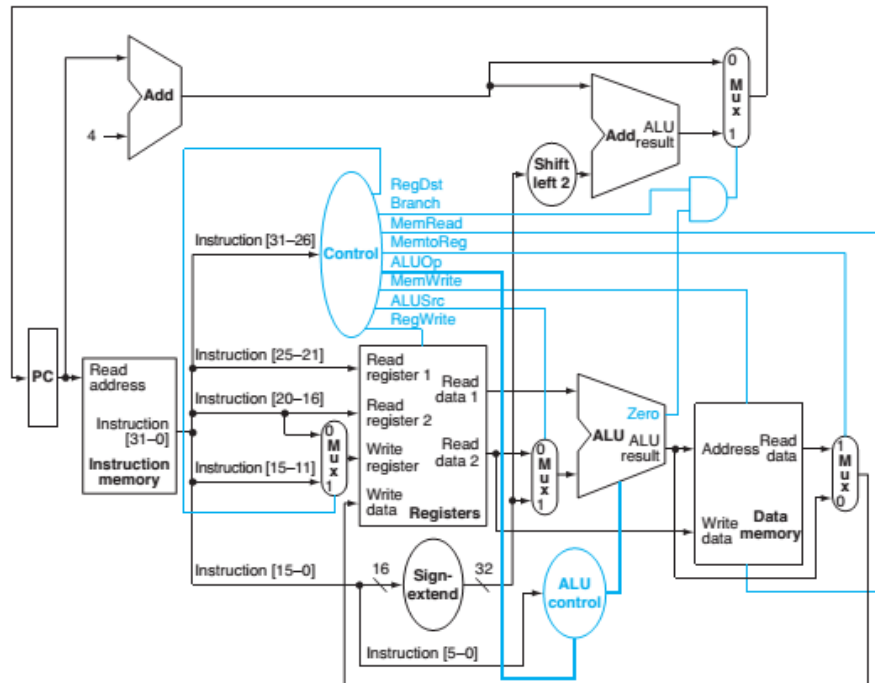


Bài Tập (Datapath)

---oOo---

Các bài tập chương này được trích dẫn và biên soạn lại từ:

Computer Organization and Design: The Hardware/Software Interface,
Patterson, D. A., and J. L. Hennessy, Morgan Kaufman, **Third Edition**, 2011.



Hình 1.

Bài 1. (4.1 – sách tham khảo)

Cho 2 lệnh như sau:

	Lệnh	Ý nghĩa
a.	add rd, rs, rt	$\text{Reg}[\text{rd}] = \text{Reg}[\text{rs}] + \text{Reg}[\text{rt}]$
b.	lw rt, offs(rs)	$\text{Reg}[\text{rt}] = \text{Mem}[\text{Reg}[\text{rs}] + \text{offs}]$

Với từng lệnh trong bảng này:

1. Giá trị các tín hiệu điều khiển từ khối “Control” sẽ như thế nào?

Answer:

a.

RegOst	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
1	0	0	0	10	0	0	1

b.

RegOst	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
0	0	1	1	00	0	1	1

2. Các khối nào trong datapath hình 1 cần thiết, khối nào không cần thiết?

Answer:

a. “Data Memory”, bộ cộng dùng cho lệnh nhảy, “shift left 2”, “sign-extend” và cổng logic AND

b. Bộ cộng dùng cho lệnh nhảy, “shift left 2” và cổng logic AND

3. Khối nào trong datapath hình 1 có output đầu ra, nhưng output này không được sử dụng cho lệnh? Khối nào không có output?

Answer:

a. Các khối có output đầu ra nhưng output không được sử dụng: bộ cộng dùng cho lệnh nhảy, “shift left 2”, ngõ Zero của ALU, “sign-extend”.

Khối không có output: Data Memory

b. Khối có output nhưng không được sử dụng: bộ cộng dùng cho lệnh nhảy, “shift left 2”, ngõ Zero của ALU, ngõ Read Data 2 của Registers.

