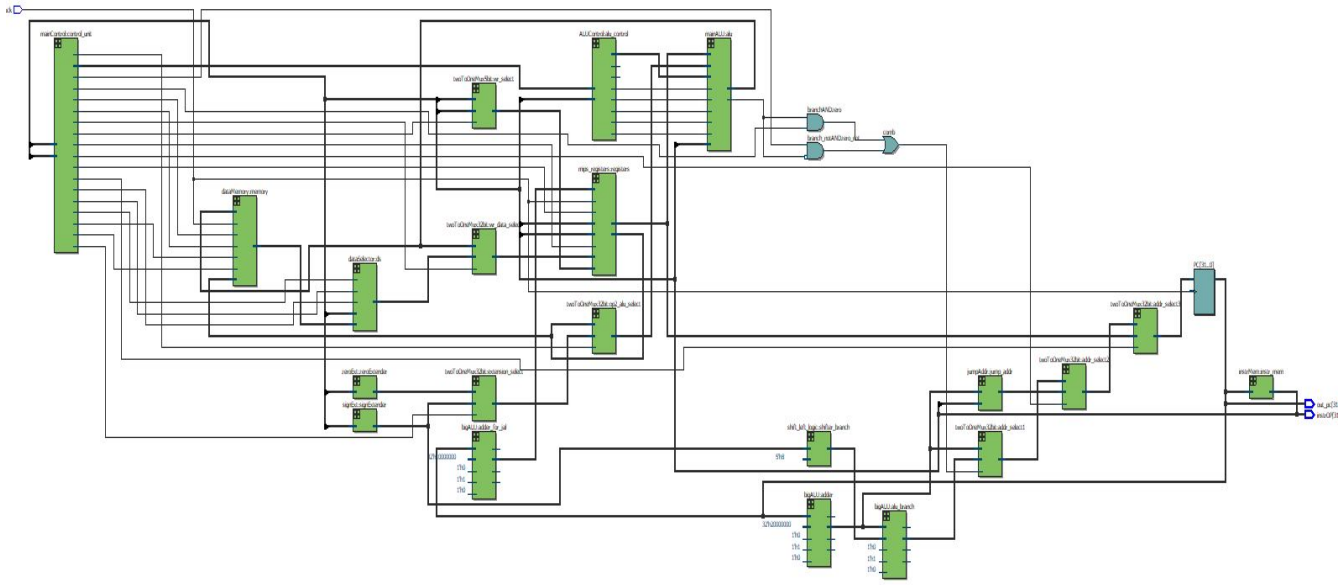
CSE 331
Computer Organization

Project 4 Report

Name / Surname : Göktuğ Ali Akın
Student ID : 161044018

Single cycle processor circuit diagram

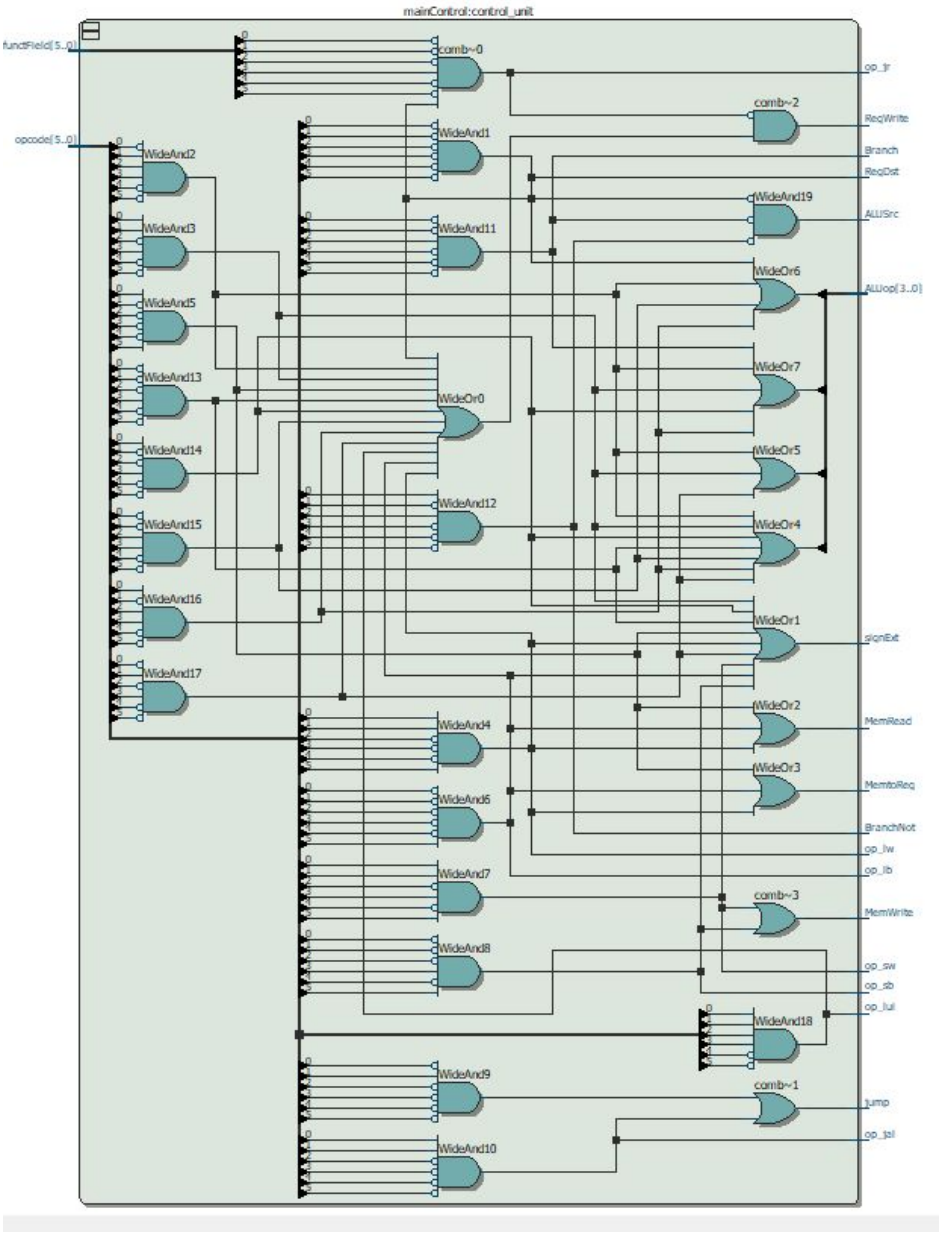


Notes : Some instructions assumed as same instructions:

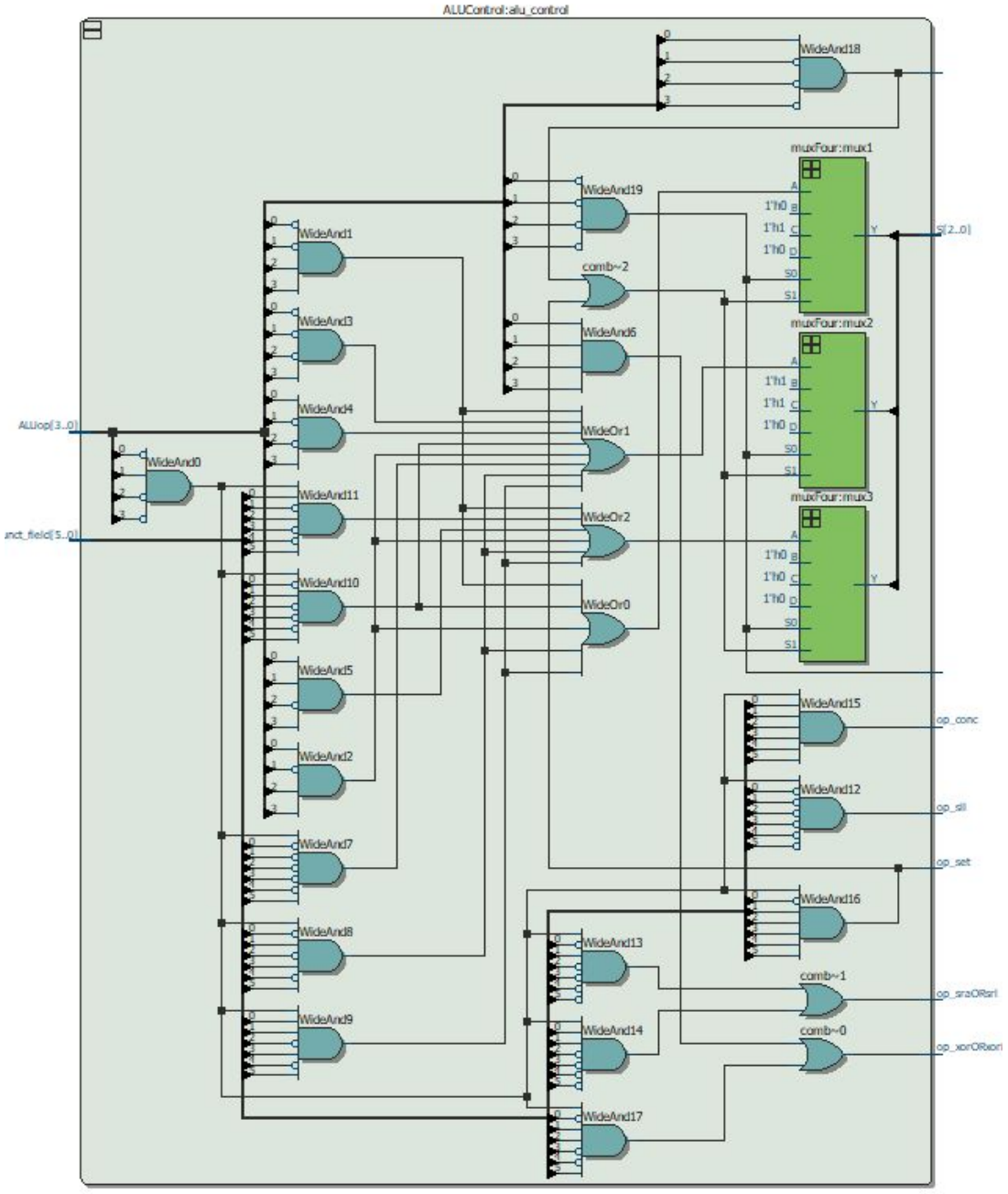
- slt & sltu considered as same, slti & sltiu considered as same.
- addi & addiu considered as same.
- srl & sra considered as same.

