** Cairo University**

**Faculty of Engineering**

**Project Arch**

**Phase 1**

|  |  |  |
| --- | --- | --- |
| Name | Section | B.N |
| Aly Ramzy Hassan | 1 | 36 |
| Mouhamed Khier | 2 | 12 |
| Nour Ahmed | 2 | 31 |
| Hager Ahmed | 2 | 34 |

**Instruction format of your design**

1. Opcode of each instruction

One Operand

|  |  |
| --- | --- |
| Group1(one operand) | Opcode |
| Nop | 00000 |
| Not Rdist | 00001 |
| Inc Rdst | 00010 |
| DEC  Rdst | 00011 |
| OUT  Rdst | 00100 |
| IN  Rdst | 00101 |

Two Operand

|  |  |
| --- | --- |
| Group2(Two operand) | Opcode |
| SWAP Rsrc, Rdst | 01000 |
| ADD Rsrc1, Rsrc2, Rdst | 01001 |
| IADD Rsrc1,Rdst,Imm | 01010 |
| SUB  Rsrc1, Rsrc2, Rdst | 01011 |
| AND  Rsrc1, Rsrc2, Rdst | 01100 |
| OR  Rsrc1, Rsrc2, Rdst | 01101 |
| SHL Rsrc, Imm | 01110 |
| SHR Rsrc, Imm | 01111 |
|  |  |

Memory Operation

|  |  |
| --- | --- |
| Group3 (Memory Operation) | Opcode |
| PUSH  Rdst | 10000 |
| POP  Rdst | 10001 |
| LDM  Rdst, Imm | 10010 |
| LDD  Rdst, EA | 10011 |
| STD Rsrc, EA | 10100 |

Branch /Control Operations

|  |  |
| --- | --- |
| Group4(branch) | opcode |
| JZ  Rdst | 11000 |
| JMP  Rdst | 11001 |
| CALL  Rdst | 11010 |
| RET | 11011 |
| RTI | 11100 |

1. **Instruction bits details**

**One Operand**

NOOP

|  |
| --- |
| OP Code 5 bit  31:27 |

NOT/INC/DEC/OUT/IN

|  |  |
| --- | --- |
| OP Code 5 bit 31:27 | Register dst index 3 bit 26:24 |

**Two Operand**

ADD/SUB/AND/OR

|  |  |  |  |
| --- | --- | --- | --- |
| OP Code 5 bit 31:27 | Register src1 3 bit  26:24 | Register src2 3 bit  23:21 | Register dst index 3 bit  20:18 |

SHL/SHR

|  |  |  |
| --- | --- | --- |
| OP Code 5 bit  31:27 | Register src1 3 bit  26:24 | Immediate value 16 bit 23:8 |

SWAP

|  |  |  |
| --- | --- | --- |
| OP Code 5 bit 31:27 | Register src1 3 bit  26:24 | Register src2 3 bit  23:21 |

IADD

|  |  |  |  |
| --- | --- | --- | --- |
| OP Code 5 bit 31:27 | Register src1 3 bit  26:24 | Register dst 3 bit  23:21 | Immediate Value 16 bit 20:5 |

**Memory Operation**

PUSH/POP

|  |  |
| --- | --- |
| OP Code 5 bit  31:27 | Register dst 3 bit  26:24 |

LDM

|  |  |  |  |
| --- | --- | --- | --- |
| OP Code 5 bit  31:27 | Register src1 3 bit  26:24 | Empty  23:21 | Immediate Value 16 bit 20:5 |

LDD/STD

|  |  |  |
| --- | --- | --- |
| OP Code 5 bit  31:27 | Register src1 3 bit  26:24 | Effective Address 20 bit 23:4 |

