

**Project Report**

Computer Architecture (EECS316)

**Team Members:**

|  |  |
| --- | --- |
| Name | ID |
| Ahmed Tarek Abdelaal | 1200088 |
| Omar Ahmed Desouki | 1200200 |
| Ziad Mohamed Ghanem | 1200472 |
| Osama Youssef ElHattab | 1200116 |

Table of Contents

[Instruction format: 3](#_Toc164298901)

[One Operand 3](#_Toc164298902)

[Two Operands 3](#_Toc164298903)

[Memory Operations 4](#_Toc164298904)

[Branching 4](#_Toc164298905)

[Input signals 4](#_Toc164298906)

[Signals: 5](#_Toc164298907)

# Instruction format:

Opcode of each instruction and Instruction bits details

## One Operand

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | [15: 14] | [13: 10] | [9: 7] | [6: 4] | [3: 1] | [0] |
| Mnemonic | opcode (2) | opcode (4) | Src1(3) | Src2 (3) | Dest (3) | ImmFlag |
| **NOP** | 00 | 0000 | XXX | XXX | XXX | 0 |
| **NOT Rdst** | 00 | 0001 | Rdst | XXX | Rdst | 0 |
| **NEG Rdst** | 00 | 0010 | Rdst | XXX | Rdst | 0 |
| **INC Rdst** | 00 | 0011 | Rdst | XXX | Rdst | 0 |
| **DEC Rdst** | 00 | 0100 | Rdst | XXX | Rdst | 0 |
| **OUT Rdst** | 00 | 0101 | Rdst | XXX | XXX | 0 |
| **IN Rdst** | 00 | 0110 | XXX | XXX | Rdst | 0 |

## Two Operands

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | [15: 14] | [13: 10] | [9: 7] | [6: 4] | [3: 1] | [0] |
| Mnemonic | opcode (2) | opcode (4) | Src1 (3) | Src2 (3) | Dest (3) | ImmFlag(1) |
| **MOV Rdst, Rsrc** | 01 | 0000 | XXX | Rsrc1 (alshan el LDM) | Rdst | 0 |
| **SWAP Rdst, Rsrc1** | 01 | 0001 | Rsrc1 | Rdst | XXX | 0 |
| **ADD Rdst, Rsrc1, Rsrc2** | 01 | 0010 | Rsrc1 | Rsrc2 | Rdst | 0 |
| **SUB Rdst, Rsrc1, Rsrc2** | 01 | 0011 | Rsrc1 | Rsrc2 | Rdst | 0 |
| **AND Rdst, Rsrc1, Rsrc2** | 01 | 0100 | Rsrc1 | Rsrc2 | Rdst | 0 |
| **OR Rdst, Rsrc1, Rsrc2** | 01 | 0101 | Rsrc1 | Rsrc2 | Rdst | 0 |
| **XOR Rdst, Rsrc1, Rsrc2** | 01 | 0110 | Rsrc1 | Rsrc2 | Rdst | 0 |
| **CMP Rsrc1, Rsrc2** | 01 | 0111 | Rsrc1 | Rsrc2 | XXX | 0 |
|  |  |  |  |  |  |  |
| **ADDI Rdst, Rsrc1, Imm** | 01 | 1000 | Rsrc1 | XXX (Imm (15: 0) NI) | Rdst | 1 |
| **SUBI Rdst, Rsrc1, Imm** | 01 | 1001 | Rsrc1 | XXX (Imm (15: 0) NI) | Rdst | 1 |