Modules :

1. **mainControl :** Control unit of the datapath.
   Circuit diagram:



2. **ALUControl :** Control module of ALU.
   Circuit diagram :


ALUControl:alu_control
WideAnd18
WideAnd19
WideAnd1
WideAnd3
WideAnd4
WideAnd0
WideAnd11
WideAnd10
WideAnd5
WideAnd2
WideAnd7
WideAnd8
WideAnd9
comb~2
WideAnd6
WideOr1
WideOr2
WideOr0
muxFour:mux1
muxFour:mux2
muxFour:mux3
S[2..0]
WideAnd15
WideAnd12
WideAnd16
WideAnd13
WideAnd14
WideAnd17
comb~1
comb~0
op_conc
op_sll
op_set
op_sraORsrl
op_xorORxori
ALUop[3..0]
inct_field[5..0]

Test cases

1.satir, op : R type, function : add
instruction : 00000000000000010001000000100000
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 00010 (2), register.mem de 6.satir
00000000000000000000000000000011
00000000000000000000000000000001
add result : 00000000000000000000000000000100
R[Rd] : 00000000000000000000000000000100
calisiyor.

---
2. satir, op R type, function : subb
instruction : 00000000000000010001100000100010
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 00011 (3), register.mem de 7.satir
00000000000000000000000000000011
00000000000000000000000000000001
sub result : 00000000000000000000000000000010
R[Rd] : 00000000000000000000000000000010
calisiyor.

---
3. satir, op R type, function : and
instruction : 00000000000000010010000000100100
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 00100 (4), register.mem de 8.satir
00000000000000000000000000000011
00000000000000000000000000000001
and : 00000000000000000000000000000001
R[Rd]: 00000000000000000000000000000001
calisiyor.

---
4. satir, op R type, function : or

instruction : 00000000000000010010100000100101
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 00101 (5), register.mem de 9.satir
00000000000000000000000000000011
00000000000000000000000000000001
or : 00000000000000000000000000000011
R[Rd]: 00000000000000000000000000000011
calisiyor.

---
5. satir, op R type, function : slt
instruction : 00000000000000010011000000 101010
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 00110 (6), register.mem de 10.satir

slt result : 00000000000000000000000000000000
R[Rd] : 00000000000000000000000000000000
calisiyor.

6.satir, op R type, function : xor
instruction : 00000000000000010011100000 100110
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 00111 (7), register.mem de 11. satir

xor result : 00000000000000000000000000000010
R[Rd] : 00000000000000000000000000000010
calisiyor.

