國立交通大學 98 學年度碩士班考試入學試題

科目：計算機組織(1006)　　考試日期:98 年 3 月 15 日 第 4 節

系所班別：資訊系所跨組聯招　　組別：資訊聯招　　第 1 頁,共 4 頁

【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

須依題號作答，題號出現在後者不予計分。

1. **(08%)** Assume that *$s1* and *$s2* are used for the input and both initially contain the integers *a* and *b*, respectively. Assume that *$v0* is used for the output.

8000 4000h

```
        add     $t0, $zero, $zero
loop:   beq     $s2, $zero, finish
        add     $t0, $t0, $s1
        sub     $s2, $s2, 1
        j       loop
finish: addi    $t0, $t0, 100
        add     $v0, $t0, $zero
```

(i)(2%) Please describe in one mathematic formula what it computes (in terms of *a*, *b*).

(ii)(6%) If we assume the above code starting at location 8000 4000h in memory, and the register numbers of *$s1* and *$zero* are 17 and 0, respectively. What are the MIPS machine codes for the ***beq*** (OP code is 4) and ***j*** (OP code is 2) instructions above? Please answer both in hexadecimal format.

2. **(08%)** Suppose there was a 16-bit IEEE 754 floating-point format with 5 exponent bits.

(i)(2%) What would be the bias in the exponent?

(ii)(3%) Show the binary representation of the number 20.5_{10} using this 16-bit IEEE 754 floating-point format.

(iii)(3%) Which one would be the range of the numbers this format could represent?

(a) $1.0000\ 0000\ 00 \times 2^{0}$ to $1.1111\ 1111\ 11 \times 2^{31}$, 0

(b) $\pm 1.0000\ 0000\ 0 \times 2^{-14}$ to $\pm 1.1111\ 1111\ 1 \times 2^{15}$, ± 0, $\pm\infty$, NaN

(c) $\pm 1.0000\ 0000\ 00 \times 2^{-14}$ to $\pm 1.1111\ 1111\ 1 \times 2^{15}$, ± 0, $\pm\infty$, NaN

(d) $\pm 1.0000\ 0000\ 00 \times 2^{-15}$ to $\pm 1.1111\ 1111\ 1 \times 2^{14}$, ± 0, $\pm\infty$, NaN

3. **(12%)** Given a series of cache references as word addresses: ***16, 17, 18, 19, 20, 21, 22, 23, 48, 49, 17, 48, 49, 17, 5, 6,* and 7**. How many of references will cause compulsory miss and how many of them will cause conflict miss? Please give your answer for each cache separately.

(i)(6%) A direct-mapped cache with 4-word blocks and a total size of 16 words.

(ii)(6%) A 2-way set associative cache with 2-word blocks and a total size of 16 words. (using LRU replacement).

4. **(06%)** Consider a virtual memory system with the following properties:
   - 12-bit virtual byte address
   - 256-byte pages
   - 65536 bytes of physical memory

   The system uses a single level page table. The contents of the page table are ***partially*** shown below (where VPN: Virtual page number and PPN: Physical page number).

VPN	Valid	PPN	VPN	Valid	PPN
0	0	-	6	1	6
1	1	0xfd	7	0	7
2	1	0x48	8	0	-
3	0	-	9	1	0xfe
4	1	0x55	A	1	0xf2
5	1	0x32	B	0	-

   (i)(3%) What is the total size of the page table for each process on this machine, assuming that the valid, protection, dirty, and use bits take a total of 4 bits and that all the virtual pages are in use?

   (ii)(3%) Please convert the following virtual addresses into physical addresses: 0xae2, 0x258.

5. **(08%)** Virtually addressed cache is accessed with a virtual address, while physically addressed cache is accessed with a physical address.

   (i)(2%) Which kind of cache takes the TLB out of normal cache access path, reducing cache latency?

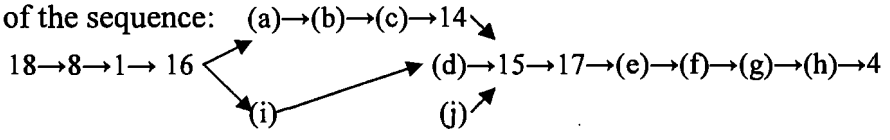
   (ii)(2%) Which kind of cache has aliasing problem?

   (iii)(2%) Assume *physically addressed* cache is used, is it possible to have TLB **miss**, but page table and cache **hit** for a data access?

