資訊工程研究所　碩士班考試　計算機結構

If some questions are unclear or not well defined to you, you can make your own assumptions and state them clearly in the answer sheet.

1. Short Questions: (24%)

Answer YES or NO with brief explanation to the following questions. Credit will be given only if explanation is provided.

(a) For a CPU with caches and TLB, there may require none or at most two memory references to access a data.

(b) Superscalar machines, unlike VLIW, have dynamic issue capability in hardware that can issue different number of instructions in a clock cycle. Therefore, compiler technology cannot improve the average ***IPC*** (instructions per clock cycle) of superscalar processors.

(c) Consider a 5-stage pipelined MIPS machine without forwarding mechanism, the execution of the following codes used to swap the values in two memory location will have no hazard.

```
lw    $t4, 0($t2)
lw    $t6, 4($t2)
nop
nop
sw    $t4, 4($2)
sw    $t6, 0($2)
```

(d) To get a speedup of 5 from 10 processors means that the percentage of the original program that was sequential would have to be 10% or less.

(e) Pipelining can reduce the overall execution time of program but cannot reduce the execution time of individual instructions.

(f) A multimedia program (such as MP3) can run faster on a CPU with MMX facility than the same machine code on a CPU without MMX support mainly because of faster clock rate.

(g) A ROM with the state register can be used to implement a finite-state machine to control the operation of CPU. Suppose these are possible 16 states in this finite-state machine with 12 inputs from instruction opcode, and 18 datapath control outputs, the size of ROM required will be $2^{(12+\log 16)} \times 18$ bits.

(h) Among three cache miss categories, compulsory misses (cold-start misses) can be reduced by increasing block size.

2. Computer Datapath (30%)

Fig. 1 represents the architecture of a simple version of pipelined MIPS machine.

(a) Discuss when the control hazard occurs, and evaluate the pipeline stall caused by this type of hazard. (6%)

(b) Brief discuss what dynamic branch prediction is. Can branch prediction help reducing the pipeline stall caused by branch instruction for the machine in Fig.1? (6%)

(c) Data forwarding can be used to reduce the pipeline stall caused by data hazard. Show the forwarding paths of the data such that the data hazard stall will be not larger than one cycle. (6%)

(d) Suppose we want to add a new instruction to the MIPS such that it can support the post-increment addressing mode for LOAD instruction. For example, the execution of the instruction **lw $t0, 4($t2++)** will load the memory data from ($t2+4) to register $t0, and it also increases the value of register $t2 by one. This function is helpful for the application that will read data at sequential addresses. Discuss the impact of this modification from the following aspects: CPI, MIPS, and application's execution time. Also discuss how to modify the datapath in order to support this new instruction. (12%)

