CSE331-HW4 Report

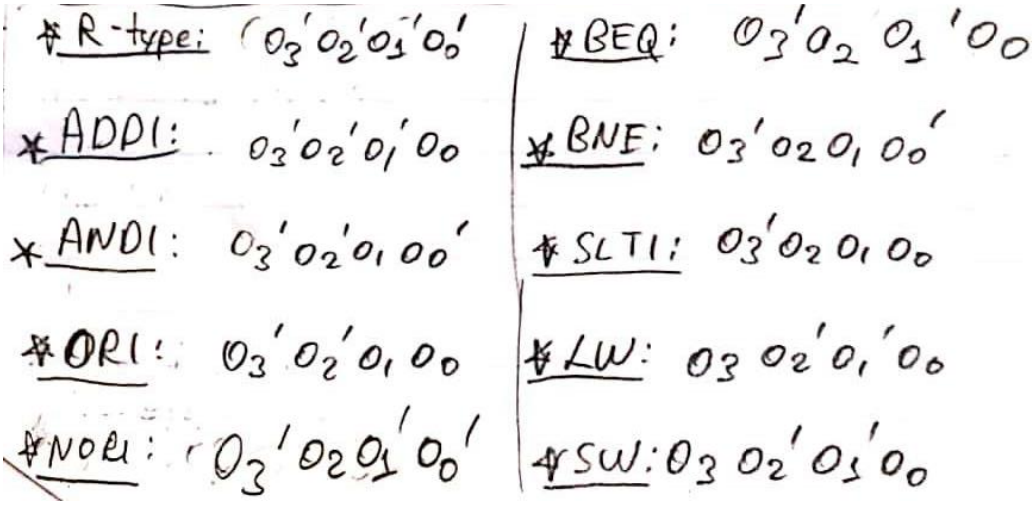
1-) Explanations

mainController Explanations

My first step of building this processor was determining the main controller states. The main controller is a combinational circuit that produces necessary signals according to operation opcodes for all the other parts of the processor.

Opcode	Operations	MemRead	MemReg	MemWrite	RegWrite	RegDst	Branch	Bne	ALUSrc	ALUOp	
0000	→ AND	0	0	0	1	1	0	0	0	000	✓
0000	→ ADD	0	0	0	1	1	0	0	0	000	✓
0000	→ SUB	0	0	0	1	1	0	0	0	000	✓
0000	→ XOR	0	0	0	1	1	0	0	0	000	✓
0000	→ NOR	0	0	0	1	1	0	0	0	000	✓
0000	→ OR	0	0	0	1	1	0	0	0	000	✓
0001	→ ADDI	0	0	0	1	0	0	0	1	001	✓
0010	→ ANDI	0	0	0	1	0	0	0	1	010	✓
0011	→ ORI	0	0	0	1	0	0	0	1	011	✓
0100	→ NORI	0	0	0	1	0	0	0	1	100	✓
0101	→ BEQ	0	0	0	0	0	1	0	0	101	✓
0110	→ BNE	0	0	0	0	x	0	1	0	101	✓
0111	→ SLTI	0	0	0	1	0	0	0	1	110	✓
1000	→ LW	1	1	0	1	0	0	0	1	001	✓
1001	→ SW	0	0	1	0	0	0	0	1	001	✓

I have determined all the necessary signals for all the operations. Afterwards, I have calculated the boolean expressions for operations according to their opcodes.



Afterwards, I have calculated signals' and ALUOP's boolean expressions using operation expressions.

* RegDst : R-type
* Branch : BEQ
* Bne : BNE
* MemRead : LW
* MemReg : LW
* Memwrite : SW
* RegWrite : R-type + ADDI + ANDI + ORI + NORI + SLTI + LW
* ALUSRC : ADDI + ANDI + ORI + NORI + SLTI + LW + SW
* ALUOP[2] : NORI + BEQ + BNE + SLTI
* ALUOP[1] : ANDI + ORI + SLTI
* ALUOP[0] : ADDI + ORI + BEQ + BNE + LW + SW

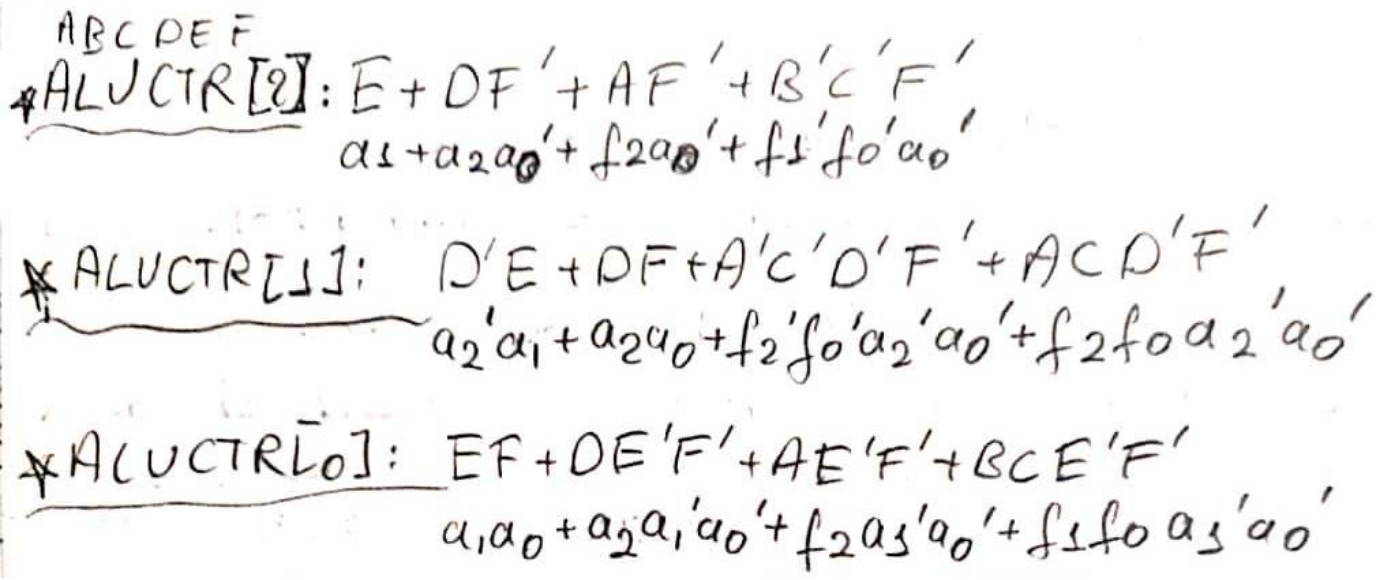
Operations	Fundion	ALUOP	Action	ALUCTR
→ AND	000	000	and	110
→ ADD	001	000	add	000
→ SUB	010	000	sub	010
→ XOR	011	000	xor	001
→ NOR	100	000	nor	101
→ OR	101	000	or	111
→ ADDI	xxx	001	add	000
→ ANDI	xxx	010	and	110
→ ORI	xxx	011	or	111
→ NORI	xxx	100	nor	101
→ BEQ	xxx	101	sub	010
→ BNE	xxx	101	sub	010
→ SLTI	xxx	110	slt	100
→ LW	xxx	001	add	000
→ SW	xxx	001	add	000

aluController Explanations

My second step was building the aluController. aluContoller is a combinational circuit that produces necessary alu control signals for alu to do the right calculations for the given operation.

I have determined all the right operations the ALU needs to do for each operation.

Afterwards, I have calculated the boolean expressions for ALUCTR according to the function codes and the opcodes.



instructionMemory Explanations

instructionMemory.v is used to fetch the needed instruction according to the current PC. At the start, it reads all the instructions from a file called instruction.mem. It stores them inside. In each positive clock, it retrieves the instruction with the PC number.

NOTE: I have chosen to read from the file here instead of the testbench and sending the instructions as an input at first because when I tried that, the compiler gave an error saying that I cannot put a 2d array as an input without changing some settings. I did not want to change anything without knowing therefore I have done all reading and writing inside the modules.

registers Explanation

registers.v is used to fetch data from the given two register addresses. At the start, it reads all the register data from a file called registers.mem. It does fetching whenever a new address is given. It also does writing to the given register address when the regWrite signal is 1. It does writing when the clock turns from 1 to 0. The writing happens to a file called registersOutput.mem.

memory Explanation

memory.v is used to fetch data from the given memory address. At the start, it reads all memory data from a file called memory.mem. It does fetching when memRead signal is 1. It also does wiritng to the given memory address when the memWrite signal is 1. It does writing when the clock turns from 1 to 0. The writing happens to a file calles memoryOutput.mem

miniMips Explanation

miniMips.v is a module capable of doing and, add, sub, xor, nor, or, addi, andi, ori, nori, beq, bne, slti, lw and sw operations. It includes a PC counter mechanism, instructionMemory, registers, mainController, aluController, signExtend module, alu, branching mechanism,

memory, muxes inside. Some of these are already explained above. The alu is the same alu I have used in HW3. All it's information and testbenches are included in HW3.

PC counter mechanism: In each switch of clock from 0 to 1, PC is incremented by 1. However, if there is a branch then the PC is incremented by 1+signExtendedImmediate.

Branching mechanism: The equality of rs and rt is checked. It is anded with branch signal. Also the reverse of the equality result is anded with bne signal. At the end these two are ored. If the result is 1 then it means there will be a branch.

General structure:
-Firstly the instruction is fetched. It is divided into parts.
-mainController and aluController produces signals.
-Fetching from register module is done.
-Alu operations are done with the fetched data and produced signals.
-Rs and rt equality is checked.
-PC for next operation is calculated.
-If needed, necessary read, write from/to memory is done.
-If needed, needed write to register is done.
-Repeat.

IMPORTANT NOTE: For the code to work, I have written "C:/altera/13.1/workspace/hw1/" to file directory whenever I read from a file/write to file. You might need to change these directories. You need to check
-instructionMemory.v
-registers.v
-memory.v
modules for that.

2-) Tests

mainController.v test

```
VSIM 5> step -current
# time =  0, opcode =0000, memRead=0, memReg=0, memWrite=0, regWrite=1, regDst=1, branch=0, bneSig=0, aluSrc=0, aluOp=000
# time = 20, opcode =0001, memRead=0, memReg=0, memWrite=0, regWrite=1, regDst=0, branch=0, bneSig=0, aluSrc=1, aluOp=001
# time = 40, opcode =0010, memRead=0, memReg=0, memWrite=0, regWrite=1, regDst=0, branch=0, bneSig=0, aluSrc=1, aluOp=010
# time = 60, opcode =0011, memRead=0, memReg=0, memWrite=0, regWrite=1, regDst=0, branch=0, bneSig=0, aluSrc=1, aluOp=011
# time = 80, opcode =0100, memRead=0, memReg=0, memWrite=0, regWrite=1, regDst=0, branch=0, bneSig=0, aluSrc=1, aluOp=100
# time = 100, opcode =0101, memRead=0, memReg=0, memWrite=0, regWrite=0, regDst=0, branch=1, bneSig=0, aluSrc=0, aluOp=101
# time = 120, opcode =0110, memRead=0, memReg=0, memWrite=0, regWrite=0, regDst=0, branch=0, bneSig=1, aluSrc=0, aluOp=101
# time = 140, opcode =0111, memRead=0, memReg=0, memWrite=0, regWrite=1, regDst=0, branch=0, bneSig=0, aluSrc=1, aluOp=110
# time = 160, opcode =1000, memRead=1, memReg=1, memWrite=0, regWrite=1, regDst=0, branch=0, bneSig=0, aluSrc=1, aluOp=001
# time = 180, opcode =1001, memRead=0, memReg=0, memWrite=1, regWrite=0, regDst=0, branch=0, bneSig=0, aluSrc=1, aluOp=001

VSIM 6>
```

aluController.v test

```
# time = 0 func=000, aluOp=000, aluContol=110
# time =20 func=001, aluOp=000, aluContol=000
# time =40 func=010, aluOp=000, aluContol=010
# time =60 func=011, aluOp=000, aluContol=001
# time =80 func=100, aluOp=000, aluContol=101
# time =100 func=101, aluOp=000, aluContol=111
# time =120 func=000, aluOp=001, aluContol=000
# time =140 func=000, aluOp=010, aluContol=110
# time =160 func=000, aluOp=011, aluContol=111
# time =180 func=000, aluOp=100, aluContol=101
# time =200 func=000, aluOp=101, aluContol=010
# time =220 func=000, aluOp=110, aluContol=100
# time =240 func=000, aluOp=001, aluContol=000
```

instructionMemory.v test

```
VSIM 5> step -current
# time= 0, PC=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, instruction=0000010011001000, expected=xxxxxxxxxxxxxxxx
# time=10, PC=00000000000000000000000000000000, instruction=0000010011001000, expected=0000010011001000
# time=30, PC=00000000000000000000000000000001, instruction=0000001011010000, expected=0000001011010000
# time=50, PC=00000000000000000000000000000010, instruction=0000010011011001, expected=0000010011011001
# time=70, PC=00000000000000000000000000000011, instruction=0000011101100001, expected=0000011101100001
# time=90, PC=00000000000000000000000000000100, instruction=0000100010101010, expected=0000100010101010
# time=110, PC=00000000000000000000000000000101, instruction=0000101100110010, expected=0000101100110010
# time=130, PC=00000000000000000000000000000110, instruction=0000101100001011, expected=0000101100001011
# time=150, PC=00000000000000000000000000000111, instruction=0000001101010011, expected=0000001101010011
```