   (iv)(2%) Assume *virtually addressed* cache is used, is it possible to have TLB **miss**, but page table and cache **hit** for a data access? Please explain your answer.

6. **(08%)** About RAID levels 1, 3, 4, and 5, please answer each of following statements as True or False.
(i)RAID systems rely on redundancy to achieve high availability and throughput.
(ii)RAID 1 (mirroring) has the highest redundancy overhead.
(iii)For small writes, RAID3 has the worst throughput.
(iv)For large writes, RAID3, 4, and 5 have the same throughput.

7. **(20%)** In Patterson and Hennessy's book there is a figure of processor design for multi-cycle implementation. You can draw the figure for your convenience when you answer questions. In the implementation there are many function blocks such as **(1)**memory, **(2)**MUX after instruction register, **(3)**memory data register, **(4)**register file, **(5)**MUX after memory data register, **(6)**A register, **(7)**ALU control, **(8)**MUX after PC, **(9)**sign extend, **(10)**MUX after B register, **(11)**B register, **(12)**shifter for ALU, **(13)**shifter for Jump, **(14)**MUX after A register, **(15)**ALU, **(16)**instruction register, **(17)**ALU out, **(18)**PC, and **(19)**MUX after ALU.
In the implementation there are many control lines such as **(20)**IorD, **(21)**ALUSrcA, **(22)**2-bit ALUOp, **(23)**PCWriteCond, **(24)**2-bit ALUSrcB, **(25)**MemRead, **(26)**IRWrite, **(27)**PCWrite, **(28)**2-bit PCSource, **(29)**MemWrite, **(30)**MemtoReg, **(31)**RegDst, and **(32)**RegWrite.
There are many actions in the implementation such as **(33)**A<=Reg[IR[25:21]], **(34)**ALUOut<=A+sign-extend (IR[15:0]), **(35)**Memory[ALUOut]<=B, **(36)**ALUOut<=PC+(sign-extend (IR[15:0])<<2), **(37)B**<=Reg[IR[20:16]], **(38)**if(A==B) PC<=ALUOut, **(39)**IR<=Memory[PC], **(40)**MDR<=Memory[ALUOut], **(41)**Reg[IR[20:16]]<=ALUOut, **(42)**ALUOut<=A op B, **(43)**PC<=PC+4, **(44)**Reg[IR[15:11]]<=ALUOut, **(45)**PC<={PC[31:28],(IR[25:0],2'b00)},and **(46)**Reg[IR[20:16]]<=MDR.
In following questions your answers should be in the form like **a=11** or **b=21A**.
(i)(10%) For a load instruction, it will go through **(18)** and other function blocks in sequence. Please reference above numbers and use **ONLY** numbers to complete unknowns of the sequence:



(ii) (03%) For R-type instructions, they will go through **(39)** and other actions in sequence. Please reference above numbers and use **ONLY** numbers to complete unknowns of the sequence: 39→(a)→(33&37 parallel) →(b)→(c).

(iii)(07%) Assume a *branch-on-equal* instruction is taken. It will use **(23D)** control line and other control lines in sequence. The symbol 'D' means the line is deasserted, and if a line is asserted the 'A' also must be indicated. You do not need to append 'D' or 'A' for 2-bit control lines. Please reference above numbers and use **ONLY** numbers and possible 'D' or 'A' to complete unknowns of the sequences: 23D→(a)→(b)→(21D&22 parallel)→20→28→(c)→ ((d)&22 parallel) →20→((e)&22 parallel)→(f)→28→(g).

8. **(30%)** Forwarding unit and data hazard detection unit are important parts of pipeline design. In pipeline we use IF/ID.RegisterRs, IF/ID.RegisterRt, IF/ID.RegisterRd, ID/EX.RegisterRs, ID/EX.RegisterRt, ID/EX.RegisterRd, EX/MEM.RegisterRs, EX/MEM.RegisterRt, EX/MEM.RegisterRd, MEM/WB.RegisterRs, MEM/WB.RegisterRt, and MEM/WB.RegisterRd to specify different source registers and destination registers in different stages. Like ID/EX.MemRead, some control lines are also prefixed by IF/ID, ID/EX, EX/MEM, or MEM/WB, though others are not prefixed. Let IF/IDWrite, ID/EXWrite, EX/MEMWrite, and MEM/WBWrite to determine whether pipeline registers get inputs or not.

(i) (18%) Draw the design of forwarding unit by using part of above signals and two groups of MUX selection lines (ForwardA and ForwardB).

(ii) (12%) Draw the design of data hazard detection unit by using part of above signals.