TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI **VIỆN ĐIỆN TỬ - VIỄN THÔNG** **Đề số: 1 Tổng số trang: 2**		**ĐỀ THI MÔN: KIẾN TRÚC MÁY TÍNH** *Lần thi: 1 Ngày thi: 29/12/2014* *Thời gian làm bài: 90 phút* ***(Được sử dụng tài liệu tham khảo, nộp đề thi cùng với bài làm)***
Ký duyệt	Trưởng nhóm Môn học:	Trưởng Bộ môn:

Câu 1: (3điểm)

1/ Hiển thị quá trình thực thi lệnh của bộ xử lý MIPS 5 giai đoạn đường ống khi thực thi đoạn mã lệnh và phân tích xung đột dữ liệu và điều khiển trong 1 vòng lặp.

(1) Loop: lw $S1, 0($S2); #$S1 = Mem[0 +$S2]

(2) lw $S3, 1000($S2); #$S3 = Mem[1000+$S2]

(3) add $S1, $S1, $S3; #$S1 = $S1 + $S3;

(4) sw $S1, 2000($S2); #Mem[2000+$S2] = $S1;

(5) lw $S4, -4($S2); #$S4 = Mem[$S2 - 4];

(6) lw $S5, 996($S2); #$S5 = Mem[$S2 + 996];

(7) add $S4, $S4, $S5; #$S4 = $S4 + $S5

(8) sw $S4, 1996($S2); #Mem[$S2 + 1996] = $S4

(9) addi $S2, $S2, -8; #$S2 = $S2 - 8

(10) bne $S2, $zero, Loop; #$S2 < 0 nhảy đến Loop

2/ Hãy tính CPI trong trường hợp giải quyết xung đột bằng phương pháp dừng và đợi, giả sử để giải quyết xung đột điều khiển chỉ cần 2 chu kỳ chờ, biết giá trị khởi tạo của thanh ghi $S2 = 64, các chu kỳ chờ khi có xung đột biểu diễn bằng dấu *.

Trả lời

$S2 = 64 -> chương trình thực thi 8 vòng lặp.

Số lệnh trong chương trình là : 8 * 10 = 80 lệnh

Số chu kỳ thực thi khi giải quyết xung đột bằng phương pháp dừng và đợi : 162 chu kỳ

CPI = 162/80 = 2.025 chu kỳ.

Sơ đồ phân tích xung đột

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	lw	I	D	E	M	W												
2	lw		I	D	E	M	W											
3	add			I	D	E	M	W										
4	sw				I	D	E	M	W									
5	lw					I	D	E	M	W								
6	lw						I	D	E	M	W							
7	add							I	D	E	M	W						
8	sw								I	D	E	M	W					
9	addi									I	D	E	M	W				
10	bne										I	D	E	M	W			
1	lw											I	D	E	M	W		
2	lw												I	D	E	M	W	
3	…			…														

Sơ đồ giải quyết xung đột bằng phương pháp dừng và đợi

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	lw	I	D	E	M	W																			
2	lw		I	D	E	M	W																		
3	add			*	*	I	D	E	M	W															
4	sw						*	*	I	D	E	M	W												
5	lw									I	D	E	M	W											
6	lw										I	D	E	M	W										
7	add											*	*	I	D	E	M	W							
8	sw														*	*	I	D	E	M	W				
9	addi																	I	D	E	M	W			
10	bne																		*	*	I	D	E	M	W
1	lw																					*	*	I	D
2	lw																					Chờ 2 chu kỳ			I
3	…																								

3/ So sánh trong trường hợp giải quyết xung đột bằng phương pháp chuyển tiếp dữ liệu và rẽ nhánh chậm.

Trả lời:

Số chu kỳ thực thi : 84 chu kỳ

CPI = 84/80 = 1.05 chu kỳ

4/ Máy tính có tham số bộ nhớ như sau: Kích thước bộ nhớ chính 4Gbyte, kích thước bộ đệm 64Kbyte, kích thước 1 khối bộ đệm là 32byte, bộ nhớ ánh xạ trực tiếp. Xác định số bit của trường Tag, block offset, word offset.

