國立中山大學101學年度碩士暨碩士專班招生考試試題

科目：計算機結構【資工系碩士班甲組、乙組】

題號：4078

NOTE: Please simply your computation into decimal numbers whenever possible.

1 (15%) You are going to enhance a machine, and there are two possible improvements: either make multiply instructions run four times faster than before, or make memory access instructions run two times faster than before. You repeatedly run a program that takes 200 seconds to execute. Of this time, 50% is used for multiplication, 20% for memory access instructions, and 30% for other tasks. Determine the speedup for the corresponding cases. Copy Table 1 to your answer sheet and fill your answers.

Enhancement cases:

C1: Improve only multiplication

C2: Improve only memory access

C3: Improve both multiplication and memory access

Table 1.

Enhancement case	Speedup
C1	
C2	
C3	

2 (5%) The dynamic power of a processor is related to the capacitive load (C), operation frequency (F) and voltage (V) with this equation $Power=CFV^2$. Suppose we developed a new, simpler processor that has 80% of the capacitive load of the more complex older processor. Further, we can reduce the voltage of the new processor by 20% and let the new processor run at 20% faster frequency. What's the dynamic power of the new processor (P_{new}) with respect to the dynamic power of the old processor (P_{old})?

3 (10%) The following C procedure `swap` is compiled into the corresponding MIPS assembly code. Assume that the parameters `v` and `k`, and variable `temp` are in registers `$a0`, `$a1` and `$t0`, respectively. Other registers that can be used are `$t1` and `$t2`. C procedure:

```
void swap(int v[], int k)
{
   int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
 }
```

MIPS assembly code:

```
Swap:  sll   $t1, $a1,  2
       add   $t1, $a0, OPA
       lw    $t0, 0($t1)
       lw    $t2, 4(OPB)
       sw    OPC, 0($t1)
       sw    OPD, 4($t1)
       jr    $ra
```

Please determine proper values for the operands (OPA, OPB, OPC, OPD). Copy the following table (Table 3) to your answer sheet and fill in the operand values.

Table 3.

Operand	Value
OPA	
OPB	
OPC	
OPD	

4 (10%) Figure 4.1 is a basic multiplication hardware and Figure 4.2 is an improved version of the basic one.

Figure 4.1: Basic multiplication hardware

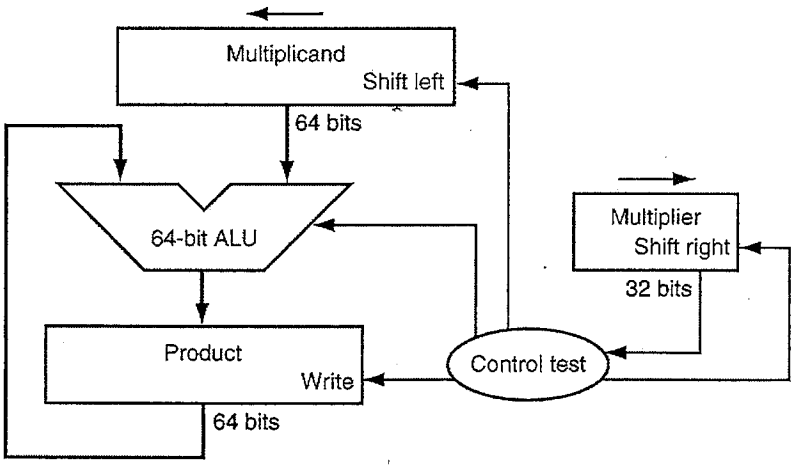
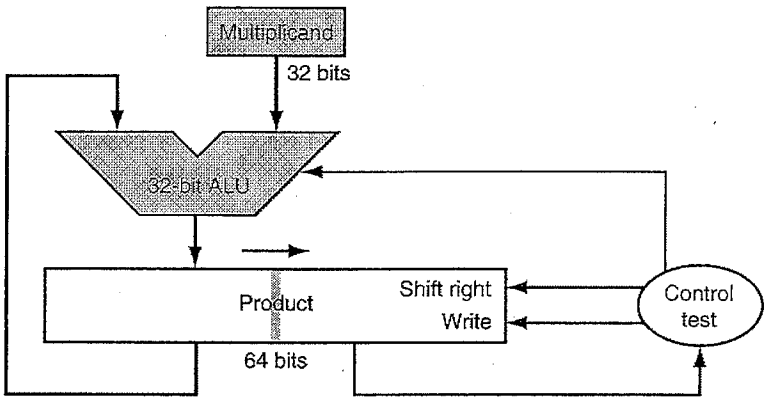


