

Cairo University

Faculty of Engineering

CMP N301

Computer Architecture

Design Document

Phase-2

Made by:

Sahar Hossam 1142335

Ahmed Mohamed Ismail 1142374

Ahmed Osman 1148161

Kamal Aly 1141322

### **Instruction Format (16 bits)**

R-Type :

Add/SUB/AND/OR/MOV

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | | | Function | | | | Rsrc | | | Rdst | | |

RLC/RRC/NOT/NEG/INC/DEC

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | | | Unused | | | | | | | Rdst | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mnemonic** | **Function** |  | Flags affected | | | |
|  |  |  | Carry flag | Zero flag | Negative flag | Overflow flag |
| MOV Rsrc, Rdst | 0000 |  | 0 | 0 | 0 | 0 |
| ADD Rsrc, Rdst | 0001 |  | 1 | 1 | 1 | 1 |
| SUB Rsrc, Rdst | 0010 |  | 1 | 1 | 1 | 1 |
| AND Rsrc, Rdst | 0011 |  |  |  |  |  |
| OR Rsrc, Rdst | 0100 |  |  |  |  |  |
| RLC Rdst | 0101 (0x5) |  |  |  |  |  |
| RRC Rdst | 0111 (0x7) |  |  |  |  |  |
| SHL Rsrc, Imm, Rdst | 0x8 |  |  |  |  |  |
| SHR Rsrc, Imm, Rdst | 0x9 |  |  |  |  |  |
| SETC | xxxx(do not use alu) |  |  |  |  |  |
| CLRC | xxxx (do not use alu) |  |  |  |  |  |
| NOT Rdst | 1010(0xA) |  |  |  |  |  |
| NEG Rdst | 1011(0xB) |  |  |  |  |  |
| INC Rdst | 1100(0xC) |  |  |  |  |  |
| DEC Rdst | 1101(0xD) |  |  |  |  |  |
|  |  |  |  |  |  |  |

SHL/SHRl

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | | | function | | | | Rsrc | | | Rdst | | |
| Immediate | | | | | | | | | | | | | | | |

SETC/CLRC

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | | Unused | | | | | | | | | | |

I-type:

RET/RTI

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | |  | | | | | | | | | | |

LDM/LDD/STD

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | | | Unused | | | | | | | Rdst | | |
| Immediate/EA | | | | | | | | | | | | | | | |

PUSH/POP/OUT/IN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | | | Unused | | | | | | | Rdst | | |

J-TYPE:

NOP

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | | | Unused | | | | | | | | | |

JZ/JN/JMP/CALL/JC

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| OP-CODE | | | | | | Unused | | | | | | | Rdst | | |

### OP-CODE Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| **Inst.No** | **Instruction** | **OP-Code** | **Function** |
| 1 | MOV | 000000 | 0000 |
| 2 | ADD | 000000 | 0001 |
| 3 | SUB | 000000 | 0010 |
| 4 | AND | 000000 | 0011 |
| 5 | OR | 000000 | 0100 |
| 6 | NOT | 000000 | 1010 |
| 7 | NEG | 000000 | 1011 |
| 8 | INC | 000000 | 1100 |
| 9 | DEC | 000000 | 1101 |
| 10 | RLC | 000000 | 0101 |
| 11 | RRC | 000000 | 0111 |
| 12 | SHL | 000000 | 1000 |
| 13 | SHR | 000000 | 1001 |
| 14 | IN | 010011 | X |
| 15 | OUT | 010010 | X |
| 16 | SETC | 110000 | X |
| 17 | CLRC | 110001 | X |
| 18 | LDM | 010110 | X |
| 19 | LDD | 010111 | X |
| 20 | STD | 011000 | X |
| 21 | PUSH | 010000 | X |
| 22 | POP | 010001 | X |
| 23 | JMP | 100010 | X |
| 24 | JZ | 100100 | X |
| 25 | JC | 100010 | X |
| 26 | JN | 100101 | X |
| 27 | CALL | 100011 | X |
| 28 | RET | 010100 | X |
| 29 | RTI | 010101 | X |
| 30 | NOP | 010110 | X |

### Memory Design

**HARVARD Design,** as it avoids structural hazards in pipelining because there is no conflicts between instruction fetch and memory access.

### Pipeline Stages Design

**The pipeline has 5 stages:**

Fetch, Decode, Execute, Memory and Write-Back.

### Hazard Handling

* **Structural Hazards:**
  + We used the Harvard Architecture so that no 2 stages will request the same memory space simultaneously.
  + Writing in Registers will occur in the first half of clk cycle and Reading will occur in the second half.
* **Data Hazards:** A data forwarding unit that forwards data for the next instructions (FULL FORWARDING).
* Forwarding from id/if to alu for immediate value
* **Control Hazards:** Static prediction that the branch instructions will not be taken.

|  |  |
| --- | --- |
| Control signal |  |
| Pc\_select |  |
| Pc\_adder\_mux\_select |  |
| Rdest\_signal |  |
| jumpcall |  |
| destination\_select |  |
| wbsignal |  |
| wb\_signal\_select |  |
| Alu enable |  |
| DEnable |  |
| MR |  |
| MW |  |
| Read\_Address\_Select |  |
| Mem\_Write\_Address\_Select |  |
| CCR\_Select1 |  |
| CCR\_Select2 |  |
| Rdst\_signal |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |