**Computer architecture project**

**Create a microprocessor**



**Team members:**

Mohamed Taha Salah 192000280

Abdulrahman Gaber 192000272

Salwa Hisham Ismail 192000277

Mohamed Heshmat 192000255

**Submitted to:**

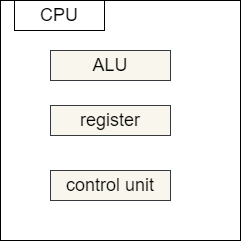
DR: Amr Zamel

TA: Nancy El Arabawy

**Introduction:**

It is required to design and build a simple processor (non- pipelined) with all the components introduced throughout computer architecture course, size of instruction = 16 bits

**CPU Content:**



**CPU:** The central processing unit (CPU) is that portion of a computer that fetches and executes instructions. It consists of an ALU, control unit and registers in a system with a signal processing unit.

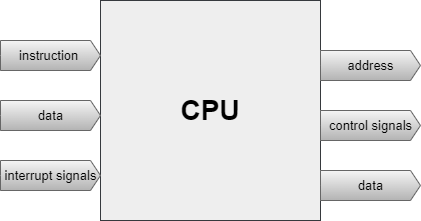
**ALU:** arithmetic and logic unit (ALU), which performs simple arithmetic and logical operations.

**Register:** a register file is an array of processor registers in a central processing unit.

**Control unit:** all computer operations are controlled by the control unit.

**CPU Connection:**

The processor contains a single general-purpose register called an accumulator (AC), a Program Counter (PC), a Memory Address Register (MAR) and a Memory Buffer Register (MBR), Instruction register (IR).



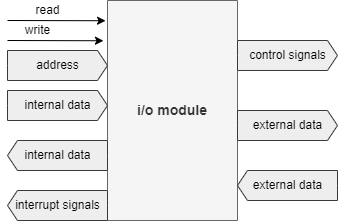
**Fetch Cycle**

1. Instruct stored into pc.
2. Pc sends it to MAR.
3. MAR sends the address to the main memory.
4. Main memory sends content of address to MDR.
5. MDR sends address content to IR.
6. Pc increment by one.

**Execute Cycle:**

1. IR sends the location that stores in ROM to MAR.
2. Accumulator sends content of address to ALU.
3. All of this happened under management of the control unit.

**Input / Output Connection:**



**Main memory:**

**Data memory**: (ROM) read the instruction.

**Instruction memory**: (RAM) stores the instructions of a program given an address.



**Memory Constraints:**

Instructions >>>>> 16 Bit

Number of instructions >>>>> 16

Opcode >>>>> 4 Bit

Address >>>>> 12 Bit

Data >>>>> 16 Bit

|  |  |
| --- | --- |
| Opcode (binary) | operation |
| 0000 | Load AC from memory |
| 0001 | Store AC into memory |
| 0010 | Add to AC from memory, Result in AC |
| 0011 | Subtract memory from AC, Result in AC |
| 0100 | Logical AND memory and AC , Result in AC |
| 0101 | Logical OR memory and AC , Result in AC |
| 0110 | Logical XOR memory and AC , Result in AC |
| 0111 | Logical NOT memory , Result in AC |
| 1000 | Logical SHL memory , Result in AC |
| 1001 | Logical SHR memory , Result in AC |
| 1010 | Logical ROL memory , Result in AC |
| 1011 | Logical ROR memory , Result in AC |
| 1100 | MOV memory int AC, Result in AC |
| 1101 | Increment memory , Result in AC |
| 1110 | Decrement memory , Result in AC |
| 1111 | Reset memory , Result in AC |

**Main Memory (RAM):**

|  |  |
| --- | --- |
| Address (Hexa decimal) | Content |
| E2E | 0002 |
| E2F | 0001 |
| E30 | 0008 |
| E31 | 0007 |
| E32 | 0005 |
| E33 | 0003 |
| E34 | 0F0F |
| E35 | 000A |
| E36 | 0000 |

**Instructions in (ROM):**

|  |  |
| --- | --- |
| Index | Instructions |
| 0 | 0E2E |
| 1 | 2E35 |
| 2 | 3E30 |
| 3 | 1E32 |
| 4 | DE32 |
| 5 | 1E2F |
| 6 | EE2F |
| 7 | 1E33 |

