**NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY**

School of Electrical Engineering and Computer Sciences

***COAL ASSIGNEMNT DOCUMENTATION***

***CLASS: BSCS-11-A***

***COURSE: COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE***

***SUBMITTED TO: SIR IMRAN ABEEL***

***DATE OF SUBMISSION: DEC-04-22***

***GROUP MEMBERS:***

***AZKA BASIT***

***RAMEEN AAMIR***

***HARAM NASIR***

***VIRTUAL 8086 PROCESSOR***

**REGISTERS (08):**

|  |  |  |  |
| --- | --- | --- | --- |
| **AX** | **BX** | **CX** | **DX** |
| **SI** | **DI** | **IP** | **BP** |

**MEMORY LOCATIONS (16):**

|  |  |  |  |
| --- | --- | --- | --- |
| **[00000]** | **[00001]** | **[00002]** | **[00003]** |
| **[00004]** | **[00005]** | **[00006]** | **[00007]** |
| **[00008]** | **[00009]** | **[0000A]** | **[0000B]** |
| **[0000C]** | **[0000D]** | **[0000E]** | **[0000F]** |

|  |  |
| --- | --- |
| **Registers:** | **Opcode:** |
| **AX** | 000 |
| **BX** | 011 |
| **CX** | 001 |
| **DX** | 010 |
| **SI** | 110 |
| **DI** | 111 |
| **BP** | 101 |
| **SP** | 100 |

**OPCODE OF REGISTERS:**

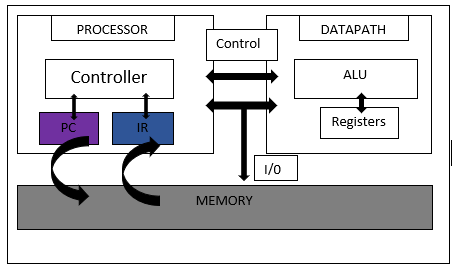
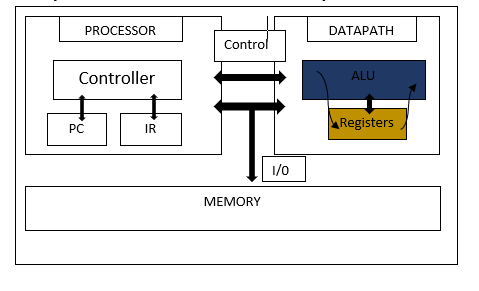
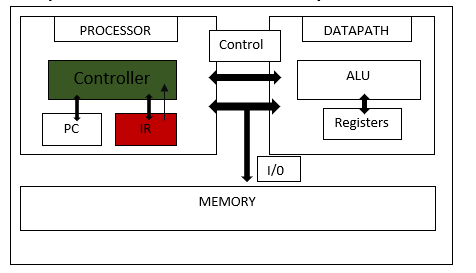
|  |  |
| --- | --- |
| **Operations:** | **Opcode:** |
| **MOV** | 100010 |
| **ADD** | 000000 |
| **SUB** | 000101 |
| **DEC** | 111111 |
| **INC** | 111111 |
| **NOT** | 111101 |
| **AND** | 001000 |
| **OR** | 000010 |
| **CMP** | 001110 |
| **NEG** | 111101 |
| **XOR** | 000110 |
| **NOP** | 100100 |

**OPCODE OF INSTRUCTIONS:**

***1.MOV INSTRUCTION***

**“MOV”**

**(Register to register) – 3 clock cycles**



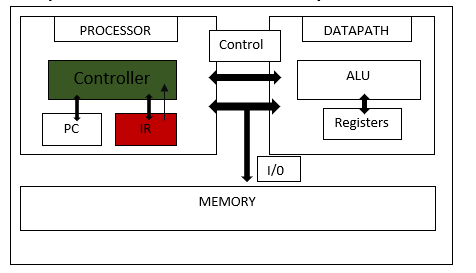
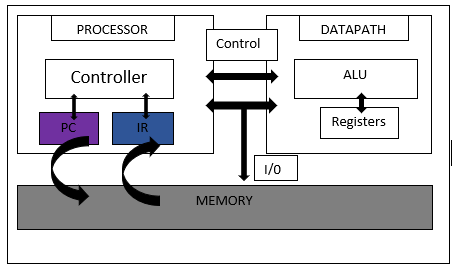
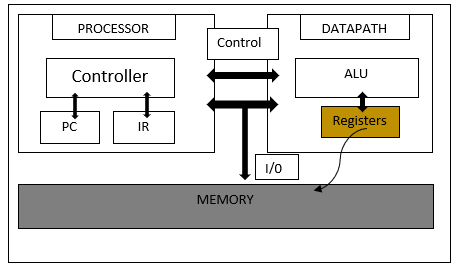
FETCH

