Homework Assignment 2

CS/ECE 6810: Computer Architecture

January 31, 2018

Pipelining

Due Date: 2/13/2018

(90 points)

Important Notes:

- Solutions turned in must be your own. Please, mention references (if any) at the end of each question. Please refrain from cheating.
- All solutions must be accompanied by the equations used/logic/intermediate steps. Writing only the final answer will receive **zero** credit.
- All units must be mentioned wherever required.
- Late submissions (**after 11:59 pm on 02/13/2018**) will not be accepted
- All submissions must be in PDF only. Scanned copies of handwritten solutions will not be accepted as a valid submission.

1. **Pipelining Performance.**Consider a single-cycle processor in which the execution of every instruction completes in 10ns. The goal is to execute a user application that consists of 1000 instructions.
   i. Ifthe single cycle processor can execute one instruction per cycle, find the CPU time. **(5 points)**
   ii. Assuming a 1ns additional delay for pipeline registers and perfect circuit partitioning and pipelining, find the speedup gained by making the processor 10-stage pipelined. Assume that 2 bubbles (NOPs) are necessary every 10 cycles for load and branch instructions.**(15 points)**

Answer

(i) CPU Time = IC x CPI x CT
= 1000 x 1 x 10 = 10000 ns = 10 μs

(ii) New CT = $(CT/N + T_{ovh}) = T_{ovh}$ = 1 ns and N = 10 : New CT = 2
New CPI = (2 NOP per 10 cycles) = 1.2
New CPU Time = 1000 x 1.2 x 2 = 2400 ns
Speedup = Old CPU Time/New CPU Time = 10000/2400 = 4.167

2. **Control Hazards.**Consider an in-orderfive-stage pipeline that determines the branch target by the end of the 3rd pipeline stage (i.e., the execute stage). To avoid branch-

related stalls, the instruction set architecture (ISA) defines two branch delay slots. Assume that (a) all stalls in the processor are branch-related, (b) 20% of all instructions are branches, (c) a branch is taken 60% of the time, and (d) you could move two instructions from the taken side into the branch delay slot. What is the expected CPI for this processor?**(15 points)**

Answer

80% non branch
20% branches
60% taken = 12% taken and 2 NOPs filled with instructions so no delay slot
40% not taken = 8% instructions have 2 delay slots
= total delay cycles = 0.08*2 = 0.16
CPI = 1+0.16 = 1.16

3. **Multi-cycle Instructions.**A pipelined architecture comprises instruction fetch (IF), instruction issue (IS), register read (RR), execute (EX), and write-back (WB) stages. Except EX, each stage requires one clock cycle to complete. The EX stage includes 4 functional units, each of which can perform a floating-point operation—e.g., ADD, SUB, MULT, DIV, load, and store. The table below shows the latency of the operations in terms of the number of clock cycles. The pipeline implements all necessary forwarding paths for the ALU operations.

	ADD	SUB	MULT	DIV	Load	Store
Latency	1	1	3	4	1	1

```
Load    F6, 20(R5)
Load    F2, 28(R5)
MUL     F0, F2, F4
SUB     F8, F6, F3
DIV     F10, F0, F6
ADD     F6, F8, F2
Store   F8, 50(R5)
```

i. Create a timing diagram for the following code showing the execution of the code in time (clock cycles). **(15 points)**

Answer

- Even though there is register forwarding, extra logic hardware and clock cycles will be required to decide whether the value read from a register is stale and the "fresh" value is available in the latch. This is particularly applicable to the Load(2) feeding the MUL(3) where you need to wait for 1 clock cycle and MUL feeding DIV where you need to wait for 2 clock cycles.
- A stall propagates down the pipeline in the same clock cycle it occurs
- This pipeline allows for out of order completion of instructions

Instruction	Clock Cycle														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Load F6, 20(R5)	IF	IS	RR	L	W										
Load F2, 28(R5)		IF	IS	RR	L	W									
MUL F0, F2, F4			IF	IS	*	RR	M1	M2	M3	W					
SUB F8, F6, F3				IF	*	IS	RR	S	W						
DIV F10, F0, F6					*	IF	IS	*	*	RR	D1	D2	D3	D4	W
ADD F6, F8, F2							IF	*	*	IS	RR	A	W		
Store F8, 50(R5)								*	*	IF	IS	RR	S	W	

ii. Identify all structural and data hazards in the following code.**(20 points)**

Answer
Data Hazards from the timing diagram above

1. Read After Write (RAW) between
   ```
   Load    F2, 28(R5)
   MUL     F0, F2, F4
   ```
2. Read After Write (RAW) between
   ```
   MUL     F0, F2, F4
   DIV     F10, F0, F6
   ```
3. **Solved:** Write After Read (WAR) between
   ```
   DIV     F10, F0, F6
   ADD     F6, F8, F2
   ```
   This hazard does not occur as the Register Read stage of DIV has already copied the value of F6 and so ADD does not need to wait for DIV to finish. If this stage was not present in the pipeline, there ADD would have to wait until the execution of DIV finished.

Data Hazards from Code

RAW (Totally 7) F6, F2, F0 and F8 (Remember Store is reading the value of F8)
```
Load    F6, 20(R5)
Load    F2, 28(R5)
MUL     F0, F2, F4
SUB     F8, F6, F3
DIV     F10, F0, F6
ADD     F6, F8, F2
Store   F8, 50(R5)
```
WAR (Totally 2)
```
SUB     F8, F6, F3
DIV     F10, F0, F6
ADD     F6, F8, F2
```
WAW (1)
```
Load    F6, 20(R5)
ADD     F6, F8, F2
```

Structural Hazards

- If you assume separate instruction and data memory, there will be no structural hazard in cycle 4. Otherwise a data load will compete with an Instruction Fetch
- If you assume separate half cycles for register read and register write, there will be no structural hazard in cycles 6 and 10. Else register write back will compete with register read.

4. **Points of Production and Consumption**.Consider an un-pipelined processor where it takes 36ns to go through the circuits and 0.5ns for the latch overhead. Assume that the point of production and point of consumption in the un-pipelined processor are separated by 12ns. Assume that half the instructions do not introduce a data hazard and half the instructions depend on their preceding instruction.
   i. What is the throughput of the processor with the un-pipelined architecture **(10 points)**
   ii. What is the throughput of the processor with a 12-stage pipeline? **(10 points)**

Answer

i. Un-pipelined Processor

Cycle Time = 36 + 0.5 = 36.5 ns
IPC = 1
Throughput(IPS) = 1/36.5 = **0.0274 BIPS**

Since this is an unpipelined processor, the point of production and consumption is the end and the beginning of the pipeline

ii. Pipelined Processor

Point of Production and Consumption separated by 12 ns = 4 cycles

Recall slide 4 and 5 from Topic ILP
IPC = 2/5 = 0.4

Throughput (IPS) = IPC/Clock Cycle Time
Clock Cycle Time = T_{ovh} + (Old CT/Pipeline Stages) = 0.5 + 36/12 = 3.5 ns

Throughput = 0.4/3.5 = **0.114 BIPS**

Alternate Method

12-stage pipeline, each stage is 3 ns and a latch is present between 2 stages so 1 stage takes 3.5 ns.

No Hazard:
Throughput = 1/3.5 BIPS

With Data Hazard (number of stages = 12/3 = 4)
Throughput = 1/(3.5*4) = 1/14 BIPS

Since half the instructions introduce a data hazard and the other half doesn't

Throughput = 1/(0.5*14 + 0.5*3.5) = 2/(14+3.5) = **0.114 BIPS**