

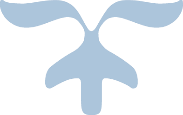
Credit Hours System

CMPN301 – Computer Architecture

Cairo University Faculty of Engineering



Phase 1 -Arch Project



|  |  |
| --- | --- |
| Mostafa Ashraf Ahmed | 1180406 |
| Nada Abdelrahman Ahmed | 1180511 |
| Sandy Samir | 4190288 |
| Sarah Osama | 1180422 |

# Types of Instructions (32bits)

Opcode (5 bits):

|  |  |
| --- | --- |
| 2 bits (Type of instruction) | 3 bits (its number in this type) |

00 🡪 R-Type

01 🡪 I-Type

10 🡪 J-Type

We have 8 registers, so we need 3 bits to identify which register in the register file. (Rs, Rt, Rd).

Each register is 32bits.

There are 2 additional registers: PC and SP.

## R-Type: (total so far: 14 out of 32)

|  |  |  |  |
| --- | --- | --- | --- |
| OpCode (5) | Rs1 (3) | Rs2 (3) | Rdst (3) |

## I-Type: (total so far: 27 out of 32)

|  |  |  |  |
| --- | --- | --- | --- |
| OpCode (5) | Rs1 (3) | Rs2 (3) | Immediate value or address (16) |

## J-Type: (total so far: 21 out 32)

|  |  |
| --- | --- |
| OpCode (5) | Target (number of jumped instructions) (16) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction |  | Control Signals | | | | | | | | | | | | | | |  | Type | OpCode |
| X | Reg  Write1 | | Reg  Write2 | InPort | OutPort | Reg/Imm | Reg  Dst | ALU Op (3bits) | JmpCond  (2 bits) | Branch | Sp/Heap | Mem Write | Mem  Read | Mem/ALU  toReg | PC/RD2 |  |  | X | X |
| NOP | 0 | | 0 | 0 | 0 | 0 | 0 | 000 | 00 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |
| HLT |  | |  |  |  |  |  | 000 |  |  |  |  |  |  |  |  |  |  |  |
| SETC | 0 | | 0 | 0 | 0 | 0 | 0 | 001 | 00 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |
| NOT Rdst | 1 | | 0 | 0 | 0 | 0 | 1 | 011 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  | R |  |
| INC Rdst | 1 | | 0 | 0 | 0 | 0 | 1 | 010 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  | R |  |
| OUT Rdst | 0 | | 0 | 0 | 1 | 0 | 0 | 000 | 00 | 0 | 0 | 0 | 0 | 0 | 0 |  |  | R |  |
| IN Rdst | 1 | | 0 | 1 | 0 | 0 | 0 | 000 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  | R |  |
| MOV Rsrc, Rdst | 1 | | 0 | 0 | 0 | 0 | 0 | 000 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  |  |  |
| SWAP Rsrc, Rdst | 1 | | 1 | 0 | 0 | 0 | 1 | 000 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  |  |  |
| ADD Rdst,  Rsrc1, Rsrc2 | 1 | | 0 | 0 | 0 | 0 | 0 | 101 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  | R |  |
| SUB Rdst,  Rsrc1, Rsrc2 | 1 | | 0 | 0 | 0 | 0 | 0 | 100 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  | R |  |
| AND Rdst,  Rsrc1, Rsrc2 | 1 | | 0 | 0 | 0 | 0 | 0 | 111 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  | R |  |
| IADD Rdst, Rsrc  ,Imm | 1 | | 0 | 0 | 0 | 1 | 0 | 110 | 00 | 0 | 0 | 0 | 0 | 1 | 0 |  |  | I |  |
| PUSH Rdst | 0 | | 0 | 0 | 0 | 0 | 0 | 000 | 00 | 0 | 1 | 1 | 0 | 0 | 0 |  |  | I |  |
| POP Rdst | 1 | | 0 | 0 | 0 | 0 | 1 | 000 | 00 | 0 | 1 | 0 | 1 | 0 | 0 |  |  | I |  |
| LDM Rdst, Imm | 1 | | 0 | 0 | 0 | 1 | 1 | 000 | 00 | 0 | 0 | 0 | 1 | 0 | 0 |  |  | I |  |
| LDD Rdst,  offset(Rsrc) | 1 | | 0 | 0 | 0 | 1 | 1 | 110 | 00 | 0 | 0 | 0 | 1 | 0 | 0 |  |  | I |  |
| STD Rsrc1,  offset(Rsrc2) | 0 | | 0 | 0 | 0 | 1 | X | 110 | 00 | 0 | 0 | 1 | 0 | X | 0 |  |  | I |  |
| JZ Imm | 0 | | 0 | 0 | 0 | 1 | X | 000 | 11 | 1 | 0 | 0 | 0 | X | 0 |  |  | J |  |
| JN Imm | 0 | | 0 | 0 | 0 | 1 | X | 000 | 10 | 1 | 0 | 0 | 0 | X | 0 |  |  | J |  |
| JC Imm | 0 | | 0 | 0 | 0 | 1 | X | 000 | 01 | 1 | 0 | 0 | 0 | X | 0 |  |  | J |  |
| JMP Imm | 0 | | 0 | 0 | 0 | 1 | X | 000 | 00 | 1 | 0 | 0 | 0 | X | 0 |  |  | J |  |
| CALL Imm | 0 | | 0 | 0 | 0 | 1 | X | 000 | 00 | 1 | 1 | 1 | 0 | X | 1 |  |  | J |  |
| RET |  | |  |  |  |  |  | 000 |  |  |  |  |  |  |  |  |  |  |  |
| INT index |  | |  |  |  |  |  | 000 |  |  |  |  |  |  |  |  |  |  |  |
| RTI |  | |  |  |  |  |  | 000 |  |  |  |  |  |  |  |  |  |  |  |
| Reset |  | |  |  |  |  |  | 000 |  |  |  |  |  |  |  |  |  |  |  |
| Interrupt |  | |  |  |  |  |  | 000 |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| ALU Operation | ALU OpCode (3Btis) |
| No operation | 000 |
| Set carry | 001 |
| Add 1 (INC) | 010 |
| NOT | 011 |
| Subtract | 100 |
| ADD 2 Reg | 101 |
| Add Imm | 110 |
| AND | 111 |