Løsningsforslag eksamen TDT4160 våren 2006

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

a) Forklar forskjellen mellom little endian og big endian.
Hvorfor kan det være viktig å vite hvilken endian en prosessor bruker?

Ord som er på mer enn en byte kan lagres på to måter. Man kan enten lagre disse med mest signifikante byte først og minst signifikante byte sist (big endian) eller omvendt (little endian). For eksempel vil 0x12 34 56 78 bli lagret 78 56 34 12 i little endian.

Det er viktig å vite hvilken type endian som er brukt når man skal overføre data fra en prosessor(-familie) til en annen. Hvis en prosessor leser 78 56 34 12 (eksempel over) og ikke vet om det er little eller big endian, vet den ikke om tallet faktisk er 0x12 34 56 78 eller 0x78 56 34 12.

b) 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.

To ting er viktig å innse: Vi trenger ikke uavhengig tilgang til alle bits (#1) og vi trenger ikke samtidig tilgang til alle bits (#2). Minnebrikker blir derfor (konseptuelt) organisert med D-flip-flops i en todimensjonal matrise. Alle bits i en rad leses/skrives samtidig (#1) og på et gitt tidspunkt er kun en rad aktiv (#2). Minnebrikken trenger derfor bare datapinner nok til en rad. I tillegg trengs pinner for å oppgi radnummer (adresse).

c) Forklar forskjellen mellom RAID 4 og RAID 5.

Begge lagrer data i striper og bruker paritet som redundans. Forskjellen er at RAID 4 har alle paritetsstriper på en disk, mens de er fordelt jevnt utover diskene i RAID 5. RAID 5 unngår derfor en flaskehals ved skriveoperasjoner.

d) Forklar fordeler og ulemper ved direkte avbildning (direct mapping) versus fullt assosiativ avbildning. Hvorfor er sett-assosiativ avbildning et godt kompromiss?

Fordeler med direkte avbildning: Raskt oppslag, enkel hardware, enkelt å finne hvor nye data skal inn.
Fordeler med fullt assosiativ avbildning: God utnyttelse av plass, høyere treffrate.

Ved direkte avbildning kan et ord fra hovedlagret kun ligge på en linje i hurtigbufferet. Ved fullt assosiativ avbildning kan ord ligge hvor som helst i hurtigbufferet. Ved sett-assosiativ avbildning kan et ord kun ligge i et sett, men det er plass til flere ord i hvert sett. Dette gjør at det er fort å finne frem samtidig som man har litt fleksibilitet når det gjelder plassering.

Oppgave 2 – Dataoverføring – 10 %

a) Hva gjør en DMA-kontroller? Hvilke fordeler oppnås ved å bruke en slik?

En DMA-kontroller er ansvarlig for å utføre dataoverføringer til/fra hovedlager uten innblanding fra prosessor. Prosessoren gir bare beskjed hva som skal overføres, hvor mye og hvor dataene skal skrives. Så utfører DMA-kontrolleren jobben slik at prosessoren kan gjøre noe annet i mens.

b) 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.

Ved vanlig parallell overføring er det for eksempel 8 bit fra et og samme ord som overføres samtidig på 8 linjer. Dermed er det viktig at bitene kommer frem samtidig, noe som kan være et problem ved høye klokkehastigheter (korte klokkepulser).

PCI Express bruker serielle forbindelser slik at 8 bit fra et ord overføres etter hverandre og ikke samtidig. Med 8 slike forbindelser (lanes) kan man overføre 8 bit samtidig, men dette er da 8 bit fra forskjellige ord. Overføringene over disse 8 forbindelsene trenger derfor ikke være synkronisert med hverandre.

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

a) Forklar forskjellen mellom statisk og dynamisk forgreningspredikering.
Gi ett eksempel på teknikker for hver av disse to kategoriene.

Forgreningspredikering er å gjette om en forgrening i programkoden vil bli tatt eller ikke. Dette er viktig for å kunne hente inn instruksjoner så tidlig som mulig i samlebånd.

