



Hình 2

**CÂU HỎI TỰ LUẬN (Đề HK I năm 2014-2015)**

1. Cho đoạn chương trình sau được thực thi trong kiến trúc pipeline 5 tầng:

```
sub $s1, $s2, $t5
sw $s2, 0($s1)
lw $s1, 4($s2)
add $s2, $s2, $s1
```

Sử dụng lệnh *nop* để giải quyết nếu có xung đột dữ liệu trong hai trường hợp sau:

- Không dùng kỹ thuật “forwarding”. Tổng số chu kỳ cần thiết để hoàn thành 4 lệnh trên là bao nhiêu? (0.75đ)
  - Dùng kỹ thuật “forwarding”. Tổng số chu kỳ cần thiết để hoàn thành 4 lệnh trên là bao nhiêu? (0.75đ)
- Chú ý: Vẽ rõ ràng hình ảnh các chu kỳ pipeline khi đoạn lệnh trên thực thi

**Đáp án:**

- Không dùng kỹ thuật “forwarding”

```
sub
nop
nop
sw
lw
nop
nop
add
```

Hình ảnh các chu kỳ pipeline khi đoạn lệnh thực thi:

Chu kỳ:	1	2	3	4	5	6	7	8	9	10	11	12
sub	IF	ID	EX	MEM	WB							
nop												
nop												
sw				IF	ID	EX	MEM	WB				
lw					IF	ID	EX	MEM	WB			
nop												
nop												
add								IF	ID	EX	MEM	WB

## ĐỀ 001

Tổng số chu kỳ cần thiết để hoàn thành 4 lệnh trên là : 12

b. Dùng kỹ thuật “forwarding”

*sub*

*sw*

*lw*

*nop*

*add*

Hình ảnh các chu kỳ pipeline khi đoạn lệnh thực thi:

Clock	1	2	3	4	5	6	7	8	9
sub	IF	ID	EX	MEM	WB				
sw		IF	ID	EX	MEM	WB			
lw			IF	ID	EX	MEM	WB		
nop									
add					IF	ID	EX	MEM	WB

Tổng số chu kỳ cần thiết để hoàn thành 4 lệnh trên là: 9

-----

2. Một bộ xử lý MIPS 32 bits có datapath và control ở hình 2 thực thi đoạn chương trình assembly như sau: (Biết khi bắt đầu thanh ghi  $\$t0 = 0x64$  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? (0.5đ)

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: (1đ)

Ngõ vào		Điều khiển		Kết quả	
Thanh ghi	Giá trị	Tín hiệu	Giá trị	Ngõ	Giá trị
<i>Instruction[25-21]</i>		<i>RegDst</i>		<i>ALUResult</i>	
<i>Instruction [20-16]</i>		<i>RegWrite</i>		<i>WriteData</i> (của khối <i>Registers</i> )	
<i>Instruction [15-11]</i>		<i>ALUSrc</i>		<i>WriteData</i> (Của khối <i>Data</i> <i>Memory</i> )	
<i>ReadData1</i>		<i>Branch</i>			
<i>ReadData2</i>		<i>MemtoReg</i>			
		<i>MemWrite</i>			
		<i>MemRead</i>			

## ĐỀ 001

### Dáp án:

- a. Giá trị output của khối “Instruction memory” khi bộ xử lý trên thực thi ở câu lệnh thứ 3 là mã máy của lệnh “*sw \$t9, 12(\$s0)*”: 0xAE19000C

$$= 101011100001100100000000000001100_{(2)}$$

b.

Ngõ vào		Điều khiển		Kết quả	
Thanh ghi	Giá trị	Tín hiệu	Giá trị	Ngõ	Giá trị
<i>Instruction[25-21]</i>	16/10000 <sub>2</sub>	<i>RegDst</i>	X	<i>ALUResult</i> (của ALU)	0x10001108
<i>Instruction [20-16]</i>	25/11001 <sub>2</sub>	<i>RegWrite</i>	0	<i>WriteData</i> (của khối Registers)	X
<i>Instruction [15-11]</i>	0/00000 <sub>2</sub>	<i>ALUSrc</i>	1	<i>WriteData</i> (Của khối Data Memory)	0x64
<i>ReadData1</i>	0x100010FC	<i>Branch</i>	0		
<i>ReadData2</i>	0x64	<i>MemtoReg</i>	X		
		<i>MemWrite</i>	1		
		<i>MemRead</i>	0		

X tức là bằng 0 hay 1 đều được, không quan tâm vì giá trị này sẽ không được sử dụng, không ảnh hưởng đến lệnh đang chạy.

Lưu ý:

Nếu *ALUResult* là ngõ ra của bộ cộng sau khối “Shift left 2”, trả lời *ALUResult* = 0xC + PC (tại lệnh thứ 3) + 4

Dữ liệu cần thiết:

## MIPS Reference Data

①



### CORE INSTRUCTION SET

NAME, MNEMONIC	FOR-MAT	OPERATION (in Verilog)	OPCODE / FUNCT (Hex)
Add	add R	$R[rd] = R[rs] + R[rt]$	0 / 20 <sub>hex</sub>
Or	or R	$R[rd] = R[rs]   R[rt]$	0 / 25 <sub>hex</sub>
Store Word	sw I	$M[R[rs]+SignExtImm] = R[rt]$	2b <sub>hex</sub>

### BASIC INSTRUCTION FORMATS

R	opcode	rs	rt	rd	shamt	funct
	31	26 25	21 20	16 15	11 10	6 5
I	opcode	rs	rt	immediate		
	31	26 25	21 20	16 15	0	
J	opcode	address				
	31	26 25	0			

### Registers:

NAME	NUMBER	USE
\$zero	0	The Constant Value 0
\$at	1	Assembler Temporary
\$v0-\$v1	2-3	Values for Function Results and Expression Evaluation
\$a0-\$a3	4-7	Arguments
\$t0-\$t7	8-15	Temporaries
\$s0-\$s7	16-23	Saved Temporaries
\$t8-\$t9	24-25	Temporaries
\$k0-\$k1	26-27	Reserved for OS Kernel
\$gp	28	Global Pointer
\$sp	29	Stack Pointer
\$fp	30	Frame Pointer
\$ra	31	Return Address

Duyệt đề của Khoa/Bộ môn

Nguyễn Minh Sơn

Giảng viên ra đề

Nguyễn Minh Sơn