CSE331

Computer Organization

Homework - 4

Nevzat Seferoglu

171044024

*Project Purpose:

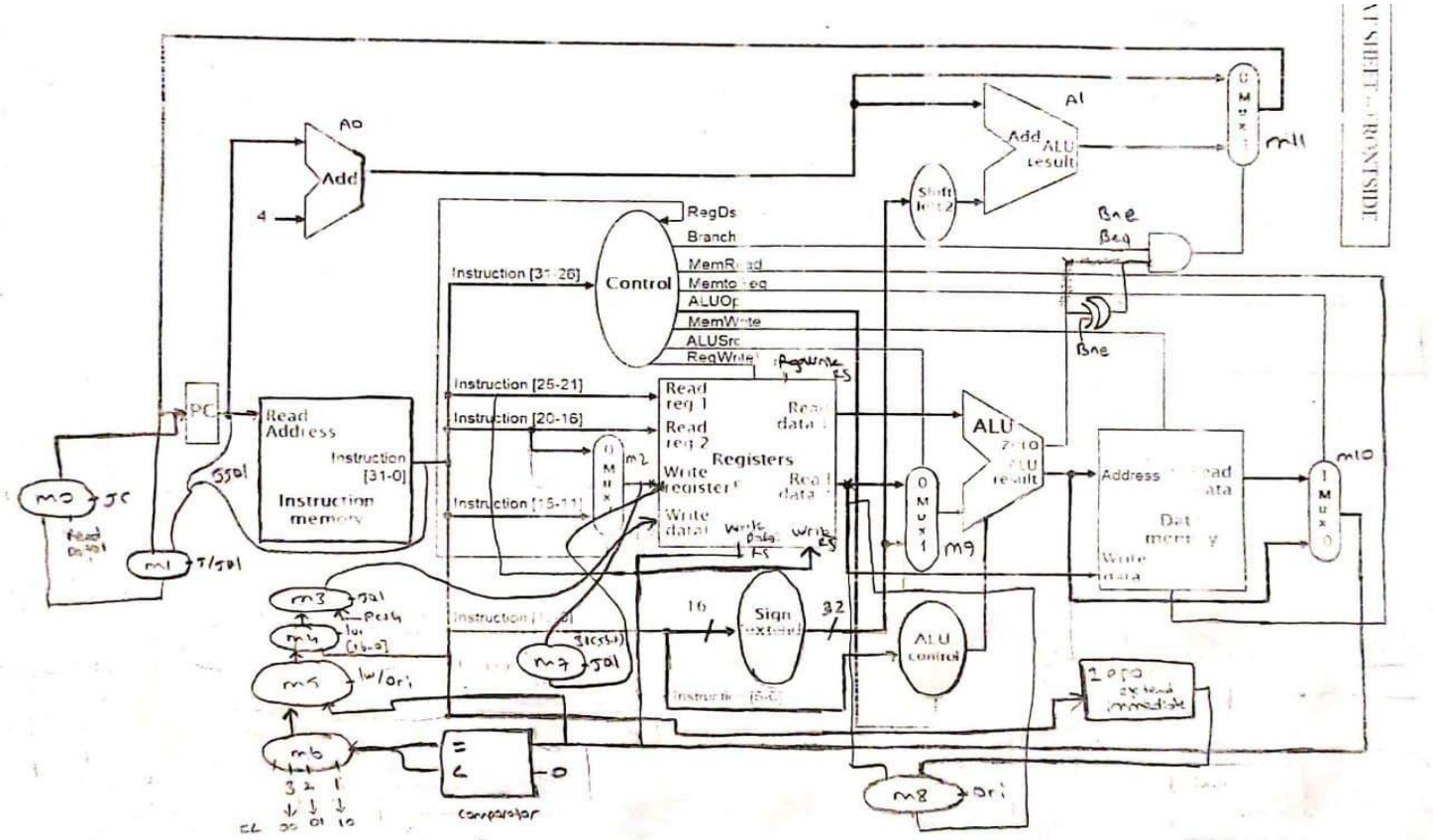
In this project, we implemented a different version of 32-bit MIPS processors. The block that we design get no inputs from the outside and can take several file inputs from the outside to test part.