DECODE

EXECUTEE

**“MOV”**

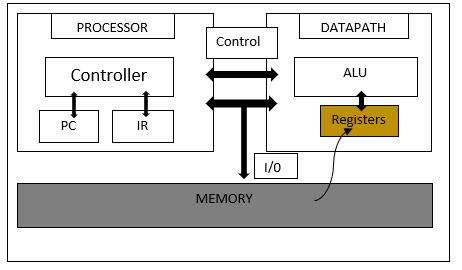
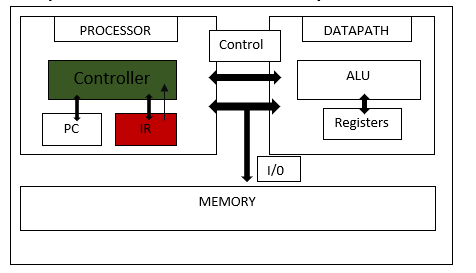
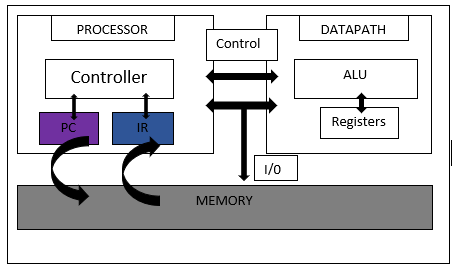
**(Register to memory) – 3 clock cycles**



STORE

**“MOV”**

**(Memory to register) – 3 clock cycles**



LOAD

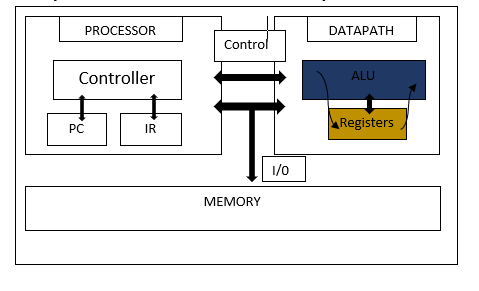
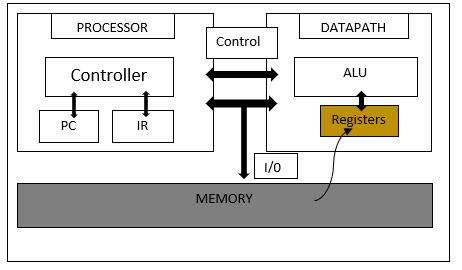
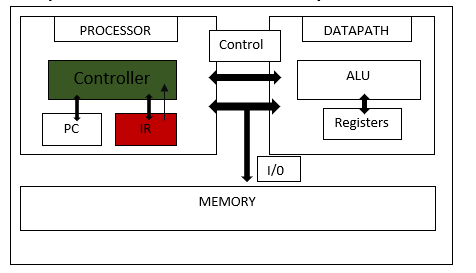
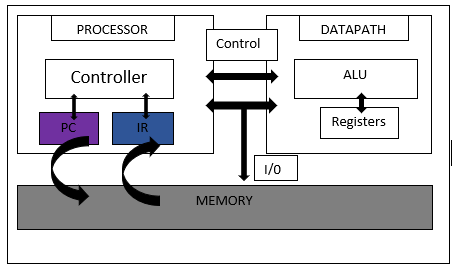
***2.ADD INSTRUCTION***

