Q1(b)(i), (ii), (iii).

CS2100 - Tutorial 5 - MIPS: Datapath & Control

Week 7

1

i. `0x0285c822: sub $25, $20, $5`
ii. `0x8df80000: lw $24, 0($15)`
iii. `0x1023000C: beq $1, $3, 12`

(a)

result of ALU op

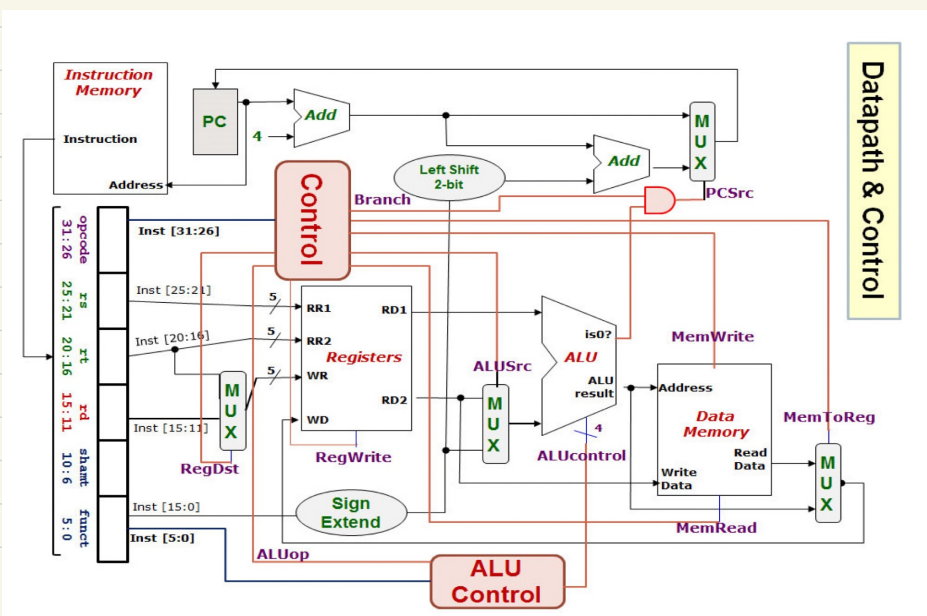
[RD2]

	Registers File				ALU		Data Memory	
	RR1	RR2	WR	WD	Opr1	Opr2	Addr	Write Data
(i)	$20	$5	$25	[$20] - [$5]	[$20]	[$5]	[$20]-[$25]	[$5]
(ii)	$15	$24	$24	Mem + 0 ([$15])	[$15]	0	[$15]+0	[$24]
(iii)	$1	$3	$3 or $0	[$1]-[$3] or random value	[$1]	[$3]	[$1]-[$3]	[$3]

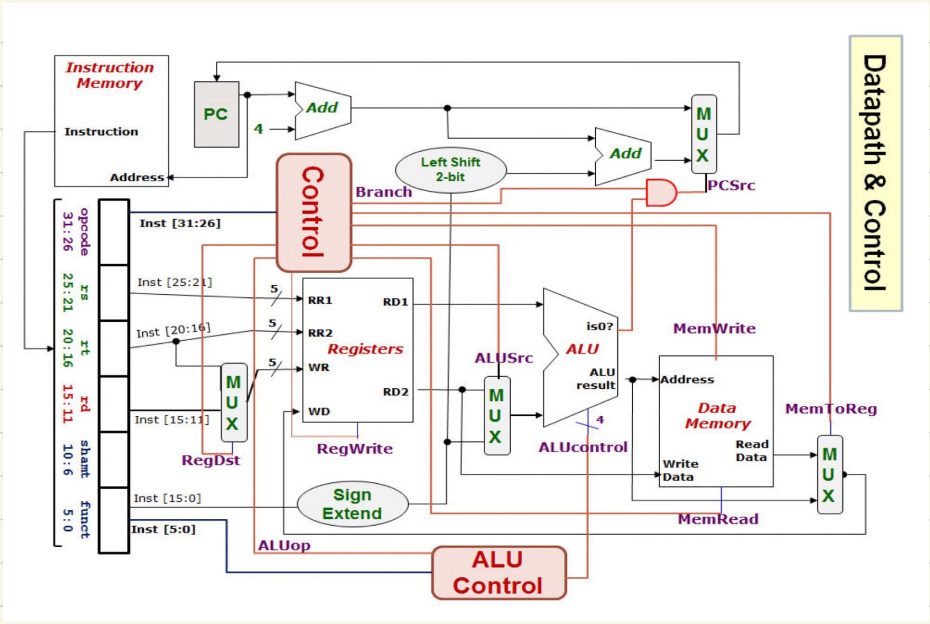
[Wr = Write; Rd = Read; M = Mem; R = Reg]

