**Bora Ersoy 280201024**

I might have made some typos in the instructions but it should be true in my actual dat files

**ISA**

|  |  |
| --- | --- |
| opcode | instruction |
| **024** | **R-type** |
| **025** | **Lw** |
| **026** | **sw** |
| **027** | **jump** |
| **028** | **Beq** |
| **029** | **Bne** |
| **030** | **addi** |
| **031** | **nor** |

**Instructions ordered by execution order**

R-Type Instruction (e.g., add):

Opcode: 24

Functionfield = 32

Binary Instruction: 00011000 1110 1100 1110 000000 100000

Hex Representation: 00ECE016

Detail:

Type: R-Type

Opcode: 000000

Rs (Source Register): $t7 (register 14)

Rt (Target Register): $t4 (register 12)

Rd (Destination Register): $t7 (register 14)

Shamt: 0

Funct: 100110 (corresponding to ADD operation)

8 + 10 = 18 register 14 is 18 now

Opcode: 25

Binary Instruction: 00011001 1100 1101 0000000000001110

Hex Representation: 19DC000E

Detail:

Operation: Load Word

Registers:

Rt (Target Register): $t5 (register 13)

Rs (Base Register): $t4 (register 12)

Offset: 14

Data Flow: $t5 = Memory[$t4 + 14]

SW (Store Word) Instruction:

memory(10 + 14) = 292 register 13 = 292

Opcode: 26

Binary Instruction: 00011010 1110 1101 0000000000000010

Hex Representation: 1AED0002

Detail:

Operation: Store Word

Registers:

Rt (Source Register): $t5 (register 13)

Rs (Base Register): $t6 (register 14)

Offset: 0

Data Flow: Memory[18 + 2] = $t5 = 37

register 13 is mem[20] = 292

ADDI (Add Immediate):

Opcode: 30

Binary Instruction: 00011110 1110 1101 0000 0000 0010 0000

Hex Representation: 1EED0020

Detail:

Operation: Add Immediate

Registers:

Rt (Destination Register): $t5 (register 13)

Rs (Source Register): $t6 (register 14)

Immediate Value: 32

Data Flow: reg13 = 18 + 32 = 50

R-Type Instruction

Opcode: 24

funcfield = 31

Binary Instruction: 00011000 1110 1100 1110 000000 011111

Hex Representation: 18ECE01F

Detail:

Type: R-Type

Opcode: 000000

Rs (Source Register): $t7 (register 14) nors s and t 1010000

Rt (Target Register): $t4 (register 12) 0001010 = 111...0100101

Rd (Destination Register): $t7 (register 14)

Shamt: 0

Funct: 011111 (corresponding to nor operation)

reg 14 = 111...0100101

Jump (e.g., j):

Opcode: 27

Binary Instruction: 00011011 0000 0000 0000 0000 0001 1000

Hex Representation: 1B000030

Detail:

Operation: Jump

Target Address: 24

Data Flow: PC = 24

BEQ (Branch Equal):

Opcode: 28

Binary Instruction: 00011100 1101 1110 0000 0000 0000 1111

Hex Representation: 1CDE000F

Detail:

Operation: Branch if Equal

Registers:

Rs (Source Register): $t6 (register 14)

Rt (Target Register): $t5 (register 13)

Offset: 15

Data Flow: if $t6 == $t5 then PC = PC + (15)

BNE (Branch Not Equal):

Opcode: 29

Binary Instruction: 00011101 1101 1110 0000 0000 0000 1111

Hex Representation: 1DDE000F

Detail:

Operation: Branch if Not Equal

Registers:

Rs (Source Register): $t6 (register 14)

Rt (Target Register): $t5 (register 13)

Offset: 15

Data Flow: if $t6 != $t5 then PC = PC + (15)