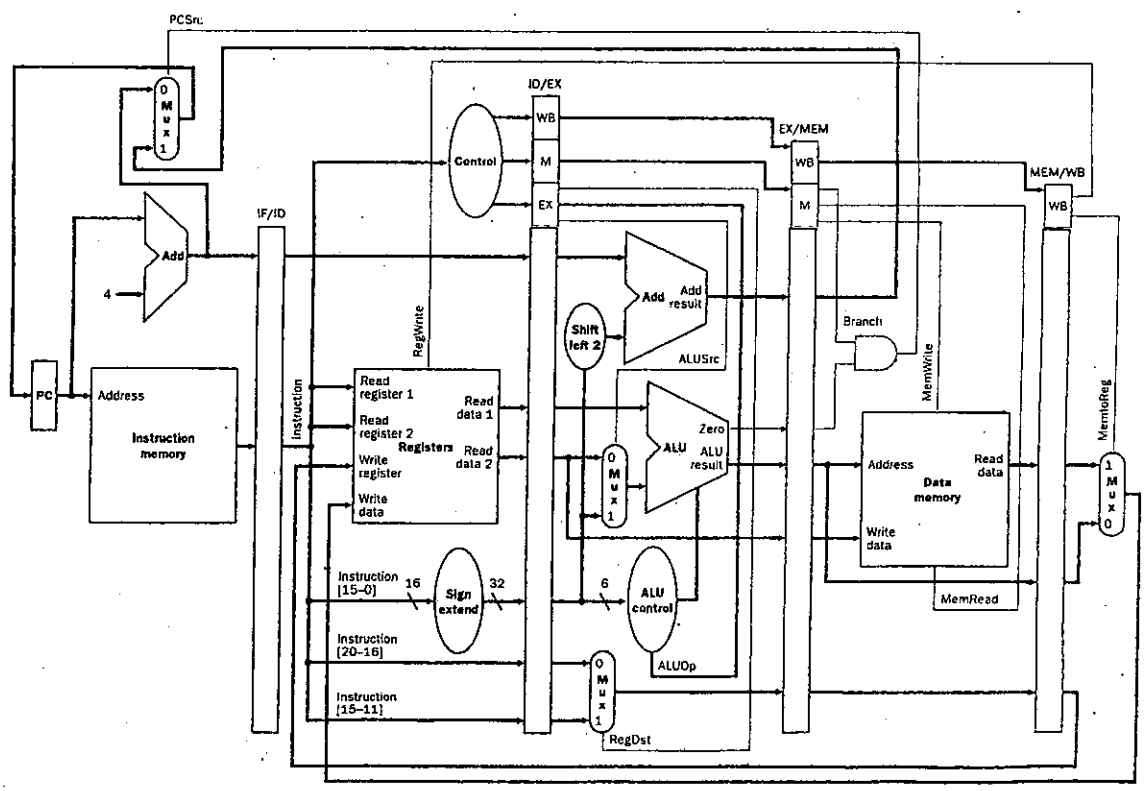


Fig. 1

3. Cache (26%)

Fig.2 illustrates an instruction cache architecture used for one commercial DSP Processor. The instruction for this DSP processor is 24-bit wide.

(a) Calculate the overall size of the cache. (6%)

(b) How do you characterize this cache organization in terms of associativity and block size? (6%)

(c) Suppose a series of address reference: 1, 4, 8, 17, 40, 7, 8, 33, 72, 40 is given to get access to this cache, label each reference in the list as s hit or a miss and show the final contents of the cache. (8%)

(d) The basic concept of cache takes advantage of temporal and spatial locality in the execution of application programs. Show a short segment of application code in C or Assembly that will have high temporal and spatial locality. (6%)

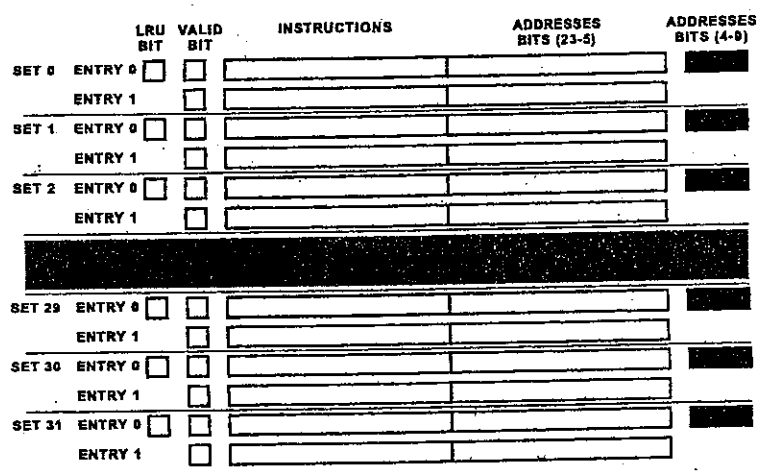


Fig. 2

4. Computer arithmetic: (20%)

(a) Derive an 8-bit two's complement representation for –13. (4%)

