CSE 331/503
Computer Organization
Final Project –
MiniMIPS Design Report

Samet NALBANT

1801042614

Instructions

Instr	Opcode	Func
AND	0000	000
ADD	0000	001
SUB	0000	010
XOR	0000	011
NOR	0000	100
OR	0000	101
ADDI	0001	XXX
ANDI	0010	XXX
ORI	0011	XXX
NORI	0100	XXX
BEQ	0101	XXX
BNE	0110	XXX
SLTI	0111	XXX
LW	1000	XXX
SW	1001	XXX

Truth Table for Alu Control Bits

INSTR	OPCODE	FUNC	ALU OPERATION	ALU OP CODE	ALU CONTROL
ADD	0000	000	ADD	011	000
AND	0000	001	AND	011	110
SUB	0000	010	SUB	011	010
XOR	0000	011	XOR	011	001
NOR	0000	100	NOR	011	101
OR	0000	101	OR	011	111
ADDI	0001	XXX	ADD	000	000
ANDI	0010	XXX	AND	110	110
ORI	0011	XXX	OR	111	111
NORI	0100	XXX	NOR	101	101
BEQ	0101	XXX	SUB	010	010
BNE	0110	XXX	SUB	010	010
SLTI	0111	XXX	SLT	100	100
LW	1000	XXX	ADD	000	000
SW	1001	XXX	ADD	000	000

MODULES

Alu32 Module

Alu32 module taken from the previous homework and used without any changes.

Register Module

Register module takes read_reg1, read_reg2, write_reg, write_data, reg_write, and clocks as an input and produce read_data1 and read_data2 outputs.

Because of the meaningless reasong there could be file I/O error. To resolve it path must be added!

Register zero is prohibited to write.

Instruction Module

Instruction module takes adress as an input and produce instruction from related adress in “instructions.mem” file.

Because of the meaningless reasong there could be file I/O error. To resolve it path must be added!

Data Memory Module

Data memory module takes adress, write,data, mem_write,mem_read and clock as an input and produce read_data input.

Because of the meaningless reasong there could be file I/O error. To resolve it path must be added!

Sign Extender Module

Sign extender module takes 6 bit immediate input and extend it to 32 bit according to most significant bit.

Alu Control Module

Alu control module takes function and alu operataion code. According to inputs it produce proper outputs which is based on the truth table to perform alu operations.

Control Unit Module

Control unit module takes op code as an input and it produces proper outputs which is based on truth table, according to instruction type.

Mini MIPS Module

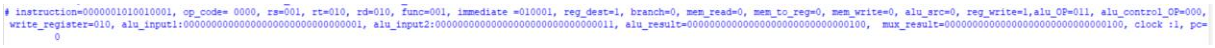
Mini MIPS module takes clock and program counter as an input. It runs the instruction module to get proper instruction. After that, It runs the control unit module with the coming instruction. Using the outputs of the control unit, register module runs. Immediate field of instruction sends to sign extender and the result puts into multiplexer with readed data from register. Alu control module runs and alu32 module runs with the op code produced from control module. After the proper operation realized, data memory module is starts and the if the writing result to register is necesseary, again register module runs.

Missing Parts of Project

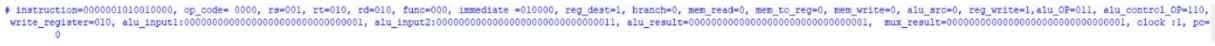
- BEQ
- BNE
- This instructions are not implemented. Necessary outputs are producing but they are not added to system.

R TYPE INSTRUCTIONS TEST RESULTS

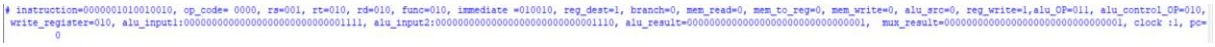
1- ADD Instruction: 0000 001 010 010 001



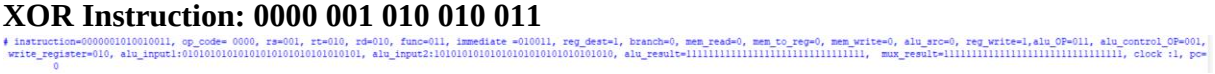
2- AND Instruction: 0000 001 010 010 000