Our processor will support 14 different instruction. These are, **lw, sw, j, jal, jr, beq, bne, addn, subn, xorn, andn, orn, ori, lui**. Also, we have **data memory**, **instruction memory** and **registers** as storage components.

*Design Details:

There was same changing in our project which is different than standard Mips processors. Data memory addressing bits must be 18 bits. And the data memory size must be 256KB which is 262.144 bytes. Our addressing (18 bits) supplies us a range of numbers from 0 to (2^{18}-1) which is 262.143 as a total number of spaces, we have 262.144 that matches with our size which is requested before. Therefore, **each line must be 8 bits in our data memory file** which means byte addressing. I ruled that situation and make data memory and instruction memory byte addressed.

*My overall design:



*Controller Unit Table:

Opcode	000000 R-type	100011 lw	101011 sw	000100 beq	000101 bne	001101 ori	001111 lui	000010 j	000011 jal	001000 jr
RegDst	1	0	X	X	X	0	0	X	X	X
Beq	0	0	0	1	0	0	0	0	0	0
Bne	0	0	0	0	1	0	0	0	0	0
MemRead	0	1	0	0	0	0	0	0	0	0
MemtoReg	0	1	X	X	X	0	X	X	X	X
ALUOp1	1	0	0	0	0	1	X	X	X	X
ALUOp0	0	0	0	1	1	1	X	X	X	X
MemWrite	0	0	1	0	0	0	0	0	0	0
ALUSrc	0	1	1	0	0	0	X	X	X	X
RegWrite1	1	1	0	0	0	1	1	0	0	1
RegWrite2	1	0	0	0	0	0	0	0	0	0
ori	0	X	X	0	0	1	X	X	X	X
lui	0	0	X	X	X	0	1	X	X	X
j	0	0	0	0	0	0	0	1	0	0
jal	0	0	0	0	0	0	0	0	1	1
lw	0	1	X	X	X	X	X	X	X	X
jr	0	0	0	0	0	0	0	0	0	0

*Alu Table:

	AluOp	Function Field	Output
lw, sw	00	XXXXXX	010
beq, bne	01	XXXXXX	110
add	10	100000	010
sub	10	100010	110
and	10	100100	000
or	10	100101	001
xorn	10	101000	011
ori	11	XXXXXX	001

CS Scanned with CamScanner

***Note1**: For creating **jr** signal, I used official function field of jr as a opcode which is sent to CU.

***Note2**: As a **function field** of **xorn**, I specified 101000.

- Other instructions have same properties which is denoted by Mips green sheet.

*Tests:

In the tests part, I tested each instruction for two times in a different test. I used different test because of readability of the homework. All register views are representing after tests situation.

```
Instructions in instruction memory:
------------
addn R4, R2, R3 -> 00000000
                   01000011
                   00100000
                   00100000
addn R7, R5, R6 -> 00000000
                   10100110
                   00111000
                   00100000
--------
Initial contents:

R2 content = 00000000000000000000000000000101
R3 content = 00000000000000000000000000000011
R5 content = 00000000000000000000000000000110
R6 content = 00000000000000000000000000001000
-------------------------------------------------------
*Note = All the other registers are filled with zero.
```

