

Tentamen i kursen EDA330

# Datorsystemteknik D

15/1 1999

**Tentamensdatum:** Fredag 15/1 1999 kl. 8.45 i sal MN

**Examinator:** Jonas Vasell

**Institution:** Datorteknik

**Förfrågningar:** Jonas Vasell (ankn. 1689)

**Lösningar:** Anslås måndag 18/1 på institutionens anslagstavla utanför laboratoriet

**Resultat:** Anslås senast torsdag 28/1 på institutionens anslagstavla utanför laboratoriet

**Rättningsgranskning:** Torsdag 28/1 klockan 11.30-12.00 hos Jonas Vasell

**Betygsgränser:** 3: 24-35 poäng, 4: 36-47 poäng, 5: 48-60 poäng

**Tillåtna hjälpmedel:** inga

**Allmänt:** För full poäng på en uppgift krävs både ett korrekt svar och en motivering. En bra motivering är minst lika viktig som ett korrekt svar. Redovisa noggrant alla gjorda antaganden utöver de som anges i uppgiftstexten. Skriv tydligt och använd gärna figurer. Maximal poäng på varje deluppgift anges inom parentes efter uppgiftstexten.

**Välgångsönskning:** Lycka till!

## Uppgifter (1-5):

1.

- a. Ange instruktionskoderna i binär form för det minsta antal MIPS-instruktioner som krävs för att ladda flyttalskonstanten 9,375 i register 4. Konstanten ska representeras i flyttalsformatet IEEE 754 med enkel precision (32 bitar). Exponenten i detta flyttalsformat är 8 bitar med bias 127. Konstanten får ej lagras i datamminnet. Använd endast de MIPS-instruktioner som anges i bilaga 1. (6 p)
- b. Visa med hjälp av figurer uppbyggnaden hos en 32-bitars ALU som kan utföra operationerna addition, subtraktion, logisk och, logisk eller, samt "mindre än"-jämförelse (som den är definierad i MIPS). Använd grundläggande grindkomponenter (och, eller, inverterare), multiplexers, och/eller heladdera-re. Ange vilka styrsignaler som krävs. Du får poäng även för en delvis korrekt lösning. (6 p)

2.

- a. Antag att vi har ett fyra-vägs partiellt associativt (four-way set associative) cacheminne med total storlek 16 ord och blockstorlek 1 ord (1 ord = 4 byte). Utbytesalgoritmen är LRU och cacheminnet är ursprungligen tomt. Följande sekvens av referenser till byte-adresser görs: 12, 30, 13, 76, 5, 14, 44, 116, 15, 61, 101, 76, 6, 40, 31. Ange för var och en av dessa referenser om den ger träff eller miss i cacheminnet. Ange också i tabellform innehållet i varje cache-block efter varje referens (ange tag för det block som ligger där), samt vilken hit rate (träffsannolikhet) som gäller för den aktuella sekvensen av referenser. (6 p)
- b. Vid undersökning av en viss dators uppbyggnad har man kommit fram till följande fakta om ett cacheminne: Det finns totalt 80 KB minne för hela cacheminnet, d.v.s. för såväl data som lagring av tags och statusbitar. Minnet förefaller uppbyggt som ett två-vägs partiellt associativt (two-way set associative) cacheminne. Typen av buss som förbinder cacheminnet med primärminnet indikerar att blockstorleken måste vara 1, 2, eller 4 ord (d.v.s. 4, 8, eller 16 byte). Adresserna till cacheminnet är 30 bitar. Din uppgift är nu att beräkna cacheminnets datakapacitet, d.v.s. hur många block det kan lagra och vilken blockstorlek som används. Gör beräkningen under antagandet att endast en valid-bit behöver lagras för varje block (inga andra statusbitar). Antag också att allt tillgängligt cacheminne (dvs alla 80 KB) utnyttjas och att datakapaciteten är en jämn tvåpotens. Undersök också om det finns mer än en möjlig lösning. (6 p)

3. Följande frågor gäller de MIPS-instruktioner som är sammanfattade i bilaga 1.
- För MIPS-instruktioner finns fyra olika adresseringsmoder (sätt referera till data och instruktioner). Beskriv var och en av dessa. (4 p)
  - I ett program vill man på ett ställe hoppa till adressen DEST om talet i register 15 är mindre än ( $<$ ) konstanten 0. Visa hur detta kan kodas i MIPS-assembly med den instruktionsuppsättning som finns i bilaga 1. Visa också hur man kodar om testvillkoret är "lika med" ( $=$ ), "större än" ( $>$ ), eller "större än eller lika med" ( $\geq$ ). Kodsekvenserna ska vara så korta som möjligt. (4 p)
  - Hur beräknas hoppadressen för ett villkorligt hopp? Hur långt kan man hoppa med en `beq`-instruktion? Hur kan man göra om man vill ha längre villkorliga hopp än vad som tillåts av `beq`? (4 p)
4. Många inbyggda datorsystem är tidskritiska vilket gör det mycket viktigt att kunna analysera deras realtidsegenskaper. En sådan analys är att ta reda på den längsta möjliga exekveringstiden för ett givet avsnitt kod (t.ex. en subrutin), så kallad WCET (Worst Case Execution Time). Ofta är det dock omöjligt att exakt förutsäga WCET eftersom exekveringstiden kan bero på faktorer man inte känner till, och man strävar då att hitta en minsta säkra gräns för WCET, d.v.s. en uppskattning av exekveringstiden som är så liten som möjligt men ändå inte underskrider WCET.

