

Ôn Tập Giữa Kỳ

Hà Lê Hoài Trung

Nội dung – Hình thức

- Thời gian: 60' – 90'
- Số câu: 4 – 5 câu
- Nội Dung ôn tập: Chương 0, Chương 1, Chương 2
- Tham khảo tài liệu: tài liệu mở - 1 tờ giấy A4

Chương 0

- Hệ thống số
 - Nhị phân
 - Bát phân
 - Thập phân
 - Thập lục phân
- Chuyển đổi giữa các hệ thống số
- Thực hiện phép toán $+$, $-$, $*$, số bù 2
- Biểu thức luận lý – Mạch luận lý
 - Mạch giải mã – mạch mã hóa
 - Mạch phân kênh – mạch dồn kênh

Chương 0

- Thực hiện phép chuyển đổi giữa các hệ thống số

Decimal	Binary	Octal	Hexadecimal
43			
	1100110		
		237	
			2EA

Chương 0

- Chuyển các số thập phân sau sang dạng số nhị phân, thực hiện các phép toán:

$$23 + 34$$

$$35 - 22$$

$$125 * 123$$

- Biểu diễn số âm sau sang dạng số bù 2:

$$- 234$$

$$- 112$$

Chương 0

- Xây dựng bộ Mux 16 – 1, chỉ sử dụng các bộ mux 8 – 1

Chương 1

1. Introduction
2. Below your program
3. Under the Covers
 - Resolution
 - Main Memory
 - Ethernet network
4. Performance

Chương 1

4. Performance (thời gian thực thi): số lệnh, tần số xung, số xung clock cho 1 lệnh.

$$\text{Performance}_x = \frac{1}{\text{Execution time}_x}$$

$$\text{CPU execution time for a program} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}} \times \text{Clock cycle time}$$

Alternatively, because clock rate and clock cycle time are inverses,

$$\text{CPU execution time for a program} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

$$\text{CPU clock cycles} = \text{Instructions for a program} \times \text{Average clock cycles per instruction}$$

Chương 1

4. Performance (thời gian thực thi): số lệnh, tần số xung, số xung clock cho 1 lệnh.

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

or, since the clock rate is the inverse of clock cycle time:

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}}$$

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

$$\text{MIPS} = \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

Chương 1

- Consider two different implementations of the same instruction set architecture. There are four classes of instructions, A, B, C, and D. The clock rate and CPI of each implementation are given in the following table.

	Clock rate	CPI Class A	CPI Class B	CPI Class C	CPI Class D
P1	3 Ghz	1	2	3	4
P2	4 Ghz	2	2	2	2

- Given a program with 10^8 instructions divided into classes as follows: 10% class A, 20% class B, 40% class C, and 30% class D, which implementation is faster?

Chương 1

- What is the global CPI for each implementation?
- Find the clock cycles required in both cases

	Clock rate	CPI Class A	CPI Class B	CPI Class C	CPI Class D
P1	3 Ghz	1	2	3	4
P2	4 Ghz	2	2	2	2

Chương 1

- Assuming that arith instructions take 1 cycle, load and store 5 cycles, and branches 2 cycles, what is the execution time of the program in a 3 GHz processor?

Arith	Store	Load	Branch	Total
500	25	100	25	650

- Find the CPI for the program
- If the number of load instructions can be reduced by one half, what is the speedup and the CPI?

Chương 2

- Hiểu được các lệnh biểu diễn trong máy tính
- Toán hạng (địa chỉ - mips)
- Cách biểu diễn số có dấu và không có dấu trong máy tính.
- Các phép toán luận lý
- Các câu lệnh điều khiển
- Hàm, thủ tục
- Dạng bài tập:
 - Viết được đoạn chương trình bằng tập lệnh ngôn ngữ cấp cao chuyển sang ngôn ngữ MIPS (for, while, do ... while, if ... else, switch ... case, procedure).
 - Cho lệnh MIPS chuyển sang biểu diễn dạng mã nhị phân.
 - Đọc kết quả của các phép toán.
 - Biểu diễn số trong máy tính – số âm, số dương.
 - Xác định địa chỉ toán hạng.

Chương 2

Operands of the computer hardware

1. Register Operands
2. Memory Operands
3. Constant or Immediate Operands

Chương 2

- **Instructions for Making Decisions**
 - Chuyển từ ngôn ngữ cấp cao – assembly MIPS
 - Giá trị thanh ghi

Conditional branch	branch on equal	beq \$s1,\$s2,25	if (\$s1 == \$s2) go to PC + 4 + 100	Equal test; PC-relative branch
	branch on not equal	bne \$s1,\$s2,25	if (\$s1 != \$s2) go to PC + 4 + 100	Not equal test; PC-relative
	set on less than	slt \$s1,\$s2,\$s3	if (\$s2 < \$s3) \$s1 = 1; else \$s1 = 0	Compare less than; for beq, bne
	set on less than unsigned	sltu \$s1,\$s2,\$s3	if (\$s2 < \$s3) \$s1 = 1; else \$s1 = 0	Compare less than unsigned
	set less than immediate	slti \$s1,\$s2,20	if (\$s2 < 20) \$s1 = 1; else \$s1 = 0	Compare less than constant
	set less than immediate unsigned	sltiu \$s1,\$s2,20	if (\$s2 < 20) \$s1 = 1; else \$s1 = 0	Compare less than constant unsigned
Unconditional jump	jump	j 2500	go to 10000	Jump to target address
	jump register	jr \$ra	go to \$ra	For switch, procedure return
	jump and link	jal 2500	\$ra = PC + 4; go to 10000	For procedure call

