Solutions to Problems Marked with a * in
Logic and Computer Design Fundamentals, 4th Edition

Chapter 10

10-2.*

a)
```
LD    R1, E
LD    R2, F
MUL   R1, R1, R2
LD    R2, D
SUB   R1, R2, R1
LD    R2, C
DIV   R1, R2, R1
LD    R2, A
LD    R3, B
ADD   R2, R2, R3
MUL   R1, R1, R2
ST    Y, R1
```

b)
```
MOV   T1, A
ADD   T1, B
MUL   T1, C
MOV   T2, E
MUL   T2, F
MOV   T3, D
SUB   T3, T2
DIV   T1, T3
MOV   Y, T1
```

c)
```
LD    E
MUL   F
ST    T1
LD    D
SUB   T1
ST    T1
LD    A
ADD   B
MUL   C
DIV   T1
ST    Y
```

10-3.*

a) (A - B) x (A + C) x (B - D) = A B - A C +x B D -x

b, c)

PUSH A	PUSH B	SUB	PUSH A	PUSH C	ADD
A	B	A-B	A	C	A+C
	A		A-B	A	A-B
				A-B	

MUL	PUSH B	PUSH D	SUB	MUL	POP X
(A-B)x(A+C)	B	D	B-D	(A-B)x(A+C)x(B-D)	
	(A-B)x(A+C)	B	(A-B)x(A+C)		
		(A-B)x(A+C)			

10-6.*

a) X = 195 – 208 – 1 = –14 b) X = 1111 1111 1111 0010

The number is negative because the branch is backwards. The – 1 assumes that the PC has been incremented to point to the address after that of the address word of the instruction.

10-10.*

a) 3 Register Fields x 4 bits/Field = 12 bits. 32 bits - 12 bits = 20 bits. 2^{20} = 1048576

b) 64 < 100 < 128 => 7 bits. 2 Register Fields x 4 bits/Field => 8 bits. 32 bits - 7 bits - 8 bits => 17 Address Bits

10-14.*

a)
```
ADD   R0, R4
ADC   R1, R5
ADC   R2, R6
ADC   R3, R7
```

b) $R0 \leftarrow 7B + 4B$, R0 = C6, C = 0

$R1 \leftarrow 24 + ED + 0$, R1 = 11, C = 1

$R2 \leftarrow C6 + 57 + 1$, R2 = 1E, C = 1

$R3 \leftarrow 1F + 00 + 1$, R3 = 20, C = 0

10-17.*

OP	Result Register	C
	0110 1001	1
SHR	0011 0100	1
SHL	0110 1000	0
SHRA	0011 0100	0
SHLA	0110 1000	0
ROR	0011 0100	0
ROL	0110 1000	0
RORC	0011 0100	0
ROLC	01101000	0

10-19.*

$$\text{Smallest Number} = 0.5 \times 2^{-255}$$

$$\text{Largest Number} = (1 - 2^{-26}) \times 2^{+255}$$

10-20.*

E	e	$(e)_2$
+8	15	1111
+7	14	1110
+6	13	1101
+5	12	1100
+4	11	1011
+3	10	1010
+2	9	1001
+1	8	1000
0	7	0111
–1	6	0110
–2	5	0101
–3	4	0100
–4	3	0011
–5	2	0010
–6	1	0001
–7	0	0000

10-23.*

```
TEST  (0001)16, R     (AND Immediate 1 with Register R)
BNZ   ADRS            (Branch to ADRS if Z = 0)
```

10-25.*

a) A = 0101 1101 93
B = 0101 1100 - 92
A – B = 0000 0001 1

b) C (borrow) = 0, Z = 0

c) BA, BAE, BNE

10-27.*

	PC	SP	TOS
a) Initially	2000	4000	5000
b) After Call	0502	4001	2002
c) After Return	2002	4000	5000

10-30.*

External Interrupts:
1) Hard Drive
2) Mouse
3) Keyboard
4) Modem
5) Printer

Internal Interrupts:
1) Overflow
2) Divide by zero
3) Invalid opcode
4) Memory stack overflow
5) Protection violation

A software interrupt provides a way to call the interrupt routines normally associated with external or internal interrupts by inserting an instruction into the code. Privileged system calls for example must be executed through interrupts in order to switch from user to system mode. Procedure calls do not allow this change.