Họ tên: Vương Đinh Thanh Ngân

MSSV: 20521649

Lớp: IT006.M12

Bài 1:

1.1

- Có pipeline: 450ps

- Không pipeline đơn chu kỳ: 440 + 350 +330 + 450 +180 = 1750 ps

- Không đa chu kỳ: 450 ps

1.2

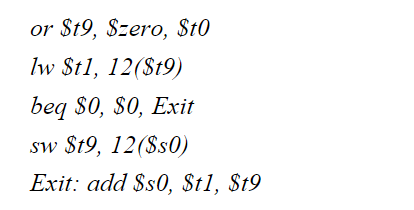
Beq có 4 stage

- beq có pipeline: 450\*5 =2250 ps

- Không pipeline đơn chu kỳ:1750 ps

- Không pipeline đa chu kỳ: 1350 ps

1.3



-Thời gian cần thiết để thực hiện lệnh đoạn chương trình sau cho trường hợp processor có pipeline (không dùng kỹ thuật forwarding):

- Không pipeline đơn chu kỳ:

- Không pipeline đachu kỳ:

Bài 2:

- Proccessor thiết kế theo pipeline 5 tầng không dùng kỹ thuật nhìn trước :chu kỳ xung clock 400ps

a.

sw $2, 12($1)

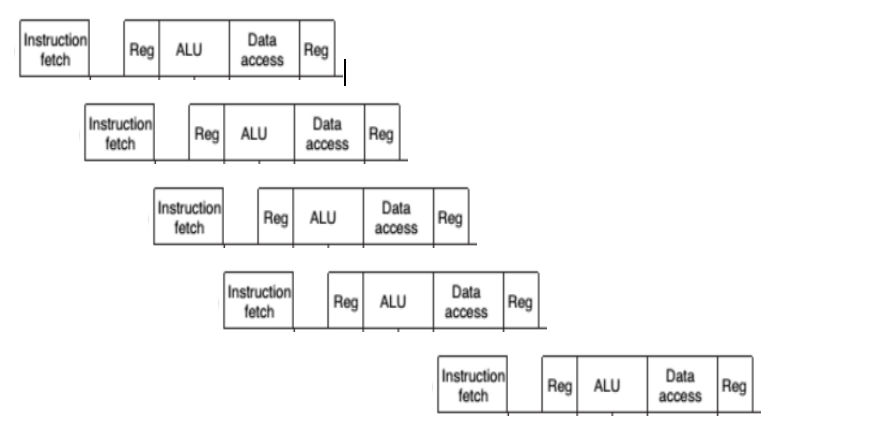
add $3 $2, $1

lw $1, 16($2)

add $1, $3, $2

sw $1, 4($2)

=> Thời gian thực thi là: 11 \*400=4400 ps



b.

lw $3, 8($2)

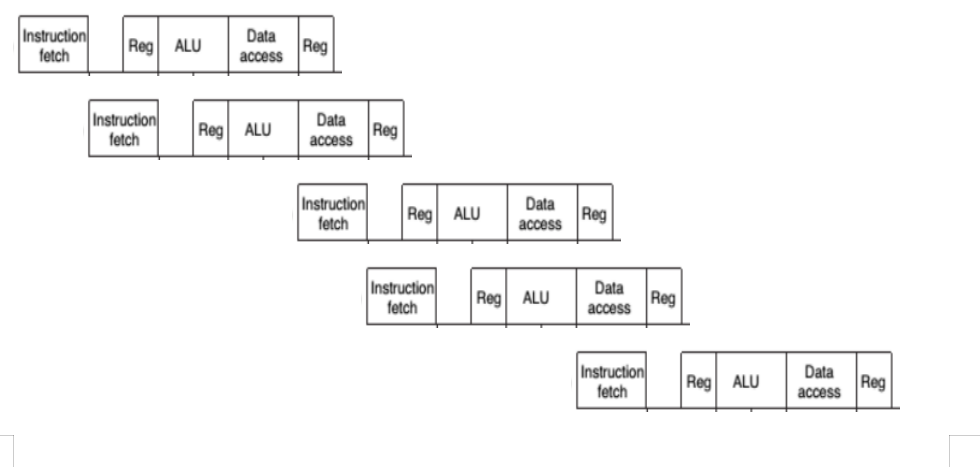
add $3, $2, $1

sw $1, 28($3)

add $1, $1, $3

sw $2, 4($1)

* Thời gian thực thi là: 13 \*400=5200 ps



- Có kỹ thuật nhìn trước (full-forwarding): chu kỳ xung clock 430ps

a.

