

**Gebze Technical University
Computer Engineering**

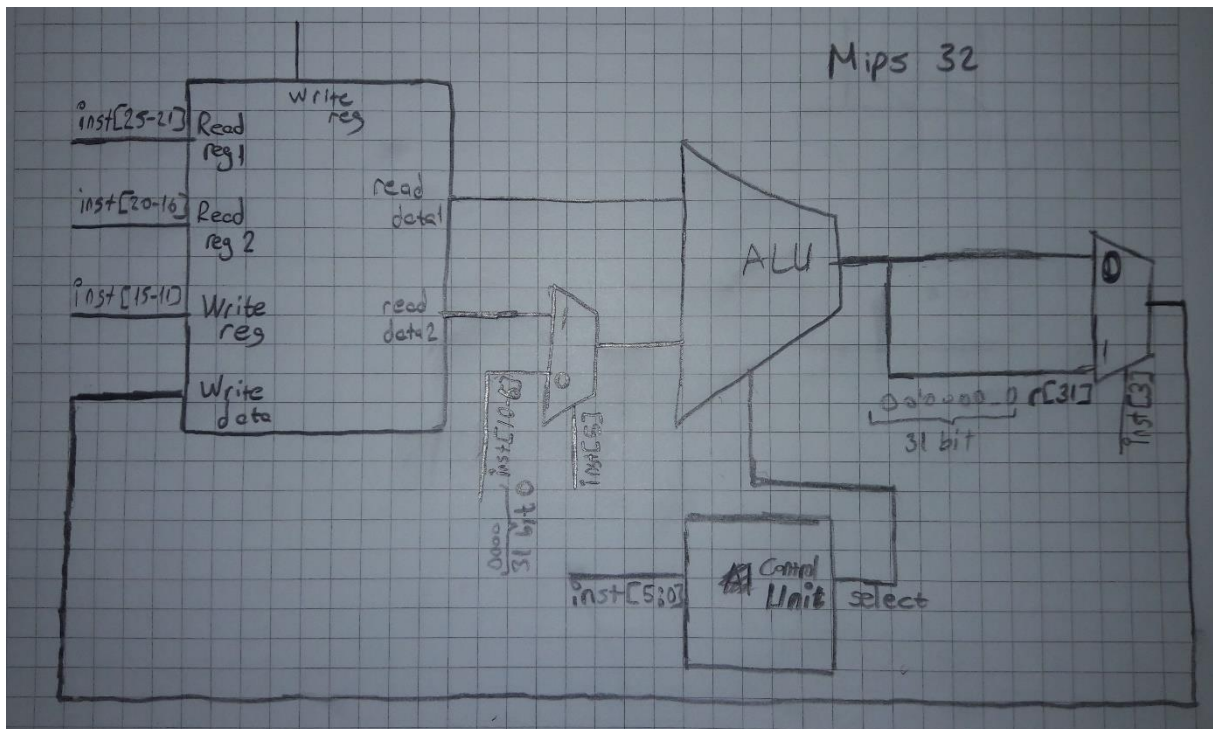
CSE 331 - 2018

ASSIGNMENT 3 REPORT

**CELAL CAN KAYA
161044014**

Course Assistant: Fatma Nur Esirci

ŞEMATİK TASARIMLAR

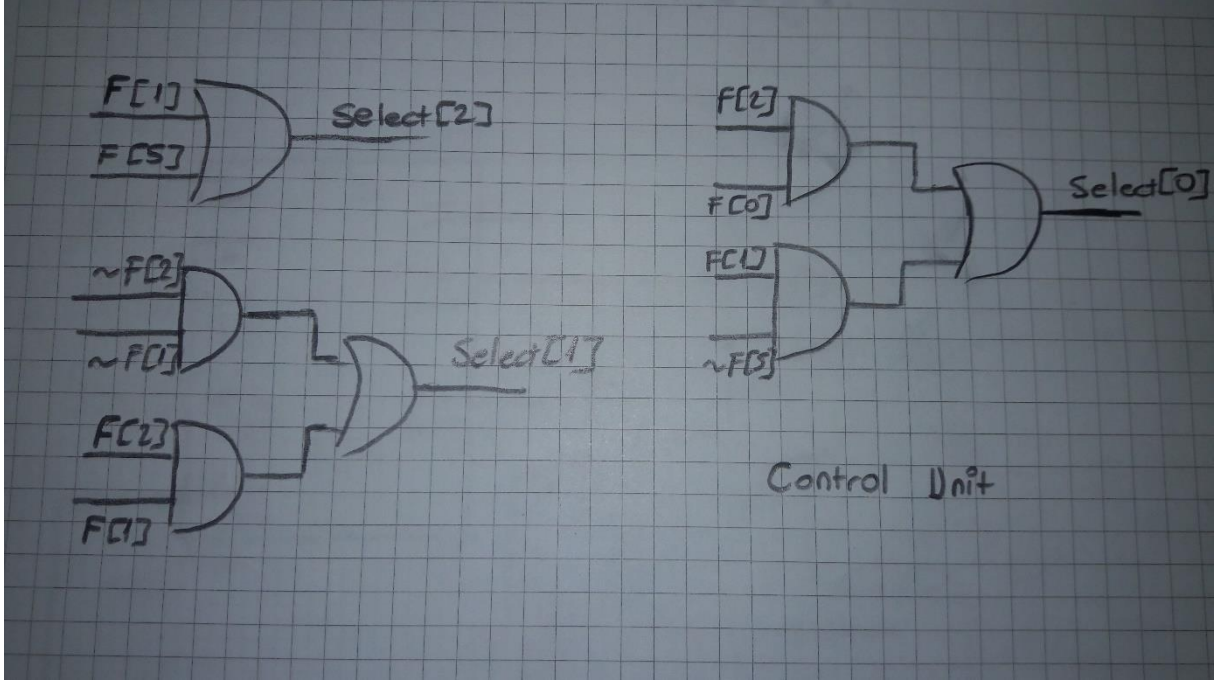


Şekil 1- Mips32

	F5	F4	F3	F2	F1	F0	Operation	Select
add	1	0	0	0	0	0	add	010
addu	1	0	0	0	0	1	add	010
sub	1	0	0	0	1	0	sub	100
subu	1	0	0	0	1	1	subu	100
and	1	0	0	1	0	0	and	000
or	1	0	0	1	0	1	or	001
nor	1	0	0	1	1	1	nor	111
sll	0	0	0	0	0	0	sll	110
srl	0	0	0	0	1	0	srl	101
sllv	0	0	1	0	1	1	sllv	100

$\text{Select} \langle 0 \rangle = (F2 \& F0) + (!F5 \& F1)$
 $\text{Select} \langle 1 \rangle = (!F2 \& !F1) + (F2 \& F1) + (F5 \& F1)$
 $\text{Select} \langle 2 \rangle = F1 + !F5 = !F5$

Şekil 2- Control Unit Doğruluk Tablosu



Şekil 3- Control Unit

VERİLOG MODÜLLERİ

Mips32 – Top level modülüm. Register, alu, control unit ve 32Bitlik 2:1 Muxlar kullanılarak yapılmış modülüm. Muxları instruction shift veya sltu olduğunda doğru şekilde write data'ya aktarmak için kullanıyorum.

Control_unit – 5 Bitlik Function code'un doğruluk tablosunu kullanarak Select bitinin hangi değişkenlere bağlı olduğunu bulup formülünü çıkardım. Daha sonra o formülleri kullanarak alu için select bitlerini üreterek Alu'ya gönderdim.