7.satir, op R type, function : sll
instruction : 00000000000000010100000010000000
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 01000 (8), register.mem de 12. satir

sll result : 00000000000000000000000000001100
R[Rd] : 00000000000000000000000000001100
calisiyor.

---
8. satir, op R type, function : sra
instruction : 00000000000000010100100010000011
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 01001 (9), register.mem de 13. satir

sra result : 00000000000000000000000000000000
R[Rd]   : 00000000000000000000000000000000
calisiyor.

---
9. satir, op R type, function : srl
instruction : 00000000000000010101000010000010
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 01010 (10), register.mem de 14.satir

srl result : 00000000000000000000000000000000
R[Rd]   : 00000000000000000000000000000000
calisiyor.

---
10. satir, op R type, function : sltu
instruction : 00000000000000010101100010101011
Rs : 00000 (0), R[Rs] : 00000000000000000000000000000011
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001

Rd : 01011 (11), register.mem de 15. satir
sltu result : 00000000000000000000000000000000
R[Rd]   : 00000000000000000000000000000000
calisiyor. (slt ve sltu ayni kabul edildi.)

---
11. satır, op : xori
instruction : 00111000001011010000000000101011

zero ext imm : 00000000000000000000000000101011
rs = 00001, (0), R[Rs] = 00000000000000000000000000000001
rt = 01101, (13), register.mem de 17. satir

xor result : 00000000000000000000000000101010
R[rt] : 00000000000000000000000000101010
calisiyor.

---
12. satir, op sltiu
instruction : 00101100001011100000000000000000

sign ext imm : 00000000000000000000000000000000
rs = 00001, (0), R[Rs] = 00000000000000000000000000000001
rt = 01110, (14), register.mem de 18. satir

sltiu result : 00000000000000000000000000000000
R[Rt] : 00000000000000000000000000000000
calisiyor.

---
13. satir, op : lw
instruction : 10001100001011110000000000000010

zero ext imm : 00000000000000000000000000000010
rs : 00001, (1), R[Rs] = 00000000000000000000000000000001
rt : 01111, (15), register.mem de 19. satir
Mem[R[Rs] + signExtImm] = 00000000000000000000000000011001
R[Rt] = 00000000000000000000000000011001
calisiyor.

---
14. satir, op : sw
instruction : 10101100001000010000000000000011

sign ext imm : 00000000000000000000000000000011
rs : 00001, (1), R[Rs] = 00000000000000000000000000000001
rt : 00001, (1), R[Rt] = 00000000000000000000000000000001
addres: 00000000000000000000000000000100 (4)
R[Rt] = 00000000000000000000000000000001
Mem[R[Rs] + signExtImm] = 00000000000000000000000000000001
calisiyor.

---
15. satir, op : addi
instruction : 00100000001100000000000000001001

rs : 00001, (1), R[Rs] = 00000000000000000000000000000001
signExtImm : 00000000000000000000000000001001
rt : 10000, (16), register.mem de 20. satir

addi result = 00000000000000000000000000001010
R[rt] = 00000000000000000000000000001010
calisiyor.

---
16. satir, op : addiu
instruction : 00100100001100010000000000001101

rs : 00001, (1), R[Rs] = 00000000000000000000000000000001
signExtImm : 00000000000000000000000000001101
rt : 10001, (17), register.mem de 21.satir

addi result = 00000000000000000000000000001110
R[rt] = 00000000000000000000000000001110
calisiyor.

---
17. satir, op : andi
instruction : 00110000001100100000000000000000

rs : 00001, (1), R[Rs] = 00000000000000000000000000000001
zeroExtImm = 00000000000000000000000000000000

rt : 10010, (18), register.mem de 22.satir

andi result = 00000000000000000000000000000000
R[Rt] = 00000000000000000000000000000000
calisiyor.

---
18. satir, op : ori
instruction : 00110100001100111111111111111111
rs : 00001, (1), R[Rs] = 00000000000000000000000000000001
zeroExtImm = 00000000000000001111111111111111

rt : 10011, (19), register.mem de 23.satir

ori result = 00000000000000001111111111111111
R[Rt]   = 00000000000000001111111111111111
calisiyor.

