

Oppgave 1 – Flervalgsspørsmål ("multiple choice") – 15 %

Denne oppgaven skal besvares på eget svarark sist i oppgavesettet. Dersom du finner flere alternativer som synes å passe, setter du kryss for det ene som passer best. For å unngå at gode tippere blir belønnet, vil et galt svar gi færre poeng enn om oppgaven forblir ubesvart.

- a) Hvilket av alternativene er **ikke** en grunn til å ha hierarkiske busser i en datamaskin?
1. Det er ønskelig å skille trege enheter fra raske enheter.
 2. En lang buss får gjerne høy transmisjonsforsinkelse.
 3. Med flere busser kan flere enheter overføre data samtidig.
 4. Det kan være en fordel med spesialiserte busser mot enkelte enheter.
 5. Lokalitetsprinsippet gjør det mindre sannsynlig at fjerne enheter blir brukt.
- b) Anta at en datamaskin har et hovedlager på 256 MB med aksesstid på 50 ns, og et 1. nivå hurtigbuffer på 16 KB med aksesstid på 5 ns. og treffrate på 90 %. 50 % av minneaksessene som ikke treffer dette hurtigbufferet, kan isteden hente data fra et 2. nivå hurtigbuffer på 128 KB. Gå ut ifra at miss i et hurtigbuffer medfører aksess mot hovedlageret. Hva må aksesstiden på 2. nivå-hurtigbufferet være for at aksesstiden for minnesystemet under ett skal være 10 ns?
1. 10 ns.
 2. 15 ns.
 3. 20 ns.
 4. 25 ns.
 5. Ingen av disse.
- c) Hvorfor er det gjerne ønskelig med flere nivå med hurtigbuffer?
1. Det forenkler tilbakeskriving fra hurtigbuffer til hovedlager.
 2. Hvis man ønsker en rask hurtigbuffer, er det begrenset hvor stor denne kan være.
 3. Det gjør det mulig å kjøre større programmer uten bruk av virtuelt minne.
 4. Det forenkler design av prosessorbrikker.
 5. Prosessoren slipper å arbitrere systembussen selv.
- d) Hvilket av alternativene er **ikke** en grunn til å lagre styreprogrammer til mikrokontrollere i flashminne?
1. Flashminne er overskrivbart slik at programvaren kan oppdateres.
 2. Innholdet bevares uten strømtilførsel.
 3. Kan slettes byte for byte slik at små feil kan fjernes enkelt.
 4. Prisen er lavere enn for EEPROM.
 5. Krever kun en transistor per bit slik at det kreves mindre plass enn for EEPROM.
- e) Gitt en 32 bits buss mellom prosessor og hovedlager med en klokkefrekvens på 33 MHz. Dersom du skulle lage en 64 MB minnemodul for dette systemet, hvilken alternativ ville du valgt?
1. Modul bestående av 4 stk. 16 Mbit brikker – hver brikke organisert som 4M x 4 bit.
 2. Modul bestående av 4 stk. 16 Mbit brikker – hver brikke organisert som 1M x 16 bit.
 3. Modul bestående av 32 stk. 16 Mbit brikker – hver brikke organisert som 4M x 4 bit.
 4. Modul bestående av 32 stk. 16 Mbit brikker – hver brikke organisert som 16M x 1 bit.
 5. Ingen av disse.