sw $2, 12($1)

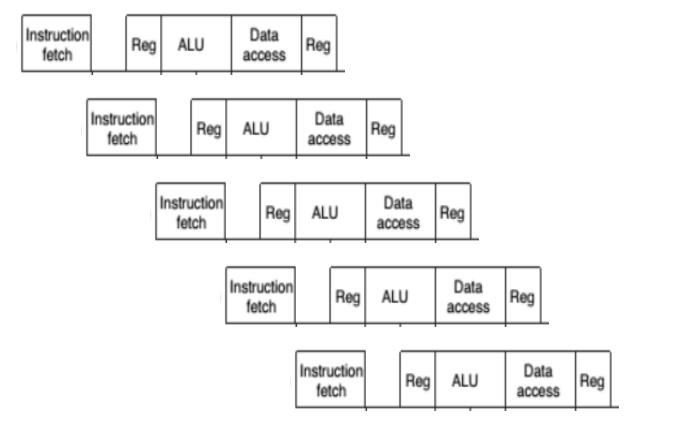
add $3 $2, $1

lw $1, 16($2)

add $1, $3, $2

sw $1, 4($2)

=> Thời gian thực thi là: 9 \*430=3870 ps



b.

lw $3, 8($2)

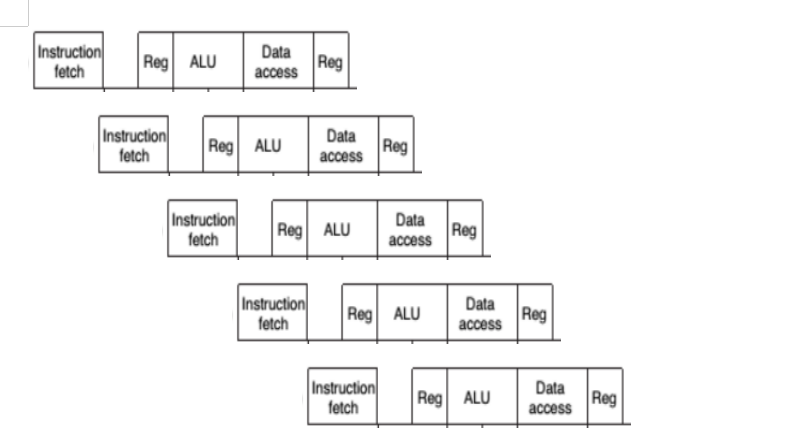
add $3, $2, $1

sw $1, 28($3)

add $1, $1, $3

sw $2, 4($1)

=> Thời gian thực thi là: 9 \*430=3870 ps



Datapath

Bài 1:

1.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| K | Regwrite | Memread | Memwrite | ALUOp | ALUSrc | Memtoreg | Branch |
| a | 1 | 0 | 0 | add | 0 (reg) | 0 (ALU) | 0 |
| b | 1 | 1 | 0 | add | 1 (imm) | 1 (mem) | 0 |

1. Có thể không cần khối “ Data Memory” và bộ cộng dùng trong lệnh nhảy ( Shift left 2, sign - extend và cổng AND), còn lại đều được sử dụng.
2. Bộ cộng dùng cho lệnh nhảy (Shift left 2, cổng AND) có thể không cần dùng, còn lại đều được sử dụng.

3.

|  |  |  |
| --- | --- | --- |
|  | Các khối có output nhưng không sử dụng | Các khối không có output |
| a | Khối lệnh nhảy | Khối Memory |
| b | Khối lệnh nhảy | Không ( tất cả các khối đều có output) |

4.

a.Độ trễ lớn nhất: 400 + 200 + 30+ 120 + 30 + 200 = 980ps

Critical path: I- Mem, Regs, Mux, ALU, Mux, Regs

b.Thời gian trễ lớn nhất là: 500 + 220 + 100 + 180 + 100 + 220 = 1320ps

Critical path: I-Mem, Regs, Mux, ALU, Mux, Reg.

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

a)

+ Critical path: I-Mem, Mux, Regs, Mux, ALU, D-Mem, Mux, Reg.

+ Thời gian trễ lớn nhất là: 400 + 30 + 200 + 30 + 120 + 350 + 30 + 200 = 1360

b)

+ Critical path: I-Mem, Mux, Regs, Mux, ALU, D-Mem, Mux, Reg.

+ Thời gian trễ lớn nhất là: 500 + 100 + 220 + 100 + 180 + 1000 + 100 + 220 = 2420

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

a)

+ Critical path: I-Mem, Mux, Regs, Mux, ALU, Mux.