---
19. satir, op : slti
instruction : 00101000001101000000000000000000
rs : 00001, (1), R[Rs] = 00000000000000000000000000000001
signExtImm   = 00000000000000000000000000000000

rt : 10100, (20), register.mem de 24. satir

slti result = 00000000000000000000000000000000
R[Rt]   = 00000000000000000000000000000000
calisiyor.

---
20. satir, op : set (kendi ekledigim instruction 1)
instruction : 000000 00001 00100 10101 00000 111110
Rs = 00001, (1) , R[Rs] = 00000000000000000000000000000001
Rt = 00100, (4) , R[Rt] = 00000000000000000000000000000001
Rd = 10101, (21), register.mem de 25. satir

set result = 00000000000000000000000000000001
R[Rd]   = 00000000000000000000000000000001
calisiyor.

---
21. satir, op : conc (kendi ekledigim instruction 2) (concatenate)
instruction : 000000 00000 00001 10110 00000 111111
Rs = 00000, (0), R[Rs] = 00000000000000000000000000000011
Rt = 00001, (1), R[Rt] = 00000000000000000000000000000001
Rd = 10110, (22), register.mem de 26. satir

conc result = 00000000000000110000000000000001
R[Rd]   = 00000000000000110000000000000001
calisiyor.

---
22. satir, op : lb
instruction : 100000 00001 10111 0000000000000111

Rs = 00001, R[Rs] = 00000000000000000000000000000001
zeroExtImm = 00000000000000000000000000000111
addr = 00000000000000000000000000001000
{24'b0,M[R[Rs] + zeroExtImm](7:0)}

rt = 10111, (23), register.memde 27.satir
M[R[Rs] + zeroExtImm] = 00000000000000000000001110001000
MSB(7:0) = 11001101
res = 00000000000000000000000010001000
R[rt] = 00000000000000000000000010001000
calisiyor.

---
23. satir, op : lh
instruction : 100001 00001 11000 0000000000000110
Rs = 00001, R[Rs] = 00000000000000000000000000000001
zeroExtImm = 00000000000000000000000000000110
addr = 00000000000000000000000000000111

rt = 11000, (24), register.mem 28. satir
M[R[Rs] + zeroExtImm] = 00000000000000000000000000000111
MSB(15:0) = 0000000000000111

res = 00000000000000000000000000000111
R[rt] = 00000000000000000000000000000111
calisiyor.

---
24. satir, op : sb
instruction : 101000 00001 00000 0000000000010100
Rs = 00001, R[Rs] = 00000000000000000000000000000001
signExtImm = 00000000000000000000000000010100
addr = 00000000000000000000000000010101

rt = 00000 , (0) , register.mem de 4.satir
R[rt] = 00000000000000000000000000000011
R[rt](7:0) = 00000011
M[R[rs] + signExt Imm] = 11110000000000000000000000000011
M[R[rs] + signExt Imm](7:0) = 00000011
calisiyor.

---

25. satir, op : beq
instruction : 000100 00000 00101 0000000000000001

Rs = 00000, (0) , R[Rs] = 00000000000000000000000000000011
Rt = 00101, (5) , R[Rt] = 00000000000000000000000000000011
it will branched.

pc = 24 = 00000000000000000000000001100000 (25.satir)
= 00000000000000000000000000000100
pc + 4 = 00000000000000000000000001100100
shift sign ext = 00000000000000000000000000000100 (signExtended)

branch addr = 00000000000000000000000001101000 ===> 26 ==> it means jump 27. instruction.
instruction.mem de 27. satira atla, 26 yi calistirma.

26. satirdaki instruction :
000000 11101 00001 11110 00000100101, R type, or operation.

Rs : 11101 (29), R[Rs] : 11111111111111111111111111111111
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 11110 (30), register.mem de 34.satir

eger 27. satira(instruction'a) branch ederse, 26 calismayacaktir.
26 calismayacaksa, R[30] != 11111111111111111111111111111111 olmalidir.(rd = 30)
R[30] 32 bit 1 den olusursa, branch etmemistir.

27. satirdaki instruction :
000000 11101 11100 11100 00000100101, R type, or operation.
Rs : 11101 (29), R[Rs] : 11111111111111111111111111111111
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 11100 (28), register.mem deki 32. satir

eger 26 calismayip 27 calisirsa, 28. register 32 bit 1 den olusamlidir.

Test :
R[30] after branch execution : 00000000000000000000000000000000
R[28] after branch execution : 11111111111111111111111111111111

calisiyor.
ekrana bastirilan 32 bitlik instructionlardan da branch islemi takip edilebilir.

