# Gebze Technical University Computer Engineering CSE 331

### **ASSIGNMENT 4 REPORT**

FERHAT ÇELİK 151044014

Course Assistant: Fatma Nur Esirci



	R		I		J	
	op es et	rd showt func	op rs r	t add/im	op to	rget oddress
_	6 5 5	5 5 6		5 16	6	26
	, , ,		,			
19		, add, sub, and	المرار مرارية عاد راموx		i, jal,	
	95,500,50	1,511,5160	sur, sto, beq, bre, addi			
14			addin , and i jost , s. Hi			
	200000	func degisir	lut		_	
		agg 100000				
1.5						
14	ond	000000	andi	000001	j 0	10001
	٥٢	000001	irc	000010	-	10010
	Xoc	000010	xorī	000011	fisht 1	1 111 1 > skip next
	add	0 0 0 0 1 1	addi	000100		111241
	sub	0 00100	addiu	0,00101		ALu
14	slt	000101	slti	000110	and	000
	sra	000110	sltīu	000111	or	001
	srl	000111	lw	100011	add	010
	sll	0 0 1 0 0 0	lh	100001	Xoc	old gnew
	sltu	001001	16	100000	shift r	100 → new
	jr ,	001010	luí	001111	shiftl	101 -> new
-			sω	101011	Sub	110
			sb	101000	SIE	111
			beq	001110	l	
	Marine Propulsion Systems					
10			Celik	111110	~ 13 (59	nim ise is xor imm

Transmissions - Hybrid Ready Transmissions - Control Systems - Joystick Maneuvering Systems
Fixed Pitch Propellers - SPP - Sail Drive Systems - Surface Drive Systems - FPP and CPP Tunnel Thrusters

Well Mounted / Deck Mounted / Stem Mounted / Retractable / Contra Rotation / Shallow Draught Azimuth Thrusters

#### Eklenen yeni 2 instruction:

frht (j type) → 1 instruction ileri atlar. Yani bir sonraki instruction'ı atlayarak sonrakine geçer.

Celik (i type) → eğer rs signimm değerinden küçükse rs ile signimm değerini xorlar.Değilse 0 döner.

#### Yeni eklenen modüller

İkinci ödevde yaptığım 32 bitlik alu modülünü kullandım. Ancak xor ve shift modüllerini de ekledim.

Alucontrol → r type ise function code'a bakarak i type ise opcode'a göre aluopcode üretiyor.

```
I Type
and (i_type_op[0],~and_,~add_,~sub_);
and (i_type_op[1],~and_,~or_);
and (i_type_op[2],~and_,~or_,~add_,~xor_);
R Type
and (r_type_op[0],~r_and,~r_add,~r_shiftr,~r_sub);
and (r_type_op[1],~r_and,~r_or,~r_shiftr,~r_shiftl);
and (r_type_op[2],~r_and,~r_or,~r_add,~r_xor);
```

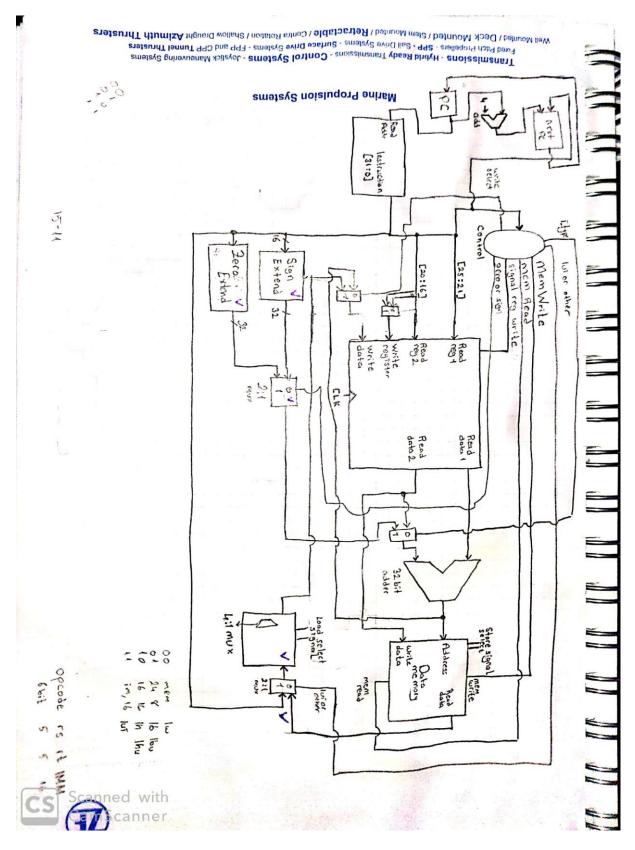
32 bit 2:1 mux ile instruction tipine göre aluop belirlenir.

instruction\_memory → 32 bit bir memory.Program counter 'a göre sırayla instructionları gönderir.

jump\_address → jump instructionı geldiği zaman 26 bitlik jump adresini program counter ile birleştirip jump edilecek adresi oluşturur.

next\_pc → program counter'ı clock atımlarına göre arttırır.Jump veya brach instructionları geldiğinde pc 'ı hanginin geldiğine göre değiştirir.

# Şema



# 3. ödevde bulunup bu ödevde bulunmayan instructionlar da çalışmaktadır.

## Test Sonuçları

