1. Given the 4-bit restoring division block diagram below. Please divide 7_{10} by 2_{10} and show the value of each register for each of the steps. (15%)
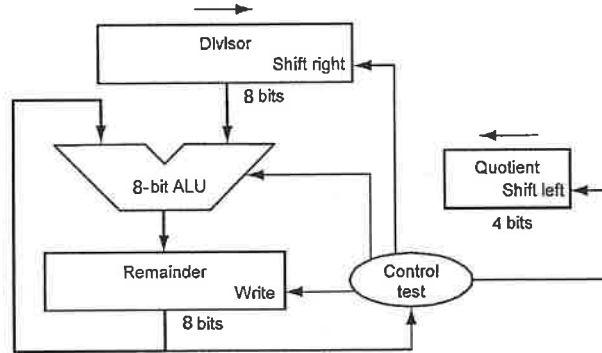
Iteration	Quotient	Divisor	Remainder
0 (initial values)	0000	0010 0000	0000 0111
1			
2			
3			
4			
5			

2. Given the code sequence below:

```
lw   $t1, 8($t7)        //assume mem[$t7+8] contains (+72)10
addi $t2, $zero, #10
nor  $t3, $t1, $t2
beq  $t1, $t2, Label
add  $t4, $t2, $t3
sw   $t4, 108($t7)
Label:  ...
```

Step Name	Action for R-type Instructions	Action for Memory-Reference Instructions	Action for branches	Action for jumps
Instruction fetch		IR ← Memory[PC] PC ← PC+4		
Instruction decode/register fetch		A ← Reg[IR[25-21]] B ← Reg[IR[20-16]] ALUOut ← PC + sign-extend(IR[15-0]) << 2		
Execution, address computation, branch/jump completion	ALUOut ← A op B	ALUOut ← A + sign-extend (IR[15-0])	If (A==B) then PC ← ALUOut;	PC ← {PC[31-28], (IR[25-0]<<2)}
Memory Access or R-type completion	Reg[IR[15-11]] ← ALUOut	Load: MDR ← Memory[ALUOut] or Store: Memory[ALUOut] ← B		
Memory read completion		Load: Reg[IR[20-16]] ← MDR		

According to the multi-cycle implementation scheme (see the above table),
(A) How many cycles will it take to execute this code? (3%)
(B) What is going on during the 19th cycle of execution? (3%)
(C) In which cycle does the actual addition of 108 and $t7 take place? (4%)

3. In the pipelined processor datapath design below, there is a serious problem. Please correct this problem. (5%)
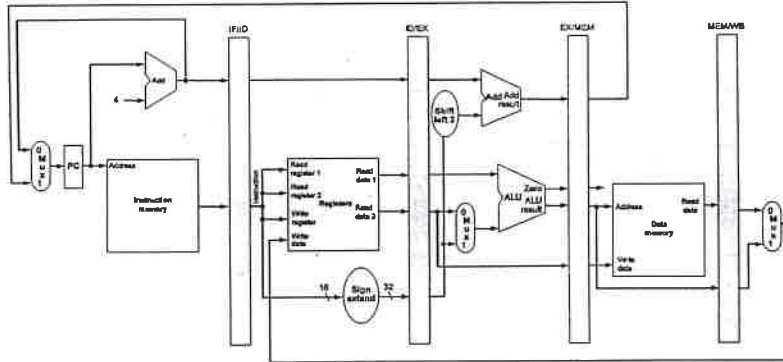
4. Given a 2^S-byte, 2^L-byte-per-line cache in an M-bit, byte-addressable memory system (i.e., total 2^M bytes in the memory system),
(A) What is the range of the index field size (number of bits), in a memory address while accessing the cache? (5%)
(B) What is the range of the tag field size (number of bits) in a memory address while accessing the cache? (5%)