28.satir, op : bne
instruction : 000101 00000 000001 0000000000000001
Rs = 00000, (0), R[Rs] = 00000000000000000000000000000011
Rt = 00001, (1), R[Rt] = 00000000000000000000000000000001
rs rt esit degil, branch etmeli.

pc = 27   = 00000000000000000000000001101100
           00000000000000000000000000000100
pc + 4   = 00000000000000000000000001110000
signExtImm = 00000000000000000000000000000100 (shifted)
branch addr= 00000000000000000000000001110100 ==> 29 ==> 30. instruction,
30. satira branch etmeli.

29. satirdaki (29. instruction) calismayacaktir.
29. satirdaki instruction :
000000 11101 00001 11011 00000100101, R type, or operation.

Rs : 11101 (29), R[Rs] : 11111111111111111111111111111111
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 11011 (27), register.mem de 31.satir

eger 29. instruction calisirsa, Reg[27] 32 bit 1 den olusmalidir.

30. satirdaki (30. instruction) calisacaktir. (28 den buraya atlayacaktir.)
30. satirdaki instruction :
000000 11101 00001 11010 00000100101, R type, or operation.

Rs : 11101 (29), R[Rs] : 11111111111111111111111111111111
Rt : 00001 (1), R[Rt] : 00000000000000000000000000000001
Rd : 11010 (26), register.mem de 30.satir

eger 30. instruction calisirsa, Reg[26] 32 bit 1 den olusmalidir.
--
sonuc olarak, reg[26] 32 bit 1 ise, reg[27] degismiyor ise, branch etmistir, 29 u atlayıp 30 a gecmistir.
test ;
reg[26] = 11111111111111111111111111111111
reg[27] = 00000000000000000000000000000000
calisiyor.

---
31. satir, op : jr
instruction :
000000 11001 00000 0000000000001000
000000 11001 00000 0000000000001000
Rs = 11001, (25) , R[Rs] = 00000000000000000000000010000100, register.mem de 29. satir

PC = R[Rs] ==> PC 33 olmali, 34. satirdan devam etmeli.

32. ve 33. instruction : op : sw
101011 00001 00000 0000000000000000

Rs : 00001 (1), R[Rs] : 00000000000000000000000000000001
Rt : 00000 (0), R[Rt] : 00000000000000000000000000000011

addr = 00000000000000000000000000000001
M[1] = R[rt] olmali, M[1] (5.satir data.memde)
00000000000000000000000000000011 olmali.
Eger 32 ve 33 calisirsa m[1] degismelidir. degismezse, calismamistir.

34. satirdaki instruction, jump edecegi instruction.
instruction : 101011 01011 00000 0000000000000000
Rs : 01011 (1) R[Rs] = 00000000000000000000000000000000
Rt : 00000 (0) R[Rt] = 00000000000000000000000000000011

addr = 00000000000000000000000000000000
M[0] = R[rt] olmali, M[0] (4.satir data.mem de)00000000000000000000000000000001 olmali.
ozer;
eger jump ederse, 32 ve 33 calismayacagindan m[1]
00000000000000000000000000000011 olmamalidir.
34. instr calisacagindan m[0] 00000000000000000000000000000011 olmalidir.

test ;
m[0] : 00000000000000000000000000000011
m[1] : 00000000000000000000000000000000
calisiyor.

---
35. satir, op : jal
instruction : 000011 00000000000000000000100101

38. satira jump yapmasi gerekiyor. 36 ve 37 ye nop instructionlar ekledim, egeer bunlari
36. ve 37. instruction ;
11111111111111111111111111111111
11111111111111111111111111111111
ekrana basmazsa testbencthe, bunlari atlamistir.

38. satir; op : sw
101011 00000 00001 0000000000000010
Rs = 00000, (0) , 00000000000000000000000000000011
signExtImm = 00000000000000000000000000000010
addr = 00000000000000000000000000000101 (5)

Rt = 00001, (1) , R[Rt] = 00000000000000000000000000000001
M[5] = 00000000000000000000000000000001 olmali. (data.mem 9. satir)
calisiyor.

ayni zamanda, pc + 8 i R[31] e kaydetmeli.
R[31] = 00000000000000000000000010010000
kaydedilmis.

---
j instructionu test edilmedi, jal ile ayni, R[31] e kaydetmek disinda.