Chương 2

- **MIPS Addressing Mode Summary:**
 - **Immediate addressing**, where the **operand** is a **constant** within the instruction itself
 - **Register addressing**, where the operand is a **register**
 - **Base or displacement addressing**, where the operand is at the memory location whose address is *the sum of a register and a constant* in the instruction
 - **PC-relative addressing**, where the branch address is **the sum of the PC and a constant** in the instruction
 - **Pseudo direct addressing**, where the **jump** address is the **26 bits** of the instruction concatenated with the upper bits of the PC

Chương 2

- Chuyển từ ngôn ngữ cấp cao sang ngôn ngữ cấp thấp:
 - Tập lệnh – 2 trang đầu tiên, phụ lục tham khảo B của sách, hoặc tối thiểu là tập lệnh giới thiệu trong slide.
 - Truy xuất phần tử trong mảng
 - Chỉ số mảng * 4 $\rightarrow$ xác định khoảng cách từ địa chỉ base đến địa chỉ ô nhớ cần xác định.
 - Địa chỉ base + khoảng cách.

Chương 2

- Chuyển từ ngôn ngữ cấp cao sang ngôn ngữ cấp thấp (tt):

- Chuyển lệnh lặp

```
for (int i = 0; i < n; i++)  
{  
    Statement;  
}
```

- Chuyển lệnh if

```
if( condition){  
    Statement 1;  
} else {  
    Statement 2;  
}
```

Chương 2

- Chuyển từ ngôn ngữ cấp cao sang ngôn ngữ cấp thấp (tt):
 - Thủ tục, hàm
 - Put **parameters** in a place where the procedure can access them. $\$a0 - \$a3$
 - Transfer **control** to the procedure. *jal name_procedure*
 - Acquire the **storage resources** needed for the procedure. Lưu các tham số cần thiết và địa chỉ trả về cho hàm vào trong stack, stack pointer sẽ giảm.
 - Put the **result** value in a place where the **calling program** can access it. $\$v0 - \$v1$: **two value registers** in which to return values
 - **Return** control to the point of origin. *jr \$ra*: **one return address register** to return to the point of origin – địa chỉ này lấy từ stack pointer.

Chương 2

- Bài tập
 - Biểu diễn đoạn mã C sau qua lệnh assembly MIPS

a.	<pre>for(i=0; i<a; i++) a += b;</pre>
b.	<pre>for(i=0; i<a; i++) for(j=0; j<b; j++) D[4*j] = i + j;</pre>

- Vẽ lược đồ thực thi của lệnh C
- Chuyển mã trên sang dạng mã hợp ngữ MIPS.
Dùng lệnh tối thiểu. Giả sử: a, b, i, j nằm trong các thanh ghi \$s0, \$s1, \$t0, và \$t1. Địa chỉ nền của mảng D nằm ở thanh ghi \$s2

Chương 2

Chương 2

- Representing Instructions in the Computer
 - R-type(for register) or R-format
 - I-type (for immediate) or I-format

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

Instruction	Format	op	rs	rt	rd	shamt	funct	address
add	R	0	reg	reg	reg	0	32 _{ten}	n.a.
sub (subtract)	R	0	reg	reg	reg	0	34 _{ten}	n.a.
add immediate	I	8 _{ten}	reg	reg	n.a.	n.a.	n.a.	constant
lw (load word)	I	35 _{ten}	reg	reg	n.a.	n.a.	n.a.	address
sw (store word)	I	43 _{ten}	reg	reg	n.a.	n.a.	n.a.	address

Chương 2

- Representing Instructions in the Computer
 - Cho 1 chuỗi các bit nhị phân sau:
1010 1101 0010 1000 0000 0100 1000 0000
 - Chuỗi trên nếu biểu diễn số nguyên không dấu. Giá trị bao nhiêu.
 - Chuỗi trên nếu biểu diễn số nguyên có dấu (dạng bù 2). Giá trị bao nhiêu.
 - Nếu chuỗi trên lưu trữ trong thanh ghi lệnh, thì thực hiện lệnh MIPS gì? Lệnh đó biểu diễn như thế nào

Chương 2

- **Logical Operations**
 - Shift
 - And
 - Or
 - Not
 - Nor

Chương 2

- Given \$t1:

0011 1111 1111 1000 0000 0000 0000 0000_{two}

- Give \$t0:

0010 0100 1001 0010 0100 1001 0010 0100

What is the value of \$t2 after the following instructions?

```
    slt  $t2, $t0, $t1
    beq  $t2, $zero, ELSE
    j     DONE
ELSE: addi $t2, $zero, 2
DONE:
```

Chương 2

- Chuyển mã assembly MIPS sang dạng mã máy MIPS:

sll \$s2, \$s4, 12

addi \$t0, \$t0, -1

Chương 2

- Chuyển dạng mã máy MIPS sang dạng mã assembly MIPS

00001025_{hex}

0005402A_{hex}