(b) Draw a four-bit ALU architecture consisting of fundamental logic gates, multiplexers and full-adders corresponding to the following table. In addition to the four-bit result output, this architecture should also generate two flag outputs: **Zero** (=1 if the result is 0) and **Overflow** (=1 if the two's complement addition result is overflowed). (8%)

ALU control lines (c0, c1)	Function
00	ADD
01	NAND
10	XOR
11	OR

(c) Carry-lookahead adder as shown in Fig.3 can be used for fast generation of addition. Find out the value of C1, C2, C3, C4, G0, P1, G2 and P3 when adding two numbers unsigned number 0111010110110100 and 1000101010111011 with **Carry_In** bit equal to 0. (Hint: The result can be obtained easily from the theory of carry-lookahead addition instead of deriving complex equations.) (8%)

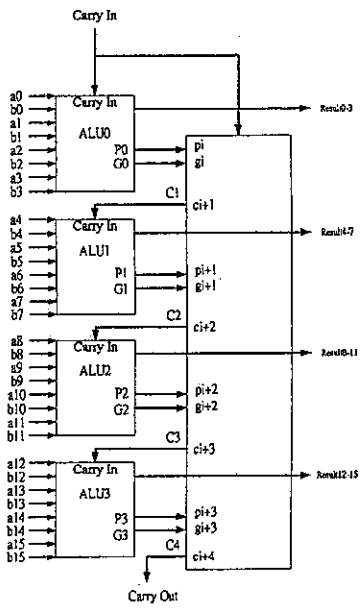


Fig.3

國立中山大學九十一學年度碩士班招生考試試題

科目： 作業系統與資料結構【資訊工程學系碩士班】

1. (10%) Suppose that a moving-head disk contains 100 tracks (numbered 0 through 99) and that the head is currently at track 0. We have the following requests in a queue:

10, 89, 73, 27, 75, 44, 77, 90, 9.

What is the total head movment needed to satisfy these requests for the following disk-scheduling algorithms ? (a) SSTF.(b) SCAN.(c) C-SCAN.(d) LOOK.(e) FCFS.

2. (10%) Assume we have a paged memory system with associative registers to hold the most active page table entries. If the page table is normally held in memory, and memory access time is 1 microsecond, what is the effective access time if 85% of all memory references find their entries in the associative registers ? How about 50% ?

3. (10%) Consider the following page reference string:

1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6.

How many page faults would occur for the following replacement algorithms, assuming 3,4,5,6 or 7 frames ? (a) LRU.(b) Optimal.

4. (12%) Suppose the following jobs arrive for processing at the times indicated. Each job will run the listed amount of time. What are the average turnaround time and the average waiting time for these jobs by using the following scheduling algorithms ? (a) Non-Preemptive Shortest-Job-First. (b) Preemptive Shortest-Job-First.

Job	Arrival-Time	Burst Time
1	0.0	8
2	0.5	5
3	1.0	2

5. (8%) (a) In the following two-process solution for mutual exclusion, what is wrong with it ?

```
repeat
  flag[i]:=true;   --(1)
  while flag[j] do no-op;  --(2)
  CS;
  flag[i] :=false;
until false;
```

(b) If we exchange the order of statements 1 and 2, will the algorithm be correct ?

6. (10%) Let $A = [a_1, a_2, \ldots, a_n]$ be an array of n distinct positive integers. A is called a *heap*, if

$$a_i \geq \max\{a_{2i}, a_{2i+1}\}, \text{ for } i = 1, 2, \ldots, \lfloor n/2 \rfloor.$$

Note that if n is even, we define $a_{n+1} = 0$ in the above definition.

(a) Show that a_1 is the maximum element in a heap A.

(b) Can you design a *linear time* algorithm to print out the elements of a heap A in increasing or decreasing order? Justify your answer.

7. (10%) Quick sort is a sorting algorithm for an array $A = [a_1, a_2, \ldots, a_n]$. It first swaps elements of the array and splits the array into two parts $A_l = [a'_1, a'_2, \ldots, a'_s]$ and $A_r = [a'_{s+1}, a'_{s+2}, \ldots, a'_n]$ so that $a_i \leq a_j$ for every $a_i \in A_l$ and every $a_j \in A_r$. It then recursively sorts A_l and A_r. Suppose that you are going to design a nonrecursive version of quick sort, and choose to use a stack to store the ranges of the array yet to be sorted. The new algorithm sorts the part of the array whose range is stored on the top of the stack until the stack is empty. Initially, the stack contains only one range, $[1..n]$, which means that the entire array is to be sorted. Show that if you store the smaller part of the array on the top of the stack after each split, then the size of the stack will be bounded by $O(\log n)$.

8. (10%) A graph $G = (V, E)$ consists of a set of vertices V and a set of edges E. Assume that the graph is simple, that is, each edge $e \in E$ connects two distinct vertices in V and no two edges connect the same pair of vertices. Also assume that the graph is connected, that is, there is a path between every pair of vertices. It is known that a simple and connected graph will have at least $|V| - 1$ edges and at most $\binom{|V|}{2}$ edges. A graph G is *dense* if the number of edges is closed to $\binom{|V|}{2}$. A graph G is *sparse* if the number of edges is closed to $|V| - 1$. Design a data structure for dense graphs and a data structure for sparse graphs so that the memory required will be as small as possible.

9. (20%) Let S be a set of n elements. The *union-find* problem on the set S consists of a sequence of `union(x,y)` and `find(z)` instructions. Initially, each element in S is a set by itself. The instruction `union(x,y)` makes the set containing x and the set containing y into one set. The `find(z)` reports the name of the set in which z belongs. The name of a set can be any elements in that set, but it must be consistent. That is, any element in the same set should get the same name, and the name cannot be changed, except it is unioned to another set. Design data structures and algorithms for the problem so that both instructions can be executed efficiently for large $|S|$. Analyze the time complexity of `union(x,y)` and `find(z)`.

國立中山大學九十一學年度碩士班招生考試試題

科目：離散數學（資訊工程學系）

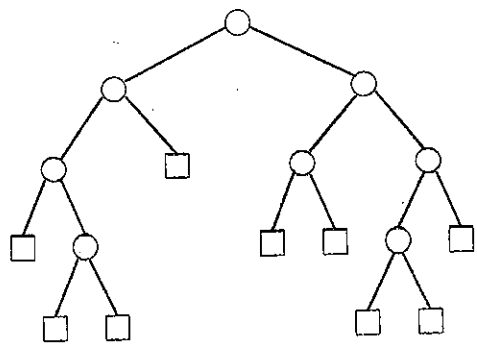
1. (a) How many different n node binary trees are there? (5%)
   (b) Proof your answer in (a). (15%)
2. Solve the following recurrence relation using characteristic root method. (15%)

$$\begin{cases} t_n = 7t_{n-1} - 12t_{n-2} + 2n + 3 \\ t_1 = 1 \\ t_0 = 0 \end{cases}$$

3. Solve the following recurrence relation using generating function method: (15%)

$$\begin{cases} f_n = f_{n-1} + f_{n-2} \quad n \geq 2 \\ f_0 = 0, f_1 = 1 \end{cases}$$

4. Show that $n! = O(n^n)$. (10%)
5. Given the extended binary tree shown below (these square nodes are called external nodes, and the others are called internal nodes),
   (a) Determine the internal path length I (i.e., the sum of the path lengths of all internal nodes) and external path length E. (8%)
   (b) Is there any association between I and E? Proof it. (12%)



6. Do any two spanning trees for a connected graph always have an edge in common? If so, give a proof. If not, give a counterexample. (10%)
7. Let x_i be an integer and $x_i \geq 0$. How many solutions are for $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 15$? (10%)