

## F.4 Chapter 4 Solutions

### 4.1 Components of the Von Neumann Model:

- (a) Memory: Storage of information (data/program)
- (b) Processing Unit: Computation/Processing of Information
- (c) Input: Means of getting information into the computer. e.g. keyboard, mouse
- (d) Output: Means of getting information out of the computer. e.g. printer, monitor
- (e) Control Unit: Makes sure that all the other parts perform their tasks correctly and at the correct time.

4.3 The program counter does not maintain a count of any sort. The value stored in the program counter is the address of the next instruction to be processed. Hence the name 'Instruction Pointer' is more appropriate for it.

4.5 (a) Location 3 contains 0000 0000 0000 0000

Location 6 contains 1111 1110 1101 0011

(b) i. Two's Complement -

Location 0: 0001 1110 0100 0011 = 7747

Location 1: 1111 0000 0010 0101 = -4059

ii. ASCII - Location 4: 0000 0000 0110 0101 = 101 = 'e'

iii. Floating Point -

Locations 6 and 7: 0000 0110 1101 1001 1111 1110 1101 0011

Number represented is  $1.1011001111111011010011 \times 2^{-114}$

iv. Unsigned -

Location 0: 0001 1110 0100 0011 = 7747

Location 1: 1111 0000 0010 0101 = 61477

(c) Instruction - Location 0: 0001 1110 0100 0011 = Add R7 R1 R3

- (d) Memory Address - Location 5: 0000 0000 0000 0110 Refers to location 6. Value stored in location 6 is 1111 1110 1101 0011

4.7 60 opcodes = 6 bits

32 registers = 5 bits

So number of bits required for IMM =  $32 - 6 - 5 - 5 = 16$

Since IMM is a 2's complement value, its range is  $-2^{15} \dots (2^{15} - 1) = -32768 \dots 32767$ .

4.9 The second important operation performed during the FETCH phase is the loading of the address of the next instruction into the program counter.

4.11 The phases of the instruction cycle are:

- Fetch: Get instruction from memory. Load address of next instruction in the Program Counter.
- Decode: Find out what the instruction does.
- Evaluate Address: Calculate address of the memory location that is needed to process the instruction.
- Fetch Operands: Get the source operands (either from memory or register file).
- Execute: Perform the execution of the instruction.
- Store Result: Store the result of the execution to the specified destination.

4.13

		F	D	EA	FO	E	SR	
x86:	ADD [eax] edx	100	1	1	100	1	100	= 303
LC3:	ADD R6, R2, R6	100	1	-	1	1	1	= 104

4.15 Once the RUN latch is cleared, the clock stops, so no instructions can be processed. Thus, no instruction can be used to set the RUN latch. In order to re-initiate the instruction cycle, an external input must be applied. This can be in the form of an interrupt signal or a front panel switch, for example.

4.17

	R/W	MAR	MDR
<b>Operation 1</b>	W	x4000	1 1 1 1 0
<b>Operation 2</b>	R	x4003	1 0 1 1 0
<b>Operation 3</b>	W	x4001	1 0 1 1 0
<b>Operation 4</b>	R	x4002	0 1 1 0 1
<b>Operation 5</b>	W	x4003	0 1 1 0 1

Before Access 1

X4000	0 1 1 0 1
X4001	1 1 0 1 0
X4002	0 1 1 0 1
X4003	1 0 1 1 0
X4004	1 1 1 1 0
X4000	1 1 1 1 0
X4001	1 0 1 1 0
X4002	0 1 1 0 1
X4003	1 0 1 1 0
X4004	1 1 1 1 0

After Access 3

After Access 5

X4000	0 1 1 0 1
X4001	1 1 0 1 0
X4002	0 1 1 0 1
X4003	0 1 1 0 1
X4004	1 1 1 1 0

- 4.19 (a) MAR: x2  
(b) MDR: 00111001

MDR: 01010000