - ii) Er denne prosessoren generell? Forklar kort.
- d. Følgjande register har gitt innhald:
 - R1 = 0x0000 AAAA,
 - R2 = 0x0000 5555,
 - R3 = 0xAAAA AAAA,
 - R4 = 0x5555 5555.

Følgjande sudo kode køyrer:

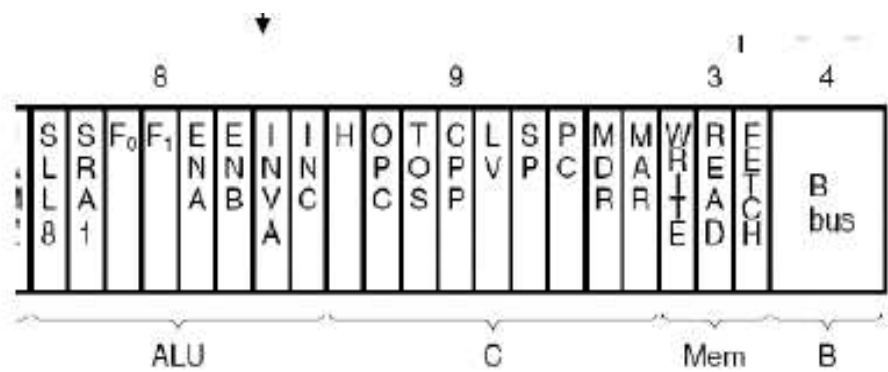
- :
- STORE R1, R3;
- STORE R2, R4;
- INV R3, R3;
- INV R4, R4;
- LOAD R1, R3;
- LOAD R2, R4;

- i) Kva er innhaldet i R1 og R2 etter at koden er køyrt?
- ii) I eit forsøk på å optimalisere endrast koden til:

- :
- STORE R1, R3;
- STORE R2, R4;
- CP R1, R2;
- CP R2, R1;

Er resultatet det samme eller ikkje? Kort forklaring.

IJVM appendix



B bus registers

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

Figur 5: Microinstruction format (IJVM).