Tất cả các khối đều có output

Cho thời gian cần để hoàn thành của từng khối trong hình 1 như sau (khối nào không có trong bảng xem như thời gian cần để hoàn thành bằng 0):

	I-Mem	Add	Mux	ALU	Regs	D-Mem
a.	400ps	100ps	30ps	120ps	200ps	350ps
b.	500ps	150ps	100ps	180ps	220	1000ps

4. Tính thời gian cần để hoàn thành lớn nhất của lệnh “and” trong kiến trúc MIPS và cho biết “critical path” của lệnh?

Chú ý: “Critical path” của một lệnh là đường đi có thời gian trễ lớn nhất trong số các đường có thể khi lệnh thực thi.

Answer:

Critical path = I-Mem/Mux/Regs/Mux/ALU/Mux/Regs

a. Time = $400 + 30 + 200 + 30 + 120 + 30 + 200 = 1010$ ps

b. Time = $500 + 100 + 220 + 100 + 180 + 100 + 220 = 1420$ ps

5. Tính thời gian cần để hoàn thành lớn nhất của lệnh “lw” trong kiến trúc MIPS và cho biết “critical path” của lệnh?

Answer:

Critical path = I-mem/Mux/Regs/ALU/D-Mem/Mux/Regs

a. Time = $400 + 30 + 200 + 120 + 350 + 30 + 200 = 1330$ ps

b. Time = $500 + 100 + 220 + 180 + 1000 + 100 + 220 = 2320$ ps

6. Tính thời gian cần để hoàn thành lớn nhất của lệnh “beq” trong kiến trúc MIPS và cho biết “critical path” của lệnh?

Answer:

Critical path = I-Mem/Regs/Mux/ALU/Mux

a. Time = $400 + 200 + 30 + 120 + 30 = 780$ ps

b. Time = $500 + 220 + 100 + 180 + 100 = 1100$ ps

---oOo---

Bài 2. (4.2 – sách tham khảo)

Giả sử tập lệnh có thêm hai lệnh mới như sau:

	Lệnh	Ý nghĩa
a.	add3 rd, rs, rt, rx	$\text{Reg}[\text{rd}] = \text{Reg}[\text{rs}] + \text{Reg}[\text{rt}] + \text{Reg}[\text{rx}]$
b.	sll rd, rt, shift	$\text{Reg}[\text{rd}] = \text{Reg}[\text{rt}] \ll \text{shift}$ (dịch trái shift bits)

Với từng lệnh trên:

1. Khối nào đang có trong hình 1 có thể sử dụng cho các lệnh này?

Answer:

a. Instruction Memory, Registers (cả 2 cổng đọc và cổng ghi), ALU

b. Instruction Memory, Registers (chỉ 1 cổng đọc và 1 cổng ghi), đường truyền số tức thời tới ALU

2. Khối mới nào cần được thêm vào?

Answer:

- a. Thêm một cổng đọc vào khối Registers và thêm một ALU để tính tổng Rx với Rs + Rt (hoặc sửa ALU đang có thành ALU với 3 input)
 - b. Đưa thêm tính năng dịch vào ALU hiện tại
3. Tín hiệu mới nào cần được thêm vào từ khối “Control” để hỗ trợ?

Answer:

- a. Thêm một tín hiệu điều khiển để điều khiển ALU mới trong trường hợp ALU câu 2.2.a chọn thêm ALU mới (hoặc thay đổi lại khối “ALU control” để điều khiển ALU 3 đầu vào trong trường hợp câu 2.2.a chọn sửa lại ALU 2 input thành 3 input)
- b. Thay đổi lại khối “ALU Control” để điều khiển ALU có thêm tính năng sll

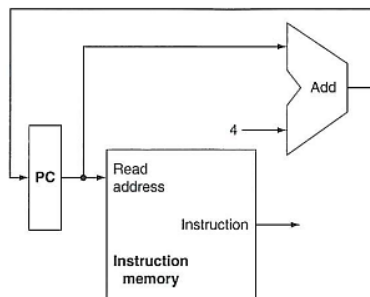
---oOo---

Bài 3. (4.6 – sách tham khảo)

Giả sử các khối trong datapath (hình 1) có độ trễ như sau:

	I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-left-2
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	2ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

1. Giả sử việc duy nhất được thực hiện trong processor chỉ là nạp lệnh liên tục (như hình bên dưới), chu kỳ xung clock cần cho thiết kế là bao nhiêu?



Answer:

- a. Chu kỳ xung clock = I-mem's time = 400 ps
- b. 500 ps

2. Giả sử processor chỉ thực hiện duy nhất mỗi lệnh nhảy (như beq nhưng không cần điều kiện bằng), chu kỳ xung clock cần cho thiết kế là bao nhiêu?

Answer:

Critical path = I-Mem/Sign-extend/shift-left-2/Add/Mux

- a. Chu kỳ xung clock = 400 + 20 + 2 + 100 + 30 = 552 ps

b. Chu kỳ xung clock = $500 + 90 + 20 + 150 + 100 = 860$ ps

3. Như câu 2, nhưng lệnh nhảy trong trường hợp này có xét đến điều kiện bằng (như beq), chu kỳ xung clock cần cho thiết kế là bao nhiêu?

Answer:

Critical path = I-mem/Mux/Registers/Mux/ALU/Mux

a. Chu kỳ xung clock = $400 + 30 + 200 + 30 + 120 + 30 = 810$ ps

b. Chu kỳ xung clock = $500 + 100 + 220 + 100 + 180 + 100 = 1200$ ps

Cho khối chức năng sau:

a.	Add 4 (bộ cộng dùng để cộng PC với 4)
b.	Data Memory

4. Dạng lệnh nào cần các khối chức năng trên

Answer:

a. Tất cả các lệnh, trừ các lệnh nhảy thuộc nhóm “not PC-relative” như j, jal, jr, jalr

b. Các lệnh liên quan đến “load” và “store” như lw, sw,...

5. Dạng lệnh nào mà các khối chức năng trên nằm trong critical path?

Answer:

a. Không có lệnh vào (Vì khối “I-mem” luôn có độ trễ cao hơn “Add 4” và tất cả các lệnh đều cần phải qua “I-mem” cho việc đọc lệnh)

b. Dạng lệnh ‘load’ và ‘store’ (lw, sw, ...)

---oOo---

Bài 4. (4.7 – Sách tham khảo)

Cho độ trễ của các khối trong datapath như sau:

	I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-extend	Shift-left-2
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

1. Chu kỳ xung clock là bao nhiêu nếu datapath chỉ hỗ trợ các lệnh thuộc nhóm logic và số học (như add, and, ...)?

Answer:

Critical path = I-mem/Mux/Regs/Mux/ALU/Mux/Regs

a. Chu kỳ xung clock = $400 + 30 + 200 + 30 + 120 + 30 + 200 = 1010 \text{ ps}$

b. Chu kỳ xung clock = $500 + 100 + 220 + 100 + 180 + 100 + 220 = 1420 \text{ ps}$

2. Chu kỳ xung clock là bao nhiêu nếu datapath chỉ hỗ trợ lệnh lw?

Answer:

Critical path = I-mem/Mux/Regs/Mux/ALU/D-Mem/Mux/Regs

a. Chu kỳ xung clock = $400 + 30 + 200 + 30 + 120 + 350 + 30 + 200 = 1360 \text{ ps}$

b. Chu kỳ xung clock = $500 + 100 + 220 + 100 + 180 + 1000 + 30 + 220 = 2420 \text{ ps}$

3. Chu kỳ xung clock là bao nhiêu nếu datapath hỗ trợ các lệnh: add, beq, lw, sw?

Answer:

a. Chu kỳ xung clock bằng với chu kỳ xung clock của lệnh có critical path dài nhất, bằng chu kỳ xung clock của lệnh lw = 1360 ps

b. Chu kỳ xung clock = 2420 ps

Giả sử tỉ lệ các lệnh được thực hiện trong một đoạn lệnh như sau (Processor không pipeline):

	add	addi	not	beq	lw	sw
a.	30%	15%	5%	20%	20%	10%
b.	25%	5%	5%	15%	35%	15%

4. Bao nhiêu phần trăm chu kỳ xung clock có sử dụng khối “Data memory”?

Answer:

“Data Memory” bị truy xuất với lệnh lw và sw

a. 30%

b. 50%

5. Bao nhiêu phần chu kỳ xung clock có sử dụng khối “Sign-extend”?

Answer:

Khối “sign-extend” đều có tính toán ra kết quả trong mỗi chu kỳ nhưng output của nó chỉ cần cho các lệnh addi, beq, lw và sw.

