

Team members:

- Mohammad Hosseini (4003613021) محمد حسینی
- Arshia Shafiei (4003623019) ارشیا شفیعی
- Behnood Obodiyat (4003623027) بهنود عبودیات

<https://github.com/1mimhe/16bit-CPU>

Instruction format(16-bit):

Every Instruction is 16-bit so they are basically one format but the bits could have different meanings.

R-type:

Opcode(4-bit)	Rs(4-bit)	Rt(4-bit)	Rd(4-bit)
---------------	-----------	-----------	-----------

I-type:

Opcode(4-bit)	Rs(4-bit)	Rt(4-bit)	Imm/Address/Offset (4-bit)
---------------	-----------	-----------	----------------------------

Instructions and Opcodes:

Instruction	Op Code	Format
add	0000	R
sub	0001	R
mul	0010	R
div	0011	R
ori	0100	I
nor	0101	R
nand	0110	R
sw	0111	I
lw	1000	I
blt	1001	I

Control Unit Truth Table:

	R-format	ori	sw	lw	blt
RegDst	1	0	X	0	X
Alusrc	0	1	1	1	0
MemToReg	0	0	X	1	X
RegWrite	1	1	0	1	0
MemWrite	0	0	1	0	0
MemRead	0	0	0	1	0
Branch	0	0	0	0	1
ExtOp	X	0	1	1	1
Aluctr	Op[2:0]	Op[2:0]	000	Op[2:0]	Op[2:0]

add, sub, mul, div, nor and nand are the same but with different Opcodes.

Control Unit test results:

```

List - Default
ps- /ControlUnit_tb/Opcode- /ControlUnit_tb/MemWrite-
delta- /ControlUnit_tb/RegDst- /ControlUnit_tb/Branch-
/ControlUnit_tb/ALUSrc- /ControlUnit_tb/ExtOp-
/ControlUnit_tb/MemtoReg- /ControlUnit_tb/ALUctr-
/ControlUnit_tb/RegWrite-

0 +0 0000 StX StX StX StX StX StX StX xxx
0 +2 0000 St1 St0 St0 St1 St0 St0 St0 000
100 +0 0001 St1 St0 St0 St1 St0 St0 St0 000
100 +2 0001 St1 St0 St0 St1 St0 St0 St0 001
200 +0 0010 St1 St0 St0 St1 St0 St0 St0 001
200 +2 0010 St1 St0 St0 St1 St0 St0 St0 010
300 +0 0011 St1 St0 St0 St1 St0 St0 St0 010
300 +2 0011 St1 St0 St0 St1 St0 St0 St0 011
400 +0 0101 St1 St0 St0 St1 St0 St0 St0 011
400 +2 0101 St1 St0 St0 St1 St0 St0 St0 101
500 +0 0110 St1 St0 St0 St1 St0 St0 St0 101
500 +2 0110 St1 St0 St0 St1 St0 St0 St0 110
600 +0 0100 St1 St0 St0 St1 St0 St0 St0 110
600 +2 0100 St0 St1 St0 St1 St0 St0 St0 100
700 +0 0111 St0 St1 St0 St1 St0 St0 St0 100
700 +2 0111 St0 St1 St0 St0 St1 St0 St1 000
800 +0 1000 St0 St1 St0 St0 St1 St0 St1 000
800 +2 1000 St0 St1 St1 St1 St0 St0 St1 000
900 +0 1001 St0 St1 St1 St1 St0 St0 St1 000
900 +2 1001 St0 St0 St0 St0 St0 St1 St0 001

```

ALU test results:

Adder and subtractor tests:

Wave - Default						
	Msgs					
+ /ALU_tb/A	10	12	35	12	10	
+ /ALU_tb/B	10	35	12	35	10	
+ /ALU_tb/Opcode	001	000	001			
+ /ALU_tb/Result	0	47	23	-23	0	
+ /ALU_tb/ZERO	St1					

Multiplier and Subtractor tests:

The screenshot shows the Waveform viewer with the ALU register selected. The waveform displays the ALU register value over time, showing a sequence of values: 0, 100, 69, 011, and 26. The ALU register is labeled as 'ALU' in the left pane.

OR, NOR and NAND tests:

Wave - Default

Signal	Value
/ALU_tb/A	0000000000...
/ALU_tb/B	0000000000...
/ALU_tb/OpCode	101
/ALU_tb/Result	1111111111...
/ALU_tb/ZERO	S0

Msgs

Time	Signal	Value
00000000000001100	/ALU_tb/Result	111111111111110010
0000000000000101	/ALU_tb/Result	1111111111111101
0000000000000100	/ALU_tb/Result	1111111111111100
0000000000000011	/ALU_tb/Result	1111111111111101