Homework10
yxu66

4.8 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
250ps	350ps	150ps	300ps	200ps

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

alu	beq	lw	sw
45%	20%	20%	15%

4.8.1 [5] <§4.5> What is the clock cycle time in a pipelined and non-pipelined processor?

In MIPS: instructions take something like 4-5 stages.
For non-pipelined processor, the clock cycle time has to allow an instruction to go through all the stages each cycle, so it determined by the sum of time of all stages:
CT = 250 + 350 + 150 + 300 + 200 = 1,250 ps

For pipelined processor, the cycle clock time is determined by the slowest stage:
CT = 350ps

4.8.2 [10] <§4.5> What is the total latency of an LW instruction in a pipelined and non-pipelined processor?

Lw instruction uses all 5 stages.
Pipelined processor takes 5 cycles at 350ps per cycle, so total latency is: 5*350 = 1,750 ps
Unpipelined processor takes 250 + 350 + 150 + 300 + 200 = 1,250 ps

4.8.3 [10] <§4.5> 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?
Split ID stage into two stages of 175ps. New clock cycle time of the processor: 300ps.

4.8.4 [10] <§4.5> Assuming there are no stalls or hazards, what is the utilization of the data memory?
Data memory is utilized only by lw and sw instruction, so the utilization is 20%+15% = 35% .

4.8.5 [10] <§4.5> Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?
The write-register may be utilized by alu and lw instruction, so the utilization is 45%+20% = 65% .

4.8.6 [30] <§4.5> Instead of a single-cycle organization, we can use a multi-cycle organization where each instruction takes multiple cycles but one instruction finishes before another is fetched. In this organization, an instruction only goes through stages it actually needs (e.g., ST only takes 4 cycles because it does not need the WB stage). Compare clock cycle times and execution times with single-cycle, multi-cycle, and pipelined organization.
(For example, sw only uses 4 cycles. For each of the three implementations (single-cycle, multi-cycle, and pipelined) find the minimum and maximum possible execution times for a program that executes 1,000,000 instructions distributed as indicated in the table.**)**

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

alu	beq	lw	sw
45%	20%	20%	15%

ALU instruction uses 4 stages(IF, ID, EX, WB).
Beq instruction uses 3 stages(IF, ID, EX).
Lw instruction uses all 5 stages.
Sw instruction uses all 4 stages(IF, ID, EX, MEM).

Clock cycle time(Alu) = 250+350+150+200 = 950 ps
Clock cycle time(Beq) = 250+350+150 = 750 ps
Clock cycle time(Lw) = 250 + 350 + 150 + 300 + 200 = 1,250 ps
Clock cycle time(Sw) = 250+350+150+300 = 1,050 ps

For single-cycle:
Clock cycle time = 250 + 350 + 150 + 300 + 200 = 1250 ps
Total execution time = 1000000 * 1250*10^-12 = 1.25ms

For pipelined organization:
Total execution time = 350*1000000*10^-12 =0.35ms

For multiple-cycle:
Alu = 250+350+150+200 = 950 ps
Beq = 250+350+150 = 750 ps
Lw = 250 + 350 + 150 + 300 + 200 = 1,250 ps
Sw = 250+350+150+300 = 1,050 ps
Total execution time = (1000000 *45%*950 + 1000000 *20%*750 + 1000000 *20%*1250 + 1000000 * 15%*1050)*10^-12 = 0.985ms