

**Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap**



EKSAMENSOPPGAVE I FAG TDT4160 – DATAMASKINER GRUNNKURS

Faglig kontakt under eksamen: Gunnar Tufte

Tlf.: 93440

Eksamensdato: 11. august 2006

Eksamenstid: 09.00 – 13.00

Tillatte hjelpemiddel: D: Ingen trykte eller håndskrevne hjelpemiddel tillatt. Bestemt, enkel kalkulator tillatt.

Språkform: Bokmål

Oppgave 1 – Lager – 25 % (5 % på a, b og c; 10 % på d)

- Forklar forskjellen mellom little endian og big endian.
Hvorfor kan det være viktig å vite hvilken endian en prosessor bruker?
- Minnebrikker bruker gjerne D-flip-flops til å lagre data. Hver D-flip-flop har 4 innganger/utganger: D (data inn), Q (data ut), CK (klokke), CLR (nullstill). Forklar hvordan en 4-Mbit minnebrikke kan være organisert slik at den slipper å ha 16 millioner (4*4M) pinner.
- Forklar forskjellen mellom RAID 4 og RAID 5.
- Forklar fordeler og ulemper ved direkte avbildning (direct mapping) versus fullt assosiativ avbildning. Hvorfor er sett-assosiativ avbildning et godt kompromiss?

Oppgave 2 – Dataoverføring – 10 %

- Hva gjør en DMA-kontroller? Hvilke fordeler oppnås ved å bruke en slik?
- PCI Express overfører data over serielle forbindelser (lanes). Enheter som trenger å overføre store mengder data, kan bruke flere slike forbindelser i parallell. Forklar hvordan denne parallelliteten skiller seg fra parallell overføring i tradisjonelle busser som for eksempel PCI.

Oppgave 3 – Mikroarkitektur – 35 % (5 % på a, 10 % på b, 20 % på c)

- Forklar forskjellen mellom statisk og dynamisk forgreningspredikering.
Gi ett eksempel på teknikker for hver av disse to kategoriene.
- Lag mikroinstruksjon(er) for følgende IJVM-operasjon (se bort fra Addr og JAM):

$$SP = TOS + OPC$$
 Se vedlegget for oversikt over utførende enhet, funksjonstabell for ALU og formatet på mikroinstruksjoner.
- Figuren under viser et scoreboard for en superskalar prosessor som bruker i-rekkefølge-tildeling og i-rekkefølge-fullføring (in-order issue, in-order completion). Hvordan vil scoreboardet se ut for klokkesyklus 5 og 6? Anta følgende:
 - Instruksjon 5 er $R7 = R1 * R2$
 - Instruksjon 6 er $R1 = R0 - R2$
 - Addisjon og subtraksjon tar 2 klokkesyklur
 - Multiplikasjon tar 3 klokkesyklur
 - Det er alltid en passende funksjonell enhet ledig
 - Maksimalt 2 instruksjoner kan dekodes / starte utføring (issue) hver klokkesyklus

Cy	#	Decoded	Iss	Ret	Registers being read								Registers being written							
					0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	1	$R3=R0*R1$	1		1	1														
	2	$R4=R0+R2$	2		2	1	1										1	1		
2	3	$R5=R0+R1$	3		3	2	1									1	1	1		
	4	$R6=R1+R4$	–		3	2	1									1	1	1		
3					3	2	1									1	1	1		
4				1	2	1	1										1	1		
				2	1	1												1		
				3																

