

Tentamen i kursen EDA330

Datorsystemteknik

22/8 1998

Tentamensdatum: Lördag 22/8 1998 kl. 8.45 i sal MG

Examinator: Jonas Vasell

Institution: Datorteknik

Förfrågningar: Jonas Vasell (ankn. 1689)

Lösningar: anslås tisdag 25/8 på institutionens anslagstavla utanför laboratoriet

Resultat: anslås senast torsdag 3/9 på institutionens anslagstavla utanför laboratoriet

Rättningsgranskning: tid och plats anslås tillsammans med resultaten

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. Översätt decimaltalet 5,25 till binär form i flyttalsformatet IEEE 754 med enkel precision (32 bitar). Exponenten i detta flyttalsformat är 8 bitar med bias 127. (2 p)
 - b. Det binära ordet $10101100010000000000000000000000_2$ representerar ett decimaltal med samma flyttalsformat som i förra deluppgiften. Beräkna vilket decimaltalet är. Ge svaret på formen heltal $\times 2^{\text{exponent}}$. (2 p)
 - c. Redogör för de fyra huvudstegen vid flyttalsaddition. Beskriv för varje steg vilken funktion det har. Definiera alla använda begrepp som är specifika för flyttalsrepresentation. (8 p)
2. Följande program körs på en MIPS-processor utan pipelining (en instruktion hämtas inte förrän föregående instruktion är helt avslutad):

```

200: lw   $3, 0($2)
204: lw   $4, 100($2)
208: slti $6, $2, 16
212: add  $5, $3, $4
216: sw   $5, 100($2)
220: addi $2, $2, 4
224: bne $6, $0, -28

```

Varje programrad ovan inleds med byte-adressen till det ord där instruktionen ligger lagrad. Alla konstanter, inklusive adresser, är decimala (bas 10).

Processorn ingår i ett datorsystem med ett gemensamt cacheminne för instruktioner och data. Cacheminnet är tvåvägs partiellt associativt (two way set associative) med blockstorleken 2 ord och total datalagringskapacitet 16 ord. Utbytesalgoritmen är LRU. Vid skrivmissar hämtas det aktuella blocket alltid in till cacheminnet.

- a. Vad gör programmet? Registret \$2 innehåller värdet noll innan programkörningen. (2 p)
- b. Lista den sekvens av minnesreferenser i form av byte-adresser som programmet ger upphov till. (2 p)
- c. Beskriv hur tag, index, och block-offset beräknas för varje minnesreferens i detta fall. (2 p)
- d. Ange för var och en av de första 10 minnesreferenserna om den ger träff eller miss i cacheminnet. Ange också i tabellform innehållet i varje cacheblock efter varje referens genom att ange tag för det minnesblock som ligger där. Utgå ifrån att cacheminnet är tomt från början. Vad blir hit rate för de första 10 minnesreferenserna? (4 p)
- e. Fortsätt som i deluppgift d för de nästkommande 20 minnesreferenserna. Starta med cacheinnehållet som det såg ut efter tionde referensen. Vad blir hit rate för de första 30 minnesreferenserna? (2 p)

3. Räkna upp fyra olika strategier för lösning av hoppkonflikter i en pipeline, beskriv hur varje strategi fungerar, samt redogör för varje strategis relativa för- och nackdelar. (12 p)
4. I ett visst datorsystem har man kopplat processorn till minnet via en asynkron buss som tillåter processorn att läsa 32 bitar åt gången från minnet. Bussen är 35 bitar bred, och kan delas upp i fyra signalgrupper: Request (1 bit), Data (32 bitar), Acknowledge (1 bit), och DataReady (1 bit). Varje läsning sköts genom ett handskakningsprotokoll mellan processor och minne. Data-signalerna på bussen används för att skicka både adresser och data.
 - a. Redogör för hur ett handskakningsprotokoll baserat på signalerna ovan fungerar då processorn ska läsa ett ord från minnet. Ställ upp en numrerad lista i tidsordning över alla tillfällen då en eller flera av signalerna på bussen ändras från det att läsningen inleds till den är helt avslutad. Beskriv vad som händer och varför vid varje sådant tillfälle. Rita också ett tidsdiagram som illustrerar förloppet. (8 p)
 - b. Ange två olika för- eller nackdelar med asynkrona bussar jämfört med synkrona bussar. (2 p)
 - c. Ange två typiska skillnader mellan en processor-minne-buss och en I/O-buss. (2 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. De minsta detaljer som kan implementeras med VLSI är idag c:a (1) 0,025 μm . (X) 0,25 μm . (2) 2,5 μm .
 - b. Antalet instruktioner som krävs för att utföra ett program på en viss processor beror på (1) kompilator och arkitektur. (X) arkitektur och implementeringens organisation. (2) implementeringens organisation och hårdvaruteknologi.
 - c. Write-invalidate är (1) en teknik för att stoppa felaktiga skrivningar. (X) ett cache-koherensprotokoll. (2) en typ av minnesåtkomst.
 - d. Hög rumslokalitet i minnessystem innebär att (1) minneskretsarna är tätt packade. (X) om en adress refereras så är det stor sannolikhet att en närliggande adress snart refereras. (2) att få adresser refereras.
 - e. En superpipelined processor (1) kan starta exekvering av flera instruktioner samtidigt. (X) har en extra lång pipeline. (2) har en speciellt kraftfull ALU.
 - f. SIMD står för (1) Single In-line Memory Device. (X) Single Instruction Multiple Data. (2) Single Interrupt Multiple Devices.
 - g. MIPS-instruktionen `jal` (1) hoppar till en adress som anges av ett register. (X) hoppar och lagrar återhoppadress. (2) används för extra långa hopp.
 - h. SCSI är en standard för (1) processor-minnesbussar. (X) bakplansbussar. (2) I/O-bussar.
 - i. För cache-minnen betyder fullt associativ att (1) varje cache-block är associerat med ett primärminnesblock. (X) varje block kan lagras var som helst i cache-minnet. (2) varje block kan lagras på exakt ett ställe i cache-minnet.
 - j. Skillnaden i snabbhet mellan minnen och processorer tenderar (1) att minska. (X) att vara oförändrad. (2) att öka.
 - k. LRU är (1) en utbytesalgoritm för cache och virtuellt minne. (X) en instruktion för subrutinanrop. (2) en skrivningsstrategi för virtuellt minne.
 - l. DMA står för (1) Dual Memory Access. (X) Direct Memory Access. (2) Dynamic Memory Access.

SLUT