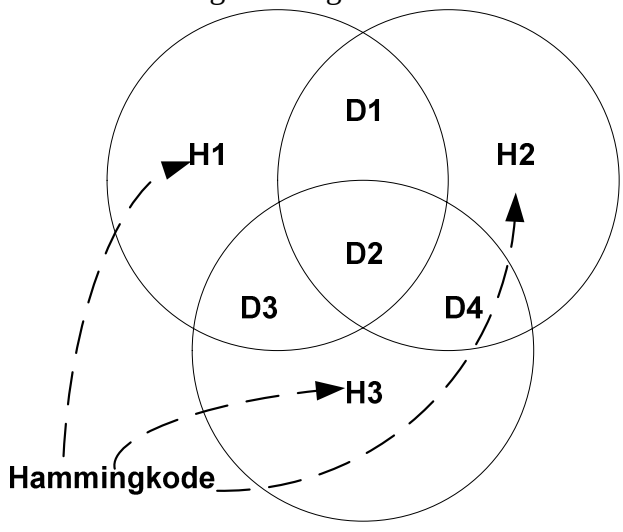
Oppgave 1 – Flervalgsspørsmål (”multiple choice”) – 15 %

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 de følgende punktene er **ikke** en konsekvens av Moores lov?
   1. Arbitrering på serielle busser bør helst skje distribuert.
   2. Økt prosessorytelse.
   3. Økt lagringskapasitet pr. minnebrikke.
   4. Økt forskjell mellom hastigheten til prosessor og minnesystem.
   5. God treffrate på hurtigbufre blir stadig mer viktig.

b) Tidsmultipleksing av data- og adresselinjer gjør at en buss:
   1. Må ha separate adresse- og datafaser i busstransaksjoner.
   2. Kan sende i begge retninger samtidig (”full duplex”).
   3. Ikke trenger kontroll-linjer.
   4. Ikke kan bruke sentralisert arbitrering.
   5. Ikke kan bruke distribuert arbitrering.

c) Figur 1 viser hvordan fire databit (D1–D4) og tilhørende Hammingkode-bit (H1–H3) er satt opp i et Venn-diagram. Gitt følgende verdier: D1 = 1, D2 = 1, D3 = 1, D4 = 1, H1 = 0, H2 = 1, H3 = 0. Hvilket utsagn er da riktig?
   1. Bit D1 er feil.
   2. Bit D2 er feil.
   3. Bit D3 er feil.
   4. Bit H3 er feil.
   5. Det finnes ingen korrigerbar enkeltbitfeil.



Figur 1: Venn-diagram for fire databit m/Hammingkode

d) Hammingkoden vist i Figur 1 utvides med en ekstra bit, H4, slik at den også kan brukes til å detektere dobbeltbitfeil. Gitt følgende verdier: D1 = 0, D2 = 0, D3 = 1, D4 = 1, H1 = 1, H2 = 1, H3 = 0, H4 = 0. Hvilket utsagn er da riktig?
   1. Bit D1 er feil.
   2. Bit D2 er feil.
   3. Bit D3 er feil.
   4. Bit H3 er feil.
   5. Det finnes ingen korrigerbar enkeltbitfeil.

e) Hvilket av alternativene under er **ikke** et kjennetegn ved SDRAM?
   1. Aksess er synkronisert med ekstern klokke.
   2. Støtter ”burst mode” for overføring av sekvensielle data.
   3. Har eget ”mode register” for konfigurering av minnebrikken.
   4. Utstrakt bruk av parallelle minnebanker for økt ytelse.
   5. Består av minnebrikker koblet til en smal men hurtig buss.

f) Moderne harddisker bruker såkalt ”multiple zoned recording”. Hva blir en konsekvens av dette?
   1. Det blir enklere å finne frem til en datablokk sammenlignet med bruk av konstant vinkelhastighet (CAV).
   2. Lesehastigheten varierer med hvor lese/skrive-hodet er på disken.
   3. Plassutnyttelsen blir bedre enn når man bruker konstant lineær hastighet (CLV).
   4. Aksesstiden blir dårligere enn for CLV, men bedre enn for CAV.
   5. Data kan lagres i flere soner samtidig.

