

컴퓨터구조론 (컴퓨터공학과 003/004, 인공지능공학과 001) Homework 1

1. 이 연습문제에서는 파이프라인의 클럭 사이클 시간에 어떤 영향을 미치는지 알아본다. 데이터패스의 각 단계(stage)는 다음과 같은 지연시간을 갖는다고 가정한다.

IF	ID	EX	MEM	WB
250 ps	350 ps	150 ps	300 ps	200 ps

또한 이 프로세서의 실행 명령어 분포는 다음과 같다고 가정한다.

ALU/Logic	Jump/Branch	Load (lw)	Store (sw)
45%	20%	20%	15%

- 1.1. 파이프라인 프로세서와 파이프라인되지 않은 프로세서의 클럭 사이클 시간은 얼마인가? (3점)

파이프라인의 클럭 사이클이 가장 긴 단계 = 350ps

파이프라인 아닌 경우 = 모든 단계 = 250 + 350 + 150 + 300 + 200 = 1250ps

- 1.2. 파이프라인 프로세스와 파이프라인되지 않은 프로세서에서 lw 명령어의 전체 지연시간은 얼마인가? (3점)

lw 5 단계를 거친다.

파이프라인에서 지연시간은 $5 \times 350 = 1750ps$

파이프라인 아닌 경우 지연시간은 $250 + 350 + 150 + 300 + 200 = 1250ps$

- 1.3. 파이프라인 데이터패스 중 하나의 stage를 지연시간이 절반인 stage 2개로 나눌 수 있다면 어떤 stage를 나누는 것이 좋을 것이며, 이 때 프로세서의 클럭 사이클 시간은 얼마가 되는가? (4점)

클럭사이클이 가장 좋은 ID 350ps를 175로 나눈다.

이러면 디렉트리서의 클럭사이클은 300ps

- 1.4. Stall이나 hazard가 없다고 가정하면 데이터 메모리의 이용률은 얼마인가? (4점)

데이터 메모리 오리는 lw, sw 명령만 사용함으로

데이터 메모리의 이용률은 클럭 사이클의 35%이다.

- 1.5. Stall이나 hazard가 없다고 가정하면 “Register File” 유닛의 Write register 포트의 이용률은 얼마인가? (4점)

R.F 유닛의 Write register 포트는
ALU와 lw 명령어에서 활용되기 때문에
이용률은 클럭 사이클 65%가 된다.

2. 이 연습문제에서는 structure hazard 및 ISA가 파이프라인 실행에 어떤 영향을 미치는지 알아본다. 다음과 같은 MIPS 코드를 살펴보자.

```

sw $t0, 12($s2)
lw $t0, 8($s2)
beq $s1, $s0, Label      # Assume $s1 != $s0
add $s1, $s3, $s0
slt $s1, $t1, $s0

```

데이터파스의 각 단계(stage)는 다음과 같은 지연시간을 갖는다고 가정한다.

IF	ID	EX	MEM	WB
200 ps	120 ps	150 ps	190 ps	100 ps

2.1. 이 문제에서는 모든 branch는 완벽하게 예측된다고 가정한다. 즉, control hazard는 발생하지 않으며, delay slot도 사용되지 않는다. 만약 하나의 메모리만 사용한다고 가정할 때 (즉, instruction과 data가 같은 메모리에 저장되어 있을 때), 어떤 명령어(instruction)이 데이터 접근을 하는 cycle에는 새로운 instruction fetch가 structural hazard로 인해 일어날 수 없다. 이렇게 하나의 메모리만 있는 5-stage 파이프라인에서 위의 명령어들을 실행하는데 걸리는 전체 cycle을 얼마나가? 우리는 앞서 data hazard의 경우 nop 명령어를 추가하여 해결하는 것을 배웠다. 이 structural hazard에서도 동일한 방법을 사용할 수 있는가? 만약 그렇다면 (혹은 그렇지 않다면) 이유는 무엇인가? (6점)

5단계 파이프라인에서 주어진 명령을 실행하는데 걸리는 전체 cycle은 11 cycle이다

 S.h에서는 nop 명령어를 추가하여 hazard를 해결할 수 있다. 다음 명령어는 마찬가지로 nop 명령어로 명령어 fetch가 필요하지 않을 때 이를 고려해두어야 한다.