+ Thời gian trễ lớn nhất là: 400 + 30 + 200 + 30 + 120 + 30 = 810

b)

+ Critical path: I-Mem, Mux, Regs, Mux, ALU, Mux.

+ Thời gian trễ lớn nhất là: 500 + 100 + 220 + 100 + 180 + 100 = 1200

Bài 2:

1.

1. Lệnh này cần sử dụng các khối: instruction memory, registers ( cả 2 cổng đọc và ghi), ALU
2. Lệnh này cần sử dụng các khối: instruction memory (chỉ cần ổng đọc hoặc ghi), registers, đường truyền tới ALU

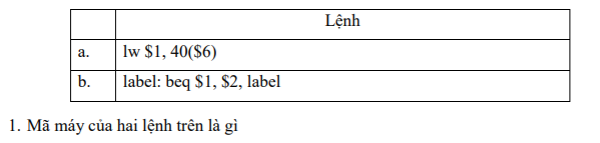
2.

1. Các khối mởi cần thêm vào: cổng đọc ( Registers), ALU để tính tổng Rs + Rt hoặc có thể sữa ALU thành ALU 3 input
2. Các khối mới cần them vào: Khả năng dịch bit của ALU
3. Các tính hiệu điểu khiển mới cần thêm vào:
4. Thêm một tín hiệu điều khiển để điều khiển ALU mới nếu như câu trên chọn thêm 1 ALU mới

Thay đổi lại khối “ ALU control” để điều khiển ALU 3 input nếu câu trên chọn sữa ALU đầu 2 thành đầu 3.

1. Các tính hiệu điểu khiển mới cần thêm vào:
2. Thay đổi khôia” ALU control” để điều khiển ALU có thêm tính năng dịch bit (sll)

Bài 5:



1. Mã máy của các lệnh trên là:

Lw thuộc lệnh I type

Opcode = 2316 = 100011

Rs = 00110

Rt = 00001

Address & const =0000000000101000

1000 1100 1100 0001 0000 0000 0010 10002 = 8CC1002816

Beq thuộc I type

Opcode = 416 = 000100

Rs = 00001

Rt = 00010

Address & const = 1111111111111111

0001 0000 0010 0010 1111 1111 1111 11112 =1022FFFF

**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 2)**

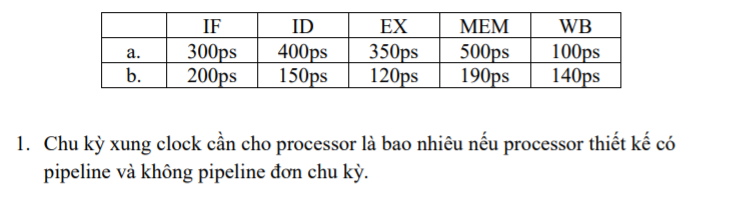
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Register 1 | Được đọc và sử dụng | Register 2 | Được đọc và sử dụng |
| a | $6 = 001102 | Thật sự được đọc và sử dụng | $1 = 000012 | Thật sự được đọc những không sử dụng |
| b | $1 = 000012 | Thật sự được đọc và sử dụng | $2 = 000102 | Thật sự được 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 2)**

|  |  |  |
| --- | --- | --- |
|  | Write register 1 | Được ghi |
| a | $1 = 000012 | Thật sự được ghi |
| B | Tuỳ thuộc vào RegDst (x) mà có thể ra 2 =000102 hoặc 31 = 111112 | Không thật sự được ghi |

PIPELINE

Bài 1: Trong bài tập này, chúng ta khảo sát pipeline ảnh hưởng như thế nào tới chu kỳ xung clock (clock cycle time) của processor. Giả sử rằng mỗi công đoạn (stage) trong pipeline có thời gian thực hiện.



Processor có pipeline: chu kỳ xung clock sẽ là 500ps ( chúng ta chọn công đoạn nhiều thời gian nhất vì có pipeline nên các công đoạn sẽ được điều chỉnh bằng nhau và bằng công đoạn tiêu tốn nhiều thời gian nhất)

Processor không có pipeline: chu kỳ xung clock sẽ bằng thời gian thực hiện tất cả cộng đoạn cộng lại = 300 + 400 + 350 + 500 + 100 = 1650ps

b.

Processor có pipeline: 200ps

Processor không có pipeline = 200 + 150 + 120 + 190 + 140 = 800ps

1. **Thời gian cần thiết để thực hiện lệnh lw cho trường hợp processor có pipeline và không pipeline đơn chu kỳ có thể là bao nhiêu.**

Ta có lệnh lw cần trải qua 5 công đoạn bao gồm: IF, ID, EX, MEM, WB.