## Memory Operations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | [15: 14] | [13: 10] | [9: 7] | [6: 4] | [3: 1] | [0] |
| Mnemonic | opcode (2) | opcode (4) | Src1 (3) | Src2 (3) | Dest (3) | ImmFlag(1) |
| **PUSH Rdst** | 10 | 0000 | XXX | Rdst | XXX | 0 |
| **POP Rdst** | 10 | 0001 | XXX | XXX | Rdst | 0 |
| **PROTECT Rsrc** | 10 | 0010 | Rsrc | XXX | XXX | 0 |
| **FREE Rsrc** | 10 | 0011 | Rsrc | XXX | XXX | 0 |
|  |  |  |  |  |  |  |
| **LDM Rdst, Imm** | 10 | 0100 | XXX | XXX (Imm (15: 0) **NI**) | Rdst | 1 |
|  |  |  |  |  |  |  |
| **LDD Rdst, EA(Rsrc1)** | 10 | 0101 | Rsrc1 | XXX (EA (15: 0) **NI**) | Rdst | 1 |
| **STD Rdst, EA(Rsrc1)** | 10 | 0110 | Rsrc1 | Rdst (EA (15: 0) **NI**) | XXX | 1 |

## Branching

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | [15: 14] | [13: 10] | [9: 7] | [6: 1] | [0] |
| Mnemonic | opcode (2) | opcode (4) | Src1 (3) | remain bits (6) | ImmFlag |
| **JZ Rdst** | 11 | 0000 | Rdst | XXXX XX | 0 |
| **JMP Rdst** | 11 | 0001 | Rdst | XXXX XX | 0 |
| **CALL Rdst** | 11 | 0010 | Rdst | XXXX XX | 0 |
| **RET** | 11 | 0011 | XXX | XXXX XX | 0 |
| **RTI** | 11 | 0100 | XXX | XXXX XX | 0 |

## Input signals

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | [15: 14] | [13: 10] | [9: 1] | [0] |
| Mnemonic | opcode (2) | opcode (4) | remain bits (9) | ImmFlag |
| **Reset** | 11 | 0101 | XXXX XXXX X | 0 |
| **Interrupt** | 11 | 0110 | XXXX XXXX X | 0 |

# Signals: **Control Signals**

## Fetch, Decode and Excute

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | Reg read | Branch | Sign Ex | D/E Reg En | ALU Selectors | ALU src2 | Reg write |
| NOP | 0 | 0 | X | 1 | XXX | XX | 0 |
| PUSH Rdst | 1 | 0 | X | 1 | XXX | 00(src2) | 0 |
| POP Rdst | 0 | 0 | X | 1 | XXX | XX | 1 |
| PROTECT Rsrc | 1 | 0 | X | 1 | XXX | XX | 0 |
| FREE Rsrc | 1 | 0 | X | 1 | XXX | XX | 0 |
| JZ Rdst | 1 | Z (0 flag) | X | 1 | XXX | 00(src2) | 0 |
| JMP Rdst | 1 | 1 | X | 1 | XXX | 00(src2) | 0 |
| CALL Rdst | 1 | 1 | X | 1 | XXX | 00(src2) | 0 |
| RET | 0 | 1 | X | 1 | XXX | XX | 0 |
| RTI | 0 | 1 | X | 1 | XXX | XX | 0 |
| RESET | X | X | X | 1 | XXX | XX | 0 |
| INTERRUPT | 0 | 1 | X | 1 | XXX | XX | 0 |
| IN Rdst | 0 | 0 | X | 1 | XXX | XX | 1 |
| OUT Rdst | 1 | 0 | X | 1 | XXX | 00 | 1 |
| SWAP Rsrc1, Rsrc2 | 1 | X | X | 1 | XXX | XX | 0 |
|  |  |  |  |  |  |  |  |
| MOV Rdst, Rsrc | 1 | 0 | X | 1 | 011 | 00(src2) | 1 |
| LDM Rdst, Imm | 1 | 0 | 1 | 0 | 011 | 01(Imm) | 1 |
|  |  |  |  |  |  |  |  |
| INC Rdst | 1 | 0 | X | 1 | 001 | 10(1) | 1 |
| ADD Rdst, Rsrc1, Rsrc2 | 1 | 0 | X | 1 | 001 | 00(src2) | 1 |
| ADDI Rdst, Rsrc1, Imm | 1 | 0 | 0 | 0 | 001 | 01(Imm) | 1 |
| LDD Rdst, EA(Rsrc1) | 1 | 0 | 0 | 0 | 001 | 01(Imm) | 1 |
| STD Rdst, EA(Rsrc1) | 1 | 0 | 0 | 0 | 001 | 01(Imm) | 1 |
|  |  |  |  |  |  |  |  |
| DEC Rdst | 1 | 0 | X | 1 | 010 | 10(1) | 1 |
| NEG Rdst | 1 | 0 | X | 1 | 010 | XX | 1 |
| SUB Rdst, Rsrc1, Rsrc2 | 1 | 0 | X | 1 | 010 | 00(src2) | 1 |
| SUBI Rdst, Rsrc1, Imm | 1 | 0 | 0 | 0 | 010 | 01(Imm) | 1 |
| CMP Rsrc1, Rsrc2 | 1 | 0 | X | 1 | 010 | 00(src2) | 0 |
|  |  |  |  |  |  |  |  |
| NOT Rdst | 1 | 0 | X | 1 | 111 | XX | 1 |
| AND Rdst, Rsrc1, Rsrc2 | 1 | 0 | X | 1 | 100 | 00(src2) | 1 |
| OR Rdst, Rsrc1, Rsrc2 | 1 | 0 | X | 1 | 101 | 00(src2) | 1 |
| XOR Rdst, Rsrc1, Rsrc2 | 1 | 0 | X | 1 | 110 | 00(src2) | 1 |

