國立臺北科技大學 102 學年度碩士班招生考試

系所組別：2300 資訊工程系碩士班

第一節　計算機系統　試題

注意事項：

1. 本試題共八題，配分共 100 分。
2. 請標明大題、子題編號作答，不必抄題。
3. 全部答案均須在答案卷之答案欄內作答，否則不予計分。

一、(12%) Among the following statements about *process management*, please indicate whether each statement is true or false. If a statement is incorrect, please explain the reasons.

1. Shortest-job-first (SJF) scheduling algorithm will not result in starvation. (2%)
2. By using the multilevel feedback queue which dynamically moves processes between different queues, we can implement a form of aging to help prevent starvation. (2%)
3. In round-robin (RR) scheduling algorithm, the average turnaround time of a set of processes always improve as the time-quantum size increases. (2%)
4. Multithreaded programs can always provide better performance than a single-threaded solution. (2%)
5. A multithreaded program using multiple user-level threads achieves better performance on a multiprocessor system than on a single-processor system. (2%)
6. Interprocess communication using shared memory is usually faster than message passing. (2%)

二、(8%) Among the following statements about *memory management*, please indicate whether each statement is true or false. If a statement is incorrect, please explain the reasons.

1. Pure paging avoids the problem of external fragmentation. (2%)
2. Pure segmentation has the problem of internal fragmentation. (2%)
3. Pure paging requires less memory overhead than pure segmentation to maintain the address translation structures. (2%)
4. Both pure paging and pure segmentation have the ability to share code across processes. (2%)

三、(8%) Consider the following page reference string:

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1.

How many page faults would occur for the following *page-replacement algorithms* assuming a memory with **three** page frames? Please show the process of page replacement in details. (Note: Remember that all frames are initially empty, so your first unique pages will cost one fault each.)

1. Optimal algorithm. (4%)
2. Least-recently-used (LRU) algorithm. (4%)

四、(10%) Regarding the following descriptions about *file management*, please indicate whether each statement is true or false. If a statement is incorrect, please explain the reasons.

1. The contiguous allocation algorithm suffers from the problems of internal fragmentation and size declaration. (2%)
2. The major problem of linked allocation is that it can be used effectively only for sequential-access files. (2%)
3. Indexed allocation method supports efficient random access to disk blocks, without suffering from external fragmentation. (2%)
4. Similar to SJF algorithm for CPU scheduling, shortest-seek-time-first (SSTF) disk scheduling algorithm is optimal in terms of the total distance in disk head movements. (2%)
5. The performance of disk scheduling algorithms is not affected by file-allocation methods. (2%)

五、(12%) Answer the following questions regarding *process coordination.*

1. Why *disabling interrupts* are not appropriate for implementing synchronization primitives such as semaphores in multiprocessor systems? Please explain your answers. (4%)
2. Why *spinlocks* are not appropriate for single-processor systems yet are often used in multiprocessor systems? Please explain your answers. (4%)
3. Please describe the possible issues if we want to provide synchronization mechanisms in a *distributed system*. (4%)

注意：背面尚有試題

六、(25%) Given the following MIPS assembly code segments in Figure 1, which contain two functions **FOO** and **FUNC**. Please answer the following five problems.

```
MIPS Code
FOO: addi $sp, $sp, -16
     sw $s0, 12($sp)
     sw $s1, 8($sp)
     sw $s2, 4($sp)
     sw $s3, 0($sp)
     add $s0, $a0, $zero
     add $s1, $a1, $zero
     addi $s2, $zero, 1
     lw $s3, 20($s1)
     slt  $t0,  $s0,  $s3
     beq  $t0,  $zero,  EXIT
     addi $a0, $s0, $0
     lw $a1, 40($s1)
     jal FUNC
     add $s2, $s0, $v0
EXIT: lw $s3, 0($sp)
     lw $s2, 4($sp)
     lw $s1, 8($sp)
     lw $s0, 12($sp)
     addi $sp, $sp, 16
     jr $ra
FUNC: add $s0, $a0, $a1
     sll $s0, $s0, 0x1
     add $v0, $s0, $0
     jr $ra
```

Figure 1. MIPS assembly code segment

1. For the **MIPS assembly code** segment above, it suffers some mistakes that violate the MIPS calling convention. What are the mistakes and how to correct this code segment to be correctly executed? (6%)

2. What is the **equivalent C code** of the above MIPS assembly code in Figure 1? Please disassemble the MIPS code into the equivalent C code that contains two functions **FOO** and **FUNC**. Assume that the function FOO's two arguments are named as **a** and **b**, FOO's two local variables (in $s2 and $s3) are named as **c** and **d**, the function FUNC's two arguments are named as **e** and **f**, and FUNC's local variable is named as **g**, in the corresponding C code. (6%)

3. Here the function **FOO** is called by passing two arguments in the registers **$a0** and **$a1**, and the register **$a0** has an integer value **1**, and the register **$a1** contains an array address, where this array consists of 20 sequential integer values as: **1, 2, 3, 4, …, 20**. What is the **returned value** after the function **FOO** being correctly executed? (3%)

