國立清華大學 105 學年度碩士班入學考試試題

系所班組別：資訊工程學系

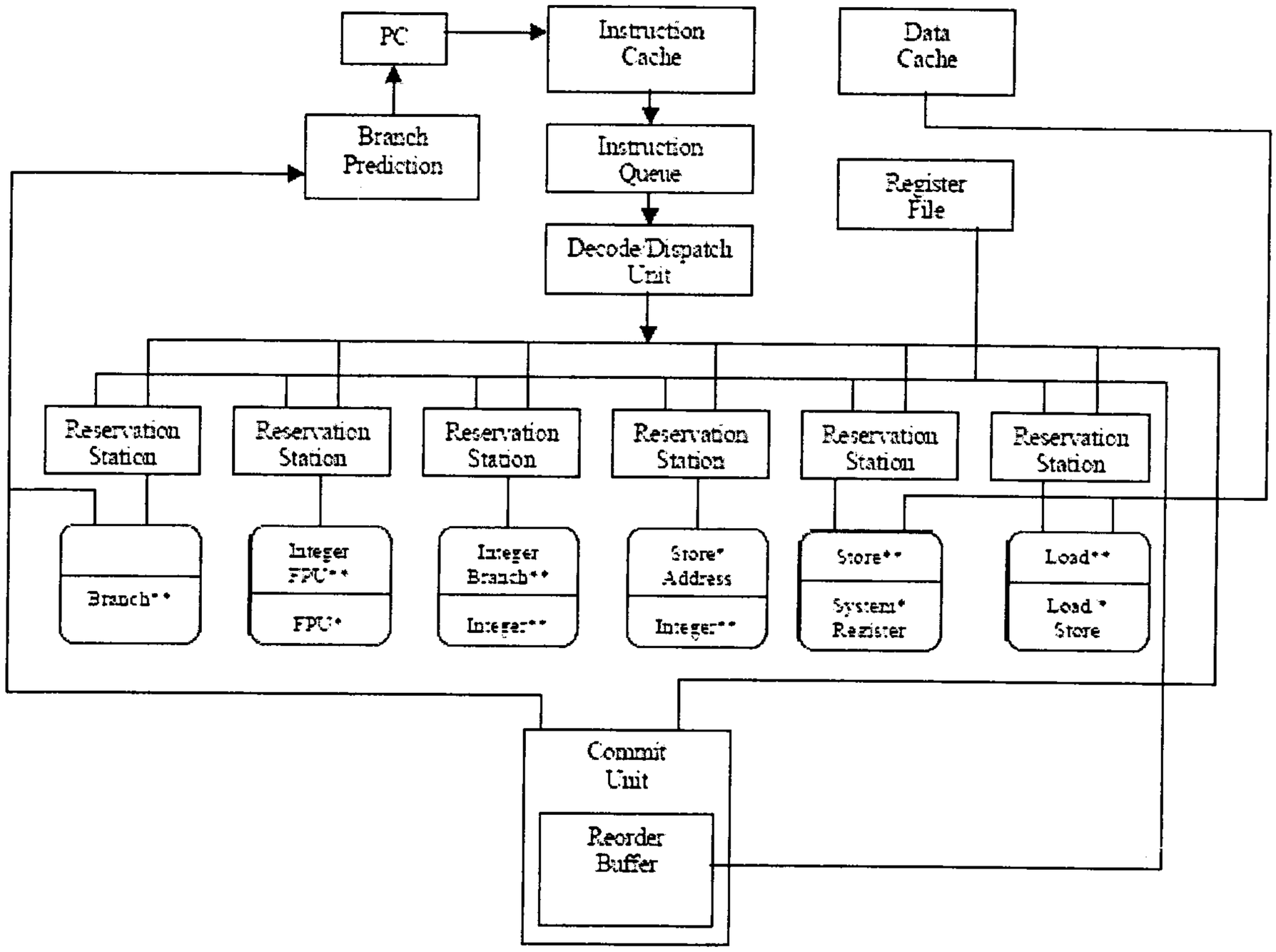
考試科目（代碼）：計算機系統(1902)

