CSE 331

HW3-RAPOR

Moduller:

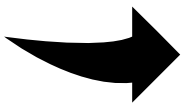
-hw2 modullerine ek olarak hw3'de 3 yeni modül var.Bunların isimleri mips_32, mips_register ve control_unit.

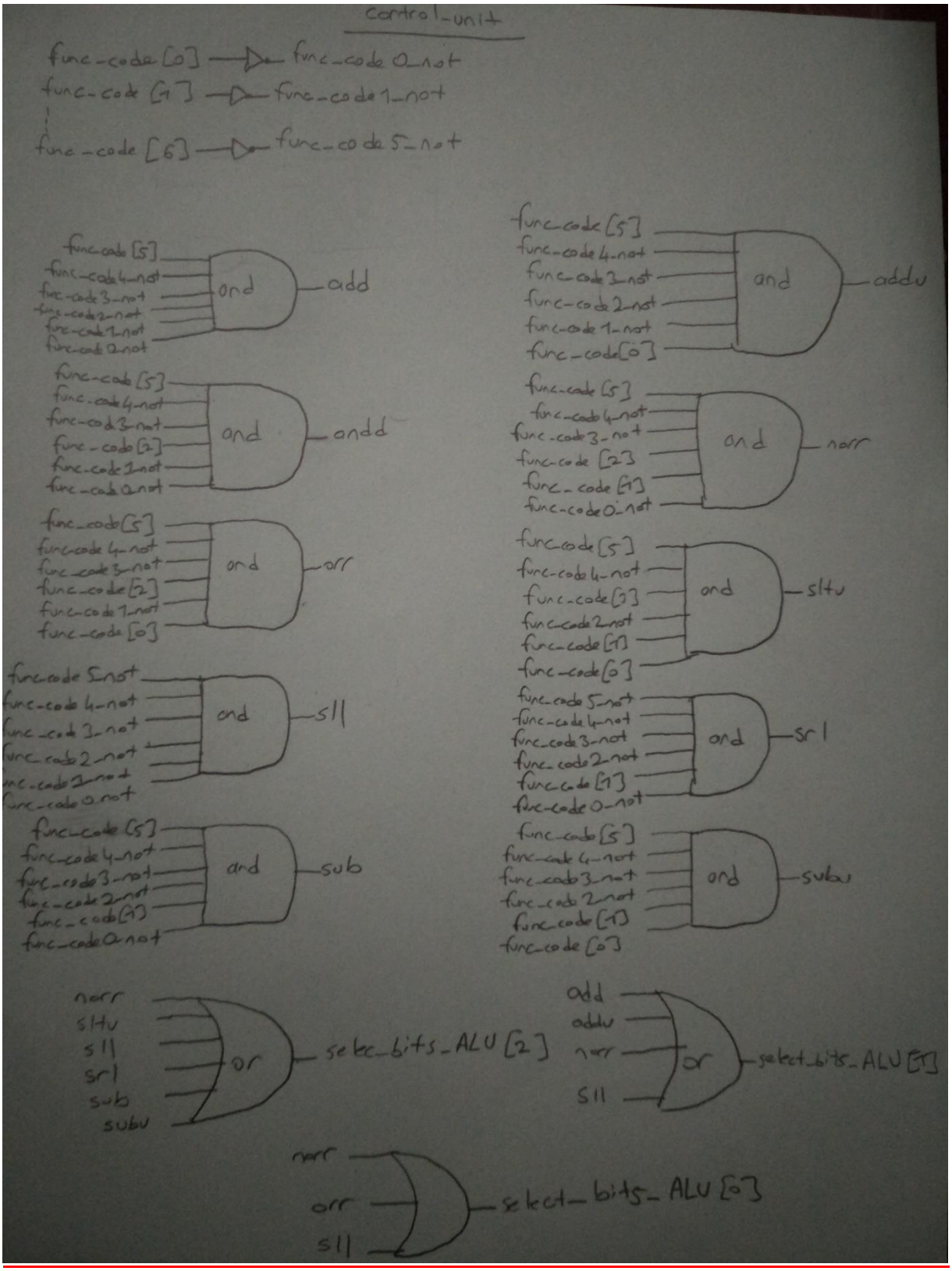
-mips_32 modulü gelişmiş bir alu32(hw2'de) modülü olarak düşünülebilir.Farklı olarak parametrelerini register olarak alır .mips_register ve control_unit modülleri bu modülün içinde kullanılır yani çağırılır.rs,rt ve rt,shamt için birer mux vardır.Bu muxların sonuçları ile control_unit çıktısı olarak gelen select bitlerine göre alu32 modulü çağırılır.Eğer işlem sltu ise alu32 nin sonucunun most significant bitinin başına 31 0 eklenerek yeni bir 32 bitlik değer elde edilir ve bu sonuç result olur aksi halde result olarak alu32 nin sonucu direk döndürülür.

