Dr. Cherif Salama
Faculty of Media Engineering and Technology

CSEN 402 Computer Organization and System Programming
Practice Assignment 11
Spring 2014

NOT to be submitted

Exercise 1:

Specify the control word that must be applied to the processor of Fig.8-2 to implement the following micro-operations.

a. $R_1 \leftarrow R_2 + R_3$
b. $R_4 \leftarrow R_4'$
c. $R_5 \leftarrow R_5 - 1$
d. $R_6 \leftarrow$ shl R_1
e. $R_7 \leftarrow$ input

Solution:

	SELA	SELB	SELD	OPR	Control Word
a.	R_2	R_3	R_1	ADD	010 011 001 00010
b.	R_4		R_4	COMA	100 xxx 100 01110
c.	R_5		R_5	DECA	101 xxx 101 00110
d.	R_1		R_6	SHLA	001 xxx 110 11000
e.	Input		R_7	TSFA	000 xxx 111 00000

Exercise 2:

Determine the micro-operations that will be executed in the processor of Fig. 8-2 when the following 14-bit control words are applied.

a. 00101001100101
b. 00000000000000
c. 01001001001100
d. 00000100000010
e. 11110001110000

Solution:

	SELA	SELB	SELD	OPR	Micro-operation
a.	R_1	R_2	R_3	SUB	$R_3 \leftarrow R_1 - R_2$
b.	Input		None	TSFA	Output$\leftarrow$Input
c.	R_2	R_2	R_2	XOR	$R_2 \leftarrow R_2 \oplus R_2$

d.	Input	R_1	None	ADD	Output←Input + R_1
e.	R_7	R_4	R_3	SHRA	R_3←Shr R_7

Exercise 3:

A stack is organized such that SP always points at the next empty location on the stack. This means that SP can be initialized to 0x4000 in Fig. 8-4 and the first item in the stack is stored in location 0x4000. List the micro-operations for the "PUSH" and "POP" operations.

Solution:

/Empty Descending Stack
PUSH:
M[SP] ← DR
SP ← SP – 1
POP:
SP ← SP + 1
DR ← M[SP]

Exercise 4:

A bus-organized CPU similar to Fig.8-2 has 16 registers with 32 bits in each, an ALU, and a destination decoder.

a. How many multiplexers are there in the A bus, and what is the size of each multiplexer?
b. How many selection inputs are needed for MUX A and MUX B?
c. How many inputs and outputs are there in the decoder?
d. How many inputs and outputs are there in the ALU for data, including input and output carries?
e. Formulate a control word for the system assuming that the ALU has 35 operations.

Solution:

a. 32 multiplexers, each of size 16 × 1.
b. 4 input each, to select one of 16 registers.
c. 4-to-16 – line decoder
d. 32 + 32 + 1 = 65 data input lines
   32 + 1 = 33 data output lines.
e.

4	4	4	6	=18 bits
SELA	SELB	SELD	OPR	

Exercise 5:

A first-in, first-out (FIFO) memory organization stores information in such a manner that the item that stored first is the first item that is retrieved. Show how a FIFO memory operates with three counters. A write counter WC holds the address for writing into memory. A read counter RC holds the address for reading from the memory. An available storage counter ASC indicates the number of words stored in FIFO. ASC is incremented for every word stored and decremented for every item is retrieved.

Solution:

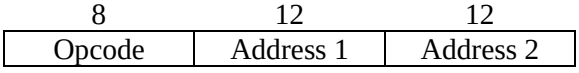
Write (if not full) :
M [WC] ← DR
WC ← WC + 1
ASC ← ASC + 1

Read : (if not empty)
DR ← M [RC]
RC ← RC + 1
ASC ← ASC –1

Exercise 6:

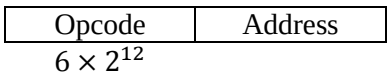
A computer has a 32-bit instructions and 12-bit addresses. If there are 250 two-address instructions, how many one-address instructions can be formulated?

Solution:



= 32 bit
Two address instructions

2^8 = 256 combinations.
256 – 250 = 6 combinations can be used for one address
For one-address instructions, one of the address fields can be used as an extension to the Opcode.



One address instructions

Maximum number of one address instruction = 6 x 2^{12} = 24,576

Exercise 7:

Write a program to evaluate the arithmetic statement:

$$\mathbf{x} = \frac{\mathbf{A} - \mathbf{B} + \mathbf{C} * (\mathbf{D} * \mathbf{E} - \mathbf{F})}{\mathbf{G} + \mathbf{H} * \mathbf{K}}$$

a. Using a general register computer with three address instructions.
b. Using a general register computer with two address instructions.
c. Using an accumulator type computer with one address instructions.

Solution:

a.
```
SUB   R1,   A,    B
MUL   R2,   D,    E
SUB   R2,   R2,   F
MUL   R2,   R2,   C
ADD   R1,   R1,   R2
MUL   R3,   H,    K
ADD   R3,   R3,   G
DIV   X,    R1,   R3
```

b.
```
MOV   R1,   A
SUB   R1,   B
MOV   R2,   D
MUL   R2,   E
SUB   R2,   F
MUL   R2,   C
ADD   R1,   R2
MOV   R3,   H
MUL   R3,   K
ADD   R3,   G
DIV   R1,   R3
MOV   X,    R1
```

c.
```
LOAD      A
SUB       B
STORE     T
LOAD      D
MUL       E
SUB       F
MUL       C
ADD       T
STORE     T
LOAD      H
MUL       K
ADD       G
STORE     T2
LOAD      T
DIV       T2
```

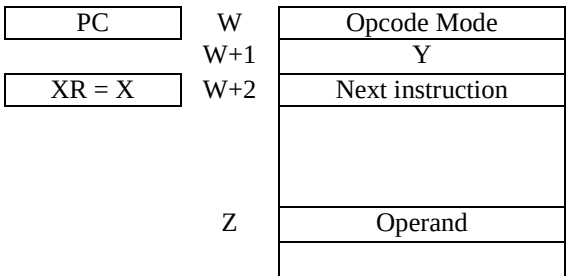
STORE X

Exercise 8:

A two-word instruction is stored in memory at an address designated by the symbol W. The address field of the instruction (stored at w+1) is designated by the symbol Y. The operand used during the execution of the instruction is stored at an address symbolized by Z. An index register contains the value X. State how Z is calculated from the other addresses if the addressing mode of the instruction is

a. Direct
b. Indirect
c. Relative
d. Indexed

Solution:



Z =Effective address

a. Direct: Z=Y
b. Indirect: Z=M[Y]
c. Relative: Z=Y+(W+2)
d. Indexed: Z=Y+X

Exercise 9:

A relative mode branch type of instruction is stored in memory at an address equivalent to decimal 750 (assume one word per instruction). The branch is made to an address equivalent to decimal 500.

a. What should be the value of the relative address field of the instruction (in decimal)?
b. Determine the relative address value in binary using 12 bits. (why must the number be in 2's complement?)
c. Determine the binary value in pc after the fetch phase and calculate the binary value of 500. Then show that the binary value in pc plus the relative address calculated in part (b) is equal to the binary value of 500.

Solution:

a. Relative address = 500 – 751 = -251
b. 251 = 000011111011, – 251 = 111100000101
c. PC = 751 = 001011101111 500 = 000111110100
   PC = 751 = 001011101111
   RA = -251 = +111100000101
   EA = 500 = 000111110100

Exercise 10:

How many times does the control unit refer to memory when it fetches and executes an indirect addressing mode instruction if the instruction is (a) a computational type requiring an operand from memory; (b) a branch type?

Solution:

Assuming one word per instruction or operand

Computational type	**Branch type**
Fetch instruction	Fetch instruction
Fetch effective address	Fetch effective address and transfer to PC
Fetch operand	
3 memory references	2 memory references

Exercise 11:

What must the address field of an indexed addressing mode instruction (with XR being the index register) be to make it the same as a register indirect mode instruction (with XR being the used register for indirect address generation)?

Solution:

The address field of an indexed mode instruction must be set to zero.