registers.v test

```
# time= 0, clock=0, writeData=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, writeReg=xxx, expected=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
#  readReg1=xxx, readData1=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, readReg2=xxx, readData2=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#
# time=10, clock=1, writeData=00000000000000000000000000000001, writeReg=001, expected=00000000000000000000000000000001,
#  readReg1=000, readData1=00000000000000000000000000000000, readReg2=001, readData2=00000000000000000000000000000001
#
# time=20, clock=0, writeData=00000000000000000000000000000001, writeReg=001, expected=00000000000000000000000000000001,
#  readReg1=000, readData1=00000000000000000000000000000000, readReg2=001, readData2=00000000000000000000000000000001
#
# time=30, clock=1, writeData=00000000000000000000000000000001, writeReg=000, expected=00000000000000000000000000000000,
#  readReg1=000, readData1=00000000000000000000000000000000, readReg2=001, readData2=00000000000000000000000000000001
#
# time=40, clock=0, writeData=00000000000000000000000000000001, writeReg=000, expected=00000000000000000000000000000000,
#  readReg1=000, readData1=00000000000000000000000000000000, readReg2=001, readData2=00000000000000000000000000000001
#
# time=50, clock=1, writeData=00000000000000000000000000000010, writeReg=010, expected=00000000000000000000000000000010,
#  readReg1=001, readData1=00000000000000000000000000000001, readReg2=010, readData2=00000000000000000000000000000010
#
# time=60, clock=0, writeData=00000000000000000000000000000010, writeReg=010, expected=00000000000000000000000000000010,
#  readReg1=001, readData1=00000000000000000000000000000001, readReg2=010, readData2=00000000000000000000000000000010
#
# time=70, clock=1, writeData=00000000000000000000000000000011, writeReg=011, expected=00000000000000000000000000000011,
#  readReg1=010, readData1=00000000000000000000000000000010, readReg2=011, readData2=00000000000000000000000000000011
#
# time=80, clock=0, writeData=00000000000000000000000000000011, writeReg=011, expected=00000000000000000000000000000011,
#  readReg1=010, readData1=00000000000000000000000000000010, readReg2=011, readData2=00000000000000000000000000000011
#
# time=90, clock=1, writeData=00000000000000000000000000000100, writeReg=100, expected=00000000000000000000000000000100,
#  readReg1=011, readData1=00000000000000000000000000000011, readReg2=100, readData2=00000000000000000000000000000100
#
# time=100, clock=0, writeData=00000000000000000000000000000100, writeReg=100, expected=00000000000000000000000000000100,
#  readReg1=011, readData1=00000000000000000000000000000011, readReg2=100, readData2=00000000000000000000000000000100
#
# time=110, clock=1, writeData=00000000000000000000000000000101, writeReg=101, expected=00000000000000000000000000000101,
#  readReg1=100, readData1=00000000000000000000000000000100, readReg2=101, readData2=00000000000000000000000000000101
#
# time=120, clock=0, writeData=00000000000000000000000000000101, writeReg=101, expected=00000000000000000000000000000101,
#  readReg1=100, readData1=00000000000000000000000000000100, readReg2=101, readData2=00000000000000000000000000000101
#
# time=130, clock=1, writeData=00000000000000000000000000000110, writeReg=110, expected=00000000000000000000000000000110,
#  readReg1=101, readData1=00000000000000000000000000000101, readReg2=110, readData2=00000000000000000000000000000110
#
# time=140, clock=0, writeData=00000000000000000000000000000110, writeReg=110, expected=00000000000000000000000000000110,
#  readReg1=101, readData1=00000000000000000000000000000101, readReg2=110, readData2=00000000000000000000000000000110
#
# time=150, clock=1, writeData=00000000000000000000000000000111, writeReg=111, expected=00000000000000000000000000000111,
#  readReg1=110, readData1=00000000000000000000000000000110, readReg2=111, readData2=00000000000000000000000000000111
#
# time=160, clock=0, writeData=00000000000000000000000000000111, writeReg=111, expected=00000000000000000000000000000111,
#  readReg1=110, readData1=00000000000000000000000000000110, readReg2=111, readData2=00000000000000000000000000000111
#
```

signExtend.v Test

```
VSIM 5> step -current
# time = 0 inpu=100101, outpu=11111111111111111111111111100101
# time =20 inpu=111001, outpu=11111111111111111111111111111001
# time =40 inpu=010101, outpu=00000000000000000000000000010101

VSIM 6>
```

memory.v Test

```
# time= 0, clock=0, writeData=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, memWrite=x, address=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, expected=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
# memRead=x, readData=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#
# time=10, clock=1, writeData=00000000000000000000000000000001, memWrite=1, address=00000000000000000000000000000001, expected=00000000000000000000000000000001,
# memRead=0, readData=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#
# time=20, clock=0, writeData=00000000000000000000000000000001, memWrite=1, address=00000000000000000000000000000001, expected=00000000000000000000000000000001,
# memRead=0, readData=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#
# time=30, clock=1, writeData=00000000000000000000000000000001, memWrite=0, address=00000000000000000000000000000001, expected=00000000000000000000000000000001,
# memRead=1, readData=00000000000000000000000000000001
#
# time=40, clock=0, writeData=00000000000000000000000000000001, memWrite=0, address=00000000000000000000000000000001, expected=00000000000000000000000000000001,
# memRead=1, readData=00000000000000000000000000000001
#
# time=50, clock=1, writeData=00000000000000000000000000000010, memWrite=1, address=00000000000000000000000000000010, expected=00000000000000000000000000000010,
# memRead=0, readData=00000000000000000000000000000001
#
# time=60, clock=0, writeData=00000000000000000000000000000010, memWrite=1, address=00000000000000000000000000000010, expected=00000000000000000000000000000010,
# memRead=0, readData=00000000000000000000000000000001
#
# time=70, clock=1, writeData=00000000000000000000000000000011, memWrite=0, address=00000000000000000000000000000010, expected=00000000000000000000000000000010,
# memRead=1, readData=00000000000000000000000000000010
#
# time=80, clock=0, writeData=00000000000000000000000000000011, memWrite=0, address=00000000000000000000000000000010, expected=00000000000000000000000000000010,
# memRead=1, readData=00000000000000000000000000000010
#
# time=90, clock=1, writeData=00000000000000000000000000000100, memWrite=1, address=00000000000000000000000000000011, expected=00000000000000000000000000000100,
# memRead=0, readData=00000000000000000000000000000010
#
# time=100, clock=0, writeData=00000000000000000000000000000100, memWrite=1, address=00000000000000000000000000000011, expected=00000000000000000000000000000100,
# memRead=0, readData=00000000000000000000000000000010
#
# time=110, clock=1, writeData=00000000000000000000000000000101, memWrite=0, address=00000000000000000000000000000011, expected=00000000000000000000000000000100,
# memRead=1, readData=00000000000000000000000000000100
#
# time=120, clock=0, writeData=00000000000000000000000000000101, memWrite=0, address=00000000000000000000000000000011, expected=00000000000000000000000000000100,
# memRead=1, readData=00000000000000000000000000000100
#
```

miniMips.v Test

There are in total of 64 instructions in the instructions.mem file. However there are only 30 legit tests are done. The others are skipped with the bne and beq instructions.