mips_registers	00000000000000000000000000000000 00000000000000000000...	Fixed...	Internal
[31]	00000000000000000000000000000000	Pack...	Internal
[30]	00000000000000000000000000000000	Pack...	Internal
[29]	00000000000000000000000000000000	Pack...	Internal
[28]	00000000000000000000000000000000	Pack...	Internal
[27]	00000000000000000000000000000000	Pack...	Internal
[26]	00000000000000000000000000000000	Pack...	Internal
[25]	00000000000000000000000000000000	Pack...	Internal
[24]	00000000000000000000000000000000	Pack...	Internal
[23]	00000000000000000000000000000000	Pack...	Internal
[22]	00000000000000000000000000000000	Pack...	Internal
[21]	00000000000000000000000000000000	Pack...	Internal
[20]	00000000000000000000000000000000	Pack...	Internal
[19]	00000000000000000000000000000000	Pack...	Internal
[18]	00000000000000000000000000000000	Pack...	Internal
[17]	00000000000000000000000000000000	Pack...	Internal
[16]	00000000000000000000000000000000	Pack...	Internal
[15]	00000000000000000000000000000000	Pack...	Internal
[14]	00000000000000000000000000000000	Pack...	Internal
[13]	00000000000000000000000000000000	Pack...	Internal
[12]	00000000000000000000000000000000	Pack...	Internal
[11]	00000000000000000000000000000000	Pack...	Internal
[10]	00000000000000000000000000000000	Pack...	Internal
[9]	00000000000000000000000000000000	Pack...	Internal
[8]	00000000000000000000000000000000	Pack...	Internal
[7]	00000000000000000000000000000011	Pack...	Internal
[6]	00000000000000000000000000001000	Pack...	Internal
[5]	00000000000000000000000000001110	Pack...	Internal
[4]	00000000000000000000000000000011	Pack...	Internal
[3]	00000000000000000000000000000011	Pack...	Internal
[2]	00000000000000000000000000001000	Pack...	Internal
[1]	00000000000000000000000000000000	Pack...	Internal
[0]	00000000000000000000000000000000	Pack...	Internal

```
Instructions in instruction memory:
------------
andn R4, R2, R3 -> 00000000
                   01000011
                   00100000
                   00100100
andn R7, R5, R6 -> 00000000
                   10100110
                   00111000
                   00100100
--------
Initial contents:

R2 content = 01010101010101010101010101010101
R3 content = 11111111111111111111111111111111
R5 content = 11111111111111100000000000000101
R6 content = 10101010101010100000000000000101
-------------------------------------------------------
*Note = All the other registers are filled with zero.
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000010		Pack... Internal
[6]	10101010101010100000000000000101		Pack... Internal
[5]	10101010101010100000000000000101		Pack... Internal
[4]	00000000000000000000000000000011		Pack... Internal
[3]	11111111111111111111111111111111		Pack... Internal
[2]	01010101010101010101010101010101		Pack... Internal
[1]	00000000000000000000000000000000		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
1: beq R1, R2, 10 -> 00010000
                     00100010
                     00000000
                     00001010

45 : addn R3, R2, R1

--------
*At the line 45(estimated jump), there is an other instruction in instruction memory to test whether
beq is works or not.

*At the line 45 there is an another helper test instruction in instruction memory that helps me tot test:

addn R3, R2, R1
00000000
01000001
00011000
00100000

Initial contents:

R1 content = 00000000000000000000000000000010
R2 content = 00000000000000000000000000000010
R3 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed... Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	00000000000000000000000000000000		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	00000000000000000000000000000000		Pack... Internal
[3]	00000000000000000000000000000011		Pack... Internal
[2]	00000000000000000000000000000100		Pack... Internal
[1]	00000000000000000000000000000010		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
1: beq R1, R2, 12 -> 00010000
                     00100010
                     00000000
                     00001100

53 : addn R3, R2, R1

--------
*At the line 53(estimated jump), there is an other instruction in instruction memory to test whether
beq is works or not.


*At the line 53 there is an another helper test instruction in instruction memory that helps me tot test:

addn R3, R2, R1
00000000
01000001
00011000
00100000

Initial contents:

R1 content = 00000000000000000000000000000011
R2 content = 00000000000000000000000000000011
R3 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	00000000000000000000000000000000		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	00000000000000000000000000000000		Pack... Internal
[3]	00000000000000000000000000000011		Pack... Internal
[2]	00000000000000000000000000000110		Pack... Internal
[1]	00000000000000000000000000000011		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
1: bne R1, R2, 10 -> 00010100
                     00100010
                     00000000
                     00001010

45 : addn R3, R2, R1

--------
*At the line 45(estimated jump), there is an other instruction in instruction memory to test whether
bne is works or not.

*At the line 45 there is an another helper test instruction in instruction memory that helps me tot test:

addn R3, R2, R1
00000000
01000001
00011000
00100000

Initial contents:

R1 content = 00000000000000000000000000000011
R2 content = 00000000000000000000000000000011
R3 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
----------------------------------------------------
```