4. Then, given a typical MIPS processor with a five-stage pipeline, including: (1).**IF**-Instruction fetch; (2).**ID**-Instruction decode and register fetch; (3).**EX**-Execution or calculate effective address; (4).**MEM**-Access data memory; and (5).**WB**-Write back to registers. Assume that the register-write is done in the first-half of a clock cycle, and the register-read can be done in the second-half of the cycle, and all branch and jump outcomes are determined in the **EX** stage. Now we want to execute **your corrected MIPS assembly code in the previous problem 1** using this pipeline. Can you find all possible **data dependencies** in **your *corrected* MIPS code**? Please state the **MIPS instructions** and related **registers** that possible **data hazards** may occur. (5%)

5. Suppose that we now have an above-mentioned five-stage pipeline with full data hazard detection and forwarding units (i.e. forward all results that can be forwarded), and the branches and jumps are decided in the ID stage with a **predict-non-taken branch predictor, but there are no delay slots.** How many clock cycles will the function **FOO (which are called with the arguments given in the previous problem 3)** take to complete its ***corrected*** **MIPS code in problem 1**, and how many stalling cycles we need due to data and control hazards? (5%)

七、(16%) Consider a single processor system with the following specification:

- Data cache size is 4 KB (for data) and is 2-way set-associative. Its block size is 32 bytes. It is physically indexed and physically tagged. It uses LRU for replacement within a set and a write allocate write-back policy for writes.
- 4-way set-associative TLB with 64 total entries and an LRU replacement policy.
- Physical addresses of 32 bits.
- Virtual addresses of 32 bits.
- Byte addressable memory.
- Page size is 256 KB.

1. For each field listed below, please indicate the bits of the virtual address that correspond to it. Please show your work.
   (a).The virtual page offset (2%)
   (b).The virtual page number (1%)
   (c).The TLB index (2%)
   (d).The TLB tag (1%)

2. For each field listed below, indicate the bits of the physical address that correspond to it. Show your work.
   (e).The physical page offset (2%)
   (f).The physical page number (2%)
   (g).The cache block offset (2%)
   (h).The cache index (2%)
   (i).The cache tag (2%)

Your answer of each problem must contain the bit requirement and indicates the bits of the virtual or physical address that correspond to it. (For example: The total require bits of the problem and indicate the bits of the address [Bit_Address_End:Bit_Address_Start].)

八、(9%) Given a processor implemented by a single-cycle control and datapath shown in Figure 2. Assume that the functional blocks adopted to implement the datapath have the following latencies:

Inst.-Mem.	Add	Mux	ALU	ALU-Ctrl	Regs. Access	Data-Mem.	Sign-extend	Shift-left-2
500ps	200ps	50ps	200ps	80ps	250ps	550ps	60ps	50ps

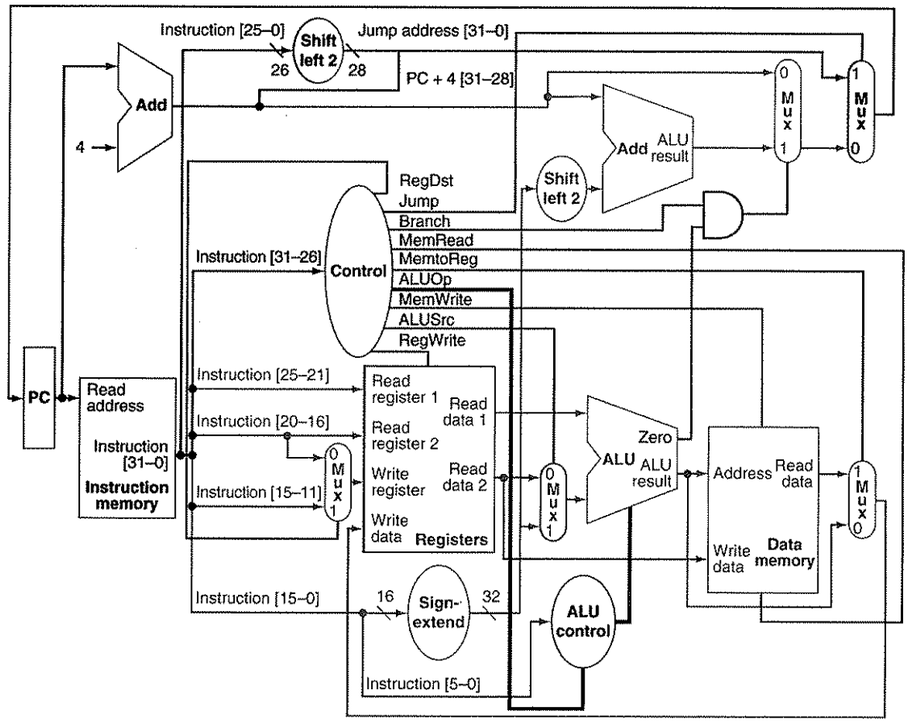

Figure 2. Diagram of the single-cycle processor of the control and datapath

1. In the control and datapath diagram shown in Figure 2, **which control signal is the most critical** that is needed to be quickly generated, and **how much time** can the control unit to generate this signal to avoid lengthening the critical path? (4%)

2. Assume that the control unit shown in Figure 2 requires the times to generate the individual control signals as the following table. What is the clock cycle time of this processor? (5%)

RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1200ps	1050ps	1050ps	700ps	1000ps	300ps	1100ps	250ps	1080ps