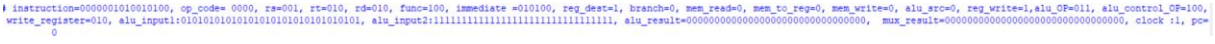
3- SUB Instruction: 0000 001 010 010 010



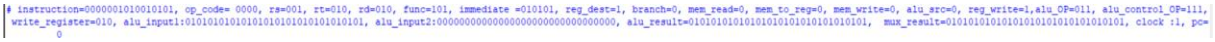
4- XOR Instruction: 0000 001 010 010 011



5- NOR Instruction: 0000 001 010 010 100

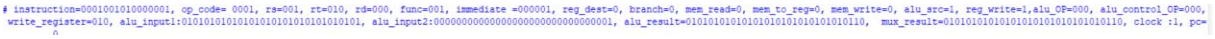


6- OR Instruction: 0000 001 010 010 101

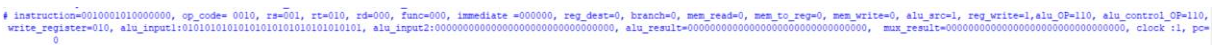


I TYPE INSTRUCTIONS TEST RESULTS

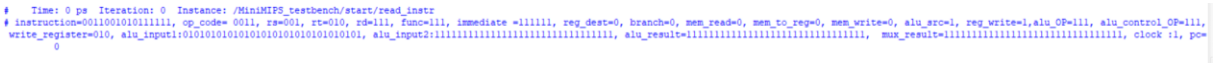
1- ADDI Instruction: 0001 001 010 000001



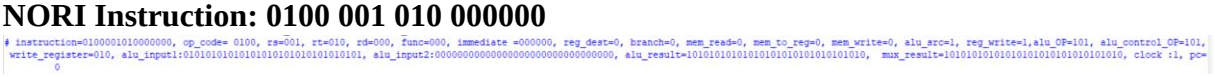
2- ANDI Instruction: 0010 001 010 000000



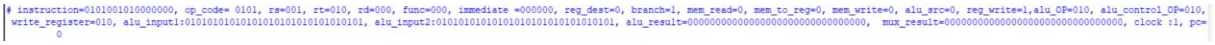
3- ORI Instruction: 0011 001 010 111111



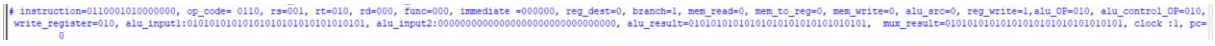
4- NORI Instruction: 0100 001 010 000000



5- BEQ Instruction: 0101 001 010 000000

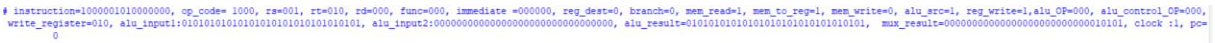


6- BNE Instruction: 0110 001 010 000000



7- SLTI Instruction: 0111010111001110

8- LW Instruction: 1000 001 010 000000



9- SW Instruction: 1001 001 010 000000

INSTRUCTION SET

```
0000001010011000
0010001110011110
0000001010010100
0000001010110011
0011001110110010
0000001110010101
0001001010000001
0000001010010000
0000001110011001
0100001111000000
0001001110001011
0011001010111111
0000001010010010
0000001110000011
1000001010000000
0000001110011010
0110110101000111
1000011010000011
0000001010010001
0110001010000000
0010001010000000
1001001000001111
0111010111001110
0100001010001010
0101001010000000
0000001000000100
0000001110110101
0101101000101000
1001001000000101
0111010111001110
```

Instruction set is created by using all instructions by twice.