- f) Hvilket av alternativene er **ikke** en egenskap ved SDRAM?
1. Innholdet kan bare slettes blokk for blokk (blokk > byte)
 2. SDRAM støtter "burst mode" som gjør at sekvensielle data kan overføres uten separate adressefaser for hver datafase.
 3. Når prosessoren sender en forespørsel til en SDRAM-modul, blir prosessoren frigjort mens forespørselen behandles.
 4. SDRAM er synkronisert mot ekstern klokke (typisk klokke på systembuss).
 5. Bruker flere minnebanker slik at muligheten for parallellitet øker.
- g) Hvilket av alternativene er riktig når det gjelder FireWire?
1. Det er en parallell buss.
 2. Alle enheter må manuelt konfigureres med en unik ID.
 3. Egner seg spesielt for overføring av store datamengder over store avstander.
 4. Kan bruke "fair" arbitrerings for å unngå at noen enheter monopoliserer bussen.
 5. Benytter skjult arbitrerings.
- h) Hva er en fordel med "big endian" sammenlignet med "little endian"?
1. Det er enklere å foreta adressekonverteringer fra f.eks. 32 til 16 bit.
 2. Raskere sortering.
 3. Aritmetikk blir enklere.
 4. Den kan brukes til å korrigere enkeltbit-feil.
 5. Tar mindre lagringsplass.
- i) Hva er den viktigste årsaken til at forgreninger er et problem for samlebånd?
1. Adressen til neste instruksjon er ikke klar før etter at forgreningsinstruksjonen er utført.
 2. Det er ikke sikkert at instruksjonene ligger etter hverandre i hovedlageret.
 3. Forgreningsinstruksjonene tar lang tid å utføre.
 4. Forgreningsinstruksjoner er ofte involvert i ut- og anti-avhengigheter.
 5. Forgreninger er bare et problem ved superskalaritet.
- j) Delvis overlappende registervindu gjør at:
1. Ut- og anti-avhengigheter ikke blir et problem.
 2. Treffraten på hurtigbuffer øker.
 3. Det blir viktig å ha så store registre som mulig.
 4. Parametere kan overføres mellom subrutiner uten ekstra kopiering av data.
 5. Ingen av delene – registervindu kan per definisjon ikke være overlappende.
- k) Fem instruksjoner skal utføres på en superskalar prosessor som bruker i-rekkefølge-tildeling og ut-av-rekkefølge-fullføring (in-order issue/out-of-order completion). For instruksjonene gjelder:
- I1 og I2 bruker den samme funksjonelle enheten.
 - I4 trenger to sykler for utføring.
- Det superskalare samlebåndet har:
- Mulighet for å hente og dekode to instruksjoner samtidig.
 - Tre funksjonelle enheter.
 - To tilbakeskrivingsenheter.
- Hvor mange klokkesykler vil prosessoren bruke for å utføre disse fem instruksjonene?
1. 5 klokkesykler.
 2. 6 klokkesykler.

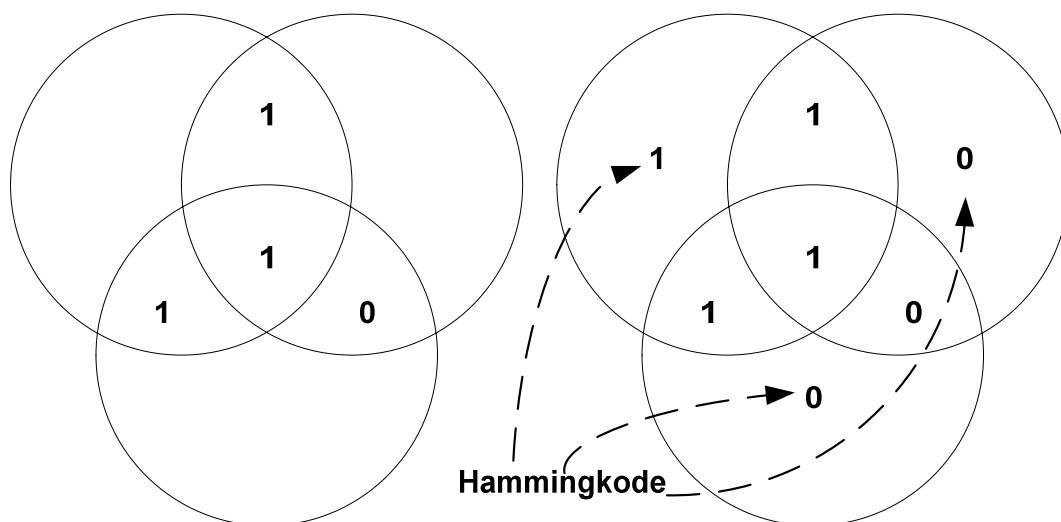
3. 6 klokkesyklar.
 4. 7 klokkesyklar.
 5. 8 klokkesyklar.
- l) Hvilken av de følgende påstandene stemmer **ikke** for Pentium 4.
1. Prosessoren inneholder en RISC-kjerne.
 2. Programmer skrevet for Pentium 4 består av RISC-instruksjoner.
 3. Samlebåndet har mange steg sammenlignet med tidligere prosessorer.
 4. Prosessoren bruker dynamisk forgreningspredikering.
 5. 1. nivås hurtigbuffer inneholder delvis dekodete instruksjoner (μops).
- m) MMX-instruksjoner er velegnet til multimedia-applikasjoner fordi:
1. De gjør det mulig med parallelle operasjoner på små datatyper.
 2. De kan brukes til overføring av store datamengder.
 3. Flyttall kan behandles med ekstra presisjon.
 4. De omfatter spesialinstruksjoner for dekodning av MP3.
 5. Prosessoren kan utføre andre instruksjoner mens MMX-instruksjonene utføres.
- n) SMP (Symmetrisk multiprosessor) regnes som:
1. SISD
 2. SIMD
 3. MISD
 4. MIMD
 5. Ingen av delene.
- o) Hvilket av de følgende alternativene er riktig for en hurtigbufferlinje i tilstanden "modified" i MESI-protokollen?
1. Innholdet i hurtigbufferlinjen er utdatert.
 2. Hovedlageret inneholder det samme som hurtigbufferlinjen.
 3. Innholdet i hurtigbufferlinjen kan finnes i andre hurtigbuffer.
 4. Eventuelle skriveoperasjoner går både til hurtigbuffer og til hovedlager.
 5. Ingen av disse alternativene er riktige.