Các cộng đoạn có thời gian thực hiện khác nhau những vì sử dụng pipeline nên các cộng đoạn sẽ được chỉnh về cho thời gian mỗi công đoạn là như nhau và bằng với công đoạn lâu nhất.

=> Thời gian thực hiện lệnh lw có pipeline là: 500\*5 = 2500ps

Thời gian thực hiện trong trường hợp không có pipeline: 300 + 400 + 350 + 500 + 100 = 1650ps ( bằng thời gian thực hiện từng công đoạn cộng lại)



Thời gian thực hiện lệnh lw có pipeline là: 200 \* 5 = 1000ps

Thời gian thực hiện trong trường hợp không có pipeline:200 + 150 + 120 + 190 + 140 = 800ps

Hỏi thêm: Thời gian cần thiết để thực hiện lệnh add cho trường hợp processor có pipeline và không pipeline đơn chu kỳ có thể là bao nhiêu.

Ta có lệnh add chỉ cần thực hiện qua 4 công đoạn: IF, ID, EX, WB bỏ qua công đoạn MEM

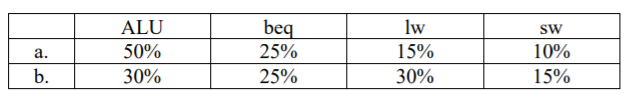
1. Thời gian thực hiện lệnh add có pipeline là: 4 \* 400 = 1600ps

Thời gian thực hiện trong trường hợp không có pipeline: 300 + 400 + 350 +100 = 1150ps

1. Thời gian thực hiện lệnh add có pipeline là: 4 \* 200 = 800

Thời gian thực hiện trong trường hợp không có pipeline: 200 + 150 + 120 + 140 = 610

=>Sai vì pipeline thì vẫn ngồi chờ rỗng trong công đoạn không lam, còn đơn chu kỳ vẫn sẽ cộng luôn thời gian của công đoạn không làm



1. **Giả sử rằng không có khoảng thời gian rỗi (stalls) hoặc xung đột (hazards), phần truy xuất bộ nhớ (MEM) và phần truy xuất ghi trên tập thanh ghi (WB) sử dụng bao nhiêu % chu kỳ của toàn chương trình.**

+ Truy xuất bộ nhớ: chỉ thực hiện trên 2 lệnh là lw and sw

1. 15% + 10% = 25%
2. 30% + 15% = 45%

+ Truy xuất ghi trên tập thanh ghi WB: thao tác truy xuất trên thanh ghi chỉ thực sự thực hiện ở ALU và lw

1. 50% + 15% = 65%
2. 30% + 30% = 60%
3. **Giả sử có thiết kế mới như sau: mỗi lệnh chỉ sử dụng đúng các giai đoạn cần có của nó, có thể lấy nhiều chu kỳ để hoàn thành, nhưng một lệnh phải hoàn thành xong thì những lệnh khác mới được nạp vào. Thiết kế này tạm gọi là thiết kế đa chu kỳ.**

**Theo kiểu này, mỗi lệnh chỉ đi qua những công đoạn mà nó thực sự cần ( sw chỉ sử dụng 4 công đoạn, không có công đoạn WB). Tính chu kỳ xung clock, so sánh thời gian thực thi của thiết kế đa chu kỳ này với thiết kế đơn chu kỳ (single cycle design) và pipeline.**

**(Chú ý: lw: sử dụng 5 stage; sw: 4 stage (không WB); ALU: 4 stage (không MEM), beq 4 stage (không WB))**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | IF | ID | EX | MEM | WB |
| lw | x | x | x | x | x |
| sw | x | x | x | x |  |
| ALU | x | x | x |  | x |
| Branch(beq) | x | x | x | x |  |

Chu kỳ xung clock :

- Có pipeline:

a - 500ps b - 200ps

- Đơn chu kỳ :

a - 1650 b - 800ps

- Đa chu kỳ:

a - 500ps b - 200ps

=> Giống với pipeline

Thời gian thực thi:

- Đơn chu kỳ:

1. Single - cycle excution time : pipeline execution time = 1650 : 500 = 3.3
2. Single - cycle excution time : pipeline execution time = 800 : 200 = 4

- Đa chu kỳ:

1. Multi - cycle excution time : pipeline excution time = 0.15 \* 5 + 0.85 \* 4 = 4.15
2. Multi - cycle excution time : pipeline excution time = 0.3 \* 5 + 0.7 \* 4 = 4,30