Exercise 12:

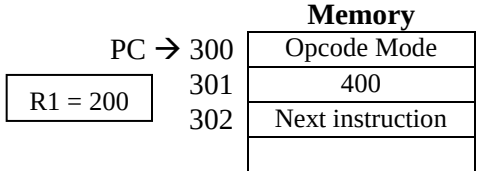
An instruction is stored at location 300 with its address field at location 301. The address field has the value 400. A processor register R1 contains the number 200. Evaluate effective address if the addressing mode of the instruction is

a. Direct
b. Immediate
c. Relative
d. Register indirect
e. Index with R1 as the index register.

Solution:

Effective address

a. Direct = 400
b. Immediate = 301
c. Relative = 302 + 400 = 702
d. Reg. Indirect = 200
e. Indexed = 200 + 400 = 600



Exercise 13:

An 8-bit computer has a register R. Determine the values of status bits C, S, Z, and V (Figure 8-8) after each of the following instructions. The initial value of register R in each case is hexadecimal 72. The numbers below are also in hexadecimal.

a. Add immediate operand C6 to R.
b. Add immediate operand 1E to R.
c. Subtract immediate operand 9A from R.
d. AND immediate operand 8D with R.
e. Exclusive-OR R with R.

Solution:

a.
$$\begin{array}{r} 01110010 \\ +\ 11000110 \\ \hline 00111000 \end{array} \qquad \begin{array}{r} 72 \\ +\ C6 \\ \hline 138 \end{array}$$

C=1, S=0, Z=0, V=0

b.
$$\begin{array}{r} 01110010 \\ +\ 00011110 \\ \hline 10010000 \end{array} \qquad \begin{array}{r} 72 \\ +\ 1E \\ \hline 90 \end{array}$$

C=0, S=1, Z=0, V=1

c. 2'complement of 9A = 01100110

$$\begin{array}{r} 01110010 \\ +\ 01100110 \\ \hline 11011000 \end{array} \qquad \begin{array}{r} 72 \\ -\ 9A \\ \hline D8 \end{array}$$

C=0, S=1, Z=0, V=1

d.
$$\begin{array}{r} 01110010 \\ \wedge\ 10001101 \\ \hline 00000000 \end{array} \qquad \begin{array}{r} 72 \\ \wedge\ 8D \\ \hline 00 \end{array}$$

C=0, S=0, Z=1, V=0

e. C=0, S=0, Z=1, V=0

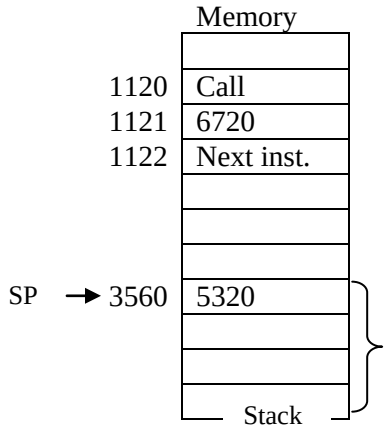
Exercise 14:

The content of the top of a *full-descending* memory stack is 5320. The content of the stack pointer SP is 3560. A two-word call subroutine instruction is located in memory at address 1120 followed by the address field of 6720 at location 1121. Note that all numbers are in decimal. What are the content of PC, SP, and the top of the stack:

a. Before the call instruction is fetched from memory?
b. After the call instruction is executed?
c. After the return from subroutine?

Solution:

a. PC = 1120 , SP=3560, top of the stack = 5320
b. PC = 6720 , SP=3559, top of the stack = 1122
c. PC = 1122 , SP=3560, top of the stack = 5320



Exercise 15:

A computer responds to an interrupt request signal by pushing onto a *full descending* stack the contents of PC and the current PSW (program status word), assume that each one will need one word. It then reads a new PSW from memory from a location given by an interrupt address symbolized by IAD. The first address of the service program is taken from memory at location IAD+1. Note: TR (temporary register could be used).

a. List the sequence of micro-operations for the interrupt cycle.
b. List the sequence of micro-operations for the return from interrupt instruction.

Solution:

a. SP ← SP-1
   M[SP] ← PC
   SP ← SP-1
   M[SP] ← PSW

TR ← IAD
PSW ← M[TR]
TR ← TR + 1
PC ← M[TR]

Go to fetch phase.

b. PSW←M[SP]
SP ← SP+1
PC←M[SP]
SP ← SP+1