**TEAM 16**

**Instruction format**

**Instructions Types:**

* **Type 0:**

instr [31:30] = 00

Instr [29:26] = function

Instr [25:0] = x

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instruction | Instr [29] | Instr [28] | Instr [27] | Instr [26] |
| **NOP** | 0 | 0 | 0 | 0 |
| **SETC** | 0 | 0 | 0 | 1 |
| **RTI** | 0 | 0 | 1 | 0 |
| **RET** | 0 | 0 | 1 | 1 |
| **HLT** | 0 | 1 | 0 | 0 |

* **Type 1:**

Instr [31:30] = 01

Instr [29:26] = function

If INT

Instr [25:23] = Index

else

Instr [25:23] = Rdst

Instr [22:0] = x

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| instruction | Instr [29] | Instr [28] | Instr [27] | Instr [26] |
| **NOT** | 0 | 0 | 0 | 0 |
| **INC** | 0 | 0 | 0 | 1 |
| **PUSH** | 0 | 0 | 1 | 0 |
| **POP** | 0 | 0 | 1 | 1 |
| **IN** | 0 | 1 | 0 | 0 |
| **OUT** | 0 | 1 | 0 | 1 |
| **JMP** | 1 | 0 | 0 | 0 |
| **JN** | 1 | 0 | 0 | 1 |
| **JZ** | 1 | 0 | 1 | 0 |
| **Jc** | 1 | 0 | 1 | 1 |
| **CALL** | 1 | 1 | 0 | 0 |
| **INT** | 1 | 1 | 0 | 1 |

* **Type 2:**

Instr [31:30] = 10

Instr [29:26] = function

Instr [25:23] = Rdst

If LDM

Instr [22:17] = x

Instr [16:1] = imm

Instr [0] = x

Else

Instr [22:20] = Rsrc1

Instr [19:0] = x

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| instruction | Instr [29] | Instr [28] | Instr [27] | Instr [26] |
| **MOV** | 0 | 0 | 0 | 0 |
| **LDM** | 0 | 0 | 0 | 1 |

* **Type 3:**

Instr [31:30] = 11

Instr [29:26] = function

If not STD:

Instr [25:23] = Rdst

Else

Instr [25:23] = x

Instr [22:20] = Rsrc1

If AND or SUB or ADD or STD:

Instr [19:17] = Rsrc2

Else

Instr [19:17] = x

If IADD or STD or LDD:

Instr [16:1] = Imm

Else

Instr [16:1] = x

Instr [0] = x

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| instruction | Instr [29] | Instr [28] | Instr [27] | Instr [26] |
| **AND** | 0 | 0 | 0 | 0 |
| **SUB** | 0 | 0 | 0 | 1 |
| **ADD** | 0 | 0 | 1 | 0 |
| **IADD** | 0 | 1 | 1 | 1 |
| **STD** | 1 | 0 | 0 | 1 |
| **LDD** | 1 | 0 | 0 | 0 |

**Instruction diagrams**

**Type 0: instruction size [31:26]**

|  |  |  |
| --- | --- | --- |
| instr [31:30] | Instr [29:26] | Instr [25:0] |

00 function x

**Type 1: instruction size [31:23]**

|  |  |  |  |
| --- | --- | --- | --- |
| instr [31:30] | Instr [29:26] | Instr [25:23] | Instr [22:20] |

01 function Rdst x

**Type 2.1: instruction size [31:20]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| instr [31:30] | Instr [29:26] | Instr [25:23] | Instr [22:20] | Instr [19:0] |

10 function Rdst Rsrc1 x

**Type 2.1: instruction size [31:7]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| instr [31:30] | Instr [29:26] | Instr [25:23] | Instr [22:7] | Instr [16:0] |

10 function Rdst imm x

**Type 3.1: instruction size [31:17]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| instr [31:30] | Instr [29:26] | Instr [25:23] | Instr [22:20] | Instr [19:17] | Instr [16:0] |

11 function Rdst Rsrc1 Rsrc2 x

**Type 3.2: instruction size [31:20]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| instr [31:30] | Instr [29:26] | Instr [25:23] | Instr [22:20] | Instr [19:4] | Instr [3:0] |

11 function Rdst Rsrc1 imm x

**Additional notes:**

(1) Memory layout is Big Endian (most significant word in lower memory location)