Mips_registers – registers.mem dosyasındaki registerları "registers" in içine atıyorum. Daha sonraki işlemler yapıldıktan sonra gelen Write data'yı Regwrite == 1 olduğu zaman Rd registerına yazıyorum.

Alu32 – 8:1 Mux Kullanarak seçilmiş olan operasyonu output olarak veriyorum.

_2mux – 1 Bitlik 2:1 Mux

_32Bit_2mux – 32 Tane 2:1 Mux Kullanılarak yapılmış 32 Bitlik 2:1 Mux

_32Bit_4mux – 3 Tane 32 Bitlik 2:1 Mux Kullanılarak yapılmış 32 Bitlik 4:1 Mux

_32Bit_8mux – 2 Tane 32 Bitlik 4:1 Mux ve 1 Tane 32 Bitlik 2:1 Mux Kullanılarak yapılmış 8:1 Mux. Alu'da select bitine göre hangi işlemin yapılacağını seçmek için yazdım.

_32Bit_and – 32 Bitlik And

_32Bit_or – 32 Bitlik Or

_32Bit_xor – 32 Bitlik Xor

_32Bit_nor – 32 Bitlik Nor

[illegible]

Tüm modüllerim testlerime göre düzgünce çalışıyor. Sonuçları “registers.mem” dosyasına tekrar yazdım. Gelen inputlar sürekli olarak değişmemesi için her instruction’un sonucunu sırayla 32. Registerdan itibaren geriye doğru yazdırdım. Inputlar olarakta genel olarak “registers.mem” dosyasındaki ilk registerları kullandım. “registers.mem” dosyasının içeriğini değiştirmeden yolladım. Modelsim üzerinden çalıştırdıktan sonra tekrar “registers.mem” dosyasına bakarak değişimleri görebilirsiniz.