1-) and $1, $2, $3
2-) and $2, $1, $3
3-) add $3, $2, $3
4-) add $4, $3, $5
5-) sub $5, $4, $2
6-) sub $6, $5, $4
7-) xor $1, $5, $4
8-) xor $2, $1, $5
9-) nor $7, $2, $1
10-) nor $1, $7, $2
11-) or $1, $1, $6
12-) or $5, $7, $4
13-) addi $0, $7, 5 (Here I show that 0 register cannot be overwritten.)
14-) addi $2, $3, 1
15-) andi $2, $3, 1
16-) andi $1, $4, 7
17-) ori $1, $1, 7

18-) ori $4, $4, 1
19-) nori $5, $5, 2
20-) nori $1, $4, 3
21-) beq $3, $4, 10
22-) beq $5, $5, 10
33-) bne $5, $5, 10
34-) bne $6, $7, 24
59-) slti $7, $6, 1
60-) slti $4, $1, 7
61-) sw $2, 2($3)
62-) sw $3, 3($4)
63-) lw $5, 2($2)
64-) lw $7, 5($2)

```
VSIM 5> step -current
# time= 0, clock=1, PC=00000000000000000000000000000000 currentInstruction=00000100111001000, aluCtr=110,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=10, clock=0, PC=00000000000000000000000000000001 currentInstruction=00000100111001000, aluCtr=110,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=20, clock=1, PC=00000000000000000000000000000001 currentInstruction=00000101111010000, aluCtr=110,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=30, clock=0, PC=00000000000000000000000000000010 currentInstruction=00000101111010000, aluCtr=110,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=40, clock=1, PC=00000000000000000000000000000010 currentInstruction=00000100111011001, aluCtr=000,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000101,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=1, chosenWriteData=00000000000000000000000000000101
#
# time=50, clock=0, PC=00000000000000000000000000000011 currentInstruction=00000100111011001, aluCtr=000,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000101,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=1, chosenWriteData=00000000000000000000000000000101
#
# time=60, clock=1, PC=00000000000000000000000000000011 currentInstruction=00000111101100001, aluCtr=000,
# readData1=00000000000000000000000000000101, chosenAluInput2=00000000000000000000000000000101, aluResult(memoryAddress)=00000000000000000000000000001010,
# memWrite=0, writeData=00000000000000000000000000000101, regWrite=1, chosenWriteData=00000000000000000000000000001010
#
# time=70, clock=0, PC=00000000000000000000000000000100 currentInstruction=00000111101100001, aluCtr=000,
# readData1=00000000000000000000000000000101, chosenAluInput2=00000000000000000000000000000101, aluResult(memoryAddress)=00000000000000000000000000001010,
# memWrite=0, writeData=00000000000000000000000000000101, regWrite=1, chosenWriteData=00000000000000000000000000001010
#
# time=80, clock=1, PC=00000000000000000000000000000100 currentInstruction=00000100010101010, aluCtr=010,
# readData1=00000000000000000000000000001010, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=00000000000000000000000000001000,
# memWrite=0, writeData=00000000000000000000000000000010, regWrite=1, chosenWriteData=00000000000000000000000000001000
#
# time=90, clock=0, PC=00000000000000000000000000000101 currentInstruction=00000100010101010, aluCtr=010,
# readData1=00000000000000000000000000001010, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=00000000000000000000000000001000,
# memWrite=0, writeData=00000000000000000000000000000010, regWrite=1, chosenWriteData=00000000000000000000000000001000
#
# time=100, clock=1, PC=00000000000000000000000000000101 currentInstruction=00000101100110010, aluCtr=010,
# readData1=00000000000000000000000000001000, chosenAluInput2=00000000000000000000000000001010, aluResult(memoryAddress)=11111111111111111111111111111110,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=11111111111111111111111111111110
#
# time=110, clock=0, PC=00000000000000000000000000000110 currentInstruction=00000101100110010, aluCtr=010,
# readData1=00000000000000000000000000001000, chosenAluInput2=00000000000000000000000000001010, aluResult(memoryAddress)=11111111111111111111111111111110,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=11111111111111111111111111111110
#
# time=120, clock=1, PC=00000000000000000000000000000110 currentInstruction=00000101100001011, aluCtr=001,
# readData1=00000000000000000000000000001000, chosenAluInput2=00000000000000000000000000001010, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=00000000000000000000000000000010
```

```
# time=130, clock=0, PC=00000000000000000000000000000111 currentInstruction=0000101100001011, aluCtr=001,
# readData1=00000000000000000000000000001000, chosenAluInput2=00000000000000000000000000001010, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=140, clock=1, PC=00000000000000000000000000000111 currentInstruction=0000001101010011, aluCtr=001,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000001000, aluResult(memoryAddress)=00000000000000000000000000001010,
# memWrite=0, writeData=00000000000000000000000000001000, regWrite=1, chosenWriteData=00000000000000000000000000001010
#
# time=150, clock=0, PC=00000000000000000000000000001000 currentInstruction=0000001101010011, aluCtr=001,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000001000, aluResult(memoryAddress)=00000000000000000000000000001010,
# memWrite=0, writeData=00000000000000000000000000001000, regWrite=1, chosenWriteData=00000000000000000000000000001010