	RegDst	RegWr	ALUSrc	MRd	MWr	MToR	Brch	ALUop	ALUctrl
(i)	1	1	0	0	0	0	0	10	0110
(ii)	0	1	1	1	0	1	0	00	0010
(iii)	X	0	0	0	0	X	1	01	0110

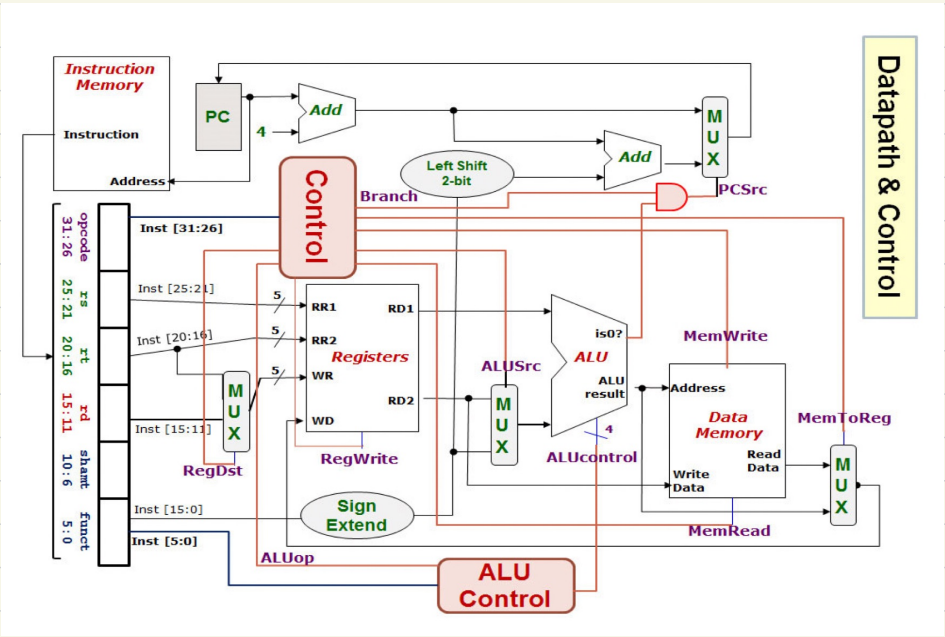
(b)(i)



(b)(ii).


Datapath & Control
Instruction Memory
Instruction
Address
PC
4
Add
Add
Left Shift 2-bit
M U X
PCSrc
Branch
Control
opcode 31:26
rs 25:21
rt 20:16
rd 15:11
shamt 10:6
funct 5:0
Inst [31:26]
Inst [25:21]
Inst [20:16]
Inst [15:11]
Inst [15:0]
Inst [5:0]
5
RR1
RR2
WR
WD
RD1
RD2
Registers
RegDst
RegWrite
ALUSrc
is0?
ALU
ALU result
4
ALUcontrol
MemWrite
Address
Data Memory
Write Data
Read Data
MemToReg
MemRead
Sign Extend
ALUop
ALU Control

(b)(iii).


Datapath & Control
Instruction Memory
Instruction
Address
PC
4
Add
Add
Left Shift 2-bit
M U X
PCSrc
Branch
Control
opcode 31:26
rs 25:21
rt 20:16
rd 15:11
shamt 10:6
funct 5:0
Inst [31:26]
Inst [25:21]
Inst [20:16]
Inst [15:11]
Inst [15:0]
Inst [5:0]
5
RR1
RR2
WR
WD
RD1
RD2
Registers
RegDst
RegWrite
ALUSrc
is0?
ALU
ALU result
4
ALUcontrol
MemWrite
Address
Data Memory
Write Data
Read Data
MemToReg
MemRead
Sign Extend
ALUop
ALU Control

2. Signals [AY1920 S1 Term Test]

(a). add $t0, $t0, $zero,

where [$t0] is any non-zero value

(b). 0 → $t1

PC = PC+4 + 0 (Imm), so

instructions carry on as per usual

3. Datapath [AY1314 S2 Term Test]

(i) (a). add opcode/funccode: 0x0/0x20 (32)

∴ add $t1, $t0, $t1

(b). $t0 == $8

Imm value should look like: $4 \times 16^3 = 16384$

0100 0000 0000 0000 ⇒ 0x4000 = 16384

∴ lw $t0, 16384($a0) (doesn't matter)

(c). same logic as above

beq $a0, $t0, 16384 All would work ∵ no WR necessary

(ii) (a). add $t2, $t0, $t1

(b). lw $t1, 0($a0)

(c). beq $a0, $t1, 0 No answer