1. (8%) For the following questions we assume that the pipeline contains 5 stages: IF, ID, EX, M, and W and each stage requires one clock cycle. A MIPS-like assembly is used in the representation.
   (a) Explains the concept of forwa rding in the pipeline design.(4%)
   (b) Identify all of the data dependencies in the following code. Which dependencies are data hazards that will be resolved via forwarding? (4%)

   ADD $2, $5, $4
   ADD $4, $2, $5
   SW $5, 100($2)
   ADD $3, $2, $4

2. (9%) Suppose out of order execution is adopted in a pipeline design. The figure below gives a generic pipeline architectures with out-of-order pipeline design.
   (a) Explain the needed work in the reservation station and reorder buffer, respectively, in order to support out of order execution. (4%)
   (b) In order to reduce power usage, we want to power off a floating point unit when it is no longer in used. The following code fragment uses a new instruction called Power-off to attempt to turn off floating point multiplier when it is no longer in used. To deal with out of order executions, what are the work needed for reservation station and reorder buffer to avoid the "Power-off @multiplier" instruction to be moved in front of the set of multiply instructions. (5%)

```
mul.d   $f2,   $f4,   $f6
mul.d   $f1,   $f3,   $f5
mul.d   $f7,   $f1,   $f2
Power-Off   @multiplier
/* Integer operations in the rest of the assembly code*/
ADD $2, $5, $4
. . .
END
```



3. (8%) A non-pipelined processor A has an average CPI (Clock Per Instruction) of 4 and has a clock rate of 40MHz.
   (a) What is the MIPS (Million Instructions Per Second) rate of processor A (2%)
   (b) If an improved successor, called processor B, of processor A is designed with four-stage pipeline and has the same clock rate, what is the maximum speedup of processor B as compared with processor A? (4%)
   (c) What is the CPI value of processor B? (2%)

4. (8%) Denote respectively s_i and c_i the Sum and CarryOut of a one-bit adder Adder(a_i,b_i), where a_i and b_i are input.

(a) For a 16-bit addition, if we use the ripple adder, how many gate delay is required for generating c_{16} after bits A, B, and c_0 are applied as input? (2%)

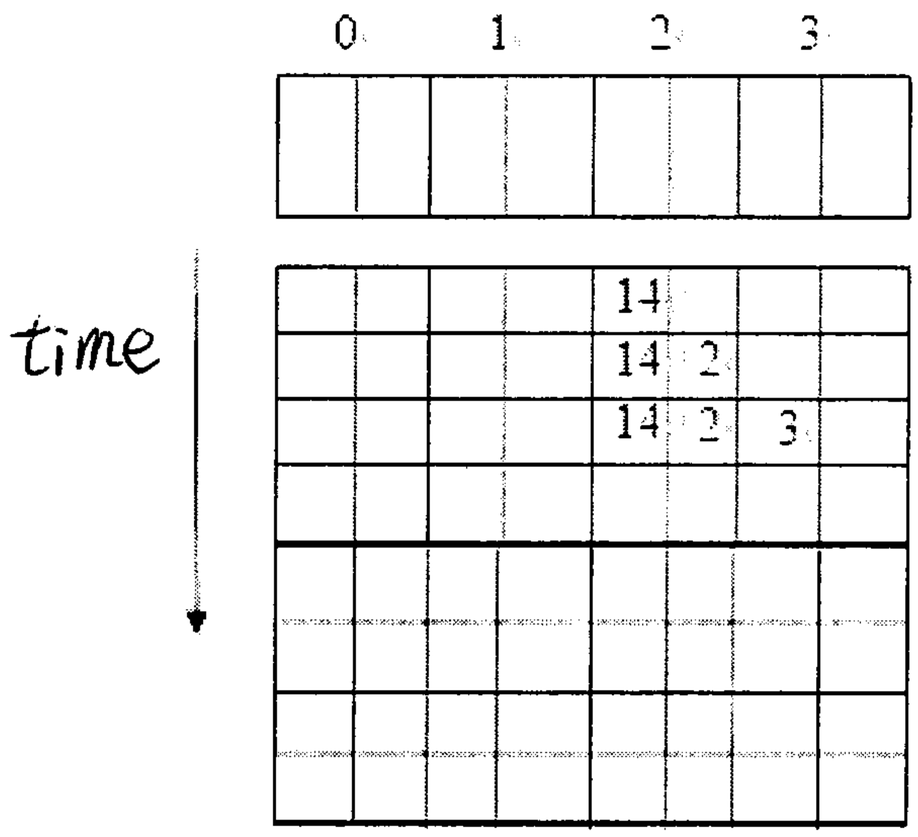
(b) If we use the carry-look-ahead adder, where the 16-bit adder is divided into four 4-bit adders. In each 4-bit adder there is a 4-bit carry-look-ahead circuit. Let $p_i = a_i + b_i$ and $g_i = a_i * b_i$ and assume there is one gate delay for each p_i and g_i and two gate delays for each carry c_4, c_8, c_{12}, c_{16} and final three gate delay for s_{15}. How many gate delay is required for generating c_{16} after A, B, and c_0 are applied as input? (3%)