#
# time=160, clock=1, PC=00000000000000000000000000001000 currentInstruction=0000010001111100, aluCtr=101,
# readData1=00000000000000000000000000001010, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=11111111111111111111111111110101,
# memWrite=0, writeData=00000000000000000000000000000010, regWrite=1, chosenWriteData=11111111111111111111111111110101
#
# time=170, clock=0, PC=00000000000000000000000000001001 currentInstruction=0000010001111100, aluCtr=101,
# readData1=00000000000000000000000000001010, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=11111111111111111111111111110101,
# memWrite=0, writeData=00000000000000000000000000000010, regWrite=1, chosenWriteData=11111111111111111111111111110101
#
# time=180, clock=1, PC=00000000000000000000000000001001 currentInstruction=0000111010001100, aluCtr=101,
# readData1=11111111111111111111111111110101, chosenAluInput2=00000000000000000000000000001010, aluResult(memoryAddress)=00000000000000000000000000000000,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=00000000000000000000000000000000
#
# time=190, clock=0, PC=00000000000000000000000000001010 currentInstruction=0000111010001100, aluCtr=101,
# readData1=11111111111111111111111111110101, chosenAluInput2=00000000000000000000000000001010, aluResult(memoryAddress)=00000000000000000000000000000000,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=00000000000000000000000000000000
#
# time=200, clock=1, PC=00000000000000000000000000001010 currentInstruction=0000001110001101, aluCtr=111,
# readData1=00000000000000000000000000000000, chosenAluInput2=11111111111111111111111111111110, aluResult(memoryAddress)=11111111111111111111111111111110,
# memWrite=0, writeData=11111111111111111111111111111110, regWrite=1, chosenWriteData=11111111111111111111111111111110
#
# time=210, clock=0, PC=00000000000000000000000000001011 currentInstruction=0000001110001101, aluCtr=111,
# readData1=00000000000000000000000000000000, chosenAluInput2=11111111111111111111111111111110, aluResult(memoryAddress)=11111111111111111111111111111110,
# memWrite=0, writeData=11111111111111111111111111111110, regWrite=1, chosenWriteData=11111111111111111111111111111110
#
# time=220, clock=1, PC=00000000000000000000000000001011 currentInstruction=0000111100101101, aluCtr=111,
# readData1=11111111111111111111111111110101, chosenAluInput2=00000000000000000000000000001010, aluResult(memoryAddress)=11111111111111111111111111111111,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=11111111111111111111111111111111
#
# time=230, clock=0, PC=00000000000000000000000000001100 currentInstruction=0000111100101101, aluCtr=111,
# readData1=11111111111111111111111111110101, chosenAluInput2=00000000000000000000000000001010, aluResult(memoryAddress)=11111111111111111111111111111111,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=11111111111111111111111111111111
#
# time=240, clock=1, PC=00000000000000000000000000001100 currentInstruction=0001111000000101, aluCtr=000,
# readData1=11111111111111111111111111110101, chosenAluInput2=00000000000000000000000000000101, aluResult(memoryAddress)=11111111111111111111111111111010,
# memWrite=0, writeData=00000000000000000000000000000000, regWrite=1, chosenWriteData=11111111111111111111111111111010
#
# time=250, clock=0, PC=00000000000000000000000000001101 currentInstruction=0001111000000101, aluCtr=000,
# readData1=11111111111111111111111111110101, chosenAluInput2=00000000000000000000000000000101, aluResult(memoryAddress)=11111111111111111111111111111010,
# memWrite=0, writeData=00000000000000000000000000000000, regWrite=1, chosenWriteData=11111111111111111111111111111010
# time=260, clock=1, PC=00000000000000000000000000001101 currentInstruction=0001011010000001, aluCtr=000,
# readData1=00000000000000000000000000000101, chosenAluInput2=00000000000000000000000000000001, aluResult(memoryAddress)=00000000000000000000000000000110,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=00000000000000000000000000000110
#
# time=270, clock=0, PC=00000000000000000000000000001110 currentInstruction=0001011010000001, aluCtr=000,
# readData1=00000000000000000000000000000101, chosenAluInput2=00000000000000000000000000000001, aluResult(memoryAddress)=00000000000000000000000000000110,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=00000000000000000000000000000110
#
# time=280, clock=1, PC=00000000000000000000000000001110 currentInstruction=0010010100000011, aluCtr=110,
# readData1=00000000000000000000000000000110, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=290, clock=0, PC=00000000000000000000000000001111 currentInstruction=0010010100000011, aluCtr=110,
# readData1=00000000000000000000000000000110, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000001010, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=300, clock=1, PC=00000000000000000000000000001111 currentInstruction=0010100001000111, aluCtr=110,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000111, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=11111111111111111111111111110110, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=310, clock=0, PC=00000000000000000000000000010000 