

ASSIGNMENT 4

COMPUTER ORGANISATION

EMRE ÖZGÜN

161044103

Bu ödevde geçen R-type işlemleri yapan mimariye ek olarak I-type işlemlerden;

- 1)andi
- 2)ori
- 3)addiu
- 4)lw
- 5)sw
- 6)beq
- 7)j

yapan mimariyi tasarladık.Bu işlemlerin datapath içerisinde gerçekleştirilmesi için memory ,register ve Pc counter bloğunu oluşturdum. **NextPc** modülünde normal işlemler için $PC + 4$ işlemini gerçekleştirirken,beq instructionunda $PC + 4 + \text{targetaddress}$,J instructionunda $PC = \text{targetAdress}$ işlemini gerçekleştirdim.

Memory bloğuda I-type instructionları için değerlerin memoryye yazıldığı veya okunduğu bölümdür. Özellikle gerçek mimaride olduğu gibi lw ve sw işlemleri bu blok ile bağlantılıdır.

Neredeyse ödevde verilen tüm instructionların işlemlerini gerçekleştirmesi için gerekli olan **signExtend**(instructionun ilk 16 bitini alıp 32 bit olarak çıkaran) modülünü kullandım.

addiu : $R[rt] = R[rs] + \text{SignExtImm}$

andi: $R[rt] = R[rs] \& \text{ZeroExtImm}$

beq : if($R[rs] == R[rt]$)
 $PC = PC + 4 + \text{BranchAddr}$

lw: $R[rt] = M[R[rs] + \text{SignExtImm}]$

sw : $M[R[rs] + \text{SignExtImm}] = R[rt]$

ori: $R[rt] = R[rs] | \text{ZeroExtImm}$

j: $PC = \text{JumpAddr}$

Control bloğundan gelen sinyal bitleri:Tüm instuctionlar için yani 7 adet sinyal biti eklendi ve bunlara ek olarak MemRead, MemtoReg, MemRead,

Branch, RegWrite, MemWrite, RegDest sinyal bitleri mimariye eklenmiştir.).

Bu sinyal bitlerinin kontrolü ve doğru sonuç için 32bitlik sonuç veren 2x1 muxlar kullanılmıştır.

Oluşturduğum mimarinin datapath şekli aşağıdadır:

SİMULASYONDA DENEME YAPTIĞIM INPUTLAR

001101000010001000000000000001101	ORI
001100000110010000000000000001100	ANDI
0010010000100010000000000000000101	ADDI
	U
101011000000001000000000000000000	SW
100011000000001000000000000000000	LW
000010000000000000000000000001000	J
0010010000100010000000000000000101	ADDI
	U
101011000000001000000000000000000	SW
100011000000001000000000000000000	LW
00010000000000001111111111111100	BEQ

--->-->-->-->>>>Çıktı sonuçları bir sonraki sayfadadır.

ÇIKTI SONUÇLARI

```
# pc =          x
#
# opcode = xxxxxx, rs = xxxxx, rt = xxxxx, rd = xxxxx, shamt = xxxxx, function = xxxxxx, alusel = xxx
# result = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
# pc =          4
#
# opcode = 000000, rs = 00001, rt = 00010, rd = 00011, shamt = 00000, function = 100101, alusel = 001
# result = 00000000000000000000000000000011
# reg_write [00011] = 000000000000000000000000000011
# pc =          8
#
# opcode = 101011, rs = 00000, rt = 00011, rd = 00000, shamt = 00000, function = 001100, alusel = 010
# result = 00000000000000000000000000001100
# memwrite [0000000c] = 00000003
#
# pc =          12
#
# opcode = 000000, rs = 00001, rt = 00000, rd = 00011, shamt = 00000, function = 100100, alusel = 000
# result = 00000000000000000000000000000000
# readdata [0000000c] = 00000003
#
# pc =          16
#
# opcode = 100011, rs = 00000, rt = 00011, rd = 00000, shamt = 00000, function = 001100, alusel = 010
# result = 00000000000000000000000000000011
# reg_write [00011] = 000000000000000000000000000011
# pc =          0
#
# opcode = 000010, rs = 00000, rt = 00000, rd = 00000, shamt = 00000, function = 000000, alusel = 010
# result = 00000000000000000000000000000000
# pc =          0
#
# opcode = 000000, rs = 00001, rt = 00010, rd = 00011, shamt = 00000, function = 100101, alusel = 001
# result = 00000000000000000000000000000011
# reg_write [00011] = 000000000000000000000000000011
# pc =          4
#
# opcode = 000000, rs = 00001, rt = 00010, rd = 00011, shamt = 00000, function = 100101, alusel = 001
# result = 00000000000000000000000000000011
# reg_write [00011] = 000000000000000000000000000011
# pc =          8
#
# opcode = 101011, rs = 00000, rt = 00011, rd = 00000, shamt = 00000, function = 001100, alusel = 010
# result = 000000000000000000000000000001100
#
#      8 tests completed.
#
# ** Note: $finish      : D:/Home/Downloads/161044103/mips32_single_cycle_testbench.v(43)
#      Time: 80 ps  Iteration: 1  Instance: /mips32_single_cycle_testbench
# 1
# Break in Module mips32 single cycle testbench at D:/Home/Downloads/161044103/mips32 single cycle testbench.v line
```



