KIỂM TRA GIỮA KỲ
MÔN KIẾN TRÚC MÁY TÍNH
Thời gian: 60 phút
Sinh viên được xem tài liệu

---

1. Nêu 6 yếu tố ảnh hưởng đến hiệu quả của một máy tính?(1đ)

   Algorithm, Programming language, Compiler, ISA, Core Organization, Technology

2. Giả sử một chương trình thực thi trên một máy tính mất 90s trong đó số lệnh load chiếm 55s, hỏi phải cải thiện thời gian thực thi số lệnh load còn bao nhiêu giây để chương trình chạy nhanh hơn 3 lần. (0.75đ)

   $T_{after} = T_{unaffected} + T_{improved}$

   ➔ $30s = 35s + T_{improved}$

   ➔ $T_{improve} < 0$

   ➔ *Mission impossible*

3. Chuyển đoạn lệnh sau sang hợp ngữ MIPS: giả sử thanh ghi $s0 chứa **địa chỉ** của biến sensor. Hai thanh ghi $a0 và $a1 chứa 2 tham số được truyền cho thủ tục speed. (2đ)

```
while(1)
{
  if ((sensor & 0x00ffff00) != 0)
      speed(100,100);
  else if ((sensor & 0xff000000) != 0)
     speed(50,150);
  else if ((sensor & 0x000000ff) != 0)
     speed(150,50);
  else
     speed(0,0);
}
```

➔ *Hợp ngữ MIPS*

```
      li $t0, 0x00ffff00
      li $t1, 0xff000000
loop: lw $t2, 0($s0)
      addi $t3,$t2,$zero
      and $t3,$t3,$t0
      beq $t3,$zero,next1
      addi $a0,$zero,100
      addi $a1,$zero,100
      jal speed
      j loop
next1:
      addi $t3,$t2,$zero
      and $t3,$t3,$t1
      beq $t3,$zero,next2
      addi $a0,$zero,50
      addi $a1,$zero,150
      jal speed
      j loop
next2:
      addi $t3,$t2,$zero
```

```
        andi $t3,$t3,0x00ff
        beq $t3,$zero,next3
        addi $a0,$zero,150
        addi $a1,$zero,50
        jal speed
        j loop
next3:
        addi $a0,$zero,0
        addi $a1,$zero,0
        jal speed
        j loop
```

Chuyển thủ tục speed sang hợp ngữ MIPS. Giả sử thanh ghi $s1 và $s2 chứa địa chỉ của 2 biến Motor_Left và Motor_Right. (1đ)

```
void speed(int left, int right)
{
  if(left < 0)
     Motor_Left = 0;
  else
     Motor_Left = left;

  if(right < 0)
     Motor_Right = 0;
  else
     Motor_Right = right;
}
```

➔ *Hợp ngữ MIPS*

```
speed:
        slt $a3, $a0, $zero
        beq $a3,$zero, left_motor
        sw  $zero, 0($s1)
        j   right
left_motor:
        sw  $a0, 0($s1)
right:
        slt $a3, $a1, $zero
        beq $a3,$zero, right_motor
        sw  $zero, 0($s2)
        jr  $ra
right_motor
        sw  $a0, 0($s2)
        jr $ra

#chú ý: nếu sử dụng các thanh ghi $t0-$t3 : cần cất giữ và phục hồi trước và sau khi sử dụng
```

4. Giả sử một chương trình đồ họa thực thi trên một máy tính mà số lệnh tính toán số học chiếm 30%, hỏi phải cải thiện thời gian thực thi các lệnh còn lại bao nhiêu lần để toàn bộ chương trình chạy nhanh gấp 1,5 lần so với trước khi cải tiến.(0.75đ)

   Theo Amdahl Law

   Speed-up = 1/[(1-F)+F/S]