currentInstruction=0010100001000111, aluCtr=110,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000111, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=11111111111111111111111111110110, regWrite=1, chosenWriteData=00000000000000000000000000000010
#
# time=320, clock=1, PC=00000000000000000000000000010000 currentInstruction=0011001001000111, aluCtr=111,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000111, aluResult(memoryAddress)=00000000000000000000000000000111,
# memWrite=0, writeData=00000000000000000000000000000010, regWrite=1, chosenWriteData=00000000000000000000000000000111
#
# time=330, clock=0, PC=00000000000000000000000000010001 currentInstruction=0011001001000111, aluCtr=111,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000111, aluResult(memoryAddress)=00000000000000000000000000000111,
# memWrite=0, writeData=00000000000000000000000000000010, regWrite=1, chosenWriteData=00000000000000000000000000000111
#
# time=340, clock=1, PC=00000000000000000000000000010001 currentInstruction=0011100100000001, aluCtr=111,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000001, aluResult(memoryAddress)=00000000000000000000000000000011,
# memWrite=0, writeData=00000000000000000000000000000010, regWrite=1, chosenWriteData=00000000000000000000000000000011
#
# time=350, clock=0, PC=00000000000000000000000000010010 currentInstruction=0011100100000001, aluCtr=111,
# readData1=00000000000000000000000000000010, chosenAluInput2=00000000000000000000000000000001, aluResult(memoryAddress)=00000000000000000000000000000011,
# memWrite=0, writeData=00000000000000000000000000000010, regWrite=1, chosenWriteData=00000000000000000000000000000011
#
# time=360, clock=1, PC=00000000000000000000000000010010 currentInstruction=0100101101000010, aluCtr=101,
# readData1=11111111111111111111111111111111, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=00000000000000000000000000000000,
# memWrite=0, writeData=11111111111111111111111111111111, regWrite=1, chosenWriteData=00000000000000000000000000000000
#
# time=370, clock=0, PC=00000000000000000000000000010011 currentInstruction=0100101101000010, aluCtr=101,
# readData1=11111111111111111111111111111111, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=00000000000000000000000000000000,
# memWrite=0, writeData=11111111111111111111111111111111, regWrite=1, chosenWriteData=00000000000000000000000000000000
#
# time=380, clock=1, PC=00000000000000000000000000010011 currentInstruction=0100100001000011, aluCtr=101,
# readData1=00000000000000000000000000000011, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=11111111111111111111111111111100,
# memWrite=0, writeData=00000000000000000000000000000111, regWrite=1, chosenWriteData=11111111111111111111111111111100
```

```
# time=390, clock=0, PC=00000000000000000000000000010100 currentInstruction=0100100001000011, aluCtr=101,
# readData1=00000000000000000000000000000011, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=11111111111111111111111111111100,
# memWrite=0, writeData=00000000000000000000000000000111, regWrite=1, chosenWriteData=11111111111111111111111111111100
#
# time=400, clock=1, PC=00000000000000000000000000010100 currentInstruction=0101011100001010, aluCtr=010,
# readData1=00000000000000000000000000000101, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=0, chosenWriteData=00000000000000000000000000000010
#
# time=410, clock=0, PC=00000000000000000000000000010101 currentInstruction=0101011100001010, aluCtr=010,
# readData1=00000000000000000000000000000101, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000010,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=0, chosenWriteData=00000000000000000000000000000010
#
# time=420, clock=1, PC=00000000000000000000000000010101 currentInstruction=0101101101001010, aluCtr=010,
# readData1=00000000000000000000000000000000, chosenAluInput2=00000000000000000000000000000000, aluResult(memoryAddress)=00000000000000000000000000000000,
# memWrite=0, writeData=00000000000000000000000000000000, regWrite=0, chosenWriteData=00000000000000000000000000000000
#
# time=430, clock=0, PC=00000000000000000000000000100000 currentInstruction=0101101101001010, aluCtr=010,
# readData1=00000000000000000000000000000000, chosenAluInput2=00000000000000000000000000000000, aluResult(memoryAddress)=00000000000000000000000000000000,
# memWrite=0, writeData=00000000000000000000000000000000, regWrite=0, chosenWriteData=00000000000000000000000000000000
#
# time=440, clock=1, PC=00000000000000000000000000100000 currentInstruction=0110101101001010, aluCtr=010,
# readData1=00000000000000000000000000000000, chosenAluInput2=00000000000000000000000000000000, aluResult(memoryAddress)=00000000000000000000000000000000,
# memWrite=0, writeData=00000000000000000000000000000000, regWrite=0, chosenWriteData=00000000000000000000000000000000
#
# time=450, clock=0, PC=00000000000000000000000000100001 currentInstruction=0110101101001010, aluCtr=010,
# readData1=00000000000000000000000000000000, chosenAluInput2=00000000000000000000000000000000, aluResult(memoryAddress)=00000000000000000000000000000000,
# memWrite=0, writeData=00000000000000000000000000000000, regWrite=0, chosenWriteData=00000000000000000000000000000000
#