**“ADD”**

**(Register, register) – 3 clock cycles**

**“ADD”**

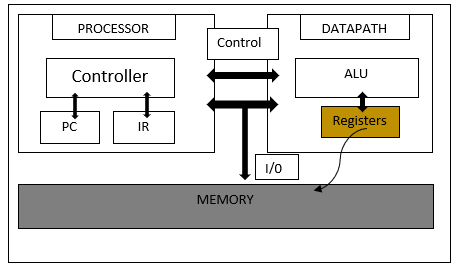
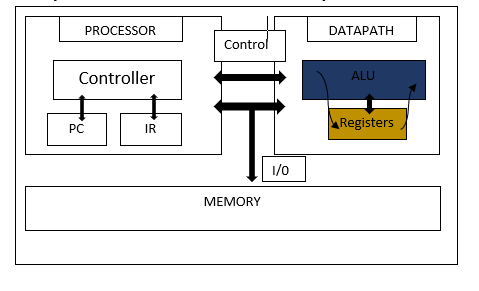
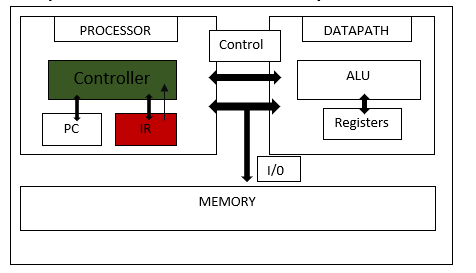
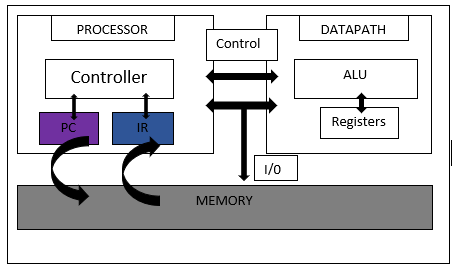
**(Register, memory) – 4 clock cycles**



