

2014-1학기 컴퓨터구조 중간시험 채점 기준표

<p>1. 10점</p>	<p>Caching(총 4점) 가격이 높지만 속도가 빠른 메모리와 가격은 낮지만 속도가 느린 메모리를 계층 구조화하고(1) 메모리 참조의 non-uniform한 특징인 temporal/spatial locality를 이용하여(1) 가격은 하위 계층의 memory에 가까우면서 속도는 상위 계층의 memory와 비슷한 저장장치라는 illusion을 제공함.(2)</p> <p>Branch prediction(총 2점) Pipeline시 Conditional jump instruction이 끝나지 않아 PC value가 결정되지 않은 상태에서, Condition이 taken 혹은 not taken 되었다고 가정하여 다음 instruction을 수행하는 작업.(2)</p> <p>공통적 특성(총 4점) branch prediction 기법 중 dynamic branch prediction의 경우.(1) caching에서와 마찬가지로 프로그램의 non-uniform한 동작 특성인 locality 정보를 이용하여 prediction을 수행함.(3)</p>
<p>2. 10점</p>	<p>Structural hazard(총 3점) 원인(1) : 같은 cycle의 다른 stage에 있는 instruction이 같은 resource를 사용하고자 할 때 발생하는 hazard. 해결책(2) : 모든 stage에서 부족하지 않도록 추가의 resource를 둬</p> <p>Data hazard(총 4점) 원인(1) : 앞의 instruction의 결과가 register에 반영되기 전에 다음 instruction이 해당 register를 사용하려고 할 때, 발생하는 hazard 해결책(3, 2가지 정확히 쓰면 만점, 한 가지에 대해 정확히 쓰면 1점) : Freezing pipeline : 이전 instruction에서 필요한 값이 해당 register에 반영되기 전까지 다음 instruction을 stall함. Forwarding : 이전 instruction에서 다음 instruction이 필요로 하는 값이 register에 반영되지 않았어도 그 값이 구해지는 시점에서 필요로 하는 instruction에 전달하여 delay되는 것을 최소화함.</p> <p>Control hazard(총 3점) 원인 : Conditional jump instruction이나 return instruction이 나오면 그 instruction을 실행하기 전까지 다음 PC값을 알 수 없음(1) 해결책: Branch prediction과 prediction이 틀렸을 시 복귀하는 방식으로 해결(1) Fetch stage를 return instruction이 실행되는 동안 stall함.(1)</p>

2014-1학기 컴퓨터구조 중간시험 채점 기준표

3.
10점

C code	IA-32 Assembly code	
<pre>switch(n){ if(n == 4){ result = 2; } else if(n == 7){ result = -1; goto L200; } else if(n == 200){ L200: result = 1; } else{ result = 0; } }</pre>	<pre>.switch : movl 12(%ebp), %edx cmpl %edx, \$4 jne .L7 movl \$2, %eax jmp .Done .L7: cmpl %edx, \$7 jne .L200 movl \$-1, %eax .L200: cmpl %edx, \$200 jne .L movl \$1, %eax jmp .Done</pre>	<pre>.L: movl \$0, %eax .Done:</pre>

C code(총 4점)

if-then-else / goto를
이용하여 정확히 c언어로 구현함(4)

Assembly code(총 6점)

n이 4인 경우 정확히 처리됨(1.5)
n이 7인 경우 정확히 처리됨(1.5)
n이 200인 경우 정확히 처리됨(1.5)
n이 그 외의 값인 경우 정확히 처리됨(1.5)

4.
10점

- ① $x > 0$
- ② $2 * x + 12$
- ③ $y > z + 1$
- ④ $x + y - 16$
- ⑤ $z > -5$ ($z \geq -4$)
- ⑥ x / y
- ⑦ $x + y + z$

10점 만점에서
각 문제 틀릴 때마다 2점씩 감점

2014-1학기 컴퓨터구조 중간시험 채점 기준표

5.
10점

IA-32 Assembly code	
.Sum_while:	
pushl	%ebp
movl	%esp, %ebp
movl	8(%ebp), %edx
xorl	%eax, %eax
cmpl	\$1, %edx
jle	.Done
.Loop:	
addl	%edx, %eax
decl	%edx
cmpl	\$1, %edx
jl	.Loop
.Done:	
popl	%ebp
ret	