mips_registers	00000000000000000000000000000000 00000000000000000000...	Fixed...	Internal
[31]	00000000000000000000000000000000	Pack...	Internal
[30]	00000000000000000000000000000000	Pack...	Internal
[29]	00000000000000000000000000000000	Pack...	Internal
[28]	00000000000000000000000000000000	Pack...	Internal
[27]	00000000000000000000000000000000	Pack...	Internal
[26]	00000000000000000000000000000000	Pack...	Internal
[25]	00000000000000000000000000000000	Pack...	Internal
[24]	00000000000000000000000000000000	Pack...	Internal
[23]	00000000000000000000000000000000	Pack...	Internal
[22]	00000000000000000000000000000000	Pack...	Internal
[21]	00000000000000000000000000000000	Pack...	Internal
[20]	00000000000000000000000000000000	Pack...	Internal
[19]	00000000000000000000000000000000	Pack...	Internal
[18]	00000000000000000000000000000000	Pack...	Internal
[17]	00000000000000000000000000000000	Pack...	Internal
[16]	00000000000000000000000000000000	Pack...	Internal
[15]	00000000000000000000000000000000	Pack...	Internal
[14]	00000000000000000000000000000000	Pack...	Internal
[13]	00000000000000000000000000000000	Pack...	Internal
[12]	00000000000000000000000000000000	Pack...	Internal
[11]	00000000000000000000000000000000	Pack...	Internal
[10]	00000000000000000000000000000000	Pack...	Internal
[9]	00000000000000000000000000000000	Pack...	Internal
[8]	00000000000000000000000000000000	Pack...	Internal
[7]	00000000000000000000000000000000	Pack...	Internal
[6]	00000000000000000000000000000000	Pack...	Internal
[5]	00000000000000000000000000000000	Pack...	Internal
[4]	00000000000000000000000000000000	Pack...	Internal
[3]	00000000000000000000000000000011	Pack...	Internal
[2]	00000000000000000000000000000110	Pack...	Internal
[1]	00000000000000000000000000000011	Pack...	Internal
[0]	00000000000000000000000000000000	Pack...	Internal

```
Instructions in instruction memory:
------------
1: bne R1, R2, 12 -> 00010100
                     00100010
                     00000000
                     00001100

53 : addn R3, R2, R1
--------
*At the line 53(estimated jump), there is an other instruction in instruction memory to test whether
bne is works or not.

*At the line 53 there is an another helper test instruction in instruction memory that helps me tot test:

addn R3, R2, R1
00000000
01000001
00011000
00100000

Initial contents:

R1 content = 00000000000000000000000000000010
R2 content = 00000000000000000000000000000011
R3 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	00000000000000000000000000000000		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	00000000000000000000000000000000		Pack... Internal
[3]	00000000000000000000000000000011		Pack... Internal
[2]	00000000000000000000000000000101		Pack... Internal
[1]	00000000000000000000000000000010		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
1: j 10 -> 00001000
           00000000
           00000000
           00001010

45 : addn R3, R2, R1

--------
*At the line 45(estimated jump), there is an other instruction in instruction memory to test whether
j is works or not.

*At the line 45 there is an another helper test instruction in instruction memory that helps me tot test:

addn R3, R2, R1
00000000
01000001
00011000
00100000

Initial contents:

R1 content = 00000000000000000000000000000101
R2 content = 00000000000000000000000000000110
R3 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000 00000000000000000000...	Fixed...	Internal
[31]	00000000000000000000000000000000	Pack...	Internal
[30]	00000000000000000000000000000000	Pack...	Internal
[29]	00000000000000000000000000000000	Pack...	Internal
[28]	00000000000000000000000000000000	Pack...	Internal
[27]	00000000000000000000000000000000	Pack...	Internal
[26]	00000000000000000000000000000000	Pack...	Internal
[25]	00000000000000000000000000000000	Pack...	Internal
[24]	00000000000000000000000000000000	Pack...	Internal
[23]	00000000000000000000000000000000	Pack...	Internal
[22]	00000000000000000000000000000000	Pack...	Internal
[21]	00000000000000000000000000000000	Pack...	Internal
[20]	00000000000000000000000000000000	Pack...	Internal
[19]	00000000000000000000000000000000	Pack...	Internal
[18]	00000000000000000000000000000000	Pack...	Internal
[17]	00000000000000000000000000000000	Pack...	Internal
[16]	00000000000000000000000000000000	Pack...	Internal
[15]	00000000000000000000000000000000	Pack...	Internal
[14]	00000000000000000000000000000000	Pack...	Internal
[13]	00000000000000000000000000000000	Pack...	Internal
[12]	00000000000000000000000000000000	Pack...	Internal
[11]	00000000000000000000000000000000	Pack...	Internal
[10]	00000000000000000000000000000000	Pack...	Internal
[9]	00000000000000000000000000000000	Pack...	Internal
[8]	00000000000000000000000000000000	Pack...	Internal
[7]	00000000000000000000000000000000	Pack...	Internal
[6]	00000000000000000000000000000000	Pack...	Internal
[5]	00000000000000000000000000000000	Pack...	Internal
[4]	00000000000000000000000000000000	Pack...	Internal
[3]	00000000000000000000000000000011	Pack...	Internal
[2]	00000000000000000000000000001011	Pack...	Internal
[1]	00000000000000000000000000000101	Pack...	Internal
[0]	00000000000000000000000000000000	Pack...	Internal