EXECUTE

**“ADD”**

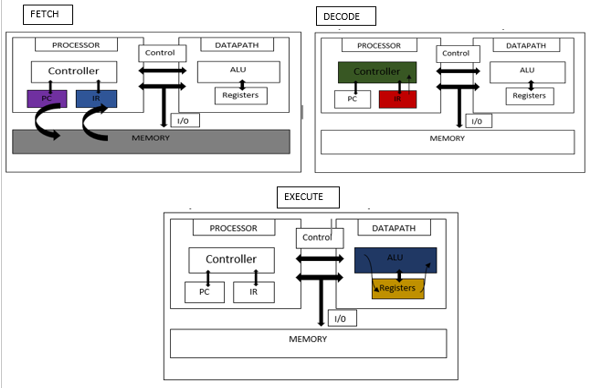
**(Memory, register) – 4 clock cycles**



EXECUTE

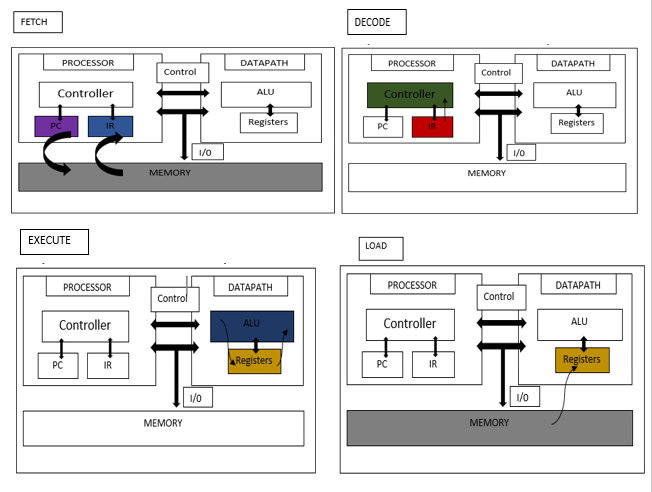
***3.SUB INSTRUCTION:***

**“SUB”**

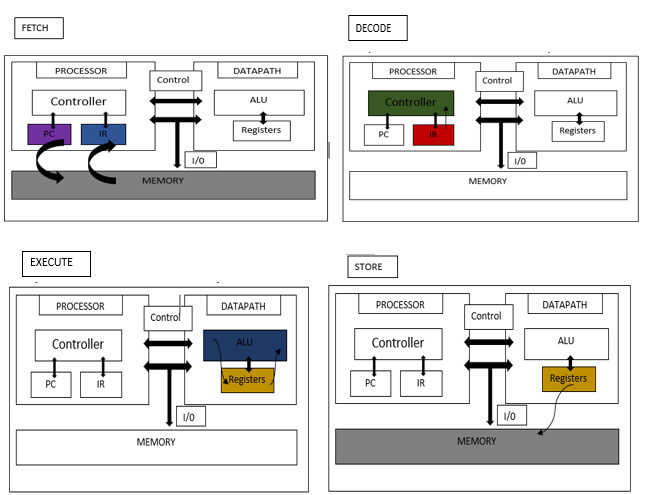
**(Register, register) – 3 clock cycles**