a. $15\% + 20\% + 20\% + 10\% = 65\%$

b. $5\% + 15\% + 35\% + 15\% = 70\%$

---oOo---

Bài 5. (4.9 – Sách tham khảo)

	Lệnh
a.	lw \$1, 40(\$6)
b.	label: beq \$1, \$2, label

1. Mã máy của hai lệnh trên là gì

Answer:

a) 100011 00110 00001 00000 00000 101000

b) 000100 00001 00010 11111 11111 111111

2. Chỉ số cung cấp cho input “Read register 1”, “Read register 2” của khối “Registers” là gì? Các thanh ghi này có thật sự được đọc và được sử dụng không? (Xem datapath hình 1)

Answer:

a) Read register 1: 00110

Read register 2: 00001

Thanh ghi Read register 2 được đọc nhưng không được sử dụng.

b) Read register 1: 00001

Read register 2: 00010

Cả hai thanh ghi đều được đọc và sử dụng

3. Chỉ số cung cấp cho input “Write register” của khối “Registers” là gì? Thanh ghi này có thật sự được ghi vào không? (Xem datapath hình 1)

Answer:

a) Write register: 00001

Được

b) Write register: Hoặc là 00010 hoặc là 11111 (không biết vì tín hiệu RegDst là ‘x’)

Thanh ghi không được ghi

---oOo---

Bài 6.

Cho một kiến trúc máy tính MIPS với datapath và tín hiệu điều khiển như hình.
Đối với lệnh: *addi Rt, Rs, Imm*

a) Lệnh addi chạy được với datapath như trên không? Những khối nào sẽ cần sử dụng cho lệnh trên, khối nào không cần sử dụng?

Answer:

Được.

Data memory, shift left 2, Add 2

b) Cho biết giá trị của các tín hiệu điều khiển?

Answer:

RegDst	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
0	0	0	0	00	0	1	1

c) Những khối nào có cho dữ liệu output nhưng dữ liệu này không sử dụng? Những khối nào không cho output?

Answer:

Những khối có cho dữ liệu output nhưng không sử dụng: Bộ cộng cho lệnh nhảy

Những khối không cho output: Data memory

d) Giả sử có lệnh mới như sau “*addi Rt, Rs, Rx, Imm*” (ý nghĩa $Rt = Rs + Rx + Imm$) thì phải thay đổi hay thêm vào hình trên những block nào? (1đ)

Answer:

Trường opcode (6 bits), 3 thanh ghi (mỗi thanh ghi 5 bits) → số bits trống còn lại trong format là 11 bits. Vậy trường Imm có thể sử dụng không quá 11 bits. Khối Sign-extend hiện tại là mở rộng số có dấu từ số tức thời 16 bits thành 32 bits, vì vậy cần sử dụng thêm một khối Sign-extend với input là n bits.

---oOo---

Bài 7.

Một bộ xử lý MIPS 32 bits có datapath như hình và thực thi đoạn chương trình assembly như sau: (Biết khi bắt đầu thanh ghi $\$t0 = 0x00000064$ và $\$t1 = 0x100010FC$)

or \$t9, \$zero, \$t0

add \$s0, \$zero, \$t1

sw \$t9, 12(\$s0)

a. Giá trị output của khối “Instruction memory” là bao nhiêu khi bộ xử lý trên thực thi ở câu lệnh thứ 3?

Answer:

101011 10000 11001 00000 00000 001100

b. Khi bộ xử lý trên thực thi ở câu lệnh thứ 3, điền các giá trị cho các thanh ghi, tín hiệu điều khiển và các ngõ input/output của datapath theo yêu cầu của bảng sau:

Ngõ vào/ra		Điều khiển		Kết quả	
Thanh ghi	Giá trị	Tín hiệu	Giá trị	Ngõ	Giá trị
<i>Instruction</i> [25-21]	10000	<i>RegDst</i>	X	<i>ALUResult</i> (Của ALU)	0x10001108
<i>Instruction</i> [20-16]	11000	<i>RegWrite</i>	0	<i>WriteData</i> (của khối <i>Registers</i>)	X
<i>Instruction</i> [15-11]	00000	<i>ALUSrc</i>	1	<i>WriteData</i> (Của khối <i>Data Memory</i>)	0x64
<i>ReadData1</i>	0x100010FC	<i>Branch</i>	0		
<i>ReadData2</i>	0x00000064	<i>MemtoReg</i>	X		
		<i>MemWrite</i>	1		
		<i>MemRead</i>	0		

---oOo---

Bài 8.

