# FE Cycle Register Questions

# In register transfer notation, the fetch-execute cycle is described as follows:

# MAR 🡨 [PC]

# PC 🡨 [PC] + 1; MBR 🡨 [Memory] addressed`

# CIR 🡨 [MBR]

# [CIR] decoded and executed

# Where: ‘[PC]’ means the contents of PC

#  ‘MAR 🡨’ means assigned to MAR

#  ‘x;y’ means operation x and operation y occur together

### Instruction Set

|  |  |  |
| --- | --- | --- |
| Hex | Instruction  | Description |
| 0 | Halt | Halts |
| 1 | Load | Loads Memory address to accumulator |
| 2 | Load# | Loads Literal Values to accumulator |
| 3 | LoadI | Loads Indirect memory address to accumulator |
| 4 | Add | Adds Memory address to accumulator |

### Memory DUMP

|  |  |
| --- | --- |
| Address | Contents (hex) 4bit opcode 4bit oprand |
| 00 | 2 6 |
| 01 | 4 5 |
| 02 | A 9 |
| 03 | 0 0  |
| 04 |  |
| 05 | 0 1 |
| 06 |  |
| 09 |  |

### Register Dry-run Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pc | MAR | MBR | CIR | Decoded Instruction | Oprand | ACC |
| 0 |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |
| 1 |  | 2 6 |  |  |  |  |
|  |  |  | 2 6 | Load#  | 6 |  |
|  |  |  |  |  |  | 6 |
|  | 1 |  |  |  |  |  |
| 2 |  | 4 5 |  |  |  |  |
|  |  |  | 4 5 | Add  | 5 |  |
|  | 5 |  |  |  |  |  |
|  |  | 0 1 |  |  |  |  |
|  |  |  |  |  |  | 7 |
|  | 2 |  |  |  |  |  |
| 3 |  | A 9 |  |  |  |  |
|  |  |  | A 9 | Store | 9 |  |
|  | 9 | 7 |  |  |  |  |

7 would be written to Memory location 9

### Instruction Set

|  |  |  |
| --- | --- | --- |
| Hex | Instruction  | Description |
| 0 | Halt | Halts |
| 1 | Load | Loads Memory address to accumulator |
| 2 | Load# | Loads Literal Values to accumulator |
| 3 | LoadI | Loads Indirect memory address to accumulator |
| 4 | Add | Adds Memory address to accumulator |
| 5 | Add# | Adds Literal Values to accumulator |
| 6 | AddI | Adds Indirect memory address to accumulator |
| 7 | Sub# | Subtracts Memory address to accumulator |
| 8 | Sub | Subtracts Literal Values to accumulator |
| 9 | SubI | Subtracts Indirect memory address to accumulator |
| A | Store | Store accumulator value to memory address |
| B | StoreI | Store accumulator value to indirect memory address |
| C | Jump | Unconditional jump |
| D | Jump0 | Jump when accumulator has value = 0 |
| E | JumpI | Jump to indirect memory address |

|  |  |
| --- | --- |
| Address | Contents (hex) 4bit opcode 8bit oprand |
| 00 |  |
| 01 |  |
| 02 |  |
| 03 |  |
| 04 |  |
| 05 |  |
| 06 |  |
| 09 |  |
| 0A |  |
| 0B |  |
| 0C |  |
| 0D |  |
| 0E |  |
| 0F |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 19 |  |
| 1A |  |
| 1B |  |
| 1C |  |
| 1D |  |
| 1E |  |
| 1F |  |