RESULTS

```
# instruction=00000001010011000, op_code= 0000, rs=001, rt=010, rd=011, func=000, immediate =011000, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=110,
 write_register=011, alu_input1:01010101010101010101010101010101, alu_input2:01010101010101010101010101010101, alu_result=01010101010101010101010101010101,  mux_result=01010101010101010101010101010101, clock :1, pc=
        0
# instruction=00100001110011110, op_code= 0010, rs=001, rt=110, rd=011, func=110, immediate =011110, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=110, alu_control_OP=110,
 write_register=110, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000011110, alu_result=00000000000000000000000000010100,  mux_result=00000000000000000000000000010100, clock :0, pc=
        1
# instruction=00000010100010100, op_code= 0000, rs=001, rt=010, rd=010, func=100, immediate =010100, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=100,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:01010101010101010101010101010101, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :1, pc=
        2
# instruction=00000010101110011, op_code= 0000, rs=001, rt=010, rd=110, func=011, immediate =110011, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=001,
 write_register=110, alu_input1:01010101010101010101010101010101, alu_input2:01010101010101010101010101010101, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :0, pc=
        3
# instruction=00110011101100010, op_code= 0011, rs=001, rt=110, rd=110, func=010, immediate =110010, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=111, alu_control_OP=111,
 write_register=110, alu_input1:01010101010101010101010101010101, alu_input2:11111111111111111111111111110010, alu_result=11111111111111111111111111110111,  mux_result=11111111111111111111111111110111, clock :1, pc=
        4
# instruction=00000011100101011, op_code= 0000, rs=001, rt=110, rd=010, func=101, immediate =010101, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=111,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000010100, alu_result=01010101010101010101010101010101,  mux_result=01010101010101010101010101010101, clock :0, pc=
        5
# instruction=00010010100000001, op_code= 0001, rs=001, rt=010, rd=000, func=001, immediate =000001, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=000, alu_control_OP=000,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000000001, alu_result=01010101010101010101010101010110,  mux_result=01010101010101010101010101010110, clock :1, pc=
        6
# instruction=00000010100010000, op_code= 0000, rs=001, rt=010, rd=010, func=000, immediate =010000, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=110,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:01010101010101010101010101010101, alu_result=01010101010101010101010101010101,  mux_result=01010101010101010101010101010101, clock :0, pc=
        7
# instruction=00000011110011001, op_code= 0000, rs=001, rt=110, rd=011, func=001, immediate =011001, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=000,
 write_register=011, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000010100, alu_result=01010101010101010101010101101001,  mux_result=01010101010101010101010101101001, clock :1, pc=
        8
# instruction=01000001111000000, op_code= 0100, rs=001, rt=111, rd=000, func=000, immediate =000000, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=101, alu_control_OP=101,
 write_register=111, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000000000, alu_result=10101010101010101010101010101010,  mux_result=10101010101010101010101010101010, clock :0, pc=
        9
# instruction=00010011100010111, op_code= 0001, rs=001, rt=110, rd=001, func=011, immediate =001011, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=000, alu_control_OP=000,
 write_register=110, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000001011, alu_result=01010101010101010101010101100000,  mux_result=01010101010101010101010101100000, clock :1, pc=
       10
```

```
# instruction=00110010101111111, op_code= 0011, rs=001, rt=010, rd=111, func=111, immediate =111111, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=111, alu_control_OP=111,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:11111111111111111111111111111111, alu_result=11111111111111111111111111111111,  mux_result=11111111111111111111111111111111, clock :0, pc=
       11
# instruction=00000010100100010, op_code= 0000, rs=001, rt=010, rd=010, func=010, immediate =010010, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=010,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:01010101010101010101010101010101, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :1, pc=
       12
# instruction=00000001110000011, op_code= 0000, rs=001, rt=110, rd=000, func=011, immediate =000011, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=001,
 write_register=000, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000010100, alu_result=01010101010101010101010101000001,  mux_result=01010101010101010101010101000001, clock :0, pc=
       13
# instruction=10000001010000000, op_code= 1000, rs=001, rt=010, rd=000, func=000, immediate =000000, reg_dest=0, branch=0, mem_read=1, mem_to_reg=1, mem_write=0, alu_src=1, reg_write=1,alu_OP=000, alu_control_OP=000,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000000000, alu_result=01010101010101010101010101010101,  mux_result=00000000000000000000000000010101, clock :1, pc=
       14
# instruction=00000001110011010, op_code= 0000, rs=001, rt=110, rd=011, func=010, immediate =011010, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=010,
 write_register=011, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000010100, alu_result=01010101010101010101010101000001,  mux_result=01010101010101010101010101000001, clock :0, pc=
       15
# instruction=01101101010000111, op_code= 0110, rs=110, rt=101, rd=000, func=111, immediate =000111, reg_dest=0, branch=1, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=010, alu_control_OP=010,
 write_register=101, alu_input1:00000000000000000000000000010100, alu_input2:00000000000000000000000000010010, alu_result=00000000000000000000000000000010,  mux_result=00000000000000000000000000000010, clock :1, pc=
       16
# instruction=10000110100000011, op_code= 1000, rs=011, rt=010, rd=000, func=011, immediate =000011, reg_dest=0, branch=0, mem_read=1, mem_to_reg=1, mem_write=0, alu_src=1, reg_write=1,alu_OP=000, alu_control_OP=000,
 write_register=010, alu_input1:01010101010101010101010101101001, alu_input2:00000000000000000000000000000011, alu_result=01010101010101010101010101101100,  mux_result=00000000000000000000000000001100, clock :0, pc=
       17
# instruction=00000010100100001, op_code= 0000, rs=001, rt=010, rd=010, func=001, immediate =010001, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=000,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:01010101010101010101010101010101, alu_result=10101010101010101010101010101010,  mux_result=10101010101010101010101010101010, clock :1, pc=
       18
# instruction=01100010100000000, op_code= 0110, rs=001, rt=010, rd=000, func=000, immediate =000000, reg_dest=0, branch=1, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=010, alu_control_OP=010,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:01010101010101010101010101010101, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :0, pc=
       19
# instruction=00100010100000000, op_code= 0010, rs=001, rt=010, rd=000, func=000, immediate =000000, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=110, alu_control_OP=110,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000000000, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :1, pc=
       20
# instruction=10010011000001111, op_code= 1001, rs=001, rt=000, rd=001, func=111, immediate =001111, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=1, alu_src=1, reg_write=0,alu_OP=000, alu_control_OP=000,
 write_register=000, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000001111, alu_result=01010101010101010101010101100100,  mux_result=01010101010101010101010101100100, clock :0, pc=
       21
```

```
# instruction=01110101011001110, op_code= 0111, rs=010, rt=111, rd=001, func=110, immediate =001110, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=011, alu_control_OP=100,
 write_register=111, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000001110, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :1, pc=
       22
# instruction=01000010100010010, op_code= 0100, rs=001, rt=010, rd=001, func=010, immediate =001010, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=101, alu_control_OP=101,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000001010, alu_result=10101010101010101010101010100000,  mux_result=10101010101010101010101010100000, clock :0, pc=
       23
# instruction=01010010100000000, op_code= 0101, rs=001, rt=010, rd=000, func=000, immediate =000000, reg_dest=0, branch=1, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=010, alu_control_OP=010,
 write_register=010, alu_input1:01010101010101010101010101010101, alu_input2:01010101010101010101010101010101, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :1, pc=
       24
# instruction=00000010000000100, op_code= 0000, rs=001, rt=000, rd=000, func=100, immediate =000100, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=100,
 write_register=000, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000000000, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :0, pc=
       25
# instruction=00000011110110101, op_code= 0000, rs=001, rt=110, rd=110, func=101, immediate =110101, reg_dest=1, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=011, alu_control_OP=111,
 write_register=110, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000010100, alu_result=01010101010101010101010101010101,  mux_result=01010101010101010101010101010101, clock :1, pc=
       26
# instruction=01011010000101000, op_code= 0101, rs=101, rt=000, rd=101, func=000, immediate =101000, reg_dest=0, branch=1, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=0, reg_write=1,alu_OP=010, alu_control_OP=010,
 write_register=000, alu_input1:00000000000000000000000000010010, alu_input2:00000000000000000000000000000000, alu_result=00000000000000000000000000010010,  mux_result=00000000000000000000000000010010, clock :0, pc=
       27
# instruction=10010010000000101, op_code= 1001, rs=001, rt=000, rd=000, func=101, immediate =000101, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=1, alu_src=1, reg_write=0,alu_OP=000, alu_control_OP=000,
 write_register=000, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000000101, alu_result=01010101010101010101010101011010,  mux_result=01010101010101010101010101011010, clock :1, pc=
       28
# instruction=01110101011001110, op_code= 0111, rs=010, rt=111, rd=001, func=110, immediate =001110, reg_dest=0, branch=0, mem_read=0, mem_to_reg=0, mem_write=0, alu_src=1, reg_write=1,alu_OP=011, alu_control_OP=100,
 write_register=111, alu_input1:01010101010101010101010101010101, alu_input2:00000000000000000000000000001110, alu_result=00000000000000000000000000000000,  mux_result=00000000000000000000000000000000, clock :1, pc=
       29
```