Một bộ xử lý MIPS 32 bits (có datapath và control như hình) thực thi đoạn chương trình assembly như sau:

addi \$t0, \$t1, 8

lw \$s0, 4(\$t0)

sw \$t0, 4(\$t0)

Biết khi bắt đầu thanh ghi PC = 0x400000; *\$t1* = 0x10010000; *\$s0* = 0x00000001; word nhớ tại địa chỉ 0x1001000c đang có nội dung (hay giá trị) bằng 0x0000ffff.

Khi bộ xử lý trên thực thi ở câu lệnh thứ hai, điền các giá trị (tín hiệu, input và output) cho từng khối vào bảng sau:

Tên khối	Ngõ	Giá trị
Instruction	Read address	0x400004
	Instruction[31-0]	100011010001000000000000000000100
Memory	Read register 1	01000 → t0
	Read register 2	10000 s0
	Write register	10000 s0
	Write data	0x0000ffff
	Read data 1	0x10010008

	Read data 2	0x00000001
ALU	Input thứ nhất của ALU	0x10010008
	Input thứ hai của ALU	0x00000004
	ALU result	0x1001000C
	Zero	0
Data Memory	Address	0x1001000C
	Write data	0x00000001
	Read data	0x0000ffff
Control	Instruction [31-26]	100011
	RegDst	0
	Branch	0
	MemRead	1
	MemtoReg	1
	ALUOp (Chỉ cần cho biết ALU thực hiện phép toán gì)	Add
	MemWrite	0
	ALUSrc	1
	RegWrite	1

---oOo---

Bài 9.

Cho một bộ xử lý MIPS 32 bits (có datapath và control như hình).

Biết $PC = 0x400000$; $\$t1 = 0x00008000$; $\$t3 = 0x00000015$; *Word nhớ tại địa chỉ* $0x00008008$ *có nội dung/giá trị bằng* $0x00000015$

Nếu đoạn chương trình sau được thực thi:

addi \$s0, \$t1, 4

lw \$t2, 4(\$s0)

beq \$t3, \$t2, ABC

add \$t2, \$t3, \$t4

ABC: sub \$t3, \$t4, \$t5

Khi bộ xử lý trên thực thi ở câu lệnh thứ ba, hỏi:

- Với khối “Instruction Memory” các ngõ “Read address” và “Instruction[31-0]” bằng bao nhiêu

Answer:

Read address : $PC + 8 = 0x400008$

Instruction[31-0] = 000100 01011 01010 00000 00000 000001

- b. Với khối “Registers”, các ngõ “Read register 1”, “Read register 2”, “Write register”, “Write data”, “Read data 1” và “Read data 2”, “RegWrite” bằng bao nhiêu?

Answer:

Read register 1: 01011
Read register 2: 01010
Write register: X
Write data: X

Read data 1: 0x15

Read data 2: 0x15

RegWrite: 0

- c. Với khối “ALU”, input thứ 1, input thứ hai, “ALU result” và “zero” bằng bao nhiêu?

Answer:

Input thứ 1: 0x15
Input thứ 2: 0x15
ALU result: 0
zero: 1

- d. Với khối “Data Memory”, “Address”, “Write data”, “Read data”, “MemWrite”, “MemRead” bằng bao nhiêu?

Answer:

Address: 0
Write data: 0x15
Read data: X
MemWrite: 0
MemRead: 0

- e. Các tín hiệu điều khiển của 3 MUX: RegDst, ALUSrc và MemToReg bằng bao nhiêu?

Answer:

RegDst: X
ALUSrc: 0
MemToReg: X

- f. Đầu vào và đầu ra của khối “Sign-extend” bằng bao nhiêu?

Answer:

Đầu vào: 0000 0000 0000 0001 (16 bits)
Đầu ra: 0000 0000 0000 0000 0000 0000 0000 0001 (32 bits)

- g. Đầu vào và đầu ra của khối “Shift left 2” bằng bao nhiêu?

Answer:

