BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING CUỐI KỲ HỌC KỲ I NĂM HỌC 2022 – 2023

ĐOÀN KHOA
CÔNG NGHỆ PHẦN MỀM
BAN HỌC TẬP

Sharing is learning

BAN HỌC TẬP

Khoa Công nghệ Phần mềm

Trường Đại học Công nghệ Thông tin

Đại học Quốc gia thành phố Hồ Chí Minh

CONTACT

bht.cnpm.uit@gmail.com

fb.com/bhtcnpm

fb.com/groups/bht.cnpm.uit

TRAINING

TỔ CHỨC - CẤU TRÚC MÁY TÍNH II

Thời gian: 19:30 thứ 3 ngày 14/02/2022

Địa điểm: Microsoft Teams

Trainers: Phan Nguyễn Tuấn Anh – KTPM2022.1

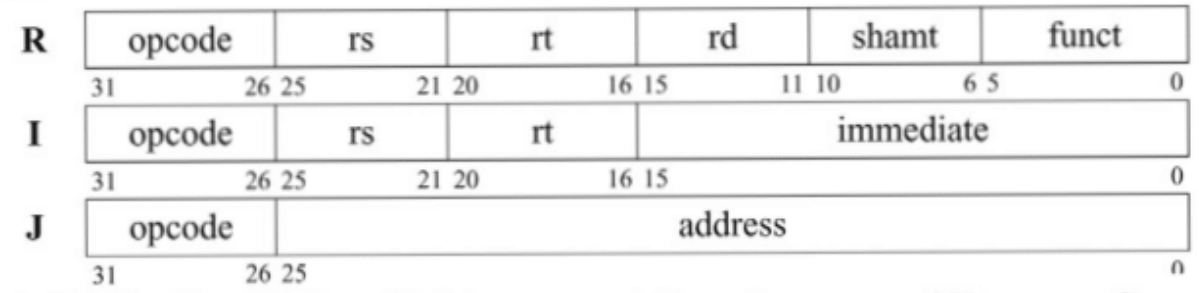
Ngô Hương Giang – KTPM2022.1

CORE INSTRUCTION SET

NAME, MNEMONIC		FOR-MAT	OPERATION (in Verilog)		OPCODE / FUNCT (Hex)
Add	add	R	R[rd] = R[rs] + R[rt]	(1)	0 / 20_{hex}
Add Immediate	addi	I	R[rt] = R[rs] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + SignExtImm	(2)	9_{hex}
Add Unsigned	addu	R	R[rd] = R[rs] + R[rt]		0 / 21_{hex}
And	and	R	R[rd] = R[rs] & R[rt]		0 / 24_{hex}
And Immediate	andi	I	R[rt] = R[rs] & ZeroExtImm	(3)	c_{hex}
Branch On Equal	beq	I	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4)	4_{hex}
Branch On Not Equal	bne	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	5_{hex}
Jump	j	J	PC=JumpAddr	(5)	2_{hex}
Jump And Link	jal	J	R[31]=PC+8;PC=JumpAddr	(5)	3_{hex}
Jump Register	jr	R	PC=R[rs]		0 / 08_{hex}
Load Byte Unsigned	lbu	I	R[rt]={24'b0,M[R[rs] +SignExtImm](7:0)}	(2)	24_{hex}
Load Halfword Unsigned	lhu	I	R[rt]={16'b0,M[R[rs] +SignExtImm](15:0)}	(2)	25_{hex}
Load Linked	ll	I	R[rt] = M[R[rs]+SignExtImm]	(2,7)	30_{hex}
Load Upper Imm.	lui	I	R[rt] = {imm, 16'b0}		f_{hex}
Load Word	lw	I	R[rt] = M[R[rs]+SignExtImm]	(2)	23_{hex}
Nor	nor	R	R[rd] = ~ (R[rs] | R[rt])		0 / 27_{hex}
Or	or	R	R[rd] = R[rs] | R[rt]		0 / 25_{hex}
Or Immediate	ori	I	R[rt] = R[rs] | ZeroExtImm	(3)	d_{hex}
Set Less Than	slt	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0		0 / $2a_{hex}$
Set Less Than Imm.	slti	I	R[rt] = (R[rs] < SignExtImm)? 1 : 0	(2)	a_{hex}
Set Less Than Imm. Unsigned	sltiu	I	R[rt] = (R[rs] < SignExtImm) ? 1 : 0	(2,6)	b_{hex}
Set Less Than Unsig.	sltu	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0	(6)	0 / $2b_{hex}$
Shift Left Logical	sll	R	R[rd] = R[rt] << shamt		0 / 00_{hex}
Shift Right Logical	srl	R	R[rd] = R[rt] >>> shamt		0 / 02_{hex}
Store Byte	sb	I	M[R[rs]+SignExtImm](7:0) = R[rt](7:0)	(2)	28_{hex}
Store Conditional	sc	I	M[R[rs]+SignExtImm] = R[rt]; R[rt] = (*atomic*) ? 1 : 0	(2,7)	38_{hex}
Store Halfword	sh	I	M[R[rs]+SignExtImm](15:0) = R[rt](15:0)	(2)	29_{hex}
Store Word	sw	I	M[R[rs]+SignExtImm] = R[rt]	(2)	$2b_{hex}$
Subtract	sub	R	R[rd] = R[rs] - R[rt]	(1)	0 / 22_{hex}
Subtract Unsigned	subu	R	R[rd] = R[rs] - R[rt]		0 / 23_{hex}

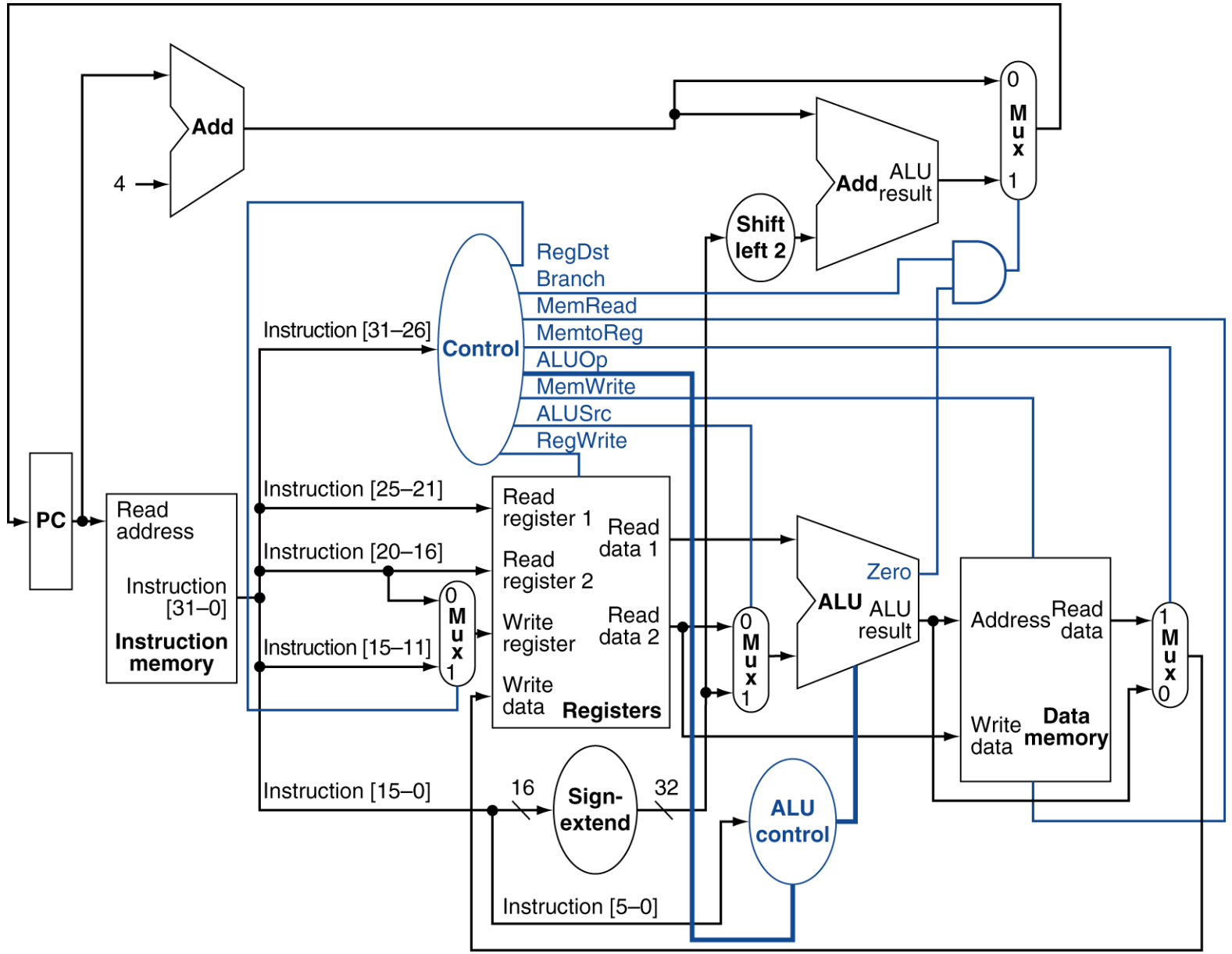
(1) May cause overflow exception
(2) SignExtImm = { 16{immediate[15]}, immediate }
(3) ZeroExtImm = { 16{1b'0}, immediate }
(4) BranchAddr = { 14{immediate[15]}, immediate, 2'b0 }
(5) JumpAddr = { PC+4[31:28], address, 2'b0 }
(6) Operands considered unsigned numbers (vs. 2's comp.)
(7) Atomic test&set pair; R[rt] = 1 if pair atomic, 0 if not atomic

BASIC INSTRUCTION FORMATS



REGISTER NAME, NUMBER, USE, CALL CONVENTION

NAME	NUMBER	USE	PRESERVED ACROSS A CALL?
$zero	0	The Constant Value 0	N.A.
$at	1	Assembler Temporary	No
$v0-$v1	2-3	Values for Function Results and Expression Evaluation	No
$a0-$a3	4-7	Arguments	No
$t0-$t7	8-15	Temporaries	No
$s0-$s7	16-23	Saved Temporaries	Yes
$t8-$t9	24-25	Temporaries	No
$k0-$k1	26-27	Reserved for OS Kernel	No
$gp	28	Global Pointer	Yes
$sp	29	Stack Pointer	Yes
$fp	30	Frame Pointer	Yes
$ra	31	Return Address	Yes


Add
4
0
M
u
x
1
Add
ALU result
Shift left 2
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign-extend
32
ALU control
Zero
ALU
ALU result
Address
Read data
Write data
Data memory

Câu 1: Lệnh "addi $t0, $s2, -8" có mã máy là bao nhiêu?

A. 0x2648FFF8

B. 0x2248FFF8

C. 0x2684FFF8

D. 0x2248F1F8

Lệnh "addi $t0, $s2, -8" thuộc định dạng lệnh I

op	rs	rt	immediate
6 bits	5 bits	5 bits	16 bits
addi	$s2	$t0	-8
0x8	18	8	-8
001000	10010	01000	1111111111111000

→ 0010 0010 0100 1000 1111 1111 1111 1000 → 0x2248FFF8

Câu 2 Mã máy "0xAD50FFF9" là cửa lệnh hợp ngữ nào sau đây?

A. Sw $t0, -7($t2)
B. Sw $s0, 7($t2)
C. Lw $s0, -7($t2)
D. Sw $16, -7($10)

0xAD50FFF9

Biểu diễn nhị phân: 1010 1101 0101 0000 1111 1111 1111 1001

Ta có:

- op (6 bits): 101011 → 0x2B. Opcode = $2B_{hex}$ có lệnh hợp ngữ tương ứng là sw

→lệnh hợp ngữ thuộc dạng I

- rs (5 bits): 01010 → 10 ($t2)
- rt (5 bits): 10000 → 16 ($s0)
- immediate (16 bits): 1111111111111001 → -7

Vậy lệnh hợp ngữ là sw $s0, -7($s2) hay sw $16, -7($10)

Câu 3: Cho $s0=0x16, sau khi thực hiện lệnh "sll $t0,$s0,2" thì giá trị $t0 là:

A. 0x04
B. 0x58
C. 0x32
D. 0x64

Biểu diễn 0x16 dạng nhị phân: 0001 0110

sll $t0,$s0,2

Tiến hành dịch trái 2 bit: 00 0101 1000 → 0x58

Câu 4 Đoạn mã hợp ngữ sau thực hiện biểu thức nào. Giả sử f,g, h, i,j được gán cho các thanh ghi $s0, $s1, $s2, $s3, $s4:

A. f = (g + h) - (i – j)
B. f = (g + h) + (i – j)
C. f = (g - h) + (i – j)
D. f = (g + h) - (i + j)

```
add $t0, $s1, $s2
sub $t1, $s4, $s3
sub $s0, $t0, $t1
```

Câu 5 Thanh ghi PC sẽ tăng bao nhiêu sau mỗi lần đọc lệnh ?

A. 1

B. 2

C. 3

D. 4

Câu 6 Lệnh MIPS nào tương đương với mã lệnh C sau đây:

```
if ($s2 < $s3)
        $s1 = 1;
else $s1 = 0;
```

A. Beq $s1, $s2, $s3

B. Slt $s1, $s2, $s3

C. Sll $s1, $s2, $s3

D. Slti $s1, $s2, $s3

Câu 7 Phát biểu nào sau đây không chính xác

A. Toán hạng thanh ghi là toán hạng mà giá trị của nó được ghi vào/đọc ra từ thanh ghi.

B. Toán hạng bộ nhớ là toán hạng mà giá trị của nó được ghi vào/đọc ra từ bộ nhớ

C. Toán hạng hằng là toán hạng mà giá trị của nó được ghi vào/đọc ra từ hằng số

D. Toán hạng hằng là toán hạng mà giá trị của nó được lấy ra từ lệnh chương trình

Câu 8: Với định dạng R-type trong kiến trúc MIPS, khi trường opcode trên hệ thập phân có giá trị 0 và trường funct có giá trị là 32. Xác định tên lệnh

A. Add
B. Addi
C. Sub
D. Lw

Câu 9 Trong câu lệnh lw, địa chỉ của write register trong mã máy là các bit từ?

A. 21-25
B. 16-20
C. 11-15
D. 7-11

Câu 10 Mã máy ngôn ngữ MIPS của lệnh sub $t3, $t1, $t2 là gì? Cho biết chỉ số của thanh ghi $t1 là 9, $t2 là 10, $t3 là 11; giá trị của trường opcode của lệnh sub là 0, trường shamt là 0, trường funct của lệnh sub là 0x22.

A. 000000 01001 01010 01011 00000 100010
B. 000000 01011 01001 01010 00000 100010
C. 000000 01011 01010 01001 00000 100010
D. 000000 01001 01010 01011 00000 010110

Câu 11: Đối với định dạng lệnh R-type của kiến trúc MIPS, khi trường opcode có giá trị 0, ta cần kết hợp với trường nào để xác định lệnh và trường này có bao nhiêu bit?

A. Shamt & 5bit
B. Funct & 5bit
C. Shamt & 6bit
D. Funct & 6bit

Câu 12 Một tín hiệu xung clock có tần số tín hiệu là 4GHz, hỏi chu kì của tín hiệu này là bao nhiêu?

a. 0.25ns
b. 0.2ms
c. 0.25ms
d. 2.5ns

Câu 13 Thành phần nào sau đây không thuộc thành phần của Datapath

a. ALU Control
b. PC
c. Instruction memory
d. Data memory

Câu 14 Công đoạn thứ 2 trong quá trình thực thi lệnh của MIPS là công đoạn nào?

a. Truy xuất bộ nhớ
b. Nạp lệnh
c. Giải mã lệnh
d. Lưu kết quả

Câu 15 Thành phần datapath nào không cần để thực thi lệnh add

a. I-MEM
b. Register
c. PC
d. D-MEM

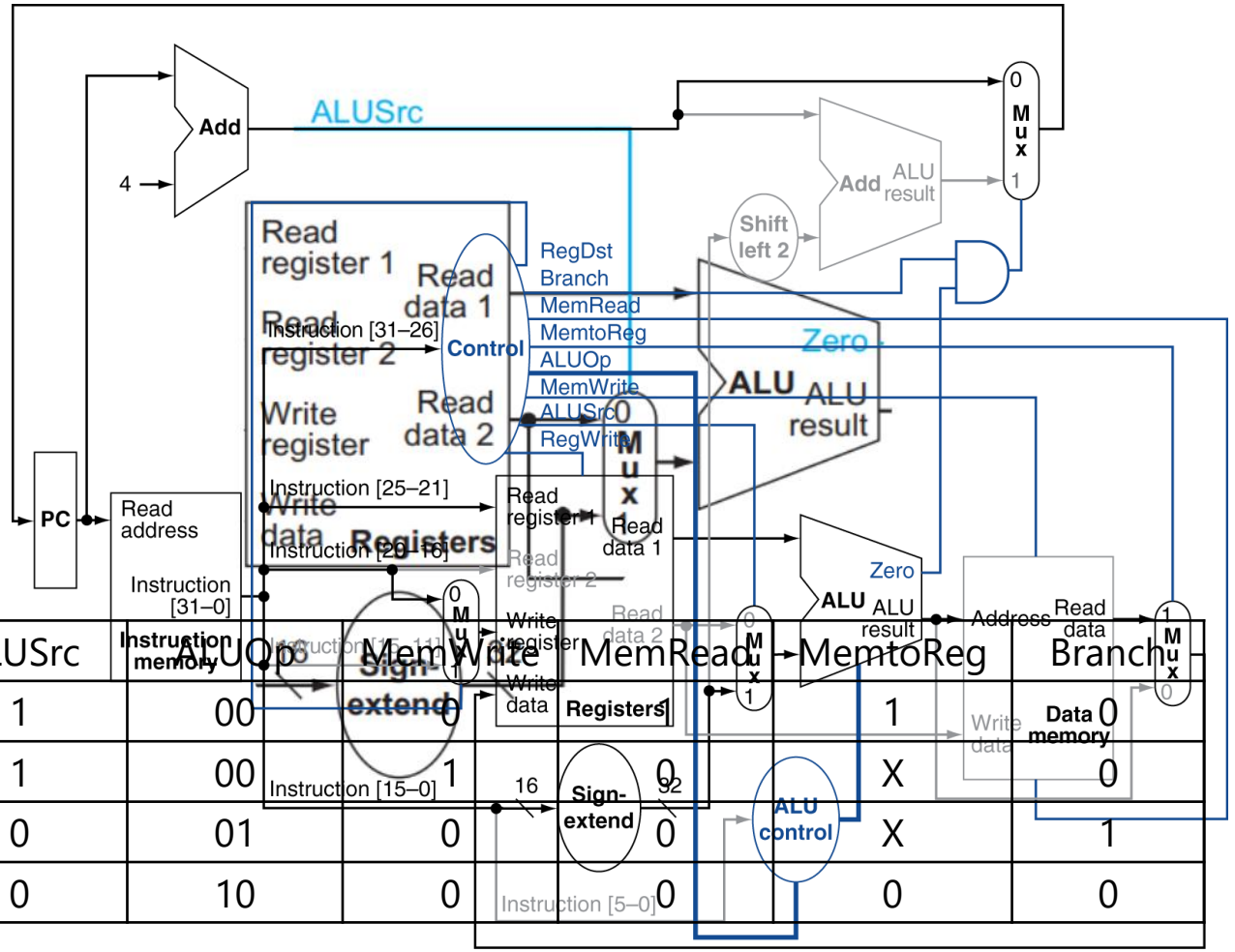
Câu 16 Trong các lệnh sau, lệnh nào có tín hiệu ALUSrc = 1

a. beq

b. lw

c. sub

d. add



opcode	RegWrite	ALUSrc	ALUOp	MemWrite	MemRead	MemtoReg	Branch
lw	1	1	00	0	1	1	0
sw	0	1	00	1	0	X	0
beq	0	0	01	0	0	X	1
Loại R	1	0	10	0	0	0	0

Câu 17 Khi thực thi lệnh nào, giá trị tín hiệu MemtoReg là tùy định

a. sw
b. lw
c. add
d. sub

opcode	RegWrite	ALUSrc	ALUOp	MemWrite	MemRead	MemtoReg	Branch
lw	1	1	00	0	1	1	0
sw	0	1	00	1	0	X	0
beq	0	0	01	0	0	X	1
Loại R	1	0	10	0	0	0	0

Câu 18 Cho hai bộ vi xử lí X và Y có tần số xung clock lần lượt là 0.8 GHz và 700 MHz. Giả xử Y thực thi một lệnh hết 4 chu kì, X thực thi một lệnh hết 5 chu kì. Hỏi khi thực thi cùng một chuong trình, bộ vi xử lí nào thực thi nhanh hơn?

a. X

b. Y

c. Bằng nhau

d. X gấp đôi Y

$$\text{Thời gian thực thi} = \frac{\text{Tổng số chu kỳ clock}}{\text{Tần số clock}}$$

Câu 19 Kỹ thuật nâng cao hiệu suất nào sau đây sửa dụng kĩ thuật Multicore

a. Thực thi đồng thời nhiều **tác vụ** bằng cách tăng số lượng đơn vị xử lý.

b. Thực thi đồng thời nhiều **lệnh** bằng cách chia chu kỳ thực thi lệnh thành các stage. Tại một thời điểm, một lệnh chỉ được thực thi một stage.

c. Thực thi đồng thời nhiều **chương trình** bằng cách tăng số lượng bộ xử lý

d. Thực thi đồng thời nhiều **lệnh** bằng cách chia chu kỳ thực thi lệnh thành các stage.

Câu 20 Một bộ vi xử lí cho clock là 600 MHz với tổng số lệnh là 3 triệu và thời và thời thực thi là 0.1s. Cần thày đổi clock cho bộ vi xử lí này bằng bao nhiêu để giảm thời gian thực thi còn một nửa?

a. 12GHz
b. 120MHz
c. 1.2GHz
d. 120000000Hz

$$\text{Thời gian thực thi} = \frac{\text{Tổng số chu kỳ clock}}{\text{Tần số clock}}$$

Câu 21 Chọn phát biểu sai về thành phần datapath D-Mem

a. Không cần thiết khi thực hiện lệnh R
b. Kết quả đầu ra có 32 bit dữ liệu
c. Có thể đọc và ghi dữ liệu
d. Thuộc loại mạch số tổ hợp

Câu 22 Cho đoạn chương trình sau

```
addi $t1, $zero, 4
add $t0, $t1, $t2
addi $t1, $t0, $t2
sw $t1, 4($s1)
```

a. 4 và 3

b. 2 và 1

c. 4 và 1

d. 3 và 4

- **Hỏi bộ nhớ lệnh và bộ nhớ dữ liệu lần được được truy cập mấy lần**

Câu 23 Máy tính A chạy ở tần số 2 Ghz cần 10s để hoàn thành chương trình P. Máy tính B chỉ cần 6s để hoàn thành chương trình P nhưng tổng số chu kỳ cần để hoàn thành chương trình P nhiều gấp 1.2 lần so với máy tính A. Máy tính B chạy ở tần số bao nhiêu?

a. 1500Mhz

b. 4Ghz

c. 3GHz

d. 5Ghz

Câu 24 Chu kì thực thi lệnh sw không trải qua giai đoạn nào dưới đây?

a. Giải mã
b. Thực thi
c. Truy xuất bộ nhớ
d. Lưu kết quả

Câu 25 Cạnh xuống của xung clock được hiểu là

a. Tại thời điểm giá trị xung clock thay đổi từ 1 xuống 0
b. Tại thời điểm giá trị xung clock thay đổi từ 0 lên 1
c. Tại thời điểm giá trị xung clock bằng 1
d. Tại thời điểm giá trị xung clock bằng 0