5. Given the single-cycle processor datapath design and the definition and formats of its instructions: (10%)
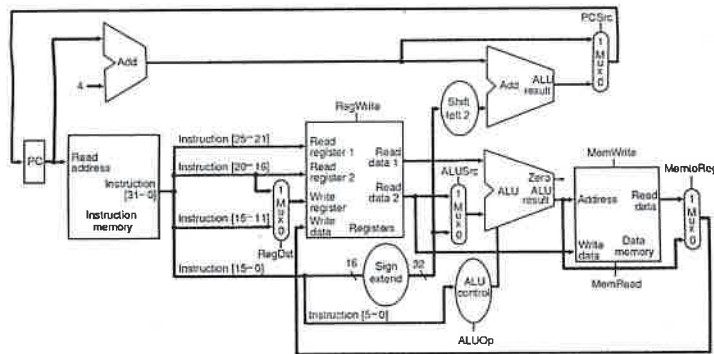

```
R-format: add $rd, $rs, $rt      //rd = $rs + $rt
I-format: lw $rt, addr($rs)      //$rt = Memory[$rs + sign-extended addr]
I-format: beq $rs, $rt, addr     //if ($rs = $rt) go to PC + 4 + 4 × addr
```

Field size	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
R-format	op	rs	rt	rd	shamt	funct
I-format	op	rs	rt	address/immediate		

Complete the following table for the setting of the control lines of each instruction. Assume that the control signal Branch and the Zero output of the ALU are ANDed together to become the control signal PCSrc.

Instruction	RegDst	ALUSrc	MemtoReg	RewWrite	MemRead	MemWrite	Branch
add							
lw							
beq							

(1: asserted of control line, 0: de-asserted of control line, x: don't care)

6. Suppose you have a machine which executes a program consisting of 50% floating point multiply, 20% floating point divide, and the remaining 30% are from other instructions.

(A) We want the machine to run 2 times faster. You can make the divide run at most 3 times faster and the multiply run at most 8 times faster. Can you meet the goal by making only one improvement? If yes, which one? If no, please explain your reason. (5%)

(B) If you make both the multiply and divide improvements, what is the speedup of the improved machine relative to the original machine? (5%)

7. This question examines the accuracy of various branch predictors for the following repeating pattern (e.g. in a loop) of branch outcomes:

T, T, NT, T
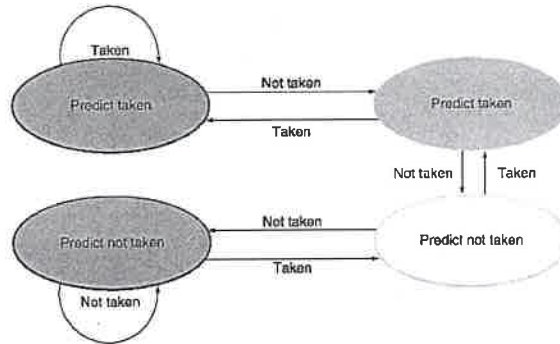
(A) What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes? (6%)

(B) What is the accuracy of the two-bit predictor for the first four branches in this pattern, assuming that the predictor starts off in the bottom left state from the above figure (predict not taken). (4%)

(C) What is the accuracy of the two-bit predictor if this pattern is repeated forever? (10%)

8. Suppose we have a processor with a base CPI = 1, assuming all references hit in primary cache, and a clock rate of 4GHz. Assume a main memory access time = 250 ns including all the miss handling.

(A) Suppose the miss rate per instruction at the primary cache = 10%. Please calculate the overall CPI. (10%)

(B) How much faster will the processor be if we add a secondary cache that has 10 ns access time for either a hit or a miss and is large enough to reduce the miss rate to main memory to 1%? (10%)

國立臺灣科技大學　　　/08學年度第 / 學期 博七412考試命題用紙　　第 3 頁共 3 頁

考試科目：Computer Architecture
☐ 研究所
☐ 大學部
☐ 工程在職進修

命題請勿超出線外