Computer Organization Exam

Chapter 3

1. (10%) Convert **-7052$_{ten}$** into a 32-bit two's complement binary number.
   Ans: 1111 1111 1111 1111 1110 0100 0111 0100
2. (10%) What decimal number does this two's complement binary number represent: **1111 1111 1111 1111 1110 0110 0100 1100$_{two}$**
   Ans: -6580
3. (20%) The Big Picture mentions that bits have no inherent meaning. Given the bit pattern:

   1000 1101 1000 1011 0000 0000 0000 0000 $_{two}$
   0111 0010 0111 0101 0000 0000 0000 0000

   What does it represent, assuming that it is
   (a) a two's complement integer?
   (b) an unsigned integer?
   (c) a single precision floating-point number?
   (d) a MIPS instruction?
   You may find the following tables useful.
   Ans:
   a. -1920270336
   b. 2374696960
   c. $-1.0859375*2^{-100}$
   d. lw $t3, 0($t4)

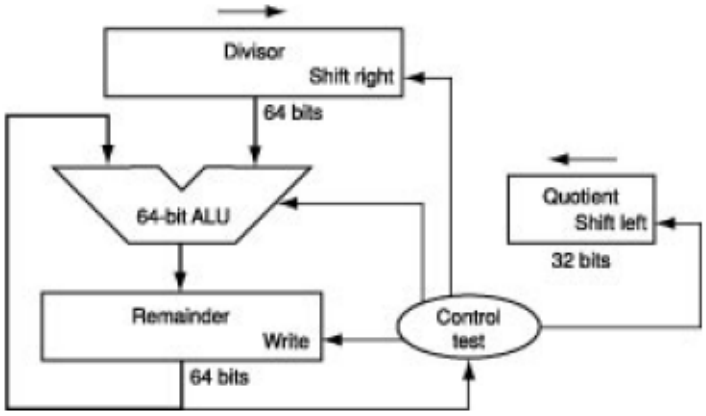
4. (20%) Show the IEEE 754 binary representation for the floating-point number **-11.5$_{ten}$** in single and double precision.
   Ans: 1100 0001 0011 1000 0000 0000 0000 0000

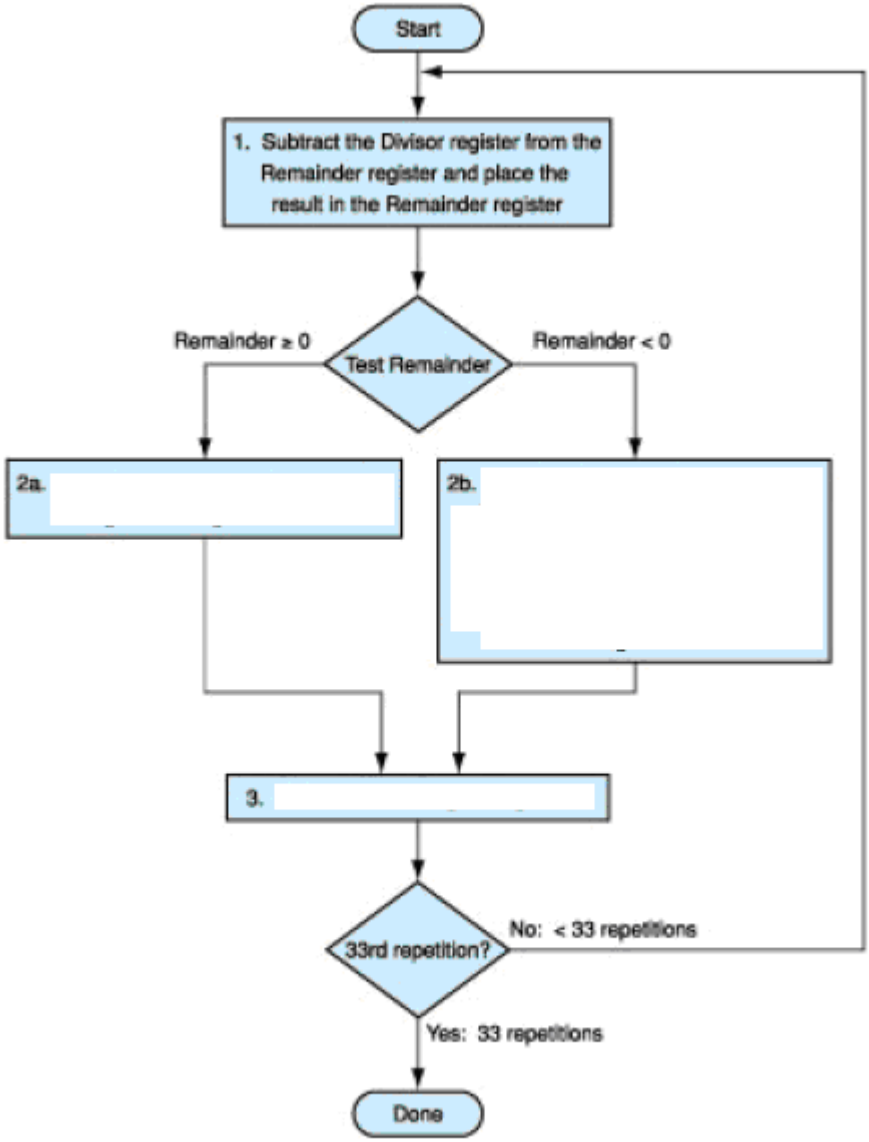
   11000000001001110000000000000000000000000000000000000000000000000