**“SUB”**

**(Register, memory) – 4 clock cycles**



**“SUB”**

**(Memory, register) – 4 clock cycles**

***4.INC INSTRUCTION:***

**“INC”**

**(register) – 3 clock cycles**

**“INC”**

**(memory) – 3 clock cycles**

****

***5.DEC INSTRUCTION:***

**“DEC”**

**(register) – 3 clock cycles**

****

**“DEC”  
(memory) – 3 clock cycles**



***6. AND INSTRUCTION:***

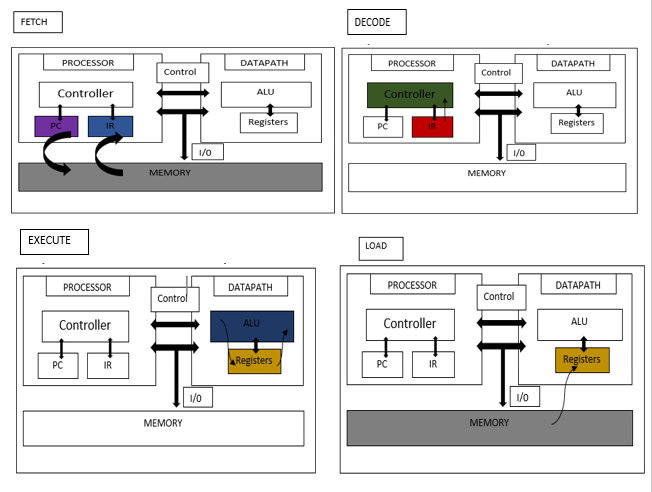
**“AND”**

**(register, register) – 3 clock cycles**

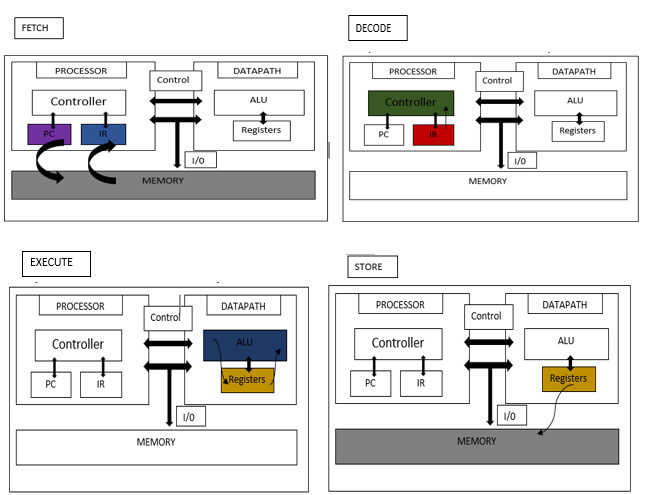


**“AND”**

**(register, memory) – 4 clock cycles**



**“AND”**

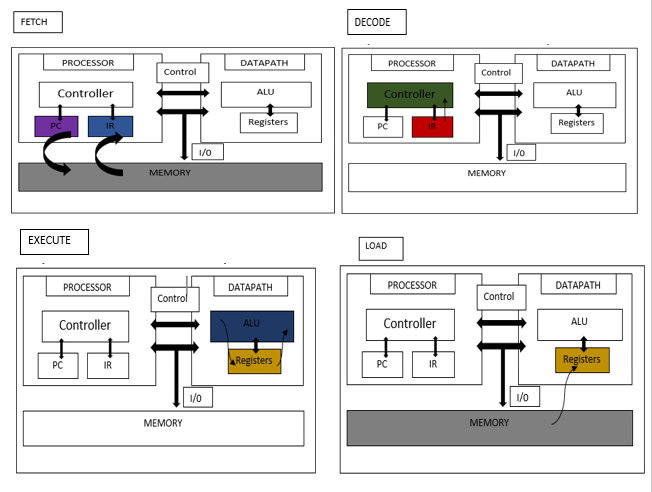
**(memory, register) – 4 clock cycles**

***7. OR INSTRUCTION:***

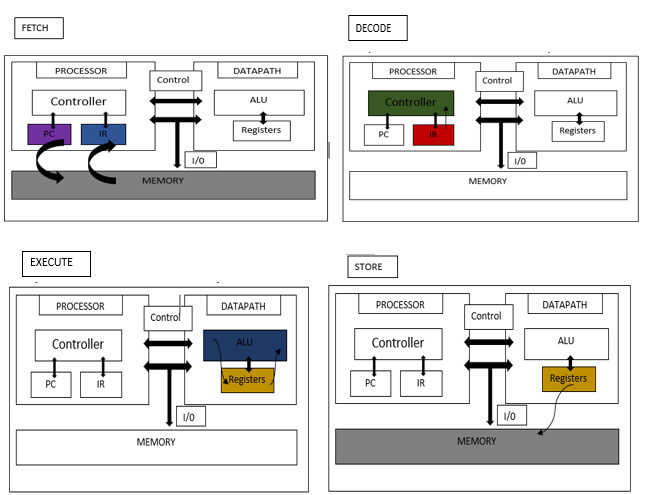
**“OR”  
(register, register) – 3 clock cycles**



**“OR”  
(register, memory) – 4 clock cycles**



**“OR”**

**(memory, register) – 4 clock cycles**

***8. NOT INSTRUCTION:***

**“NOT”  
(register) – 3 clock cycles**



**“NOT”  
(memory) – 3 clock cycles**

****

***9. NEG INSTRUCTION:***

**“NEG”  
(register) – 3 clock cycles**



**“NEG”  
(memory) – 3 clock cycles**



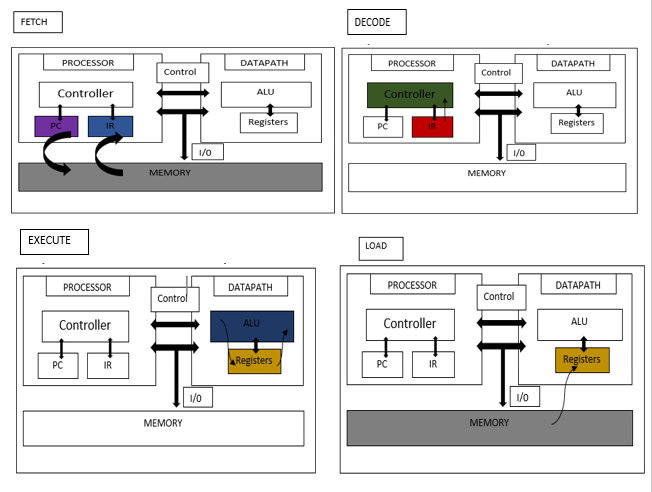
***10. XOR INSTRUCTION:***

**“XOR”**

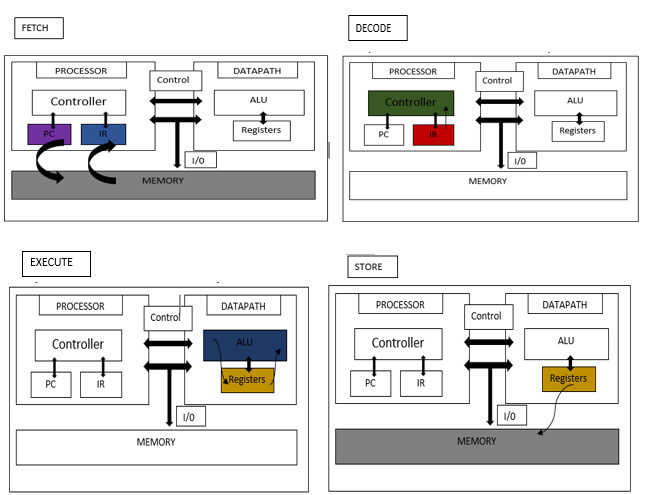
**(register, register) – 3 clock cycles**