Câu 26 Cho một máy tính có CPI = 6 và có tần số hoạt động = 1Ghz. Hỏi máy tính này thực thi một chương trình có 10 triệu lệnh hết bao nhiêu thời gian

a. 6cs
b. 1cs
c. 6ms
d. 6ds

$$\text{Thời gian thực thi} = \frac{\text{Tổng số lệnh} * \text{CPI}}{\text{Tần số clock}}$$

II. TỰ LUẬN

Câu 1: Hoàn thành bảng sau: **(0.5 điểm)**
Giả sử f, g, h được gán cho các thanh ghi $s1, $s2, $s3:

LỆNH	THỰC TẾ
add $s1, $s2, $s3	f = g + h
slti $s1, $s2, 6	f = g < 6
sub $s1, $s2, $s3	f = g - h

II. TỰ LUẬN

Câu 2: Chuyển đoạn lệnh C sau sang Assembly của MIPS **(1.5 điểm)**
Biết i, sum và avg là các số nguyên tương ứng với các thanh ghi $s0, $s1, $s2. Mảng A là mảng mà các phần tử là số nguyên, mỗi phần tử chiếm 1 từ nhớ (4 bytes) và địa chỉ nền của mảng A lưu trong thanh ghi $s3

Sum = 0;
for (i = 0; i < 4; i++)
Sum = Sum + A[i];
Avg = Sum/4;

```
add $s1, $0, $0
add $s0, $0, $0
loopFor:
slti $t0, $s0, 4
beq $t0, $0, exitFor
sll $t1, $s0, 2
add $t2, $t1, $s3
lw $t3, 0($t2)
add $s1, $s1, $t3
addi $s0, $s0, 1
j loopFor
exitFor:
srl $s2, $s1, 2
```

```
add $s1, $0, $0          #$t1 = 0 + 0 (Sum = 0)
add $s0, $0, $0          #$s0 = 0 + 0 (i=0)
loopFor:                 # Thực hiện vòng lặp For
slti $t0, $s0, 4         # $t0 = (i < 4)?
beq $t0, $0, exitFor     #Nếu $t0 = 0 (i >= 4) => Thoát vòng lặp For
sll $t1, $s0, 2          # $t1= i*4 (dịch trái i 2 bit)
add $t2, $t1, $s3        # $t2 = i*4 + $s3 (địa chỉ của phần tử A[i])
lw $t3, 0($t2)           # Nạp
add $s1, $s1, $t3        # Sum = Sum + A[i]
addi $s0, $s0, 1         # Tăng i lên 1
j loopFor                #Nhảy đến vòng lặp For
exitFor:
srl $s2, $s1, 2          # Sum/4 = dịch phải Sum 2 bit
```