Figure 4.2: Improved multiplication hardware



The following statements are possible observations for the improvement:

S1: The multiplicand register, ALU, Multiplier register and the Product register are all 32 bits wide;

S2: The Product register always shifts right two bits in each iteration;

S3: The original Multiplier register disappears and is now placed in the right half of the Product register initially;

S4: The left side of the Product register is initially set to zero.

S5: The Multiplicand register is initially set to zero.

Please copy the following table (Table 4) to your answer sheet and determine if these statements are true or false.

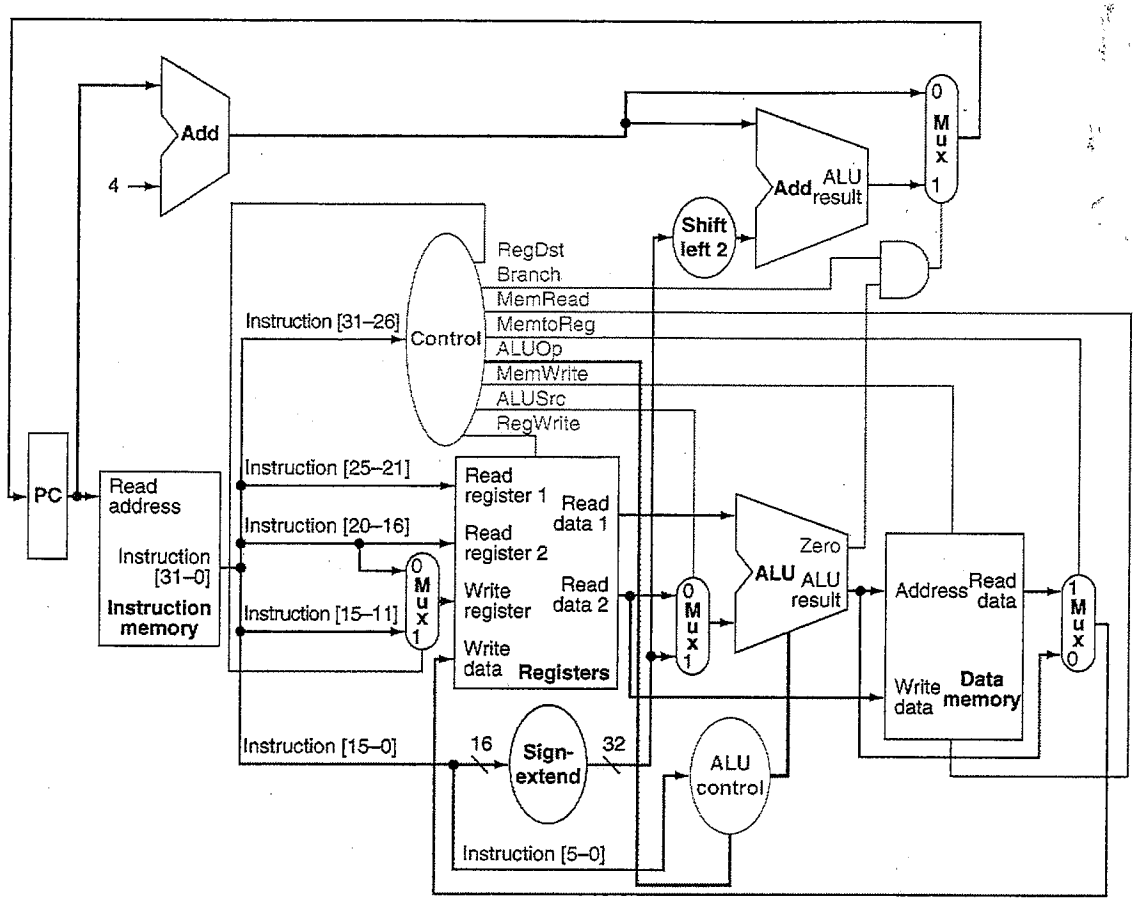
Table 4:

Statement	True (T) or False (F)
S1	
S2	
S3	
S4	
S5	

5 (20%) The following figure is a simple implementation of a MIPS subset (`R-type, lw, sw, beq`). Copy Table 5 to your answer sheet and fill in the corresponding control signal values (0, 1 or X) for instructions `lw` (load word) and `beq` (branch on equal).

Table 5:

Instruction	ALUSrc	MemtoReg	RegWrite	MemWrite	RegDst
`lw`					
`beq`					



6 (10%) There are three small caches, each consisting of four one-word blocks. The first cache is direct mapped, the second cache is two-way set associative, and the third cache is fully associative. Find the numbers of misses for each organization given the following sequence of block addresses: 8, 0, 8, 6, 0. Copy Table 6 to your answer sheet and fill in the answers. Assume that the least recently used replacement policy is used for the fully associative cache and the set-associative cache.

Table 6:

Cache organization	Number of misses
Direct mapped	
Two-way set associative	
Fully associative	

7 (Total 15%) Suppose we want to sum 100,000 numbers on a single-bus multiprocessor computer. Let's assume we have 10 processors. The first step would be to split the set of numbers into subsets of the same size. All processors start the program by running the following loop that sums their subset of numbers, where Pn is its processor index (0~9):

sum[Pn] = 0;

```
for (I = 10000 * Pn; I < 10000 * (Pn+1); I = I + 1)
    sum[Pn] = sum[Pn] + A[I]; // sum the assigned areas
```

The next step is to add these many partial sums, so we divide to conquer. Half of processors add pairs of partial sums, then a quarter add pairs of the new partial sums, and so on until we have the single, final sum. We want each processor to have its own version of loop counter variable *I*, so we must indicate that it is a "private" variable. In this problem, the two processors must synchronize before the "consumer" processor tries to read the result from the memory location written by the "producer" processor; otherwise, the consumer may read the old value of the data. Here is the code (*half* is also a private variable):

```
half = 10; //10 processors in 1-bus multiprocessor.
repeat
        synch(); // wait for partial sum completion.
        if (half%2 != 0 && Pn == 0)
            sum[0] = sum[0] + sum[half - 1]; //the case for odd number of
                                             //summing processor
        half = half / 2; // dividing line on who sums.
        if (Pn < half) sum[Pn] = sum[Pn] + sum[Pn+half];
until (half == 1); // exit with final sum in sum[0].
```

Question: according to the algorithm, find out what operations are executed by the designated processor during the designated repeat-loop iteration. (Ex: NOP or sum[0] = sum[0] + sum [4]). Copy Table 7 to your answer sheet and fill in the answers.

Table 7:

Processor index	Repeat-loop iteration	Operation
Pn=2	first	
Pn=2	second	
Pn=0	third	

8 (15% total) Consider a two-bit adder with its two operands A_i and B_i where i=0, 1 (0 is the least significant bit) and the carry-in bit C_0 and carry-out bit C_2.

8.1 (5%) Derive the Boolean equation for the carry-out bit C_2 using only A_i, B_i (where i=0, 1) and C_0.

8.2 In a carry-lookahead adder, there are two additional signals defined, $G_i=A_i \cdot B_i$ (generate), $P_i=A_i+B_i$ (propagate). Derive the Boolean equation for the carry-out bit C_2 using only G_i, P_i (where i=0, 1) and C_0.

8.3 (5%) What are the advantages of the Boolean equation in Problem 8.1 over the Boolean equation in Problem 8.2?