Oppgave 2 – Busser – 7 %

- a) Det er vanlig å skille mellom serielle og parallelle busser. Dette gjelder både interne og eksterne busser.
1. Gi eksempler på to serielle og to parallelle busser.
 2. Hvilke fordeler har serielle busser sammenlignet med parallelle busser.
- b) Hva betyr det at PCI bruker skjult og sentralisert arbitrering?

Oppgave 3 – Lager – 9 %

- a) I datamaskiner er lageret organisert i et hierarki.
1. Tegn en enkel skisse over lagerhierarkiet i en typisk datamaskin.
 2. Hva oppnår man med å organisere lagret slik?
- b) Figur 1 viser hvordan man kan lage en enkeltfeil-korrigerende kode bestående av tre bit til bitstreng 1110. Forklar hvordan man kan utvide denne koden til også å oppdage dobbeltfeil.



Figur 1: Konstruksjon av Hammingkode for enkeltfeil-korrigering

- c) Tenk deg at du skal designe et system for masselagring av data basert på harddisker.
1. Hva kan du gjøre for å oppnå høy MTTF (Mean-time-to-failure)?
 2. Hva kan du gjøre for å oppnå lav MTTR (Mean-time-to-repair)?

Oppgave 4 – Samlebånd og superskalaritet – 12 %

- a) Et n -stegs samlebånd kan teoretisk gi en ytelse som er n ganger høyere enn uten samlebånd. Gi en kort oversikt over årsaker til at denne forbedringen ikke oppnås i praksis.
- b) To teknikker for dynamisk forgreningspredikering er bruk av historebits og historietabell. Forklar den viktigste fordel historietabell har i forhold til historebits.
- c) Gitt følgende kodesnutt:
- ```

I1: R1 ← R2 + R3
I2: R2 ← R1 + 5
I3: R2 ← R2 - 10
I4: R4 ← R1 + R5

```

1. Identifiser eventuelle sanne dataavhengigheter.

2. Lag en revidert kodesnutt som bruker registeromdøping til å fjerne eventuelle ut- og anti-avhengigheter.

d) Hvorfor er fast instruksjonsformat en fordel for superskalare prosessorer?

### Oppgave 5 – IA-64 – 7 %

Gitt følgende kodesnutt:

```
st8 [r4] = r12 // Lagre 8 bit fra register r12 til minneadressen gitt av r4
ld8 r6 = [r8] // Hent 8 bit fra minneadressen gitt av r8 og legg disse i r6
add r5 = r6, r7 // Adder innholdet i r6 og r7 og legg resultatet i r5.
```

a) Hvorfor er det ønskelig å flytte ld8-instruksjonen før st8?

b) En IA-64 kompilator har endret kodesnutten til:

```
ld8.a r6 = [r8]
st8 [r4] = r12
ld8.c r6 = [r8]
add r5 = r6, r7
```

Forklar hva som vil skje når prosessoren utfører denne siste kodesnutten.