g) Gitt 6 harddisker som kjøres med RAID 4. Hva skjer når man skal utføre en skriveoperasjon?
   1. Data blir skrevet direkte til en tilfeldig disk.
   2. En datastripe og tilhørende paritetsstripe blir skrevet.
   3. Gammel paritetsstripe leses inn før oppdatert data- og paritetsstripe blir skrevet.
   4. Gammel data- og paritetsstripe leses inn før oppdatert data- og paritetsstripe blir skrevet.
   5. Ingen av alternativene over er riktige.

h) Hvilket av de følgende utsagnene er riktig når det gjelder asynkron og isokron dataoverføring på en FireWire-buss?
   1. Asynkron overføring bruker pakker med fast lengde.
   2. Isokron overføring bruker kvittering (”acknowledge”).
   3. Bare en enkelt enhet overfører pakker i hver isokrone syklus.
   4. Enheter som genererer en jevn strøm med data bør bruke asynkron overføring.
   5. Asynkron overføring kan bruke både ”fair” og ”urgent” arbitrering.

i) Hvilken av de følgende påstandene er **ikke** riktig?
   1. USB er en seriell buss.
   2. USB støtter ”plug & play”.
   3. USB regnes som en erstatning for PCI.
   4. USB-enheter kobles sammen i en trestruktur.
   5. Spesielle enheter kalt USB-hub-er brukes for å tilby flere tilkoblingspunkter.

j) Hva vil det si at en feil er transient (myk)?
   1. Det er kort tid mellom feil og deteksjon.
   2. Feilen er midlertidig – hvis operasjonen repeteres får man sannsynligvis riktig svar.
   3. Det er kort tid mellom svikt og deteksjon.
   4. Den blir maskert ved hjelp av en komparator (to moduler i duplex).
   5. Den oppstår i en overgang mellom to tilstander.

k) Hvilken fordel gir en historietabell (”branch history table”) sammenlignet med å bare lagre historiebit med tanke på dynamisk forgreningspredikering?
   1. Dersom man kun lagrer historebit gjetter man gjerne feil siste gang en løkke blir kjørt.
   2. Hvis vi gjetter på hopp, slipper vi å vente til etter dekoding av operander før vi vet måladressen.
   3. Historietabellen har mulighet til å lagre mer detaljert historieinformasjon.
   4. Bruker man kun historiebit, vet man ikke hva man skal gjette første gang man kommer til en forgreningsinstruksjon.
   5. Bruker man for mange historiebit blir tilstandsdiagrammet for komplisert.

l) Gitt en RISC-prosessor med et firestegs samlebånd (hent instruksjon, les registre, utfør ALU-operasjon og skriv registre, gjør minneaksess). En kompilator har generert følgende kode til denne prosessoren:

```
I1: Load A ← M        ; Hent fra minne inn i register A
I2: Load B ← M        ; Hent fra minne inn i register B
I3: NOOP              ; Gjør ingenting
I4: Add C ← A + B     ; Legg innholdet i registrene A og B sammen og
                        legg resultatet i C.
```

   Hvorfor har kompilatoren satt inn en NOOP-instruksjon?
   1. Den bruker teknikken utsatt hoppinstruksjon (”delayed branch”).
   2. Tre vanlige instruksjoner etter hverandre kan overopphete prosessoren.
   3. For å unngå at prosessoren må detektere dataavhengigheten mellom Load og Add.
   4. Dårlig kode – et resultat av det semantiske gapet.
   5. Prosessoren støtter ikke ”internal forwarding”.

m) Hvorfor er det kompilatoren som tar seg av registeromdøping (”register renaming”)?
   1. Dette er feil – det er prosessoren som gjør registeromdøping.
   2. Kompilatoren har mye bedre tid til å gjøre dette enn prosessoren.
   3. Kompilatoren har komplett oversikt over programkoden.
   4. Registeromdøping krever deteksjon av ut- og antiavhengigheter.
   5. Det forenkler designet av prosessoren.

n) Hvilken av de følgende påstandene om hurtigbufferkoherens i multiprosessorsystemer er **ikke** riktig?
   1. Kompilatorbaserte teknikker er gjerne for konservative.
   2. Inkoherens kan kun oppstå hvis man bruker tilbakeskriving (”write back”).
   3. Maskinvarebaserte løsninger innebærer detektering av koherensproblemer i sanntid.
   4. Kompilatorbaserte teknikker gir enklere maskinvare.
   5. MESI er et eksempel på en distribuert maskinvareløsning.