**Branch /Control Operations**

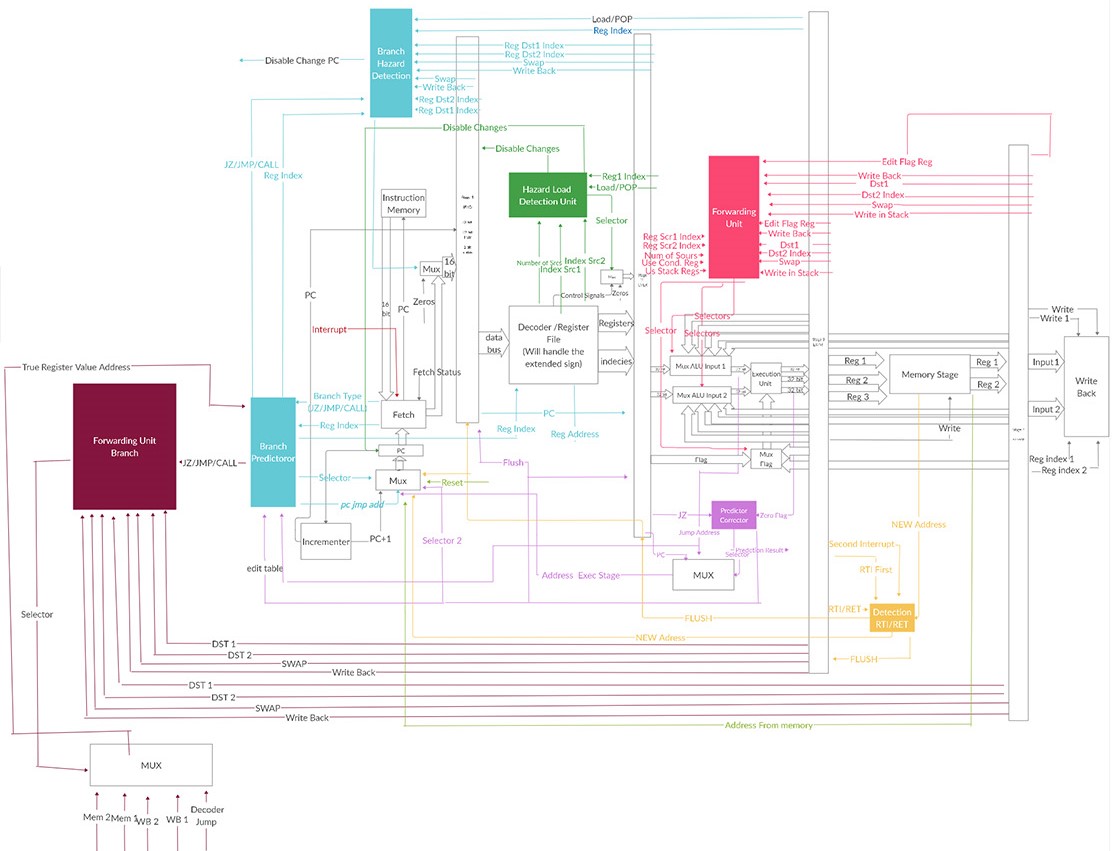
JZ/JMP/CALL

|  |  |
| --- | --- |
| OP Code 5 bit  31:27 | Register dst 3 bit  26:24 |

RET/RTI

|  |
| --- |
| OP Code 5 bit  31:27 |

**Schematic diagram of the processor with data flow details.**

**Design:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stage 1 (37 bits)** | | | | |
| **Swap** | **Wp** | **Pred result** | **Fetch status** | **instruction** |
| 1 | 1 | 1 | 2 | 32 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Stage 2 (139 bits)** | | | | | | | | | | | | | | | | | | |
| **Swap** | **Load/ pop** | **Num sources** | **JZ** | **Pred result** | **Edit flag** | **RET/ RTI** | **RTI first** | **Second interrupt** | **Reg1** | **Reg2** | **Reg3** | **R1** | **R2** | **R3** | **R4** | **Opcode** | **Selectors** | **WB** |
| 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 32 | 32 | 32 | 4 | 4 | 4 | 4 | 5 | 8 | 4 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Stage 3 (156 bits)** | | | | | | | | | | | | | | | |
| **opcode** | **reset** | **edit flag** | **Load/ pop** | **Swap** | **RET/ RTI** | **RTI first** | **Second interrupt** | **Reg1** | **Reg2** | **Reg3** | **Reg4** | **R1** | **R2** | **Selectors** | **WB** |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 32 | 32 | 32 | 32 | 4 | 4 | 4 | 4 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Stage 4 (85 bits)** | | | | | | |
| **interrupt** | **opcode** | **Edit flag** | **Swap** | **Reg1** | **Reg2** | **R1** | **R2** | **WB** |
| 2 | 5 | 1 | 1 | 32 | 32 | 4 | 4 | 4 |

**Control Unit Signals**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Reset** | **RET/RTI/Interrupt** | **Pred result** | **branch** |
| **Address from memory** | 1 | X | X | X |
| **Detection address** | 0 | 1 | X | X |
| **Predictor address** | 0 | 0 | 1 | X |
| **PC JMP address** | 0 | 0 | 0 | 1 |
| **PC + 1** | 0 | 0 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **selector** | **Use flag register (execution stage) JZ** | **edit flag register (memory stage)** | **edit flag register (write back stage)** |
| **00** | 0 | X | X |
| **00** | 1 | 0 | 0 |
| **01** | 1 | 0 | 1 |
| **10** | 1 | 1 | 0 |
| **10** | 1 | 1 | 1 |

