注意：考試開始鈴響前，不得翻閱試題，並不得書寫、畫記、作答。

國立清華大學 108 學年度碩士班考試入學試題

系所班組別：資訊工程學系

考試科目(代碼)：計算機系統(2302)

－作答注意事項－

1. 請核對答案卷（卡）上之准考證號、科目名稱是否正確。
2. 作答中如有發現試題印刷不清，得舉手請監試人員處理，但不得要求解釋題意。
3. 考生限在答案卷上標記「由此開始作答」區內作答，且不可書寫姓名、准考證號或與作答無關之其他文字或符號。
4. 答案卷用盡不得要求加頁。
5. 答案卷可用任何書寫工具作答，惟為方便閱卷辨識，請儘量使用藍色或黑色書寫；答案卡限用 2B 鉛筆畫記；如畫記不清（含未依範例畫記）致光學閱讀機無法辨識答案者，其後果一律由考生自行負責。
6. 其他應考規則、違規處理及扣分方式，請自行詳閱准考證明上「**國立清華大學試場規則及違規處理辦法**」，無法因本試題封面作答注意事項中未列明而稱未知悉。

國立清華大學 108 學年度碩士班入學考試試題

系所班組別：資訊工程學系

考試科目（代碼）：(2302)計算機系統

1. (7%) Consider the mechanisms needed to support a system call.
   a. (3%) Normally, a TRAP instruction is used to make a system call. Explain the steps involved in making a system call.
   b. (2%) How does the OS know which system call is being made?
   c. (2%) Explain why a subroutine CALL/RETURN instruction can not be used to make/return from a system call.

2. (6%) Consider process scheduling algorithms FCFS (first-come first-serve), SJF (shortest-job first), and Round-Robin (RR). Each one may be preemptive or nonpreemptive.
   a. (3%) Which algorithm(s) can potentially cause starvation? Explain by describing a scenario. Be sure to state whether preemptive or nonpreemptive.
   b. (3%) Which two of the six algorithms (preemptive or nonpreemptive FCFS, SFJ, and RR) are essentially the same? Why?

3. (6%) Consider a semaphore API where

   semaphore S(n)

   declares a semaphore named S initialized to the integer n, and functions wait(S) and signal(S) can be invoked on the semaphore.
   a. (3%) Show how to declare a semaphore L so that it works like a lock (mutex). How can acquire(L) and release(L) be implemented?
   b. (3%) Show how a semaphore can be used to ensure precedence (statement S1 of process P1 executes before statement S2 of process P2). Your pseudocode should declare a semaphore x and add the appropriate signal(x) or wait(x) call before or after statement S1 of P1 or S2 of P2.

4. (6%) Consider virtual memory support by OS.
   a. (3%) Does paging have internal fragmentation? External fragmentation? Explain.
   b. (3%) Virtual memory does not have to be implemented by paging. Describe one alternative way to implement virtual memory and give one advantage and one disadvantage compared to paging.

5. (8%) Coroutines are subroutines or tasks executing in turn. They are suitable for cooperative tasks, exceptions, event loops, and pipes. Design a coroutine in C with setjmp and longjmp functions. Try to use setjmp and longjmp to support the context-switching of a coroutine. Note that C is with setjmp and longjmp functions.
   - Setjmp(*jmp_buf env)* function saves the calling environment in *env.*
   - Longjmp(*jmp_buf env, int val*) function restores the environment saved by the most recent invocation of the respective setjmp() function. The parameter "val" is the value for the continuation of the program.

6. (4%) Grand Central Dispatch (GCD) is a technology for Apple Mac OS X and iOS. GCD identified extensions to the C and C++ language known as blocks. Explain how the GCD scheme works and the benefits of the GCD scheme.