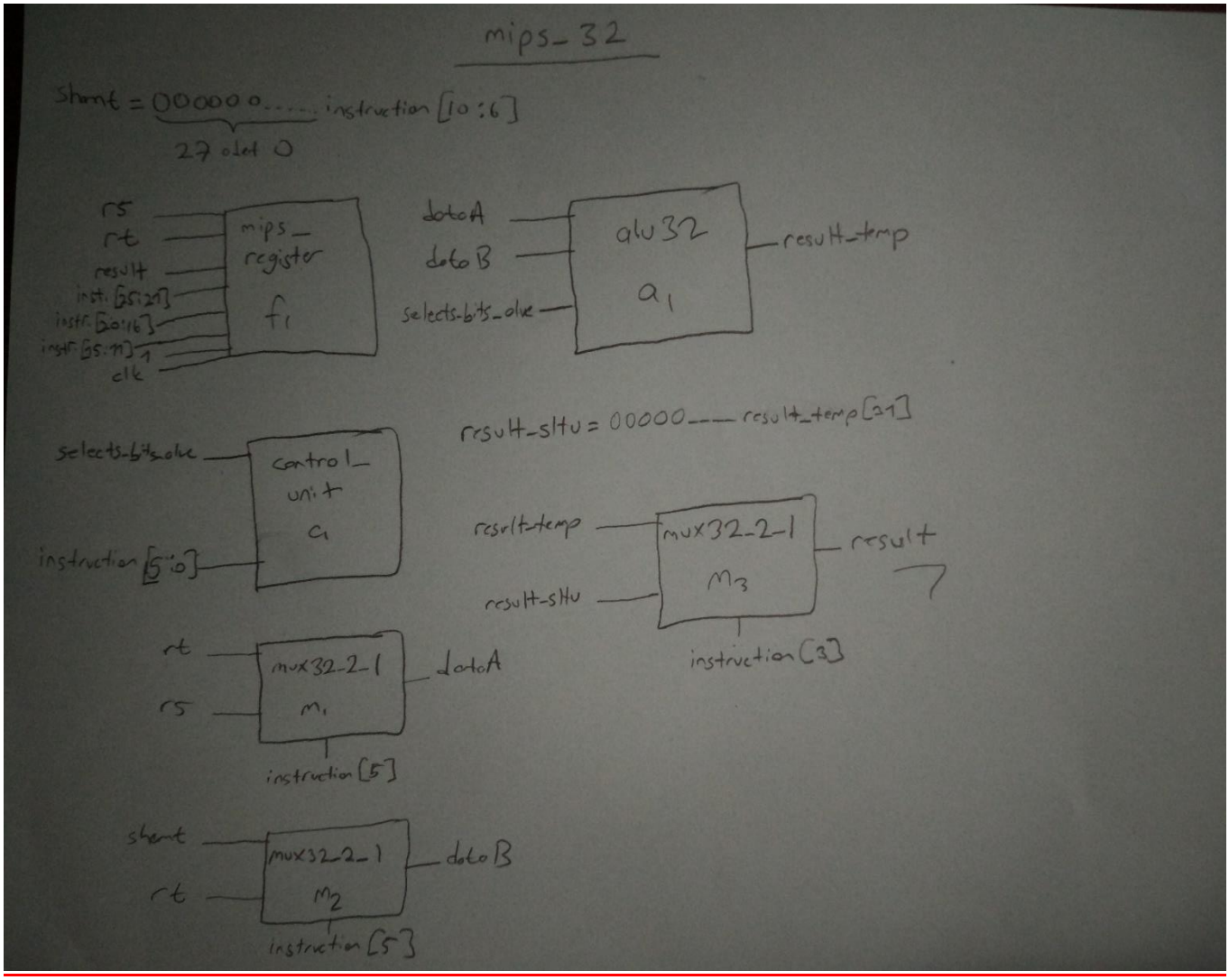
-mips_register modulü registers.mem dosyasından register datayı okur ve içerideki reg 32x32 lik registers arrayine atar.signal_reg_write sinyali 1 write_reg 0 olmadığı sürece yani 0. Registere yazılmak istenmediği sürece write_data ilgili register adresine yazılır ve register arrayi aynı dosyaya yazılır.(Bu modulün şeması çizilmemiştir çünkü behaviroal kullanılmıştır).