Trả lời :

Số khối trong bộ nhớ đệm : 64Kbyte / 32 byte = 2^{11} khối - > b = 11 bits.

Số từ trong khối: 32byte / 4byte = 8 từ -> w = 3 bits.

Dung lượng bộ nhớ chính: 4Gbyte = 2^{32} byte. Số bit địa chỉ : n = 32 bit.

Số bit trường Tag: n – w – b – 2 = 32 – 11 – 3 – 2 = 16 bit.

5/ Biết đoạn mã lệnh trên bắt đầu từ lệnh có địa chỉ 0x80000000, xác định giá trị của trường Tag, block offset, word offset trong bộ đệm lệnh và bộ đệm dữ liệu khi toàn bộ chương trình trên được ghi vào bộ đệm.

Địa chỉ bộ nhớ	Lệnh	Caches
0x80000000	1	C(0,0)
0x80000004	2	C(0,1)
0x80000008	3	C(0,2)
0x8000000C	4	C(0,3)
0x80000010	5	C(0,4)
0x80000014	6	C(0,5)
0x80000018	7	C(0,6)
0x8000001C	8	C(0,7)
0x80000020	9	C(1,0)
0x80000024	10	C(1,2)

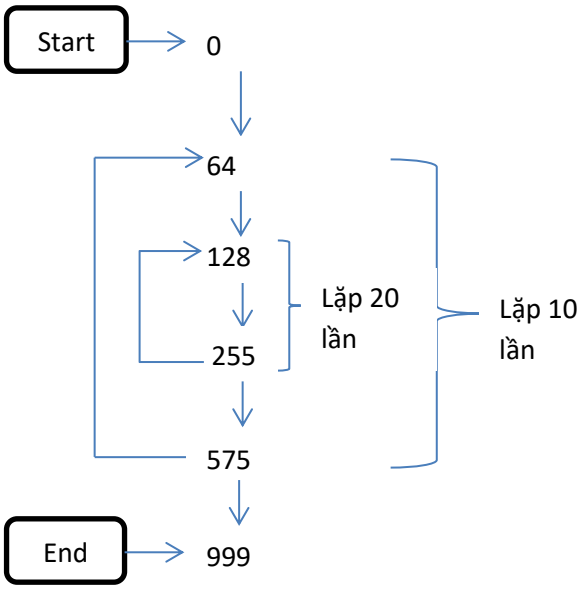
Caches :

words

Block	Tag	0	1	2	3	4	5	6	7
1	0x8000	Lệnh 1	Lệnh 2	Lệnh 3	Lệnh 4	Lệnh 5	Lệnh 6	Lệnh 7	Lệnh 8
2	0x8000	Lệnh 9	Lệnh 10						
3									
4									
5									
6									
7									
8									
9									
10									
11									
12	…….								

Câu 2: (4 điểm)

Một chương trình gồm 2 vòng lặp : Vòng lặp nhỏ bên trong và vòng lặp lớn bên ngoài, theo cấu trúc chương trình biểu diễn theo hình vẽ. Địa chỉ bộ nhớ biểu diễn trong hệ nhị phân, các địa chỉ tương ứng với địa chỉ trong bộ nhớ lệnh, thực thi tuần tự từ trên xuống dưới. Các lệnh có địa chỉ từ 128 - >255 thuộc vòng lặp nhỏ bên trong, lặp 20 lần tương ứng với mỗi vòng lặp lớn bên ngoài. Các lệnh có địa chỉ từ 64 -> 127 và 256 - > 575 thuộc vòng lặp lớn bên ngoài và mỗi lệnh lặp 10 lần tương ứng. Chương trình chạy trên máy tính có bộ nhớ đệm kiểu ánh xạ trực tiếp, bộ nhớ đệm có 64 khối (block), mỗi khối chứa 4 lệnh. Thời gian truy cập trúng bộ đệm (Hit time) là 1 chu kỳ, thời gian truy cập trượt (Miss time) là 20 chu kỳ. Giả sử thời điểm bắt đầu chạy chương trình là bộ đệm trống, không chứa nội dung.