총 10점.
올바른 위치로 부터의 argument 참조함(2)
올바른 Loop semantic 구현함(5)
올바른 결과값을 return함(3)

6.
10점

Y86 ISA state :(총 4점 : 하나 틀릴 시 1점씩 감점, 최소 0점)

① PC(Program Counter) ② program register ③ condition code ④ Memory ⑤ Program status

((가).(나).(다) 각 2점, 총 6점)

(가) $PC \leftarrow PC + 1$

(나) $PC \leftarrow PC + 6$

$M[R[rB] + D] \leftarrow R[rA]$

exception 발생 시 program status 변경 가능성이 있음

(다) $PC \leftarrow Dest$

$R[\%esp] \leftarrow R[\%esp] - 4$

$M[R[\%esp]] \leftarrow PC + 5$

exception 발생 시 program status 변경 가능성이 있음

2014-1학기 컴퓨터구조 중간시험 채점 기준표

7.
10점

OPl rA, rB
 (1) $valP \leftarrow PC + 2$
 (2) $valE \leftarrow valB \text{ OP } valA$
 (3) $R[rB] \leftarrow valE$
 (4) $PC \leftarrow valP$

Pushl rA

(5) $valB \leftarrow R[\%esp]$
 (6) $valE \leftarrow valB + (-4)$
 (7) $R[\%esp] \leftarrow valE$

10점 만점에서
 각 문제 틀릴 때마다 별 2점씩 감점

8.
10점

(각 2.5점, 예시를 통한 설명도 인정)

- ① Program이 다르면 실제 실행되는 instruction 개수도 다름.
- ② CPI는 ISA의 영향을 받음. 예를 들면 CISC는 RISC보다 더 복잡한 기능을 수행하는 instruction들을 지원하므로 이 instruction들을 사용하면 RISC ISA에 비해 CPI가 커질 수 있음.
- ③ 공정이 미세할수록 transistor의 동작속도가 빨라져서 cycle time이 짧음
- ④ 컴파일러가 동일한 작업을 다른 instruction sequence로 최적화한다면 CPI에 영향을 미칠 수 있음. 예를 들면 같은 곱하기 연산을 수행하더라도 shift 연산을 사용하는 것과 mult 연산을 사용하는 것 사이에 CPI의 차이가 존재함.

9.
10점

load/use hazard
 (E stage에 bubble이 삽입되고 F/D stage를 stall하는 이유에 대한 설명(5), 각 register의 동작(5))

0x018의 load instruction의 target register와 뒤따르는 instruction의 source register 사이에 dependancy가 존재. data hazard가 발생.
 load instruction의 경우 target의 값을 M stage에 도달해서야 얻을 수 있으므로
 internal forwarding을 하더라도 다음 instruction을 1 cycle delay를 시켜야함.
 위의 이유로 E stage에 bubble을 삽입하고, F/D stage를 stall하고, M/W stage는 진행시킴.

Condition	F	D	E	M	W
Load/Use Hazard	stall	stall	bubble	normal	normal

2014-1학기 컴퓨터구조 중간시험 채점 기준표

10. Condition A(총 5점):
 10점 발생 상황(1) : Conditional jump instruction에서 다음 PC 주소를 mispredict했는데 target에 있는 instruction이 return instruction인 경우.
 해결 방법(2) :

Condition	F	D	E	M	W
Combination A	stall	bubble	bubble	normal	normal

이유(2) :

Conditional jump instruction이 다음 PC 주소를 mispredict했을 때의 처리과정과 return instruction의 처리과정을 union하여 처리함

Condition B(총 5점):

발생 상황(1) : Load instruction의 target register가 %esp이고, 바로 뒤에 return instruction이 따라와서 %esp를 사용하는 경우.

해결 방법(2) :

Condition	F	D	E	M	W
Combination B	stall	stall	bubble	normal	normal

이유(2) :

return instruction의 처리는 D stage에서 correct한 %esp값을 얻어야 한다는 것이 필수적인 조건임. 문제의 경우에는 return instruction이 correct한 %esp값을 얻지 못하므로 1 cycle을 stall하여 correct한 %esp값을 얻을 수 있도록 처리해줌.