(c) Another way to apply look-ahead circuit is to add a second-level look-ahead circuit. With the second-level look-ahead circuit, how many gate delay is required for generating c_{16} after A, B, and c_0 are applied as input? (3%)

5. (10%) Please fill "?" with the answer of "up" or "down" in the following. For example, when associativity goes up, the access time goes up.

Design change	Effect on miss rate	Possible effects
Associativity (up)	**conflict miss ?**	**access time up**
Cache size (down)	**capacity miss ?**	**access time ?**
Block size (down)	**spatial locality ?**	**miss penalty ?**

6. (7%) In 2-way cache, there are already 3 numbers in the sequence of 14, 2, 3 in the cache. The following table shows the time sequences when 14, 2, and 3 are inserted into the 2-way cache. Please insert 6 numbers after and show the time sequence using LRU. 6, 10, 22, 42, 11, 27



7. (10%) Suppose that there are a number of processes, P_1, P_2, …, in a computer. Their arrival times are denoted by a_1, a_2, …; their CPU burst times are denoted by b_1, b_2, …. Processes are scheduled by the *Shortest-Remaining-Time-First scheduling* algorithm. Suppose that the context switching takes 0 second.

   (a) Consider the case with 2 processes where $a_1 = 0$, $b_1 = 10$, $a_2 = 4$, and $b_2 = 2$. In this case, what is the total waiting time (over all the processes) ? (5%)

   PS: Waiting time of a process is the sum of the periods spent waiting in the ready queue.

   (b) Consider the case with n processes. For each process i, $a_i < a_{i+1} < a_i + b_i$ and $b_i < b_{i+1}$. Write the total waiting time (over all the processes) in terms of a_1, a_2, …, a_n and b_1, b_2, …, b_n. (5%)

8. (10%) A process is a program in execution. A process can be further divided into a ***text*** section, ***data*** section, ***heap***, and process ***stack***. Consider the below program written in C.

```
#include <stdio.h>
#include <stdlib.h>
float float_array[10];
static double double_array[10];
int main(void)
{
    int num;
    scanf("%d",&num);
    int *int_array = (int*) malloc(num * sizeof(int));
    // Some lines are removed for space reason.
    return 0;
}
void foo1(int n)
{
    n = (n > 0)? 1 : 0;
    printf("%d", n);
}
void foo2()
{
    static int count = 0;
    count++;
    printf("%d", count);
}
```

Please fill in the following blanks with ***text***, ***data***, ***heap***, or ***stack***.

(a) The variable array **"float_array"** is stored in the _____ section.

(b) The variable array **"double_array"** is stored in the _____ section.

(c) The variable array **"int_array"** is stored in the _____ section.

(d) The variable **"n"** is stored in the _____ section.

(e) The variable **"count"** is stored in the _____ section.

9. (6%) (a) What is *Inverted page table* ? Show its hardware and explain how it works to convert a logical address to a physical address. (3%)

(b) Show the *paging hardware with TLB* (translation look-aside buffer), and explain how it works to convert a logical address to a physical address. (3%)

10. (9%) (a) What is thrashing ? (3%)

(b) What is working set (WS) concept ? (3%)

(c) Explain how the WS concept could be used to solve the thrashing problem. (3%)

11. (6%) Consider the three disk allocation schemes below: "contiguous allocation", "linked allocation" and "indexed allocation". Which scheme will you choose if the file operations are mostly consisted of "append" and "truncate". Explain your answer.

12. (9%) Explain why disk has worse I/O performance for random access, and give two solutions that can improve the random access performance of a disk.