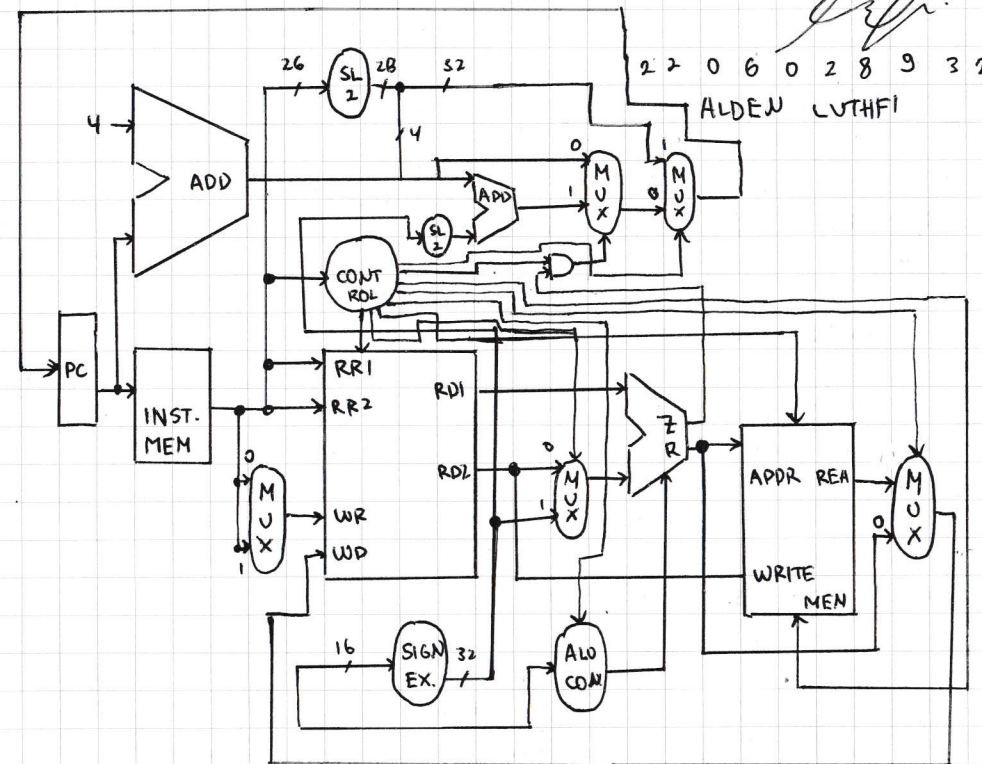


DATAPATH!! TUGAS 2 POK



①. ② R format (ex: add, sll, srl)

0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Op code						reg Source			reg target			Reg. destination				Shift Amount			

x	x	x	x	x	x
	function				

- ↳ 6 bit Opcode (semua 0)
- ↳ 5 bit rs
- ↳ 5 bit rt
- ↳ 5 bit rd
- ↳ 5 bit Shift Amount
- ↳ 6 bit function

ex: add \$t0, \$s8, \$s9

↑            ↑            ↑  
 rd        rs        rt

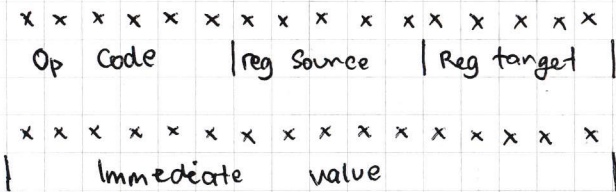
function = (32)<sub>10</sub>  
 shift = 0

on Datapath Control:

RegDst = 1, MemWrite = 0, MemRead = 0, Branch = 0

MemtoReg = 0, ALUOp = 10, Jump = 0, ALUSrc = 0, RegWrite = 1

② I format (ex: addi, beq, lw)



↳ 16 bit Immediate value

↳ 6 bit Op code

↳ 5 bit rs

↳ 5 bit rt

ex 1: addi \$t0, \$t9, 12

↑      ↑  
 rt    rs

Immediate = (12)<sub>10</sub>

Op code = (8)<sub>10</sub>

On Datapath Control:

RegDst = 0, Jump = 0, Branch = 0

MemRead = 0, MemtoReg = 0, ALUOp = 10

MemWrite = 0, ALUSrc = 1, RegWrite = 1

ex 2: lw \$t0, 12(\$t9)

↑      ↑  
 rt    rs

Immediate = (12)<sub>10</sub>

Op code = (23)<sub>16</sub>

On Datapath Control:

RegDst = 0, Jump = 0, Branch = 0

MemRead = 1, MemtoReg = 1, ALUOp = 10

MemWrite = 0, ALUSrc = 1, RegWrite = 1

③ J format

↳ 6 bit Opcode

↳ 26 bit pseudobaddress → to get real address → PC[31..26] + pseudobaddr + 00

On datapath:

RegDst = X, Jump = 1, branch = X, MemRead = X, MemtoReg = X

MemWrite = 0, ALUOp = XX, ALUSrc = X, RegWrite = 0

② a. tidak karena Immediate hanya dapat menampung 16 bit sedangkan 0xAAAAA memiliki 20 bit yang terisi

NB: MARS membolehkan hal ini dengan mengganti "addi" menjadi "lui + ori + add" namun SPIM tidak membolehkan hal ini

b. dengan syarat myLabel terdefinisi:

PC akan menjadi Label myLabel dan \$31 atau \$ra atau return address akan menjadi PC+8 sebelum diubah

c. tidak karena \$t0 bukanlah nilai Immediate yang valid

d. mengurangi isi register \$a0 dengan isi \$a1 dan menyimpannya di \$v2

e. menyimpan word sebelum word yang ada di address yang disimpan di \$t0 ke \$a0

③ a. Instruksi tersebut adalah I format karena 6 bit pertamanya bukan bernilai 0, 2 atau 3

b. Instruksi tersebut adalah R format karena 6 bit pertamanya bernilai 0, Instruksi tersebut adalah add karena 6 bit terakhirnya (20)<sub>16</sub>

c. Instruksi tersebut adalah J format karena 6 bit pertamanya bernilai 2 atau 3

d. Instruksi tersebut adalah R format karena 6 bit pertamanya bernilai 0

e. Instruksi tersebut adalah I format karena 6 bit pertamanya bukan bernilai 0, 2 atau 3



Instruksi MIPS masing-masing

2 2 0 6 0 2 8 9 5 2  
ALDEN WTHFI

(a) 0 0 0 1 0 1 0 1 0 1 0 0 1 0 0 1  
 | Opcode | rs | rt |  
 1 1 1 1 1 1 1 1 1 1 0 1 1  
 | Immediate |

Opcode 5 = bne

↳ bne \$9, \$10, -5

(b) 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0  
 | Opcode | rs | rt |  
 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0  
 | rd | shift | function |

function 32 = add

↳ add \$4, \$16, \$8

(c) 0 0 0 0 1 1 0 1 0 0 1 1 0 1 1 1  
 | Opcode | Pseudo |  
 1 1 1 1 1 1 0 1 0 0 0 0 1 0 1 0  
 | address |

Opcode 3 = jal

↳ jal ke label yang ada pada address  
 PC[31..28] + 010011011111110100001000  
 dan isi \$ra dengan PC+8

2 2 0 6 0 2 8 9 3 2  
ALDEN WTHFI

(d) 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1  
 | Opcode | rs | rt |  
 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0  
 | rd | shift amount | function |

function 0 = sll

↳ sll \$5, \$9, 3

(e) 1 0 1 0 1 1 0 0 1 0 1 0 1 0 1 0  
 | Opcode | rs | rt |  
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0  
 | Immediate |

Opcode = (2b)<sub>16</sub> = sw

↳ sw \$10, -2187(\$5)

(4) asumsi sesuai soal

lw \$a0, 0(\$a0)  
 addi \$t0, \$zero, 1

\* \$a0 sebagai loop Counter  
 \* suku ke-0

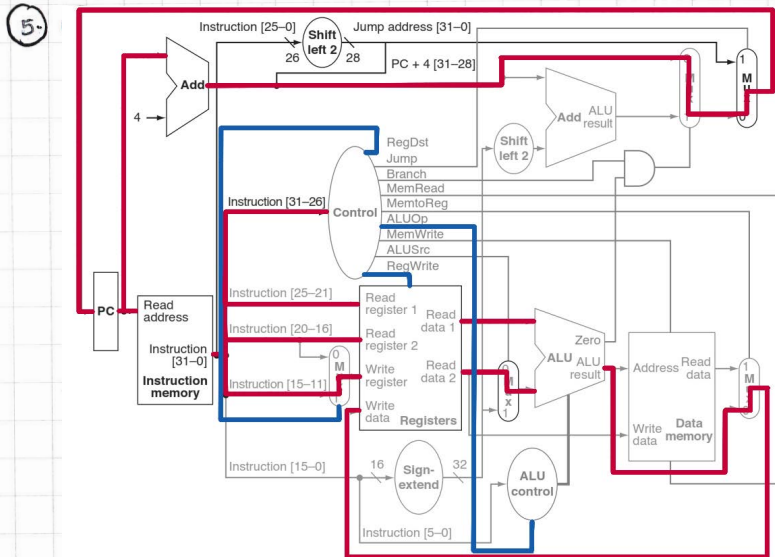
Loop :

mul \$t0, \$t0, 3  
 sw \$t0, 0(\$a1)  
 addi \$a1, \$a1, 4  
 addi \$a0, \$a0, -1  
 bne \$a0, \$zero, loop

