King Fahd University of Petroleum and Minerals
College of Computer Sciences and Engineering
Department of Computer Engineering

COE 308 – Computer Architecture (T041)

Homework # 04 (*due date: Sunday 26/12/2004*)

***** Show all your work. No credit will be given if work is not shown! *****

Problem # 1 (10 points): Using the "restoring" division algorithm, find the result of dividing the unsigned number 27 by the unsigned number 5.

Solution:
A = $(27)_{10} = 11011_2$, B = $(5)_{10} = 00101_2$

Iteration	Step		P	A	Comment
Initialization		0	00000	11011	
1	1	0	00001	1011_	Shift left P & A one bit position
	2	- 0	00101		Subtract B from P (i.e. P = P – B)
	3	- 0	00100	10110	Since result of step 2 < 0, then set LSB of A to 0
	4	0	00001	10110	Since result of step 2 < 0, then add B to P (i.e. restore P)
2	1	0	00011	0110_	Shift left P & A one bit position
	2	- 0	00101		Subtract B from P (i.e. P = P – B)
	3	- 0	00010	01100	Since result of step 2 < 0, then set LSB of A to 0
	4	0	00011	01100	Since result of step 2 < 0, then add B to P (i.e. restore P)
3	1	0	00110	1100_	Shift left P & A one bit position
	2	- 0	00101		Subtract B from P (i.e. P = P – B)
	3	0	00001	11001	Since result of step 2 > 0, then set LSB of A to 1
	4	0	00001	11001	Since result of step 2 > 0, then do nothing to P (i.e. keep current value of P)
4	1	0	00011	1001_	Shift left P & A one bit position
	2	- 0	00101		Subtract B from P (i.e. P = P – B)
	3	- 0	00010	10010	Since result of step 2 < 0, then set LSB of A to 0
	4	0	00011	10010	Since result of step 2 < 0, then add B to P (i.e. restore P)
5	1	0	00111	0010_	Shift left P & A one bit position
	2	- 0	00101		Subtract B from P (i.e. P = P – B)
	3	0	00010	00101	Since result of step 2 > 0, then set LSB of A to 1
	4	0	00010	00101	Since result of step 2 > 0, then do nothing to P (i.e. keep current value of P)

Note that the result of the division has the quotient in A and the remainder in P
(i.e. A = quotient = $00101_2 = (5)_{10}$, and P = remainder = $00010_2 = (2)_{10}$).

Problem # 2 (10 points): Representing numbers in 2's complement, find the result of multiplying +17 by +6 using the algorithm presented in class.

Solution:
A = $(+17)_{10}$ = 010001, B = $(+6)_{10}$ = 000110

Iteration	Step	P	A	a_{i-1}	Comment
Initialization		000000	010001	0	
1	1	+ 111010			Since $a_i = 1$ & $a_{i-1} = 0$, then P = P – B (note that – B = 111010)
		111010	010001	0	
	2	111101	001000	1	Shift right P & A arithmetically one bit position
2	1	+ 000110			Since $a_i = 0$ & $a_{i-1} = 1$, then P = P + B
		000011	001000	0	
	2	000001	100100	0	Shift right P & A arithmetically one bit position
3	1	+ 000000			Since $a_i = 0$ & $a_{i-1} = 0$, then P = P + 0
		000001	100100	0	
	2	000000	110010	0	Shift right P & A arithmetically one bit position
4	1	+ 000000			Since $a_i = 0$ & $a_{i-1} = 0$, then P = P + 0
		000000	110010	0	
	2	000000	011001	0	Shift right P & A arithmetically one bit position
5	1	+ 111010			Since $a_i = 1$ & $a_{i-1} = 0$, then P = P – B
		111010	011001	0	
	2	111101	001100	1	Shift right P & A arithmetically one bit position
6	1	+ 000110			Since $a_i = 0$ & $a_{i-1} = 1$, then P = P + B
		000011	001100	1	
	2	000001	100110	0	Shift right P & A arithmetically one bit position

Note that the result of the multiplication is held in the register pair P & A
(i.e. P & A = 000001100110 = $(102)_{10}$).

Problem # 3 (10 points): Representing numbers in 2's complement, find the result of multiplying +17 by – 6 using the algorithm presented in class.

Solution:
A = $(+17)_{10}$ = 010001, B = $(-6)_{10}$ = 111010

Iteration	Step	P	A	a_{i-1}	Comment
Initialization		000000	010001	0	
1	1	+ 000110			Since $a_i = 1$ & $a_{i-1} = 0$, then P = P – B (note that – B = 000110)
		000110	010001	0	
	2	000011	001000	1	Shift right P & A arithmetically one bit position
2	1	+ 111010			Since $a_i = 0$ & $a_{i-1} = 1$, then P = P + B
		111101	001000	0	
	2	111110	100100	0	Shift right P & A arithmetically one bit position
3	1	+ 000000			Since $a_i = 0$ & $a_{i-1} = 0$, then P = P + 0
		111110	100100	0	
	2	111111	010010	0	Shift right P & A arithmetically one bit position
4	1	+ 000000			Since $a_i = 0$ & $a_{i-1} = 0$, then P = P + 0
		111111	010010	0	
	2	111111	101001	0	Shift right P & A arithmetically one bit position
5	1	+ 000110			Since $a_i = 1$ & $a_{i-1} = 0$, then P = P – B
		000101	101001	0	
	2	000010	110100	1	Shift right P & A arithmetically one bit position
6	1	+ 111010			Since $a_i = 0$ & $a_{i-1} = 1$, then P = P + B
		111100	110100	1	
	2	111110	011010	0	Shift right P & A arithmetically one bit position

Note that the result of the multiplication is held in the register pair P & A
(i.e. P & A = 111110011010 = $(-102)_{10}$).

Problem # 4 (10 points): What single-precision floating-point number does the following 32-bit word represent? **11001000101001000000000000000000**.

Solution:

Sign = 1_2 = (–)ve number

Exponent field = e = 10010001_2 = 145 ⇒ Exponent = 145 – 127 = 18

Fraction field = f = $.01001000000000000000000_2$ = 0.28125

⇒ FP # represented = $-1.f \times 2^{e-127} = -1.28125 \times 2^{18} = -335{,}872$

Problem # 5 (10 points): Show the single-precision floating-point representation of **–** 5.375.

Solution:

$-5.375 = -1.34375 \times 4 = -1.34375 \times 2^2 = -1.f \times 2^{e-127}$

(–)ve number ⇒ Sign = 1_2

e – 127 = 2 ⇒ e = 129 = 10000001_2

f = 0.34375 ⇒ f = $.01011000000000000000000_2$

(Note:

0.34375 × 2	= 0.6875	⇒ .0
0.6875 × 2	= 1.375	⇒ .01
0.375 × 2	= 0.75	⇒ .010
0.75 × 2	= 1.5	⇒ .0101
0.5 × 2	= 1.0	⇒ .01011

)

⇒ FP # representation = **11000000101011000000000000000000**