Ved statisk forgreningspredikering gjetter man uten å ta hensyn til hva som har skjedd tidligere under utføring. For eksempel kan man alltid gjette hopp, aldri gjette hopp, alltid gjette hopp for BNE-instruksjoner, osv.

Ved dynamisk forgreningspredikering tar man hensyn til historie – hva som har skjedd når man har utført den aktuelle instruksjonen før. For eksempel kan man bruke en historiebit til å lagre om sist utføring medførte hopp eller ikke og bruke denne biten til å gjette neste gang. Man kan også bruke flere historiebits og historietabell.

b) 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.

Ser av den utførende enheten at TOS + OPC ikke kan utføres i en klokkesyklus. Må derfor først kopiere TOS (eller OPC) til H for så å utføre H + OPC (eller H + TOS).

Første mikroinstruksjon: ALU = B, C = H, Mem = 0, B = TOS
(eller 00010100 100000000 000 0111)

Andre mikroinstruksjon: ALU = A+B, C = SP, Mem = 0, B = OPC
(eller 00111100 000001000 000 1000)

c) 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 klokkesykler
- Multiplikasjon tar 3klokkesykler
- Det er alltid en passende funksjonell enhet ledig
- Maksimalt 2 instruksjoner kan dekodes / starte utføring (issue) hver klokkesyklus

					Registers being read								Registers being written							
Cy	#	Decoded	Iss	Ret	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	1	R3=R0*R1	1		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

Se figur 4-43 i læreboka (side 306) for fasit.

Oppgave 4 – Instruksjonssettarkitektur – 10 %

a) Forklar hvordan registeradressering fungerer.
Hvorfor er dette et populært adresseringsmodus?

Ved registeradressering inneholder instruksjonen et registernummer. Operanden ligger i registeret med det registernummeret.

Mye brukt siden registre er mye brukt (toppen av minnehierarkiet). Trenger heller ikke mange bits i instruksjonsformatet for å oppgi et registernummer (sammenlignet med for eksempel en hovedlageradresse). Dette gjør at instruksjonsformatet kan være kort og enkelt (mindre programkode, raskere å lese instruksjoner fra hovedlager, bedre utnyttelse av hurtigbuffer, …)

b) Hva brukes templatefeltet i instruksjonsformatet til IA-64 til?

To ting: Oppgi begrensninger i parallelliteten til instruksjonene og å spesifisere hvilken

instruksjon i instruksjonspakken som skal utføres av hvilken type utførende enhet. Begge deler gjør det enklere for IA-64 prosessoren å utføre instruksjonene i samlebåndet.

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

a) Forklar forskjellen mellom tolking (interpreting) og oversetting (compiling / translation). Gi ett eksempel på fornuftig bruk av hver av disse.

En datamaskin kan sees på som et hierarki av abstraksjoner der hvert nivå har sitt eget språk. For å få utført et program skrevet for det ytterste nivået, må det oversettes til språket på nivået innenfor. Dette kan skje på to måter: Ved tolking oversettes et program instruksjon for instruksjon hver gang det skal utføres, mens ved oversetting blir programmet oversatt en gang for alle.

Eksempel på tolking: Mikroprogrammert styreenhet.
Eksempel på oversetting: Kompilering fra C++ til assembly.

b) Beskriv kort de viktigste forskjellene mellom Pentium 4 og UltraSPARC III på ISA-nivået. Gjør deretter det samme for mikroarkitektur-nivået.

ISA-nivået:
- Pentium 4 er CISC, UltraSPARC III er RISC.
- Pentium 4 har få generelle registre.
- UltraSPARC III bruker registervindu.
- Pentium 4 har variabelt instruksjonsformat.
- Pentium 4 har mange adresseringsmodi.
- Pentium 4 har mange spesialiserte instruksjoner.

Mikroarkitektur-nivået:
- Pentium 4 er 32 bits, UltraSPARC III er 64 bits.
- Pentium 4 oversetter CISC-instruksjoner til uOperasjoner.
- Pentium 4 har ut-av-rekkefølge tildeling.
- UltraSPARC III har kortere samlebånd.
- UltraSPARC III har off-chip L2 cache.