Mux selectors in our design:

We have 8 multiplexers: A, B, C, D, F, E, F, G and its selectors are stated in the table below:

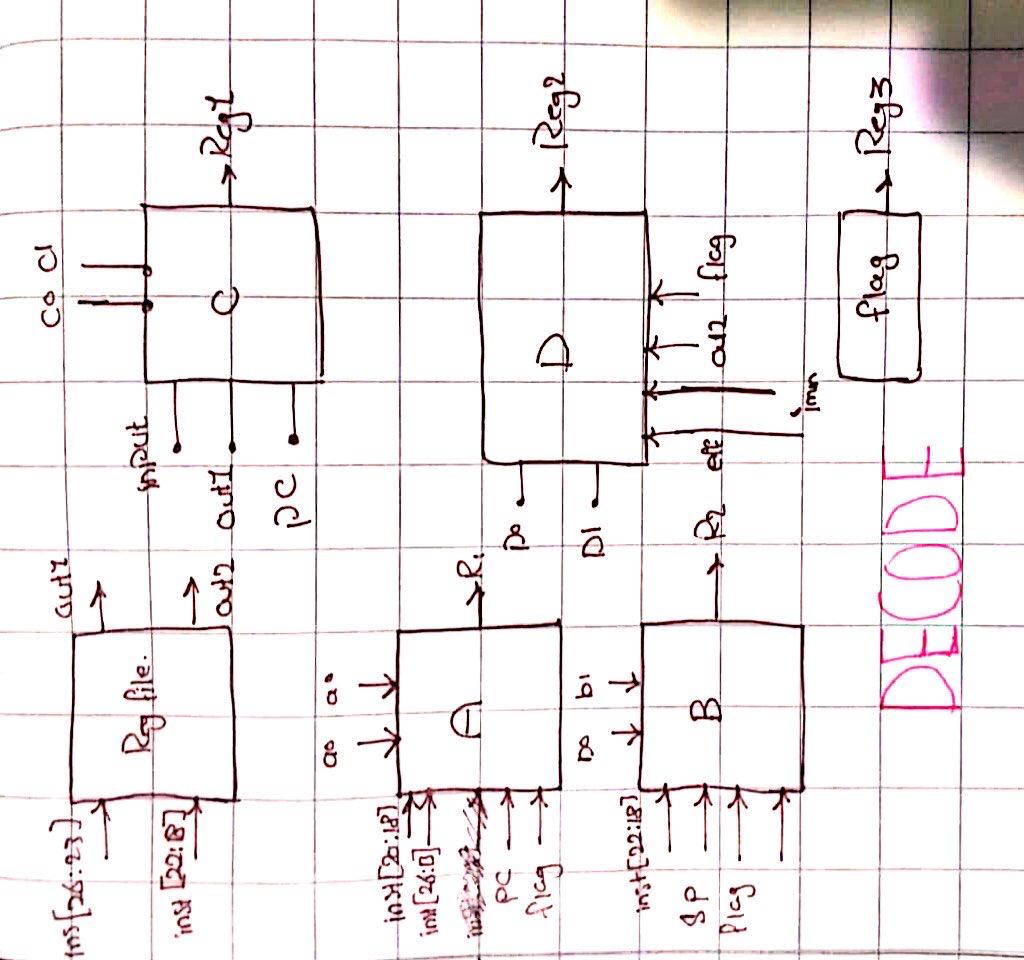
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | a1 | a0 | b1 | b0 | c1 | c0 | d1 | d0 | e0 | f1 | f0 | g1 | g0 | H1 | H0 | i0 | j0 | k |
| 1 | IADD | 1 |  | 1 |  |  |  | 1 | 1 | 1 |  | 1 |  | 1 |  |  |  |  |  |
| 2 | SHL,SHR |  |  | 1 |  |  |  | 1 | 1 | 1 |  | 1 |  | 1 |  |  |  |  |  |
| 3 | LDM |  |  |  |  |  |  | 1 | 1 |  | 1 |  |  |  |  |  |  |  |  |
| 4 | LDD |  |  |  |  |  |  |  | 1 |  |  |  |  |  | 1 |  |  |  |  |
| 5 | STD |  |  |  |  |  |  |  | 1 | 1 |  |  |  |  |  |  |  |  |  |
| 6 | NOP |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | NOT, DEC, INC |  |  | 1 |  |  |  | 1 |  | 1 |  | 1 |  | 1 |  |  |  |  |  |
| 8 | OUT |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |  |
| 9 | IN |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | SWAP |  |  |  | 1 |  |  |  |  |  | 1 |  | 1 | 1 |  |  |  |  |  |
| 11 | ADD, SUB, OR, AND |  | 1 | 1 |  |  |  |  |  | 1 |  | 1 |  | 1 |  |  |  |  |  |
| 12 | PUSH |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  |
| 13 | POP |  |  |  |  |  |  |  |  |  |  |  | 1 |  | 1 |  |  |  |  |
| 14 | JZ |  |  | 1 | 0 |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |
| 15 | JMP |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 | CALL |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  | 1 | 1 |  | 1 |
| 17 | RET |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |
| 18 | RTI)1st |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |  |
| 19 | RTI)2nd | 1 | 1 |  |  |  |  |  |  |  |  |  | 1 |  | 1 |  |  |  |  |
| 20 | INT)1st |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  | 1 |  |  |
| 21 | INT)2nd | 1 |  |  |  | 1 |  |  |  |  |  |  |  |  | 1 |  | 1 | 1 |  |
| 22 | INT)3rd | 1 | 1 |  |  |  |  |  |  |  | 1 | 1 |  |  |  |  | 1 |  |  |