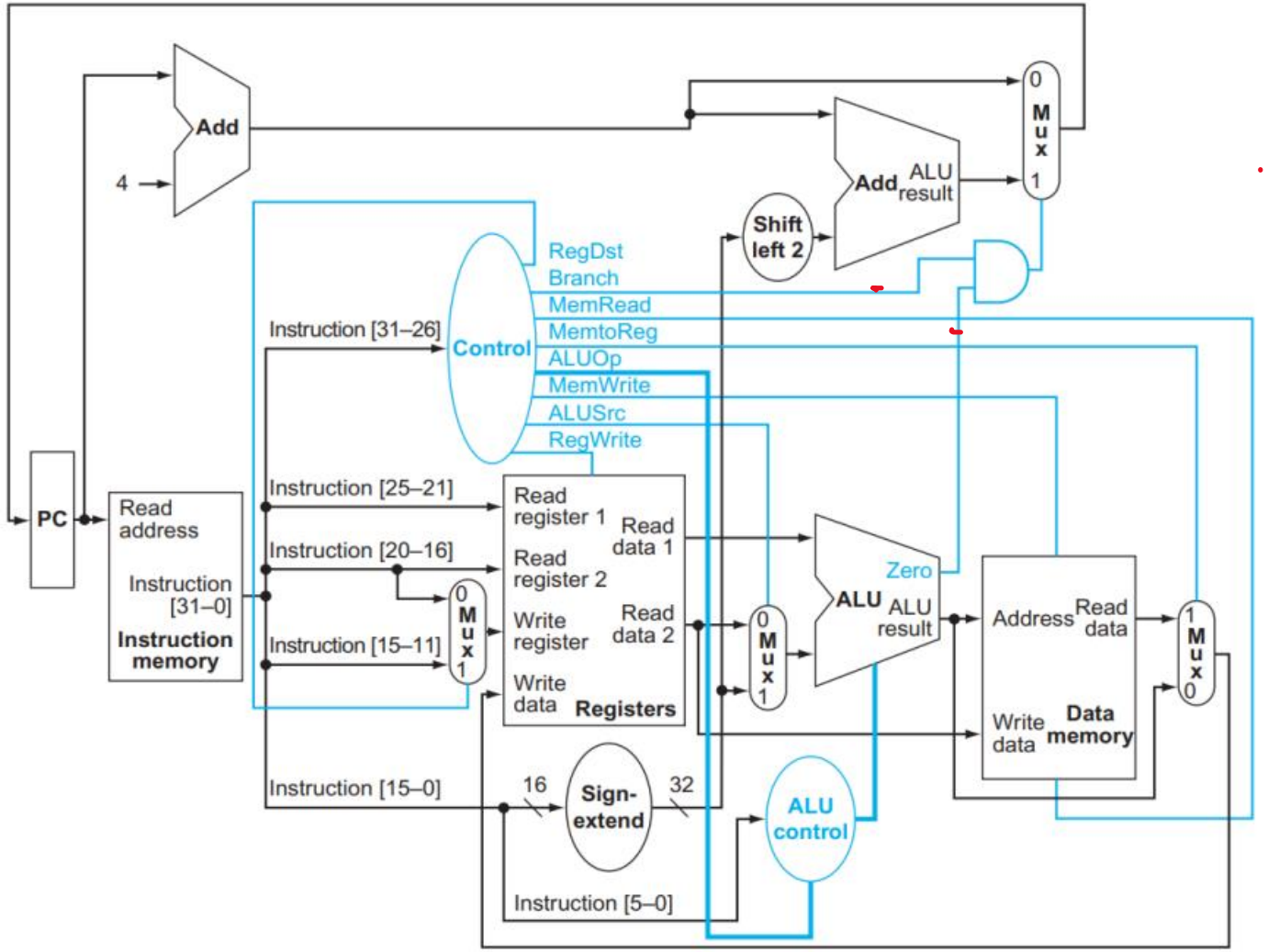
II. TỰ LUẬN

Câu 3: Xác định các giá trị theo định dạng hệ nhị phân của các tín hiệu Read Register 1, Read register 2, RegDst, ALUSrc, Regwrite khi bộ xử lí thực hiện lệnh slt $t0, $t1, $t5 (0.5 điểm)

II. TỰ LUẬN

slt $t0, $t1, $t5

Read Register 1 = 01001
Read register 2 = 01101
RegDst = 1
ALUSrc = 0
Regwrite = 1



II. TỰ LUẬN

Câu 4: Hoàn thành bảng dưới đây **(1 điểm)**

Opcode	ALUOp	Lệnh	Funct	Phép toán ALU	ALU control
lw	00	Nạp word	XXXXXX	Cộng	0010
sw	00	Lưu word	XXXXXX	Cộng	0010
beq	01	Nhảy nếu bằng	XXXXXX	Trừ	0110
Loại R	10	Cộng		Cộng	0010
		Trừ		Trừ	0110
		AND		AND	0000
		OR		OR	0001
		Thiết lập nếu nhỏ hơn		Thiết lập nếu nhỏ hơn	0111

LINK ĐIỂM DANH

BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING CUỐI KỲ HỌC KỲ I NĂM HỌC 2022 – 2023

Sharing is learning

HẾT

CẢM ƠN CÁC BẠN ĐÃ THEO DÕI
CHÚC CÁC BẠN CÓ KẾT QUẢ THI THẬT TỐT!

BAN HỌC TẬP

Khoa Công nghệ Phần mềm

Trường Đại học Công nghệ Thông tin

Đại học Quốc gia thành phố Hồ Chí Minh

CONTACT

bht.cnpm.uit@gmail.com

fb.com/bhtcnpm

fb.com/groups/bht.cnpm.uit