**“XOR”  
(register, memory) – 4 clock cycles**



**“XOR”  
(memory, register) – 4 clock cycles**



***11. CMP INSTRUCTION:***

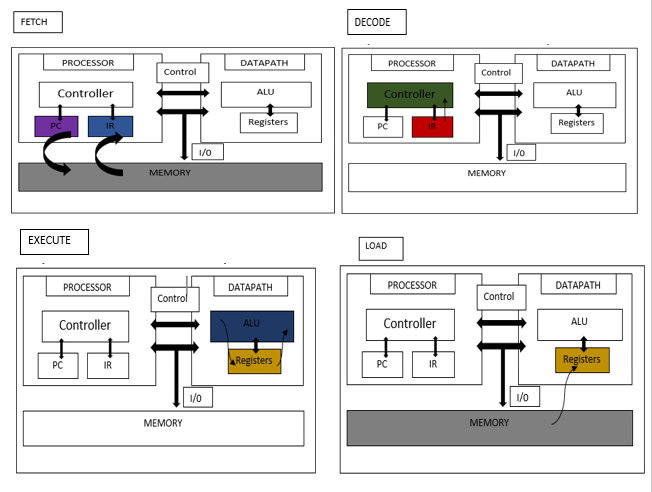
**“CMP”**

**(register, register) – 3 clock cycles**

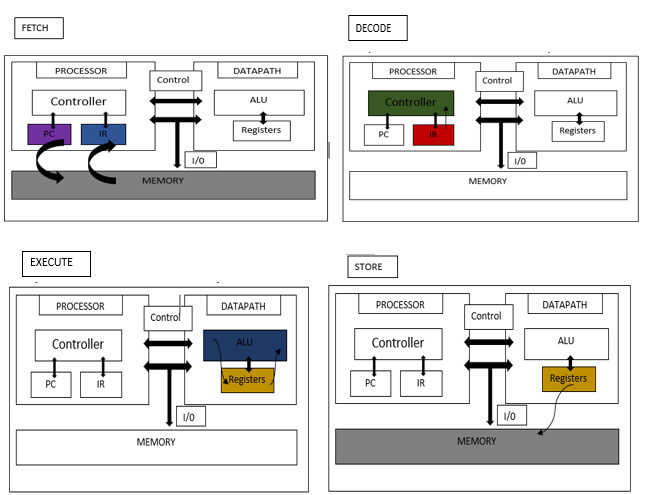


**“CMP”**

**(register, memory) – 4 clock cycles**

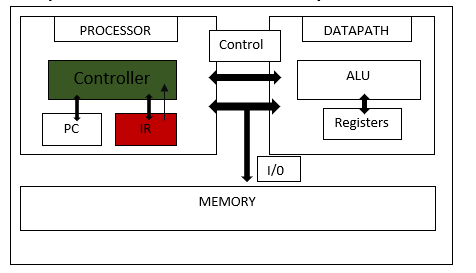
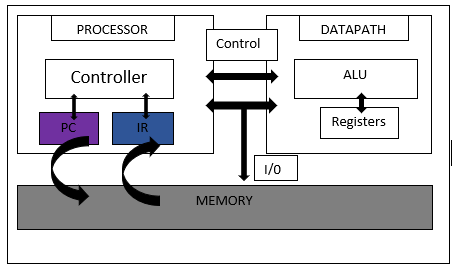


**“CMP”**

**(memory, register) – 4 clock cycles**

***12. NOP INSTRUCTION:***

**“NOP”  
(register, register) – 2 clock cycles**



**“NOP”  
(memory, register) – 2 clock cycles**

**“NOP”  
(register, memory) – 2 clock cycles**