```
Instructions in instruction memory:
------------
1: j 12 -> 00001000
           00000000
           00000000
           00001100

53 : addn R3, R2, R1

--------
*At the line 53(estimated jump), there is an other instruction in instruction memory to test whether
j is works or not.

*At the line 53 there is an another helper test instruction in instruction memory that helps me tot test:

addn R3, R2, R1
00000000
01000001
00011000
00100000

Initial contents:

R1 content = 00000000000000000000000000000111
R2 content = 00000000000000000000000000000111
R3 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	00000000000000000000000000000000		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	00000000000000000000000000000000		Pack... Internal
[3]	00000000000000000000000000000011		Pack... Internal
[2]	00000000000000000000000000001110		Pack... Internal
[1]	00000000000000000000000000000111		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
1: jal 10 -> 00001100
             00000000
             00000000
             00001010

45 : addn R3, R2, R1

--------
*First instruction in the instruction memory. Therefore R[31] should 4 after jumping and linking.
*At the line 45(estimated jump), there is an other instruction in instruction memory to test whether
jal is works or not.

*At the line 45 there is an another helper test instruction in instruction memory that helps me tot test:

addn R3, R2, R1
00000000
01000001
00011000
00100000

Initial contents:

R1 content = 00000000000000000000000000000011
R2 content = 00000000000000000000000000000010
R3 content = 00000000000000000000000000000000
R31 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000100 00000000000000000000...	Fixed...	Internal
[31]	00000000000000000000000000000100	Pack...	Internal
[30]	00000000000000000000000000000000	Pack...	Internal
[29]	00000000000000000000000000000000	Pack...	Internal
[28]	00000000000000000000000000000000	Pack...	Internal
[27]	00000000000000000000000000000000	Pack...	Internal
[26]	00000000000000000000000000000000	Pack...	Internal
[25]	00000000000000000000000000000000	Pack...	Internal
[24]	00000000000000000000000000000000	Pack...	Internal
[23]	00000000000000000000000000000000	Pack...	Internal
[22]	00000000000000000000000000000000	Pack...	Internal
[21]	00000000000000000000000000000000	Pack...	Internal
[20]	00000000000000000000000000000000	Pack...	Internal
[19]	00000000000000000000000000000000	Pack...	Internal
[18]	00000000000000000000000000000000	Pack...	Internal
[17]	00000000000000000000000000000000	Pack...	Internal
[16]	00000000000000000000000000000000	Pack...	Internal
[15]	00000000000000000000000000000000	Pack...	Internal
[14]	00000000000000000000000000000000	Pack...	Internal
[13]	00000000000000000000000000000000	Pack...	Internal
[12]	00000000000000000000000000000000	Pack...	Internal
[11]	00000000000000000000000000000000	Pack...	Internal
[10]	00000000000000000000000000000000	Pack...	Internal
[9]	00000000000000000000000000000000	Pack...	Internal
[8]	00000000000000000000000000000000	Pack...	Internal
[7]	00000000000000000000000000000000	Pack...	Internal
[6]	00000000000000000000000000000000	Pack...	Internal
[5]	00000000000000000000000000000000	Pack...	Internal
[4]	00000000000000000000000000000000	Pack...	Internal
[3]	00000000000000000000000000000011	Pack...	Internal
[2]	00000000000000000000000000000101	Pack...	Internal
[1]	00000000000000000000000000000011	Pack...	Internal
[0]	00000000000000000000000000000000	Pack...	Internal

```
Instructions in instruction memory:
------------
1: jal 12 -> 00001100
             00000000
             00000000
             00001100

53 : addn R3, R2, R1

--------
*First instruction in the instruction memory. Therefore R[31] should 4 after jumping and linking.
*At the line 53(estimated jump), there is an other instruction in instruction memory to test whether
jal is works or not.

*At the line 53 there is an another helper test instruction in instruction memory that helps me tot test:

addn R3, R2, R1
00000000
01000001
00011000
00100000

Initial contents:

R1 content = 00000000000000000000000000000101