Figur 1: Scoreboard

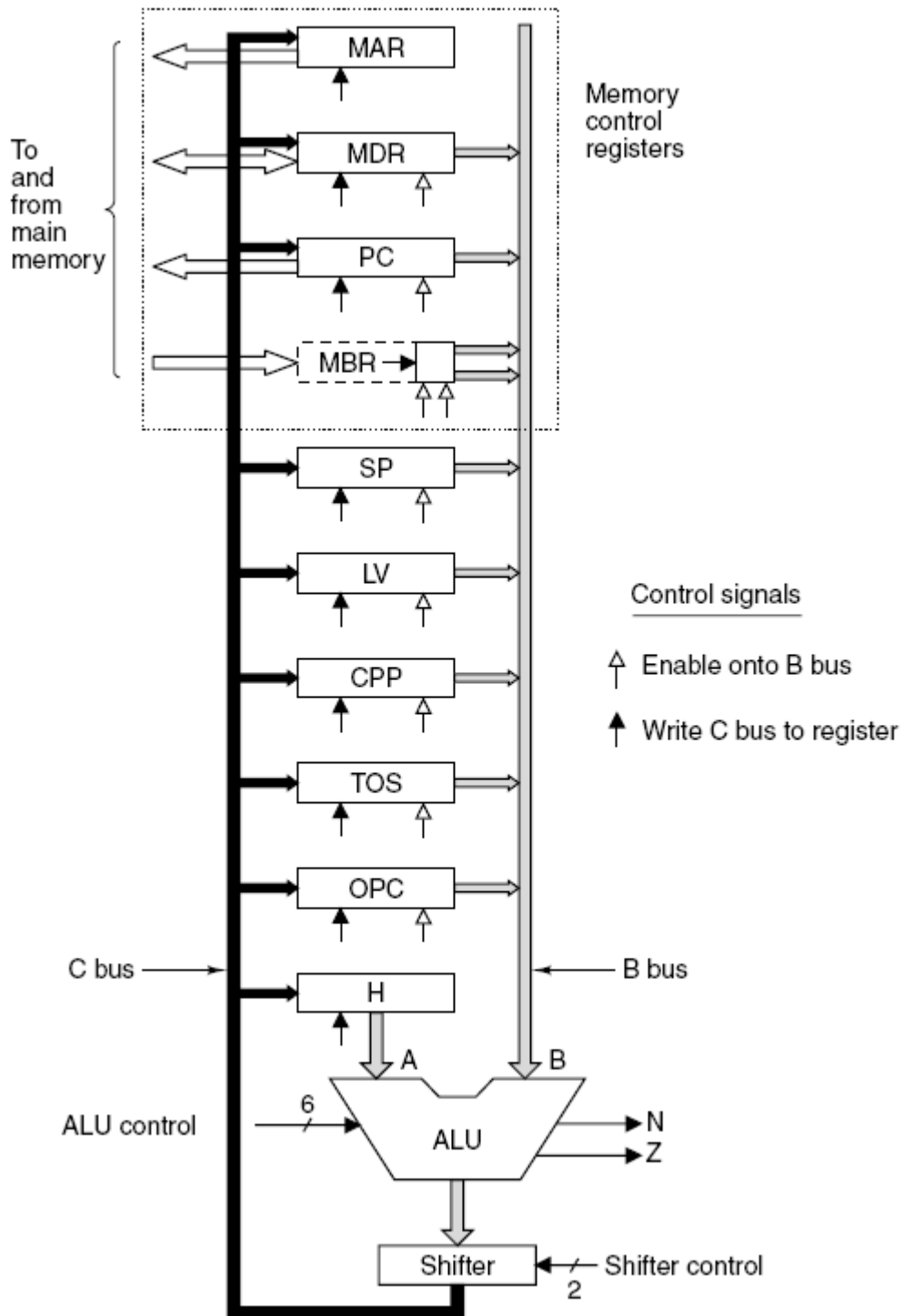
Oppgave 4 – Instruksjonssettarkitektur – 10 %

- Forklar hvordan registeradressering fungerer.
Hvorfor er dette et populært adresseringsmodus?
- Hva brukes templatefeltet i instruksjonsformatet til IA-64 til?

Oppgave 5 – Diverse – 20 % (5 % på a, 15 % på b)

- Forklar forskjellen mellom tolking (interpreting) og oversetting (compiling / translation).
Gi ett eksempel på fornuftig bruk av hver av disse.
- Beskriv kort de viktigste forskjellene mellom Pentium 4 og UltraSPARC III på ISA-nivået.
Gjør deretter det samme for mikroarkitektur-nivået.

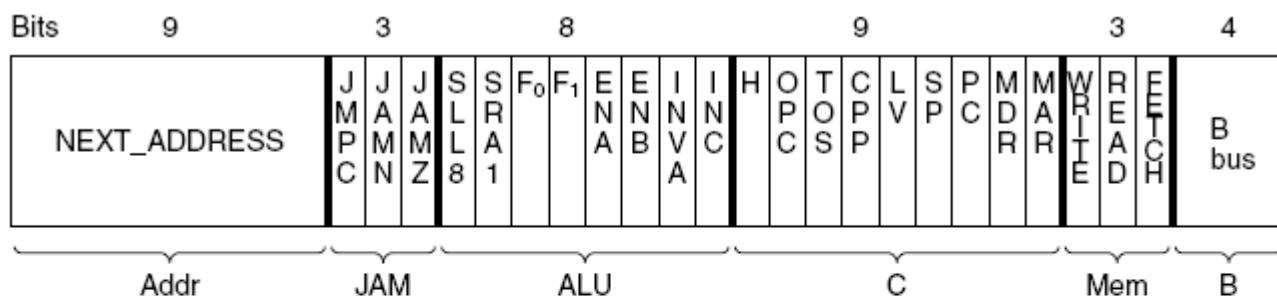
Vedlegg



Figur 2: Utførende enhet (IJVM)

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

Figur 3: Funksjonstabell for ALU (IJVM)



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 4: Mikroinstruksjonsformat (IJVM)