GÜRCAN GÜL – 260201069

CENG311 PROGRAMMING ASSIGNMENT 3

I added shift left two operations to the Datapath for the "**jump**" and "**jal**" instructions. Then I added two mux for **jump**, **jal** and **jr** transactions. I made the sizes of the mux that have out1 and out3 outputs to be 4 to do jal operation. Then I added 5 and 32 bit 0 to the 4th input of multiplexers because they are not important. I used 6 and gates in total for branch transactions. I set the aloud to four bits because I needed at least four bits to be able to control operations in alu control, since the number of instructions is more than 8. The regdest signal is two bits, I set the memtoreg signal to be 2 bits because 2 selection signals are required for the mux. I have used 3 **.dat** files. I have made it clearer in Datapath.

For initIM.dat

1) sw instruction (32'd0)

Sw $R3, 0($R12)

101011	01100	01011	0000000000000000

$R12+0 = $R3

$R3 = 56 hex 00000000000000000000000001010110 binary

Regdest	Alusrc	Regwrite	Memwrite	Aluop
00	1	0	1	0000

2) lw instruction (32'd4)

Lw $R13, 0($R12)

100011	01100	01101	0000000000000000

$R12+0 = 56 hex 00000000000000000000000001010110 binary

$R13 = $R12+0

Regdest	Alusrc	Regwrite	Memread	Memtoreg
00	1	1	1	01

3) jal instruction (32'd8)

Jal 5

000011	00000000000000000000000101

$R31 = PC+4 = 16

Jump (offset*4) = jump 32'd16 adress

Regdest	jump	Regwrite	Memtoreg	Aluop
10	1	1	10	0000

4) jr instruction

Jr $R31

000000	11111	00000000000000001000

$r31 = 16 jump 32'd12 address

Regdest	jreg	Regwrite	Memtoreg	Aluop
00	1	0	00	0010

5) bne instruction

Bne $R2, $R3, 1

000101	00010	00011	0000000000000001

$R3 = 56 hex 86 dec 00000000000000000000000001010110 binary

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

If $R2 not equal $R3 branch 1*4+PC+4

Branch 32'd20 address

Regdest	branchne	Regwrite	Memtoreg	Aluop
00	1	0	00	0001

6) bltz instruction

Bltz $R4, 1

000001	00100	00000000000000000001

$R4 = -6

If $R4 < 0 branch 1*4+PC+4

Branch 32'd28 address

Regdest	branchltz	Regwrite	Memtoreg	Aluop
00	1	0	00	1000

For initIM2.dat

7) add instruction (32'd0)

Add $R1, $R2, $R3

000000	00010	00011	00001	00000	100000

$R3 = 56 hex 86 dec 00000000000000000000000001010110 binary

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

$R1 = $R2 + $R3 = 107 dec

Regdest	Regwrite	Memtoreg	Aluop
01	1	00	0010

8) sub instruction (32'd4)

Sub $R4, $R2, $R3

000000	00010	00011	00100	00000	100010

$R3 = 56 hex 86 dec 00000000000000000000000001010110 binary

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

$R4 = $R2-$R3 = -65 dec

Regdest	Regwrite	Memtoreg	Aluop
01	1	00	0010

9) and instruction (32'd8)

And $R5, $R2, $R3

000000	00010	00011	00101	00000	100100

$R3 = 56 hex 86 dec 00000000000000000000000001010110 binary

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

$R5 = $R2 and $R3 = 00000000000000000000000000010100 binary

Regdest	Regwrite	Memtoreg	Aluop
01	1	00	0010

10) or instruction (32'd12)

Or $R6, $R2, $R3

000000	00010	00011	00110	00000	100101

$R3 = 56 hex 86 dec 00000000000000000000000001010110 binary

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

$R5 = $R2 or $R3 = 00000000000000000000000001010111 binary

Regdest	Regwrite	Memtoreg	Aluop
01	1	00	0010

11) slt instruction (32'd16)

slt $R7, $R2, $R3

000000	00010	00011	00111	00000	101010

$R3 = 56 hex 86 dec 00000000000000000000000001010110 binary

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

$R7 = 1 if $R2 < $R3

$R7 = 0 if $R2 > $R3

Regdest	Regwrite	Memtoreg	Aluop
01	1	00	0010

12) nor instruction (32'd20)

nor $R8, $R2, $R3

000000	00010	00011	01000	00000	100111

$R3 = 56 hex 86 dec 00000000000000000000000001010110 binary

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

$R8 = $R2 not or $R3 11111111111111111111111110101000 binary

Regdest	Regwrite	Memtoreg	Aluop
01	1	00	0010

13) addi instruction

Addi $R9, $R2, 18

001000	00010	01001	0000000000010010

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

18 = 12 hex 18 dec 00000000000000000000000000010010 binary

$R9 =39 dec 00000000000000000000000001010111 binary

$R9 = $R2 + 18

Regdest	Alusrc	Regwrite	Memwrite	Aluop
00	1	1	0	0110

14) andi instruction

Andi $R10, $R3, 15

001100	00011	01010	0000000000001111

15 = F hex 15 dec 00000000000000000000000000001111 binary

$R3 = 56 hex 86 dec 00000000000000000000000001010110 binary

$R10 =39 dec 00000000000000000000000000000110 binary

$R10 = $R3 + 15

Regdest	Alusrc	Regwrite	Memwrite	Aluop
00	1	1	0	0011

initIM3.dat file

15) beq instruction (32'd0)

Beq $R1, $R1, 1

000100	00001	00001	0000000000000001

$R1 = 0 hex 0 dec 00000000000000000000000000000000 binary

If $R21 equal $R1 branch 1*4+PC+4

Branch 32'd8 address

Regdest	branch	Regwrite	Memtoreg	Aluop
00	1	0	00	0001

16) j instruction (32'd8)

J 1

000010	00000000000000000000000001

Jump (offset*4) = 1*4 = 4 jump 32'd4 address

Regdest	jump	Regwrite	Memtoreg	Aluop
00	1	0	00	0000

17) bgtz instruction (32'd4)

Bgtz $R2, 2

000111	00010	00000	0000000000000010

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

If $R2 > 0 branch 2*4+PC+4

Branch 32'd16 address

Regdest	branchgtz	Regwrite	Memtoreg	Aluop
00	1	0	00	1100

18) bgez instruction (32'd24)

Bgez $R2, 1

000110	00010	00001	0000000000000001

$R2 = 15 hex 21 dec 00000000000000000000000000010101 binary

If $R2 >= 0 branch 1*4+PC+4

Branch 32'd32 address

Regdest	branchgez	Regwrite	Memtoreg	Aluop
00	1	0	00	1001