R2 content = 00000000000000000000000000000110
R3 content = 00000000000000000000000000000000
R31 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000100	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000100		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	00000000000000000000000000000000		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	00000000000000000000000000000000		Pack... Internal
[3]	00000000000000000000000000000011		Pack... Internal
[2]	00000000000000000000000000001011		Pack... Internal
[1]	00000000000000000000000000000101		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
1: jr R2 00100000
         01000000
         00000000
         00000000

44 : addn R4, R3, R1

--------
*At the line 45(estimated jump), there is an other instruction in instruction memory to test whether
jr is works or not.

*At the line 45 there is an another helper test instruction in instruction memory that helps me tot test:

addn R4, R3, R1
00000000
01100001
00100000
00100000


Initial contents:

R1 content = 00000000000000000000000000000111
R2 content = 00000000000000000000000000101100 (decimal 44) because lines are start with 1.
R3 content = 00000000000000000000000000000110
R4 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed... Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	00000000000000000000000000000000		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	00000000000000000000000000000011		Pack... Internal
[3]	00000000000000000000000000001101		Pack... Internal
[2]	00000000000000000000000000101100		Pack... Internal
[1]	00000000000000000000000000000111		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
1: jr R2 00100000
         01000000
         00000000
         00000000

53: addn R4, R3, R1

--------
*At the line 53(estimated jump), there is an other instruction in instruction memory to test whether
jr is works or not.

*At the line 53 there is an another helper test instruction in instruction memory that helps me tot test:

addn R4, R3, R1
00000000
01100001
00100000
00100000


Initial contents:

R1 content = 00000000000000000000000000000010
R2 content = 00000000000000000000000000110100 (decimal 52) because lines are start with 1.
R3 content = 00000000000000000000000000000011
R4 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed... Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	00000000000000000000000000000000		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	00000000000000000000000000000011		Pack... Internal
[3]	00000000000000000000000000000101		Pack... Internal
[2]	00000000000000000000000000110100		Pack... Internal
[1]	00000000000000000000000000000010		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
lui R4, 16'b1010101010101111 -> 00111100
                                00000100
                                10101010
                                10101111

lui R8, 16'b1111111111111111 -> 00111100
                                00001000
                                11111111
                                11111111
--------
Initial contents:

R4 content = 00000000000000000000000000000000
R8 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	11111111111111110000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	00000000000000000000000000000000		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	10101010101011110000000000000000		Pack... Internal
[3]	00000000000000000000000000000000		Pack... Internal
[2]	00000000000000000000000000000000		Pack... Internal
[1]	00000000000000000000000000000000		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
lw R5, 0(R0) -> 10001100
                00000101
                00000000
                00000000
lw R6, 4(R4) -> 10001100
                10000110
                00000000
                00000100
--------
Initial contents:

R0 content = 00000000000000000000000000000000
R4 content = 00000000000000000000000000000100
*Note = All the other registers are filled with zero.

Initial data memory denoted different in file.
-----------------------------------------------------
```