Din uppgift är att ta fram en sådan uppskattning av WCET för MIPS-koden nedan räknat i klockcykler från det att den första instruktionen hämtas (cykel 1) tills instruktionen efter den sista hämtas (cykel  $n+1$ ). Utgå från en MIPS-pipeline med stegen Instruction Fetch, Instruction Decode (inkluderar all hoppberäkning), Execute, Memory Access, och Write Back med funktionalitet som i kursboken. Antag att data forwarding tillämpas i alla fall det är möjligt och att `assume-not-taken` används som hoppstrategi (alltså inte `delayed branch` som i en riktig MIPS-implementering).

```

    addi    $4, $0, 64
L1:  lw     $8, 12($4)
    beq    $8, $0, L2
    addi   $8, $8, $5
    sw    $8, 12($4)
    j     L3
L2:  add    $6, $4, $0
    addi   $7, $7, 1
L3:  addi   $4, $4, -4
    bne   $4, $0, L1
    jr    $31

```

Antag att alla instruktions- och datareferenser går genom I-cache respektive D-cache. Båda cacheminnena är tvåvägs partiellt associativa och har 16 byte stora block. Cacheminnena är kopplade till primärminnet via en synkron buss med 32 bitars databredd och halva processorns klockfrekvens. Ett cacheminne kan reservera bussen för en hel blocköverföring. Att reservera bussen tar 1-12 busscykler beroende på kollisioner med annan busstrafik. Varje blockläsning inleds med att adressen skickas till primärminnet. Åtkomst i primärminnet av första ordet i ett block tar 6 busscykler, och åtkomst av påföljande ord tar en busscykel vardera. Överföring av ett ord data eller adress på bussen tar en busscykel. Innehållet i cacheminnena och primärminnet vid början av den aktuella MIPS-koden är okänt. (12 p)

5. Nedan följer ett antal frågor med tre svarsalternativ (1, X, 2) vardera, varav endast ett är rätt. Ställ upp svaren som en tipsrad. Varje rätt svar ger en poäng.
- a. SPEC är (1) en uppsättning benchmarks för linjära ekvationssystem. (X) ett kernel-baserat benchmark. (2) en standardiserad uppsättning benchmarks baserad på riktiga program.
  - b. Kapaciteten för de största DRAM-kretsarna ökar på tre år med ungefär en faktor (1) 2. (X) 3. (2) 4.
  - c. CPI för en processor beror på (1) kompilator och arkitektur. (X) arkitektur och implementeringens organisation. (2) implementeringens organisation och hårdvaruteknologi.
  - d. En superskalär processor (1) kan starta exekvering av flera instruktioner samtidigt. (X) har en extra lång pipeline. (2) har en speciellt kraftfull ALU.
  - e. En atomisk operation (1) är en processorinstruktion som utför en serie minnesåtkomster som inte får avbrytas av andra instruktioner. (X) är en processorinstruktion som utför en serie minnesåtkomster som får avbrytas av andra instruktioner. (2) är en processorinstruktion som utför en enda minnesläsning.
  - f. Booths algoritm (1) är en metod för flyttalsmultiplikation. (X) är en metod för heltalsmultiplikation. (2) är en metod för flyttalsdivision.
  - g. Write-update är en teknik för (1) minnekoherens. (X) skrivning i cacheminnet. (2) uppdatering av dynamiska RAM.
  - h. De minsta detaljer som kan implementeras med VLSI beräknas för 2010 vara ca (1) 0,3  $\mu\text{m}$ . (X) 0,07  $\mu\text{m}$ . (2) 0,003  $\mu\text{m}$ .
  - i. MIPS-instruktionen `slt` (1) skiftar ett tal åt vänster. (X) sätter lt-flaggan. (2) gör en jämförelse om ett tal är mindre än ett annat.
  - j. Stalling (1) är en teknik för omstrukturering av maskinkod under kompilering. (X) är en metod för pipelining av minnesåtkomster. (2) är ett sätt att lösa upp pipelinekonflikter.
  - k. Daisy-chain är benämning på (1) en metod för bussarbitrering. (X) en metod för seriekoppling av minneskretsar. (2) en teknik för utrullning av programsnurror under kompilering.
  - l. Dynamisk pipelineschemaläggning (1) är en konstruktionsmetod för pipelines. (X) innebär att maskininstruktioner kan exekveras i en annan ordning än vad som anges i maskinprogram. (2) innebär att en processor har multipla funktionsenheter för instruktionsexekvering.

SLUT

## Bilaga 1: MIPS maskininstruktioner

Se sammanställning av MIPS maskininstruktioner i kursboken.

