**Instruction Set Architecture**

Rsrc1 ; 1st operand register

Rsrc2 ; 2nd operand register

Rdst : result register

EA ; Effective address (20 bit)

Imm ; Immediate Value (16 bit)

* Two operands:

IR0 & IR1 = ’00’.

Possible formats:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SHL, SHR, | 5-bits opcode | 3-bits src1 | 8-bits don’t care | 16-bits Imm |
| SWAP, | 5-bits opcode | 3-bits src1 | 3-bits src2 | 2-bits don’t care | |
| ADD, SUB, AND, OR, | 5-bits opcode | 3-bits src1 | 3-bits src2 | 3-bits dst | 2-bits don’t care | |
| IADD, | 5-bits opcode | 3-bits src1 | 3-bits don’t care | 3-bits dst | 2-bits don’t care | | 16-bits Imm |

From IR0 -> IR4 “opcode”:

|  |  |
| --- | --- |
| Instruction | opcode |
| ADD | 00000 |
| SUB | 00001 |
| IADD | 00010 |
| AND | 00011 |
| OR | 00100 |
| SHL | 00101 |
| SHR | 00110 |
| SWAP | 00111 |

Format of src-reg code dst-reg code:

|  |  |
| --- | --- |
| Reg | Code |
| R0 | 000 |
| R1 | 001 |
| R2 | 010 |
| R3 | 011 |
| R4 | 100 |
| R5 | 101 |
| R6 | 110 |
| R7 | 111 |

* one operand:

IR0 & IR1 = ’01’

possible formats:

|  |  |  |
| --- | --- | --- |
| NOP, | 5-bits opcode | 11-bits don’t care |
| NOT, INC, DEC, OUT, IN, | 5-bits opcode | 3-bits src1 | 8-bits don’t care |

From IR0 -> IR4 “opcode”:

|  |  |
| --- | --- |
| Instruction | opcode |
| NOP | 01000 |
| NOT | 01001 |
| INC | 01010 |
| DEC | 01011 |
| OUT | 01100 |
| IN | 01101 |
| Free slot | 01110 |
| Free slot | 01111 |

Format of src-reg code dst-reg code:

|  |  |
| --- | --- |
| Reg | Code |
| R0 | 000 |
| R1 | 001 |
| R2 | 010 |
| R3 | 011 |
| R4 | 100 |
| R5 | 101 |
| R6 | 110 |
| R7 | 111 |

* Memory operations:

IR0 & IR1 = ’10’

possible formats:

|  |  |  |  |
| --- | --- | --- | --- |
| PUSH, POP, | 5-bits opcode | 3-bits dst | 8-bits don’t care |
| LDM, | 5-bits opcode | 3-bits dst | 8-bits don’t care | 16-bits Imm |
| LDD, | 5-bits opcode | 3-bits dst | 4-bits don’t care | 20-bits EA |
| STD, | 5-bits opcode | 3-bits src1 | 4-bits don’t care | 20-bits EA |

From IR0 -> IR4 “opcode”:

|  |  |
| --- | --- |
| Instruction | opcode |
| PUSH | 10000 |
| POP | 10001 |
| LDM | 10010 |
| LDD | 10011 |
| STD | 10100 |
| Free slot | 10101 |
| Free slot | 10110 |
| Free slot | 10111 |

Format of src-reg code dst-reg code:

|  |  |
| --- | --- |
| Reg | Code |
| R0 | 000 |
| R1 | 001 |
| R2 | 010 |
| R3 | 011 |
| R4 | 100 |
| R5 | 101 |
| R6 | 110 |
| R7 | 111 |

* Branch and Change of Control Operations:

IR0 & IR1 = ’11’.

possible formats:

|  |  |  |  |
| --- | --- | --- | --- |
| RET, RTI, | 5-bits opcode | 11-bits don’t care | |
| CALL, JMP, JZ | 5-bits opcode | 3-bits dst | 8-bits don’t care |

From IR0 -> IR4 “opcode”:

|  |  |
| --- | --- |
| Instruction | opcode |
| JZ | 11000 |
| JMP | 11001 |
| CALL | 11010 |
| RET | 11011 |
| RTI | 11100 |
| Free slot | 11101 |
| Free slot | 11110 |
| Free slot | 11111 |

Format of src-reg code dst-reg code:

|  |  |
| --- | --- |
| Reg | Code |
| R0 | 000 |
| R1 | 001 |
| R2 | 010 |
| R3 | 011 |
| R4 | 100 |
| R5 | 101 |
| R6 | 110 |
| R7 | 111 |