-control_unit modulünde function_code'a göre 3 bitlik bir sinyal üretilir.Bu sinyal daha sonra alu32'de opcode olarak kullanılacaktır.

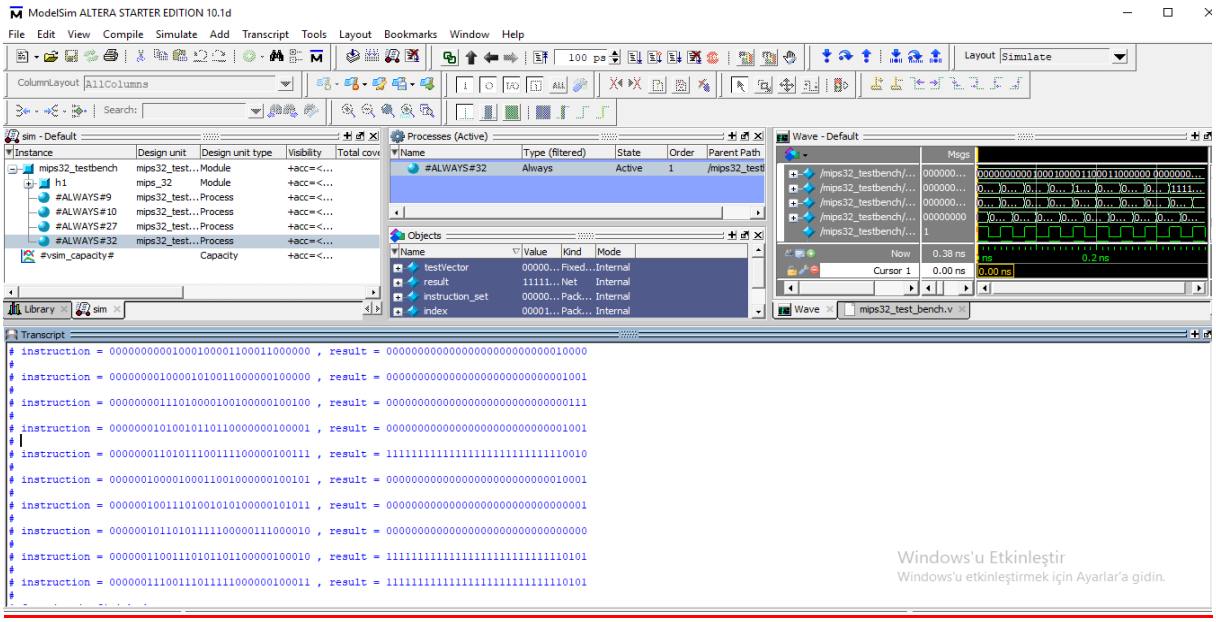
Modul çizimleri:



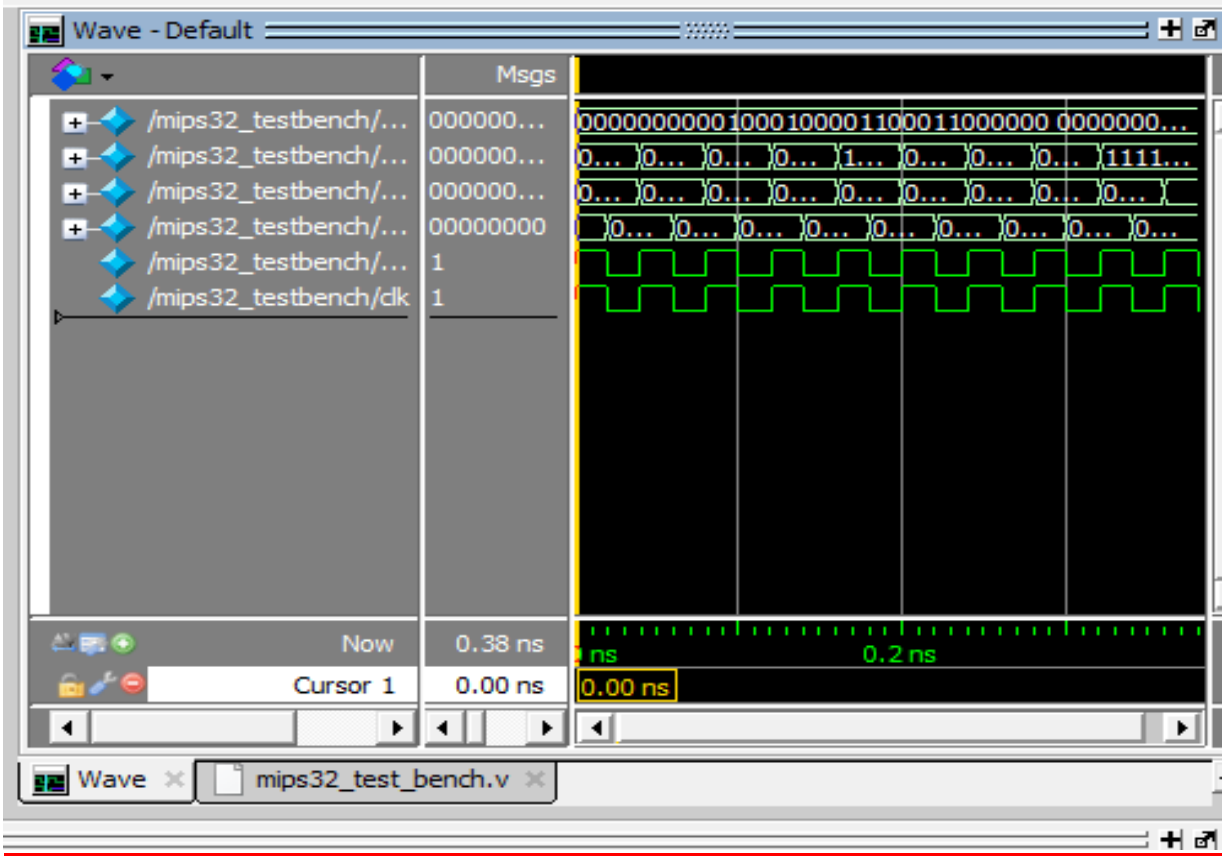

control_unit
func_code[0] func_code 0_not
func_code[1] func_code 1_not
func_code[6] func_code 5_not
func_code[5]
func_code 4_not
func_code 3_not
func_code 2_not
func_code 1_not
func_code 0_not
and
add
addu
andd
orr
slt
sll
srl
sub
subu
nor
or
select_bits_ALU[2]
select_bits_ALU[1]
select_bits_ALU[0]