Data memory before tests:

[19]	00000000	Pack...	Internal
[18]	00000000	Pack...	Internal
[17]	00000000	Pack...	Internal
[16]	00000000	Pack...	Internal
[15]	00000000	Pack...	Internal
[14]	00000000	Pack...	Internal
[13]	00000000	Pack...	Internal
[12]	00000000	Pack...	Internal
[11]	00000000	Pack...	Internal
[10]	11111111	Pack...	Internal
[9]	10101010	Pack...	Internal
[8]	11111111	Pack...	Internal
[7]	10101010	Pack...	Internal
[6]	00000000	Pack...	Internal
[5]	00000000	Pack...	Internal
[4]	00000000	Pack...	Internal
[3]	11111111	Pack...	Internal
[2]	11111111	Pack...	Internal
[1]	00000000	Pack...	Internal
[0]	00000000	Pack...	Internal

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000000		Pack... Internal
[6]	11111111101010101111111100000000		Pack... Internal
[5]	00000000000000001111111111111111		Pack... Internal
[4]	00000000000000000000000000000100		Pack... Internal
[3]	00000000000000000000000000000000		Pack... Internal
[2]	00000000000000000000000000000000		Pack... Internal
[1]	00000000000000000000000000000000		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
ori R4, R3, 16'b1111111111111111 -> 00110100
                                    01100100
                                    11111111
                                    11111111

ori R7, R6, 16'b0101010101010101 -> 00110100
                                    11000111
                                    01010101
                                    01010101
--------
Initial contents:

R3 content = 11111111111111110101010101010101
R4 content = 00000000000000000000000000000000

R6 content = 11111111111111111111111111111111
R7 content = 00000000000000000000000000000000

*Note = All the other registers are filled with zero.
-----------------------------------------------------
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	11111111111111111111111111111111		Pack... Internal
[6]	11111111111111111111111111111111		Pack... Internal
[5]	00000000000000000000000000000000		Pack... Internal
[4]	11111111111111111111111111111111		Pack... Internal
[3]	11111111111111110101010101010101		Pack... Internal
[2]	00000000000000000000000000000000		Pack... Internal
[1]	00000000000000000000000000000000		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
orn R4, R2, R3 -> 00000000
                  01000011
                  00100000
                  00100101
orn R7, R5, R6 -> 00000000
                  10100110
                  00111000
                  00100101
--------
Initial contents:

R2 content = 00000000000000001111111111111111
R3 content = 11111111111111100000000000000000
R5 content = 01010101010101010101010101010101
R6 content = 11111111111111111111111111111111
-------------------------------------------------------
*Note = All the other registers are filled with zero.
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000010		Pack... Internal
[6]	11111111111111111111111111111111		Pack... Internal
[5]	11111111111111111111111111111111		Pack... Internal
[4]	00000000000000000000000000000010		Pack... Internal
[3]	11111111111111100000000000000000		Pack... Internal
[2]	11111111111111111111111111111111		Pack... Internal
[1]	00000000000000000000000000000000		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
subn R4, R2, R3 -> 00000000
                   01000011
                   00100000
                   00100010
subn R7, R5, R6 -> 00000000
                   10100110
                   00111000
                   00100010
--------
Initial contents:

R2 content = 00000000000000000000000000011001
R3 content = 00000000000000000000000000000010
R5 content = 00000000000000000000000000000100
R6 content = 00000000000000000000000000000001
-------------------------------------------------------
*Note = All the other registers are filled with zero.
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000011		Pack... Internal
[6]	00000000000000000000000000000001		Pack... Internal
[5]	00000000000000000000000000000011		Pack... Internal
[4]	00000000000000000000000000000011		Pack... Internal
[3]	00000000000000000000000000000010		Pack... Internal
[2]	00000000000000000000000000010111		Pack... Internal
[1]	00000000000000000000000000000000		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

```
Instructions in instruction memory:
------------
sw R4, 0(R0) -> 10101100
                00000100
                00000000
                00000000
sw R6, 4(R5) -> 10101100
                10100110
                00000000
                00000100
--------
Initial contents:

R0 content = 00000000000000000000000000000100
R5 content = 00000000000000000000000000000100

R4 content = 01010101010101010101010101010101
R6 content = 11111111111111110101010101010101

*Note = All the other registers are filled with zero.
*Note = All data memory block filled with zero.
-----------------------------------------------------
```