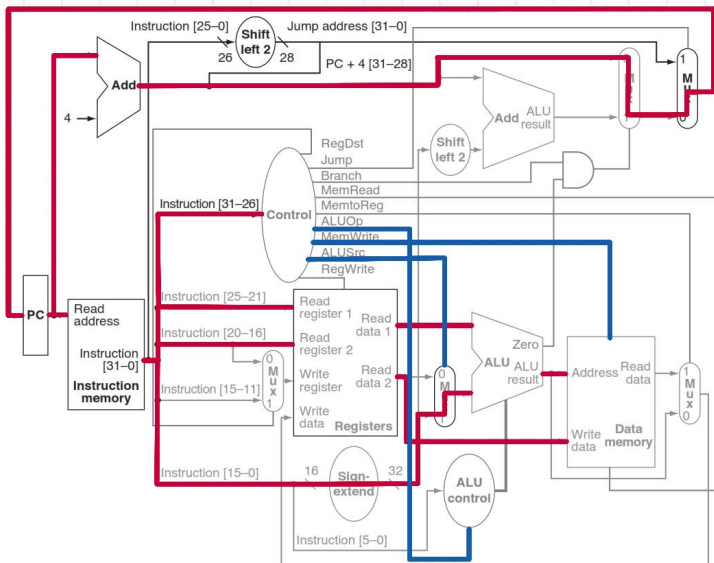
# mengalikan suku sebelumnya dengan 3  
 # store ke memory  
 # pointer suku selanjutnya  
 # decrement loop counter  
 # loop branch

2 2 0 6 0 2 8 9 3 2  
ALDEN LVTHF1

a.



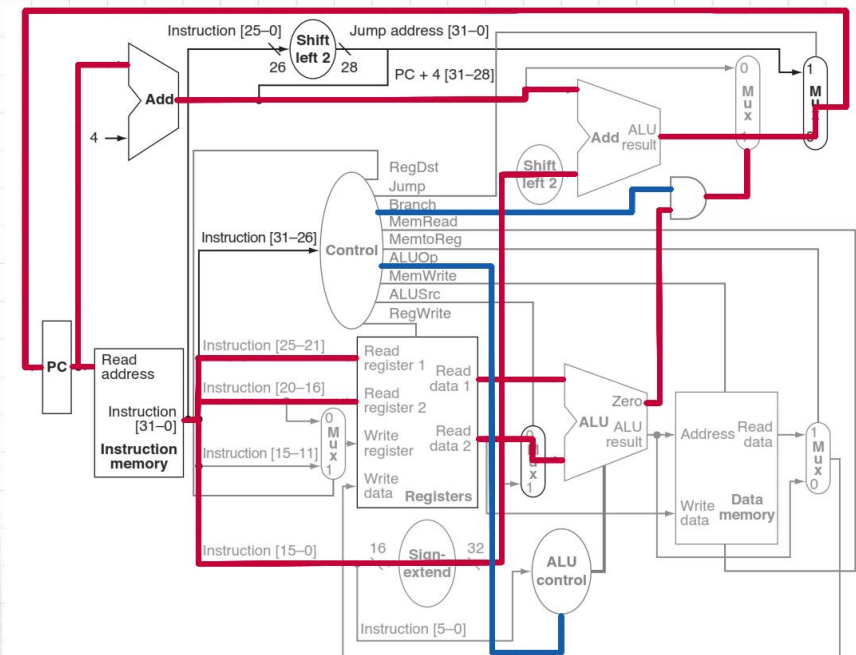
b.



6.

RegDst	= 1
RegWrite	= 1
Write Reg	= 1 0 0 0 0
Read Reg1	= 0 1 0 1 0
Read Reg2	= 0 1 1 0 0
Read Data1	= 0 x 0 3 0 4 0 C A 0
Read Data2	= 0 x 0 0 2 0 1 0 2 0
ALUSrc	= 0
ALUInput2	= 0 x 0 0 2 0 1 0 2 0
ALU Control	= 0 0 1 0
ALU RESULT	= 0 x 0 3 2 4 2 9 C 0
Zero	= 0
MemWrite	= 0
MemRead	= 0
MemtoReg	= 0

c.



22 0 6 0 2 8 9 3 2  
ALDEN LVTHF1