5. (20%) With **x = 0100 0000 1111 0000 0000 0000 0000 0000$_{two}$** and **y = 1011 1111 1011 0000 0000 0000 0000 0000$_{two}$** representing single precision IEEE 754 floating-point numbers, perform, showing all work:
   (a) **x + y** (b) **x * y**
   Ans:
   a. 0100 0000 1100 0100 0000 0000 0000 0000

b. 1100 0001 0010 0101 0000 0000 0000 0000

6. (10%) According to the division hardware provided below for positive integers, please fill in the blanks in the following flowchart.





Computer Organization - Chapter 3

Ans: 2a: Shift the Quotient register to the left, setting the new rightmost bit to 1

2b: Restore the original value by adding the Divisor to the Remainder register and place the sum in the Remainder register. Also shift the Quotient register to the left, setting the new least significant bit to 0.

3:Shift the Divisor register right 1 bit.

Operation	Operand A	Operand B	Result indicating overflow
A+B	1.	2.	3.
A+B	4.	5.	6.

A-B	7.	8.	9.
A-B	10.	11.	12.

7. (10%)Please fill in these blanks by “≧0” or “<0”.

Ans:

≧0

1. ≧0
2. ≧0
3. <0
4. <0
5. <0
6. ≧0
7. ≧0
8. <0
9. <0
10. <0
11. ≧0
12. ≧0

MIPS register conventions

Name	Register number	Usage	Preserved on call?
$zero	0	The constant value 0	n.a.
$v0-$v1	2-3	Values for results and expression evaluation	No
$a0-$a3	4-7	Arguments	No
$t0-$t7	8-15	Temporaries	No
$s0-$s7	16-23	Saved	Yes
$t8-$t9	24-25	More temporaries	No
$gp	28	Global pointer	Yes
$sp	29	Stack pointer	Yes
$fp	30	Frame pointer	Yes
$ra	31	Return address	Yes

MIPS instruction encoding

op(31:26)								
28-26 31-29	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(000)	R-format	Bltz/gez	jump	Jump&link	Branch eq	Branch ne	blez	bgtz
1(001)	Add immediate	addiu	Set less than imm.	sltiu	andi	ori	xori	Load upper imm
2(010)	TLB	FLPt						
3(011)								
4(100)	load byte	Load half	lwl	load word	lbu	lhu	lwr	
5(101)	store byte	Store half	swl	store word			swr	
6(110)	Lwc0	Lwc0						

7(111)	Swc0	Swc1						

op(31:26) = 010000(TLB),rs(25:21)								
23-21 25-24	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(00)	mfc0		cfc0		mtc0		ctc0	
1(01)								
2(10)								
3(11)								
op(31:26) =000000 (R-format),func(5:0)								
28-26 31-29	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(000)	Shift left logical		Shift right logical	sra	sllv		srlv	srav
1(001)	Jump reg.	jalr			syscall	Break		
2(010)	mfhi	mthi	mflo	Mtlo				
3(011)	Mult	multu	div	divu				
4(100)	add	addu	subtract	subu	and	or	xor	Not
5(101)			Set l. t.	sltu				
6(110)								
7(111)								