mips_data_memory...	00000000 00000000 00000000 00000000 00000000 00000000 000...	Fixed...	Internal
[19]	00000000	Pack...	Internal
[18]	00000000	Pack...	Internal
[17]	00000000	Pack...	Internal
[16]	00000000	Pack...	Internal
[15]	00000000	Pack...	Internal
[14]	00000000	Pack...	Internal
[13]	00000000	Pack...	Internal
[12]	00000000	Pack...	Internal
[11]	01010101	Pack...	Internal
[10]	01010101	Pack...	Internal
[9]	11111111	Pack...	Internal
[8]	11111111	Pack...	Internal
[7]	01010101	Pack...	Internal
[6]	01010101	Pack...	Internal
[5]	01010101	Pack...	Internal
[4]	01010101	Pack...	Internal
[3]	00000000	Pack...	Internal
[2]	00000000	Pack...	Internal
[1]	00000000	Pack...	Internal
[0]	00000000	Pack...	Internal

```
*I assigned "101000" as a function field of XORN operation.
Instructions in instruction memory:
------------
xorn R4, R2, R3 -> 00000000
                   01000011
                   00100000
                   00101000
xorn R7, R5, R6 -> 00000000
                   10100110
                   00111000
                   00101000
--------
Initial contents:

R2 content = 00000000000000001111111111111111
R3 content = 11111111111110100000000000000000
R5 content = 01010101010101010101010101010101
R6 content = 00000000000000000000000000001111
-------------------------------------------------------
*Note = All the other registers are filled with zero.
```

mips_registers	00000000000000000000000000000000	00000000000000000000...	Fixed...Internal
[31]	00000000000000000000000000000000		Pack... Internal
[30]	00000000000000000000000000000000		Pack... Internal
[29]	00000000000000000000000000000000		Pack... Internal
[28]	00000000000000000000000000000000		Pack... Internal
[27]	00000000000000000000000000000000		Pack... Internal
[26]	00000000000000000000000000000000		Pack... Internal
[25]	00000000000000000000000000000000		Pack... Internal
[24]	00000000000000000000000000000000		Pack... Internal
[23]	00000000000000000000000000000000		Pack... Internal
[22]	00000000000000000000000000000000		Pack... Internal
[21]	00000000000000000000000000000000		Pack... Internal
[20]	00000000000000000000000000000000		Pack... Internal
[19]	00000000000000000000000000000000		Pack... Internal
[18]	00000000000000000000000000000000		Pack... Internal
[17]	00000000000000000000000000000000		Pack... Internal
[16]	00000000000000000000000000000000		Pack... Internal
[15]	00000000000000000000000000000000		Pack... Internal
[14]	00000000000000000000000000000000		Pack... Internal
[13]	00000000000000000000000000000000		Pack... Internal
[12]	00000000000000000000000000000000		Pack... Internal
[11]	00000000000000000000000000000000		Pack... Internal
[10]	00000000000000000000000000000000		Pack... Internal
[9]	00000000000000000000000000000000		Pack... Internal
[8]	00000000000000000000000000000000		Pack... Internal
[7]	00000000000000000000000000000011		Pack... Internal
[6]	00000000000000000000000000001111		Pack... Internal
[5]	01010101010101010101010101011010		Pack... Internal
[4]	00000000000000000000000000000010		Pack... Internal
[3]	11111111111110100000000000000000		Pack... Internal
[2]	11111111111110101111111111111111		Pack... Internal
[1]	00000000000000000000000000000000		Pack... Internal
[0]	00000000000000000000000000000000		Pack... Internal

*Result:

According to tests that I made above, all requested instructions work fine.

*Important reminder:

For creating **jr** signal, I used official function field of jr as an opcode which is sent to CU. Therefore, I did not need to send function field to control unit.