1. Tính tổng số lệnh của chương trình và số lần truy cập vào bộ đệm lệnh (access time), và tổng số lần truy cập trượt bộ đệm lệnh.
   0 - > 63: 64 lệnh = 16 I – block -> 16 I – cache Miss
   64 - > 127: 64 lệnh = 16 I – block -> 16 I – cache Miss
   128 - > 255: 128 lệnh = 32 I – block -> 32 I – cache Miss (01 lần lặp đầu tiên của vòng lặp nhỏ bên trong sẽ bị trượt bộ đệm)
   256 - > 575: 320 lệnh = 80 I – blocks -> 80 I – cache Miss
   576 -> 999: 424 lệnh = 106 I – blocks -> 106 I – cache Miss

   Tổng số lệnh trượt bộ đệm là : 16+16+32+80+106 = 250
   Tổng số lệnh thực thi là : t1 = 64 + 64 *10 + 128 *20*10 + 320 * 10 + 424 = 29928 lệnh

2. Tính tỉ lệ trượt bộ đệm lệnh (miss rate). Biết miss rate = miss time/ access time.
   250/29928 = 0.83%

3. Tính thời gian thực thi chương trình trên tương ứng với bộ xử lý đường ống có tốc độ 2GHz. Giả sử chỉ chờ trong trường hợp trượt bộ đệm.

Tổng số chu kỳ thực thi : 29928 * 1 + 250*20 = 34928
Thời gian thực thi : 34928*0.5ns = 17464ns.

4. Nếu thay thế bộ đệm có kích thước khối lớn hơn, mỗi khối lưu trữ 8 lệnh. Tổng số khối của bộ đệm lệnh vẫn là 64. So sánh thời gian thực thi với ý 3.
   0 - > 63: 64 lệnh = 8 I – block -> 8 I – cache Miss
   64 - > 127: 64 lệnh = 8 I – block -> 8 I – cache Miss
   128 - > 255: 128 lệnh = 16 I – block -> 16 I – cache Miss (01 lần lặp đầu tiên của vòng lặp nhỏ bên trong sẽ bị trượt bộ đệm)
   256 - > 575: 320 lệnh = 40 I – blocks -> 40 I – cache Miss
   576 -> 999: 424 lệnh = 53 I – blocks -> 53 I – cache Miss

   Tổng số lệnh trượt bộ đệm là : 125
   Tổng số lệnh thực thi là : 64 + 64 *10 + 128 *20*10 + 320 * 10 + 424 = 29928 lệnh
   Tỷ lệ trượt bộ đệm lệnh : 125/29928 = 0.418%
   Thời gian thực thi: t2 = 16214ns = t1/2

Câu 3: (3 điểm)

Thêm các chú thích để mô tả quá trình thực thi lệnh trong chương trình sau. Giả sử thanh ghi $a0 chứa biến số nguyên n và thanh ghi $v0 để lưu kết quả. Viết lại chương trình dưới đây thành code C tương ứng biết thanh ghi $t0,$t1 lưu các biến a,b của chương trình. Tính giá trị kết quả trên $v0 theo n.

```
Begin:   addi $t0, $zero, 0; # $t0 = 0;
         addi $t1, $zero, 1; # $t1 = 1;
loop:    slt $t2, $a0, $t1; # Nếu n < $t1 gán $t2 = 1;
         bne $t2, $zero, finish; # Nếu $t2 khác 0 nhảy đến nhãn Finish
         add $t0, $t0, $t1;  # $t0 = $t0 + $t1;
         addi $t1, $t1, 2; # $t1 = $t1 +2;
         j loop              # nhảy tới nhãn loop
finish:  add $v0, $t0, $zero; # $v0 = $t0 + 0;
```

Trả lời :

```
int n, a, b;
a = 0;
b = 1;
while (n>= a) do
```

```
{a = a + b;
 b = b +2;
}
result = a;
```