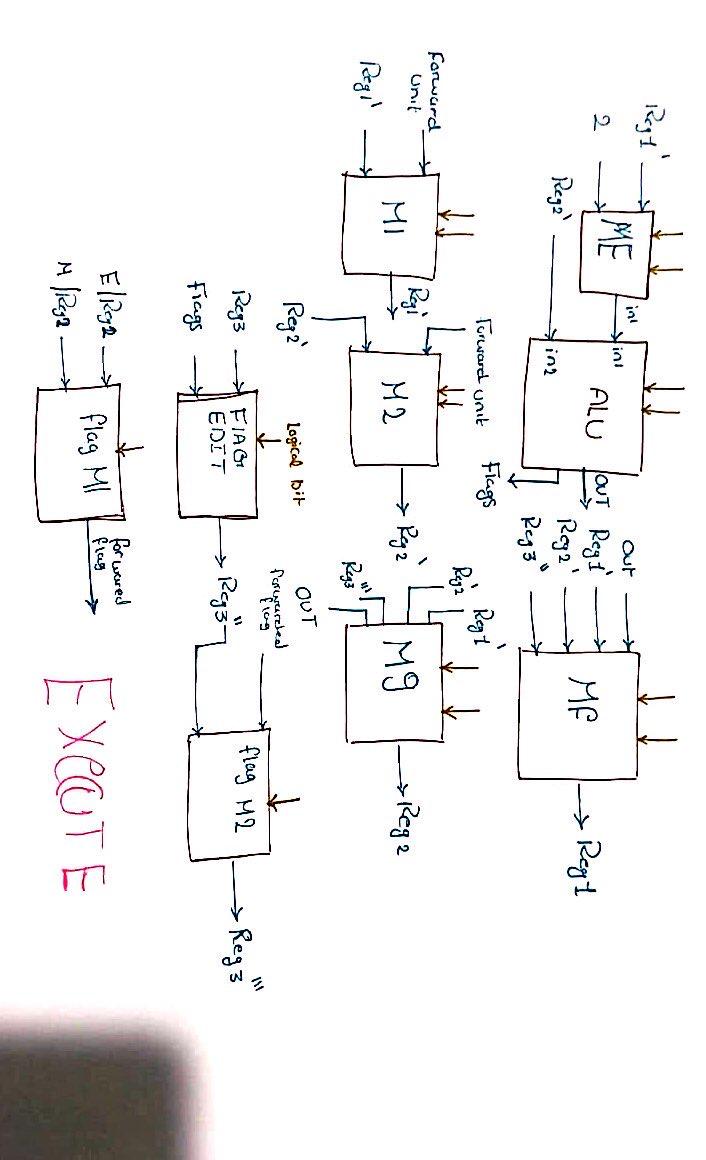
Signal table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | W0 | W2 | W1 | Right out | swap | Load  Pop  RTI second | num | wb | Prediction result | JZ | Edit  flag | RET | RTI  first | Int  sec |
| 0 | IADD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | IADD (2nd) |  | 1 | 1 |  |  |  | 01 | 1 |  |  | 1 |  |  |  |
| 2 | SHL,SHR |  | 1 | 1 |  |  |  | 01 | 1 |  |  | 1 |  |  |  |
| 3 | LDM |  |  | 1 |  |  |  |  | 1 |  |  |  |  |  |  |
| 4 | LDD |  |  | 1 |  |  | 1 |  | 1 |  |  |  |  |  |  |
| 5 | STD | 1 |  |  |  |  |  | 01 |  |  |  |  |  |  |  |
| 6 | NOP |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | NOT, DEC, INC |  | 1 | 1 |  |  |  | 01 | 1 |  |  | 1 |  |  |  |
| 8 | OUT |  |  |  | 1 |  |  | 01 |  |  |  |  |  |  |  |
| 9 | IN |  |  | 1 |  |  |  |  | 1 |  |  |  |  |  |  |
| 10 | SWAP |  | 1 | 1 |  | 1 |  | 10 | 1 |  |  |  |  |  |  |
| 11 | ADD, SUB, OR, AND |  | 1 | 1 |  |  |  | 10 | 1 |  |  | 1 |  |  |  |
| 12 | PUSH | 1 | 1 |  |  |  |  | 01 |  |  |  |  |  |  |  |
| 13 | POP |  | 1 | 1 |  |  | 1 |  | 1 |  |  |  |  |  |  |
| 14 | JZ |  |  |  |  |  |  | 10 |  | 0/1 | 1 |  |  |  |  |
| 15 | JMP |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 | CALL | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 | RET |  | 1 |  |  |  |  |  |  |  |  |  | 1 |  |  |
| 18 | RTI)1st |  | 1 |  |  |  |  |  |  |  |  |  |  | 1 |  |
| 19 | RTI)2nd |  | 1 | 1 |  |  | 1 |  |  |  |  | 1 |  |  |  |
| 20 | INT)1st | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 21 | INT)2nd |  | 1 |  |  |  |  |  |  |  |  |  |  |  | 1 |
| 22 | INT)3rd | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |

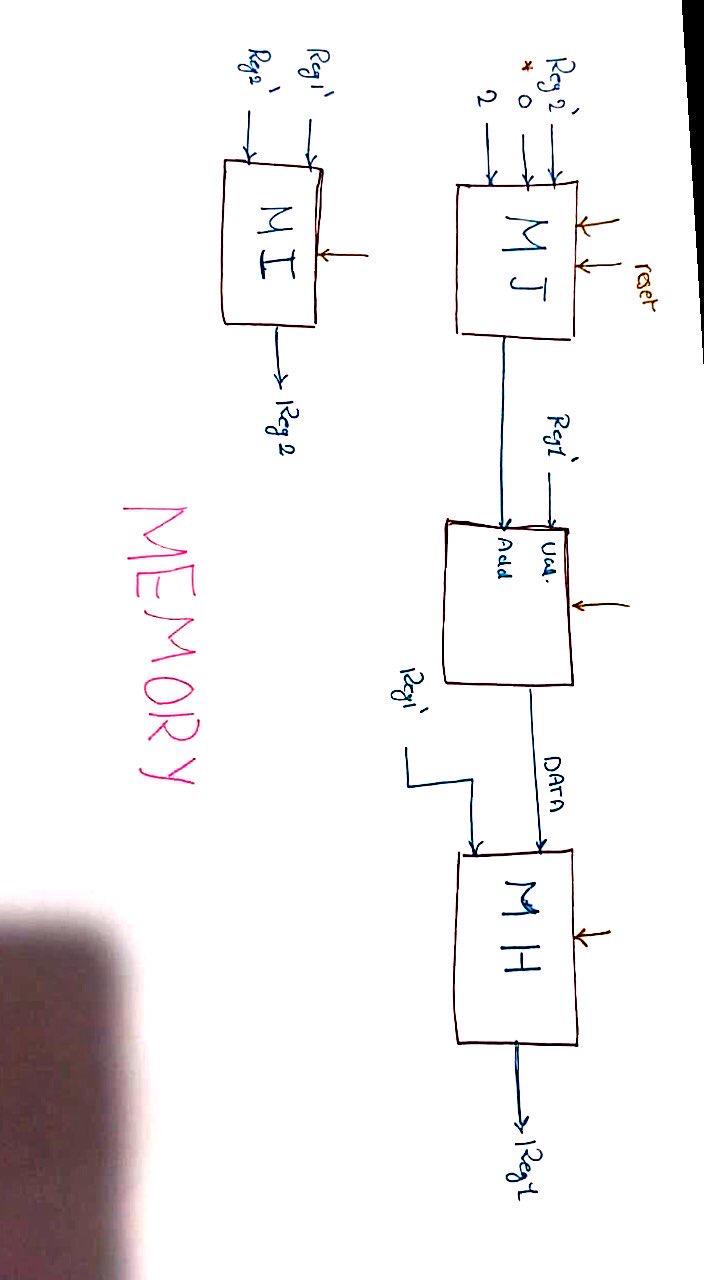
**Detailed design**

**Fetch:**

**Decode:**

**Execute:**

**Memory:**

****

**Write Back:**