Simülasyon sonuçları:



Sinyallerin yakın hali:



Konsolun yakınlaşmış hali:

```
# instruction = 00000000001000100001100011000000 , result = 00000000000000000000000000010000
#
# instruction = 00000000100001010011000000100000 , result = 00000000000000000000000000001001
#
# instruction = 00000000111010000100100000100100 , result = 00000000000000000000000000000111
#
# instruction = 00000001010010110110000000100001 , result = 00000000000000000000000000001001
#
# instruction = 00000001101011100111100000100111 , result = 11111111111111111111111111110010
#
# instruction = 00000010000100011001000000100101 , result = 00000000000000000000000000010001
#
# instruction = 00000010011101001010100000101011 , result = 00000000000000000000000000000001
#
# instruction = 00000010110101111100000111000010 , result = 00000000000000000000000000000000
#
# instruction = 00000011001110101101100000100010 , result = 11111111111111111111111111110101
#
# instruction = 00000011100111011111000000100011 , result = 11111111111111111111111111110101
#
# function is finished
#
# ** Note: $finish    : C:/Users/ibrah/Desktop/organizasyon proje 1/mips32_test_bench.v(38)
#    Time: 380 ps  Iteration: 1  Instance: /mips32_testbench
```

İnstruction listesi ve register listesi:

instruction_list.txt - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

```
//10 instructions with their definition.


//shift left logical instruction rs=1.reg ,rt=2.reg ,rd=3.reg  shamt=3
00000000001000100001100011000000
//add instruction rs=4.reg, rt=5.reg, rd=6.reg, shamt=x
00000000100001010011000000100000
//and instruction rs=7.reg, rt=8.reg, rd=9.reg, shamt=x
00000000111010000100100000100100
//addu instrouction rs=10.reg, rt=11.reg, rd=12.reg, shamt=x
00000001010010110110000000100001
//nor instruction rs=13.reg, rt=14.reg, rd=15.reg, shamt=x
00000001101011100111100000100111
//or instruction rs=16.reg, rt=17.reg, rd=18.reg, shamt=x
00000010000100011001000000100101
//sltu instruction rs=19.reg, rt=20.reg, rd=21.reg shamt=x
00000010011101001010100000101011
//shift right logical instruction rs=22.reg, rt=23.reg, rd=24.reg shamt=7
00000010110101111100000111000010
//sub instruction rs=25.reg, rt=26.reg, rd=27.reg shampt=x
00000011001110101101100000100010
//subu instruction rs=28.reg, rt=29.reg, rd=30.reg shamt=x
00000011100111011111100000100011
```

registers.mem - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

```
// memory data file (do not edit the following line - required for mem load use)
// instance=/mips32_testbench/h1/f1/registers
// format=bin addressradix=h dataradix=b version=1.0 wordsperline=1 noaddress
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000000010
00000000000000000000000000010000
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000001001
00000000000000000000000000000111
00000000000000000000000000001111
00000000000000000000000000000111
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000001001
00000000000000000000000000001001
00000000000000000000000000001100
11111111111111111111111111110010
00000000000000000000000000010000
00000000000000000000000000010001
00000000000000000000000000010001
00000000000000000000000000010011
00000000000000000000000000010100
00000000000000000000000000000001
00000000000000000000000000010110
00000000000000000000000000010111
00000000000000000000000000000000
00000000000000000000000000001111
00000000000000000000000000011010
11111111111111111111111111110101
00000000000000000000000000001111
00000000000000000000000000011010
11111111111111111111111111110101
00000000000000000000000000011111
```

-