   F = 70%, Speed-up = 1.5

   ➔ *F/S = 1/Speed-up – (1 - F)*

   ➔ *F/S = 1/1.5 – (1 – 0.7) = 11/30*

➔ *S = F/ (11/30) = 0.7 / (11/30) = 21/11 = 1.9*

5. Cho 3 CPU thực thi cùng một kiến trúc tập lệnh với bảng mô tả CPI và tốc độ clock như bảng sau:

Processor	Clock rate	CPI
P1	2.5GHz	1.6
P2	3.2GHz	2.0
P3	2.0GHz	1.2

a. CPU nào có hiệu xuất cao nhất (nếu thực thi cùng một chương trình) (0.75đ)
Vì cùng một kiến trúc tập lệnh và số lệnh thực thi (cùng chương trình) nên ta có thể dùng MIPS để so sánh
MIPS = (Clock rate)/(CPI x 10^6)
$MIPS_{P1}$ = (2.5x10^9)/(1.6x10^6) = 1562.5 MIPS
$MIPS_{P2}$ = (3.2x10^9)/(2x10^6) = 1600 MIPS
$MIPS_{P3}$ = (2.0x10^9)/(1.2x10^6) = 1666.667 MIPS
➔ *P3 có hiệu xuất cao nhất*

b. Nếu một chương trình được thực thi trong 15s. Tính số lệnh cần thực hiện cho mỗi CPU (0.75đ)

CPU_Time = Instruction_count x CPI / Clock_rate

➔ *Instruction_count = CPU_Time x Clock_rate/CPI*

➔ *IC_{P1} = 15s x 2.5 x 10^9 / 1.6 = 23.4375 x 10^9*

➔ *IC_{P2} = 15s x 3.2 x 10^9 / 2 = 24 x 10^9*

➔ *IC_{P2} = 15s x 2 x 10^9 / 1.2 = 25 x 10^9*

6. Cho bảng mô tả việc biên dịch của 2 trình biên dịch cho 1 chương trình khi thực thi trên cùng một CPU

Compiler A		Compiler B	
Số lệnh	Thời gian	Số lệnh	Thời gian
10^9 lệnh	1s	1.2x10^9 lệnh	1.4s

a. Tính CPI trung bình của chương trình cho mỗi compiler khác nhau, giả sử chu kì clock là 1ns (0.75đ)
CPI = CPU_Time/(Instruction_count x Clock_cycle)
➔ *CPI_A = 1/(10^9 x 10^{-9}) = 1*
➔ *CPI_B = 1.4/(1.2 x 10^9 x 10^{-9}) = 1.17*

b. Nếu một compiler mới (compiler C) biện dịch chương trình chỉ có 6x10^8 lệnh và CPI là 1.1 thì speed-up của compiler mới này so với A và B là bao nhiêu? (0.75đ)

CPU_Time = CPI x Instruction_count x Clock_cycle

➔ *CPU_Time_C = 1.1 x 6 x 10^8 x 10^{-9} = 0.66s*

➔ *$Speed_up_{AC}$ = 1/0.66 = 1.51*

➔ *$Speed_up_{BC}$ = 1.4/0.66 = 2.12*

7. Cho bảng sau mô tả dữ liệu của 2 benchmarks

Tên	CPI	Clock rate	SPECratio
abc	1.2	3.6GHz	15
def	1.9	3.6GHz	8.5

a. Giả sử số lệnh của mỗi chương trình tăng lên 15% (và CPI không đổi) thì thời gian thực thi của mỗi benchmark tăng lên bao nhiêu? (0.75đ)

CPU_Time = CPI x Instruction_count x Clock_cycle

➔ *Thời gian thực thi tăng lên 15%*

b. Giả sử SPECratio của chương trình abc được cải thiện thành 20, hỏi CPI và số lệnh của abc phải cải thiện bao nhiêu lần (clock rate không đổi). (0.75đ)

SPECratio = CPU_Time_{ref}/CPU_Time

➔ *$SPECratio_{old}$/$SPECratio_{new}$ = CPU_Time_{new} / CPU_Time_{old}*

➔ *CPI_{new} x IC_{new} / (CPI_{old} x IC_{old}) = 15/20 = 3/4*

➔ *Có nhiều đáp án miễn thỏa mãn đk trên*