## Execute & Memory

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | MW  enable | MR  enable | SP | Mem address | MW data | Protect | Free |
| NOP | 0 | 0 | 00 | XX | XX | 0 | 0 |
| PUSH Rdst | 1 | 0 | 01(-2) | 01 (SP) | 00 (Src2) | 0 | 0 |
| POP Rdst | 0 | 1 | 10(+2) | 01 (SP) | XX | 0 | 0 |
| PROTECT Rsrc | 1 | 0 | 00 | 00 (ALU) | XX | 1 | 0 |
| FREE Rsrc | 1 | 0 | 00 | 00 (ALU) | XX | 0 | 1 |
| JZ Rdst | 0 | 0 | 00 | XX | XX | 0 | 0 |
| JMP Rdst | 0 | 0 | 00 | XX | X | 0 | 0 |
| CALL Rdst | 1 | 0 | 01(-2) | 01 (SP) | 01 (PC+1) | 0 | 0 |
| RET | 0 | 1 | 10(+2) | XX | XX | 0 | 0 |
| RTI | 0 | 1 | 10(+2) | XX | XX | 0 | 0 |
| RESET | 0 | X | XX | 10(reset) | XX | 0 | 0 |
| INTERRUPT | 1 | 0 | 01(-2) | 01 (SP)  11 (2 inter.) | 10 (PC)  11 (CCR) | 0 | 0 |
| IN Rdst | 0 | 0 | 00 | XX | XX | 0 | 0 |
| OUT Rdst | 0 | 0 | 00 | XX | XX | 0 | 0 |
| SWAP Rsrc1, Rsrc2 | 0 | 0 | 00 | XX | XX | 0 | 0 |
|  |  |  |  |  |  |  |  |
| MOV Rdst, Rsrc | 0 | 0 | 00 | XX | XX | 0 | 0 |
| LDM Rdst, Imm | 0 | 0 | 00 | XX | XX | 0 | 0 |
|  |  |  |  |  |  |  |  |
| INC Rdst | 0 | 0 | 00 | XX | XX | 0 | 0 |
| ADD Rdst, Rsrc1, Rsrc2 | 0 | 0 | 00 | XX | XX | 0 | 0 |
| ADDI Rdst, Rsrc1, Imm | 0 | 0 | 00 | XX | XX | 0 | 0 |
| LDD Rdst, EA(Rsrc1) | 0 | 0 | 00 | 00 (ALU) | XX | 0 | 0 |
| STD Rdst, EA(Rsrc1) | 1 | 0 | 00 | 00 (ALU) | XX | 0 | 0 |
|  |  |  |  |  |  |  |  |
| DEC Rdst | 0 | 0 | 00 | XX | XX | 0 | 0 |
| NEG Rdst | 0 | 0 | 00 | XX | XX | 0 | 0 |
| SUB Rdst, Rsrc1, Rsrc2 | 0 | 0 | 00 | XX | XX | 0 | 0 |
| SUBI Rdst, Rsrc1, Imm | 0 | 0 | 00 | XX | XX | 0 | 0 |
| CMP Rsrc1, Rsrc2 | 0 | 0 | 00 | XX | XX | 0 | 0 |
|  |  |  |  |  |  |  |  |
| NOT Rdst | 0 | 0 | 00 | XX | XX | 0 | 0 |
| AND Rdst, Rsrc1, Rsrc2 | 0 | 0 | 00 | XX | XX | 0 | 0 |
| OR Rdst, Rsrc1, Rsrc2 | 0 | 0 | 00 | XX | XX | 0 | 0 |
| XOR Rdst, Rsrc1, Rsrc2 | 0 | 0 | 00 | XX | XX | 0 | 0 |

## Write Back

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instruction | Rsrc1 Data | RegW1 Enable | RegW2 Enable | RegW1 Address mux |
| NOP | XX | 0 | 0 | 0(Rdst) |
| PUSH Rdst | XX | 0 | 0 | 0(Rdst) |
| POP Rdst | 01(Mem) | 1 | 0 | 0(Rdst) |
| PROTECT Rsrc | XX | 0 | 0 | 0(Rdst) |
| FREE Rsrc | XX | 0 | 0 | 0(Rdst) |
| JZ Rdst | XX | 0 | 0 | 0(Rdst) |
| JMP Rdst | XX | 0 | 0 | 0(Rdst) |
| CALL Rdst | XX | 0 | 0 | 0(Rdst) |
| RET | XX | 0 | 0 | 0(Rdst) |
| RTI | XX | 0 | 0 | 0(Rdst) |
| RESET | XX | 0 | 0 | 0(Rdst) |
| INTERRUPT | XX | 0 | 0 | 0(Rdst) |
| IN Rdst | XX | 0 | 0 | 0(Rdst) |
| OUT Rdst | 011 | 0 | 0 | 0(Rdst) |
| SWAP Rsrc1, Rsrc2 | 10(Rsrc2) | 1 | 1 | 1(Rsrc1) |
|  |  |  |  |  |
| MOV Rdst, Rsrc | 00(ALU) | 1 | 0 | 0(Rdst) |
| LDM Rdst, Imm | 00(ALU) | 1 | 0 | 0(Rdst) |
|  |  |  |  |  |
| INC Rdst | 00(ALU) | 1 | 0 | 0(Rdst) |
| ADD Rdst, Rsrc1, Rsrc2 | 00(ALU) | 1 | 0 | 0(Rdst) |
| ADDI Rdst, Rsrc1, Imm | 00(ALU) | 1 | 0 | 0(Rdst) |
| LDD Rdst, EA(Rsrc1) | 01(Mem) | 1 | 0 | 0(Rdst) |
| STD Rdst, EA(Rsrc1) | XXX | 0 | 0 | 0(Rdst) |
|  |  |  |  |  |
| DEC Rdst | 00(ALU) | 1 | 0 | 0(Rdst) |
| NEG Rdst | 00(ALU) | 1 | 0 | 0(Rdst) |
| SUB Rdst, Rsrc1, Rsrc2 | 00(ALU) | 1 | 0 | 0(Rdst) |
| SUBI Rdst, Rsrc1, Imm | 00(ALU) | 1 | 0 | 0(Rdst) |
| CMP Rsrc1, Rsrc2 | XX | 0 | 0 | 0(Rdst) |
|  |  |  |  |  |
| NOT Rdst | 00(ALU) | 1 | 0 | 0(Rdst) |
| AND Rdst, Rsrc1, Rsrc2 | 00(ALU) | 1 | 0 | 0(Rdst) |
| OR Rdst, Rsrc1, Rsrc2 | 00(ALU) | 1 | 0 | 0(Rdst) |
| XOR Rdst, Rsrc1, Rsrc2 | 00(ALU) | 1 | 0 | 0(Rdst) |