o) Hva er en av motivasjonene for biologisk inspirert maskinvare ?
   1. Biologer er gjerne mer kreative enn teknologer slik at de kan påvirke utviklingsprosesser på en positiv måte.
   2. Enkle problemer kan gjerne løses automatisk.
   3. Slik maskinvare har gjerne et penere design.
   4. Utnytte overflødig regnekraft.
   5. Ønske om å kopiere den kontinuerlige utviklingsprosessen til naturlige systemer fordi det er denne som gjør slike systemer svært tilpasningsdyktige.

Oppgave 2 – Busser – 10 % (2,5 % pr. deloppgave)

a) Hva er fordelene og ulempene ved synkrone busser i forhold til asynkrone busser?

b) Bør lange busser være serielle eller parallelle? Begrunn svaret.

c) Hvilken fordel oppnår man ved å bruke skjult arbitrering?

d) Gi en sammenligning av egenskapene og bruksområdene til PCI og SCSI-bussen.

Oppgave 3 – Lager – 10 % (2,5 % på a og b, 5 % på c)

a) Vil innføring av hurtigbuffer på minnemodulene i hovedlageret kunne erstatte vanlig hurtigbuffer på prosessoren? Forklar hvorfor / hvorfor ikke.

b) Lagerteknologier kan grupperes etter hvilken måte data aksesseres på. Forklar kort følgende aksessmåter og gi et eksempel på en lagerteknologi for hver av dem: Sequential access, direct access, random access.

c) Gitt en datamaskin med følgende spesifikasjoner:

   - Prosessoren utfører gjennomsnittlig 2 instruksjoner pr. klokkesyklus
   - Gjennomsnittlig hver åttende instruksjon aksesserer lager.
   - Hurtigbufferen har 90% treffrate og aksesstid på 5 ns.
   - Hver hurtigbufferlinje som hentes inn fra hovedlager er på 32 byte.
   - Bussen mellom prosessor og hovedlager er på 200 MHz og er 32 bit bred.

   Hvor hurtig kan prosessoren være (i MHz) før bussen (i gjennomsnitt) ikke greier å levere data hurtig nok til å holde følge?

Oppgave 4 – Superskalaritet – 15 % (5% pr. deloppgave)

a) Denne oppgaven dreier seg om antiavhengigheter i superskalare samlebånd.

   i. Forklar hva antiavhengigheter er og hvordan de kan oppstå.
   ii. Forklar hva som kan gjøres for å unngå antiavhengigheter.

b) Denne oppgaven dreier seg om forgreningspredikering og Pentium 4.

   i. Hvorfor er det spesielt viktig med god forgreningspredikering for Pentium 4?
   ii. Forklar i korte trekk hvordan Pentium 4 utfører forgreningspredikering.

c) Gitt en RISC-prosessor med et superskalart samlebånd som har to dekodingsenheter, to heltallsenheter, en load/store-enhet, og to tilbakeskrivingsenheter. Alle heltall- og store-instruksjoner tar en klokkesyklus, mens load-instruksjoner tar to. Gå ut fra at det blir brukt registeromdøping, ut-av-rekkefølge tildeling og ut-av-rekkefølge fullføring.

   Som i læreboka, kan du anta at samlebåndet bruker ”internal forwarding”. Det vil si at hvis en instruksjon trenger resultatet av en tidligere instruksjon, holder det at de kommer etter hverandre i utføringssteget (trenger ikke vente på tilbakeskrivingssteget).

   En kompilator har generert følgende maskinkode til denne prosessoren:

```
I1: Load R4 <- [R5]      ; R5 gir minneadr. Legg i R4.
I2: R5 <- R5 + 4
I3: R7 <- R4 + R9
I4: Store [R6] <- R7     ; R6 gir minneadr. Hent fra R7.
I5: Load R4 <- [R5]
I6: R6 <- R6 + 4
I7: R7 <- R4 + R9
I8: Store [R6] <- R7
```

   Lag en tabell som viser hvordan instruksjonene beveger seg gjennom samlebåndet. Pass på å forklare hvordan du har tenkt.