# time=460, clock=1, PC=00000000000000000000000000100001 currentInstruction=0110111110011000, aluCtr=010,
# readData1=11111111111111111111111111110101, chosenAluInput2=11111111111111111111111111111110, aluResult(memoryAddress)=11111111111111111111111111110111,
# memWrite=0, writeData=11111111111111111111111111111110, regWrite=0, chosenWriteData=11111111111111111111111111110111
#
# time=470, clock=0, PC=00000000000000000000000000111010 currentInstruction=0110111110011000, aluCtr=010,
# readData1=11111111111111111111111111110101, chosenAluInput2=11111111111111111111111111111110, aluResult(memoryAddress)=11111111111111111111111111110111,
# memWrite=0, writeData=11111111111111111111111111111110, regWrite=0, chosenWriteData=11111111111111111111111111110111
#
# time=480, clock=1, PC=00000000000000000000000000111010 currentInstruction=0111110111000001, aluCtr=100,
# readData1=11111111111111111111111111111110, chosenAluInput2=00000000000000000000000000000001, aluResult(memoryAddress)=00000000000000000000000000000001,
# memWrite=0, writeData=11111111111111111111111111110101, regWrite=1, chosenWriteData=00000000000000000000000000000001
#
# time=490, clock=0, PC=00000000000000000000000000111011 currentInstruction=0111110111000001, aluCtr=100,
# readData1=11111111111111111111111111111110, chosenAluInput2=00000000000000000000000000000001, aluResult(memoryAddress)=00000000000000000000000000000001,
# memWrite=0, writeData=11111111111111111111111111110101, regWrite=1, chosenWriteData=00000000000000000000000000000001
#
# time=500, clock=1, PC=00000000000000000000000000111011 currentInstruction=0111001100000111, aluCtr=100,
# readData1=11111111111111111111111111111100, chosenAluInput2=00000000000000000000000000000111, aluResult(memoryAddress)=00000000000000000000000000000001,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=1, chosenWriteData=00000000000000000000000000000001
#
# time=510, clock=0, PC=00000000000000000000000000111100 currentInstruction=0111001100000111, aluCtr=100,
# readData1=11111111111111111111111111111100, chosenAluInput2=00000000000000000000000000000111, aluResult(memoryAddress)=00000000000000000000000000000001,
# memWrite=0, writeData=00000000000000000000000000000011, regWrite=1, chosenWriteData=00000000000000000000000000000001

# time=520, clock=1, PC=00000000000000000000000000111100 currentInstruction=1001011010000010, aluCtr=000,
# readData1=00000000000000000000000000000101, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=00000000000000000000000000000111,
# memWrite=1, writeData=00000000000000000000000000000110, regWrite=0, chosenWriteData=00000000000000000000000000000111
#
# time=530, clock=0, PC=00000000000000000000000000111101 currentInstruction=1001011010000010, aluCtr=000,
# readData1=00000000000000000000000000000101, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=00000000000000000000000000000111,
# memWrite=1, writeData=00000000000000000000000000000110, regWrite=0, chosenWriteData=00000000000000000000000000000111
#
# time=540, clock=1, PC=00000000000000000000000000111101 currentInstruction=1001100011000011, aluCtr=000,
# readData1=00000000000000000000000000000001, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000100,
# memWrite=1, writeData=00000000000000000000000000000101, regWrite=0, chosenWriteData=00000000000000000000000000000100
#
# time=550, clock=0, PC=00000000000000000000000000111110 currentInstruction=1001100011000011, aluCtr=000,
# readData1=00000000000000000000000000000001, chosenAluInput2=00000000000000000000000000000011, aluResult(memoryAddress)=00000000000000000000000000000100,
# memWrite=1, writeData=00000000000000000000000000000101, regWrite=0, chosenWriteData=00000000000000000000000000000100
#
# time=560, clock=1, PC=00000000000000000000000000111110 currentInstruction=1000010101000010, aluCtr=000,
# readData1=00000000000000000000000000000110, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=00000000000000000000000000001000,
# memWrite=0, writeData=00000000000000000000000000000000, regWrite=1, chosenWriteData=00000000000000000000000000001111
#
# time=570, clock=0, PC=00000000000000000000000000111111 currentInstruction=1000010101000010, aluCtr=000,
# readData1=00000000000000000000000000000110, chosenAluInput2=00000000000000000000000000000010, aluResult(memoryAddress)=00000000000000000000000000001000,
# memWrite=0, writeData=00000000000000000000000000000000, regWrite=1, chosenWriteData=00000000000000000000000000001111
#
# time=580, clock=1, PC=00000000000000000000000000111111 currentInstruction=1000010111000101, aluCtr=000,
# readData1=00000000000000000000000000000110, chosenAluInput2=00000000000000000000000000000101, aluResult(memoryAddress)=00000000000000000000000000001011,
# memWrite=0, writeData=00000000000000000000000000000001, regWrite=1, chosenWriteData=00000000000000000000000000001111
#
# time=590, clock=0, PC=00000000000000000000000001000000 currentInstruction=1000010111000101, aluCtr=000,
# readData1=00000000000000000000000000000110, chosenAluInput2=00000000000000000000000000000101, aluResult(memoryAddress)=00000000000000000000000000001011,
# memWrite=0, writeData=00000000000000000000000000000001, regWrite=1, chosenWriteData=00000000000000000000000000001111
#
```