2.2. 이 문제에서는 모든 branch는 완벽하게 예측된다고 가정한다. 즉, control hazard는 발생하지 않으며, delay slot도 사용되지 않는다. 만약 다음과 같이 load/store 명령어가 (offset이 없이) register 값만 주소로 사용할 수 있도록 바꾼다면, 이 명령어들은 더 이상 ALU 사용을 필요로 하지 않는다.

lw \$t0, (\$s2)

그 결과, MEM 및 EX stage는 겹쳐질 수 있으므로, 파이프라인은 4 stage로 구성 가능하다. 이렇게 변경된 파이프라인에서 실행될 수 있도록 위의 MIPS 코드를 변경하라. 이 변경은 클럭 사이클 시간에 영향을 미치지 않는다고 가정할 때, 변경된 MIPS 코드는 몇 배의 성능 향상을 달성하는가? (6점)

이 변경은 데이터 해저드 없이 전치 실행에서 한 cycle 만 save 된다. 마지막 명령이 한 cycle 뒤 일어나면 된다.
 SW 0 0000 → 9 SW 0000
 LW 00000 → LW 0000
 BEQ 00000 → BEQ 0000 → 8
 ADD 00000 → ADD 0000
 SLT 00000 → SLT 0000
 $A/B = 1.13$ 배상.

컴퓨터구조론 (컴퓨터공학과 003/004, 인공지능공학과 001) Homework 1

1. In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
250 ps	350 ps	150 ps	300 ps	200 ps

Also, assume that instructions executed by the processor are broken down as follows:

ALU/Logic	Jump/Branch	Load (lw)	Store (sw)
45%	20%	20%	15%

1.1. What is the clock cycle time in a pipelined and non-pipelined processor? (3 points)

1.2. What is the total latency of an lw instruction in a pipelined and non-pipelined processor? (3 points)

1.3. If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor? (4 points)

1.4. Assuming there are no stalls or hazards, what is the utilization of the data memory? (4 points)

1.5. Assuming there are no stalls or hazards, what is the utilization of the write-register port of the “Register file” unit? (4 points)

2. In this exercise, we examine how resource hazards, control hazards, and instruction set architecture (ISA) design can affect pipelined execution. Problems in this exercise refer to the following fragment of MIPS code:

```
sw $t0, 12($s2)
lw $t0, 8($s2)
beq $s1, $s0, Label      # Assume $s1 != $s0
add $s1, $s3, $s0
slt $s1, $t1, $s0
```

Assume that individual pipeline stages have the following latencies:

IF	ID	EX	MEM	WB
200 ps	120 ps	150 ps	190 ps	100 ps

2.1. For this problem, assume that all branches are perfectly predicted (this eliminates all control hazards) and that no delay slots are used. If we only have one memory (for both instructions and data), there is a structural hazard every time we need to fetch an instruction in the same cycle in which another instruction accesses data. What is the total execution time of this instruction sequence in the 5-stage pipeline that only has one memory? We have seen that data hazards can be eliminated by adding nops to the code. Can you do the same with his structural hazard? Why? (6 points)

2.2. For this problem, assume that all branches are perfectly predicted (this eliminates all control hazards) and that no delay slots are used. If we change load/store instructions to use a register (without an offset) as the address as following, these instructions no longer need to use the ALU.

```
lw $t0, ($s2)
```

As a result, MEM and EX stages can be overlapped and the pipeline has only 4 stages. Change this code to accommodate this changed ISA. Assuming this change does not affect clock cycle time, what speedup is achieved in this instruction sequence? (6 points)