7. (8%) Consider the OpenMP approach for process synchronization. Code fragments in Fig. 1 and Fig. 2 are examples used. Assume "result" is a shared variable by all threads and already properly declared. Please answer the following questions.
   a. (4%) Please identify the race condition in Fig. 1 and explain why the code fragment in Fig. 2 can handle race conditions.
   b. (4%) Further optimize the OpenMP code fragment in Fig. 2 to reduce the amount of critical sections.

```
Void update (int value) {
   . . .
  #pragma omp parallel for shared(result)
  for (i=0; i< 1000; i++ ) {
        result += (2*value*value + 5) *i;
  }
 }
```

Fig 1: Code fragment with OpenMP

```
Void update (int value) {
    . . .
  #pragma omp parallel for shared(result)
  for (i=0; i< 1000; i++ ) {
      #pragma omp critical
       {
          result += (2* value *value  + 5) *i;
        }
   }
  }
```

Fig 2: Code fragment with OpenMP critical linguistics

8. (5%) Explain Amdahl's law and describe how it guides the process of system performance improvement.

9. (6%) Assume the required number of parameters and required number of FP multiplication (FP MUL) computation in a convolution are expressed as below.

$$\#\,parameter = kernel_{size}{}^2 \times channel_{in} \times channel_{out} \quad (1)$$

$$\#\ FP\ MUL\ computation = \#\,parameter \times x \times y \quad (2)$$

Assume that a convolution layer *L* has the following parameters.

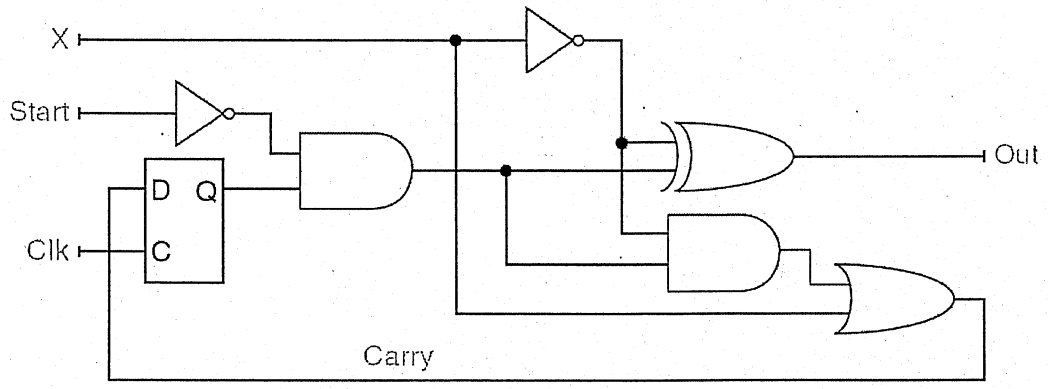
Layer	$Kernel_{size}$	$channel_{in}$	$channel_{out}$	x	y
L	5	3	100	200	200

The convolution includes three operations: FP multiplication (FP MUL), FP addition (FP ADD) and FP Load/Store (FP L/S). This convolution has the operation type breakdown as shown below. Assume the convolution is run at a processor with a 2GHz clock. Define CPO as the cycles per operation.

Convolution	FP MUL ratio	FP ADD ratio	FP L/S ratio	CPO (FP MUL)	CPO (FP ADD)	CPO (FP L/S)
L	80 %	15 %	5 %	1	0.8	0.75

How much is the computation time of this convolution?

10. a. (4%) What is the function for the following circuit diagram?



b. (4%) What is the shortest operation time for the circuit if you designed for a 64-bit operation? Assume the delay time for D Flip-Flops, NOT, AND, OR, and XOR gates are 180ps, 4ps, 120ps, 100ps, 140ps, respectively. You may ignore the flip-flop setup time.

11.(6%) Given an assembly code as below,

```
{      find:   move   $t0, $a0
               sll     $t1, $a1, 2
               add     $t1, $t1, $a0
       loop:   beq     $t0, $t1,done
               lw      $t2, 4($t0)
               sw      $t2, 0($t0)
       skip:   addi    $t0, $t0,4
               b       loop
       done:   jr      $ra
}
```

a. (4%) Write the function of the above MIPS assembly code into a pointer-based C code.

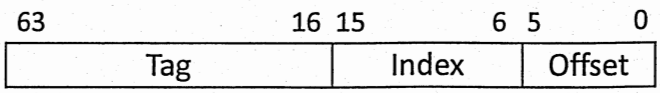
b. (2%) What is the number of temporary registers used in the above MIPS code?

12.(6%) Let $x = 1.9760 \times 10^4$, $y = -4.052734375 \times 10^{-2}$.

Calculate x*y.

Assuming x, and y are stored in the 16-bit format with sign, exponent and mantissa. The leftmost bit is the sign bit, the exponent is 5 bits wide with a bias of 31, and the mantissa is 10 bits long. A hidden 1 is assumed. Assume 1 guard, 1 round bit and 1 sticky bit, and round to the nearest even. Write the answer in 16-bit floating point format in hexadecimal representation.

13.(6%) Consider a direct-mapped cache design with a 64-bit byte-address of the following format:



a. (2%) What is the cache size in kibibytes (i.e., KiB) for the data storage? (note: 1KiB = 2^{10} bytes)

b. (2%) What is the ratio between the actual memory size over the data storage size in (a)?

c. (2%) Given the cache size in (a) as the fixed design parameter, you want to explore the different combination of Index and Offset widths. With a larger block size, will the ratio in (b) be larger or smaller? You should state a brief reason.

14. (6%) Consider the target application with the following statistics for your team to design a cache:

230 data reads per 1000 instructions;

120 data writes per 1000 instructions.

With the block size of 128 bytes, the instruction cache miss rate is 0.4%, and the data cache miss rate is 2%. Note that the minimum CPI is 1 (i.e., at most one instruction is fetched per cycle). Assume that each miss generates a request for one block.

a. (3%) The read and write bandwidths between RAM and the cache is 1 byte/cycle in the initial design. For a write-through, write-allocate cache, what is the expected CPI?

b. (3%) Your team wants to improve the CPI to less than 1.5 by increasing the read and write bandwidth between RAM and the cache. However, the bandwidth is restricted to 2^k bytes/cycle, where k is an integer and $k \geq 0$. What is the smallest k to achieve the goal? What is the resultant CPI?

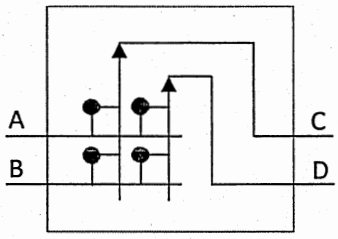
15. (6%) For one specific computation load, the execution time of the parallelizable part is $6t^2$, and that of the sequential part is $3t$, where t denotes the problem size. Assume that the parallelizable part of the computation workload can be ideally speeded up by increasing the number of processors. The execution time of the sequential part is not affected by increasing the number of processors.

a. (3%) For the problem size $t = 10$, what is the speedup with the 15-processor implementation (i.e., $n = 15$) as compared with the

single processor implementation?

b. (3%) For (a), assume that the load is not well balanced. Two of the processors equally share 20% of the parallelizable load while all the others share the rest. What is the speedup?

16. (6%) For popular multistage network topologies of multiprocessor nodes, a *fully connected network* (or *crossbar network*) allows any processor node to communicate with any other node in one pass through the network. For a crossbar network of eight processor nodes, 64 switches are needed. Considering a switch box with two inputs and two outputs in the *Omega network* as the following diagram.



a. (2%) Draw the Omega network topology of eight processor nodes (i.e., $P_0, P_1, \ldots, P_7$) with switch boxes.

b. (2%) What is the advantages and disadvantages of the Omega network as compared with the crossbar network in terms of the number of switches and the functionality?

c. (2%) The simple analysis of these networks with the number of switches ignores important practical considerations in the construction of a network. List two practical considerations that will affect the realistic implementation due to the different distance of each link in a network.