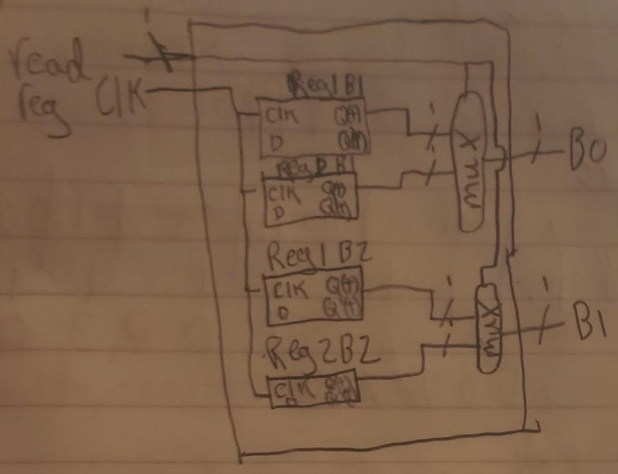
Hw4 George Kent Scheller 450626762

1.)



2.)

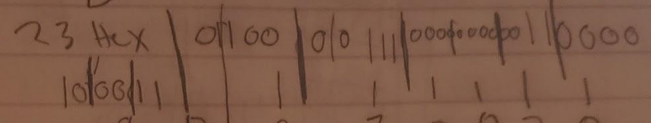
	T0	T1	T2	T3	T4	T5	T6
Out_R0	?	3	5	7	7	11	11
out_R1	?	0	2	4	4	8	8
Enable	1	1	1	0	1	0	1
IN_R2	?	3	2	5	4	7	4
OutR2	?	?	3	2	2	2	4

3.)

a.) Lw $7, 48($12)

RT im RS

Type: I | 6 opcode | 5 RS | 5 RT | Inter-16 bit



8D87 0030

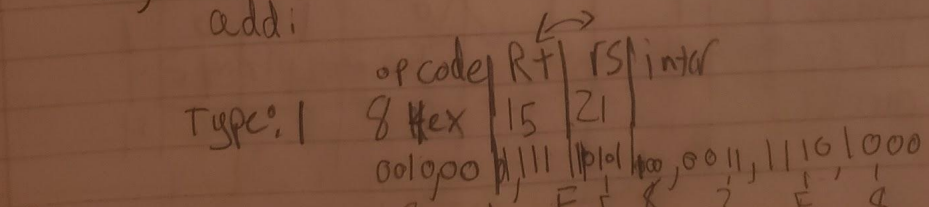
b.) Jal

Type: J | opcode | 26 bit address

Can't do would need line number of procedure

c.) addi

Type: I | opcode | Rt | rs | inter



21F583E8

opCode 5 | type I | inter

4.) beq | rs 10 | rt 11 | assuming do-4 • #lines: 4-

OP	RS	Rt	I
000101	01010	01011	1000 0001 0000 ⇐ -16

5.)

a.) 0810 0000

0, 8, 1, 0, 0, 0, 0 a=10

0000, 1000, 0001, 0000, 0000, 0000, 0000, 1010

0x 1000 0000

J 0x1000000a

b.) 8d4c0114

1000, 1101, 0100, 1100, 0000, 0001, 0100

23 0x | 13 | 24 | 01 | 1 | 4

lw $24, (0x0114)$24

6.)

BGT $4, $5, 17

4>5 →

li $t0, 1
SLT $t1, $5, $4
Beq $t0, $t1, 17

7.)

```
Li $t0, 0xEFEF
Li $t1, 0xEFFF
La $t2, (16)$9
Of $t3, $9, $t0
Of $t4, $t2, $t1
Lw $8, $t3
Lw(16)$8, $t4
```

8.)

```
Lw $9, 0xFFEFFEFF($10)
Li $t0, 0xEFEF
Sll $t1, $t0, (16)
add $t2, $t0, $t1
lui $t3, $10
add $t2, $t2, $t3
Lw $9, (0)$t2
```

9.)

1.)

RegDST = 0
branch = 0
memRead = 0
memtoReg = 1
alu OP = whichever does and
mem write = 0
aluSrc = 0
regwrite = 1

2.)

- Control
- registers
- ALU
- ALU Control
- 3 MUXES

3.)

- [illegible]
- Sign extend
- Shift left 2
- add alu
- Alu
- Data memory

no OUT

10.)

1.) • Data memory
• Alu
2.) • Read reg
• Control

2.) Can already implement it.

3.) Can already implement it, a control to specify adding to a particular part of a reg would help.

11.)

used		improve
1	I-mem 400	380
1	Add 100	+
4x	muX 30	
1	Alu 120	+ 300
1	Reg 200	
1	D-mem 350	
1	Control 100	

base t: 1300 newt: 1580

12)

1.) I-mem - add mux

200 | 70 20 = 200

2.) I-mem - add mux shift add Signex

200 20 10 70 15 = 315

3, Same

4.) branch, jump

5.) Every jump because PC has to wait for the I-mem

6.) we can get rid of: Sign extend.