The University of Alabama in Huntsville
ECE Department
CPE 431 01/01R, CPE 531 01/91
Fall 2022
Pipelining

Due September 25, 2022 1.0 (10), 2.0.1 (5), 2.0.2 (10), 3.0.1 (5), 3.0.2 (5), 3.0.3 (15), 4.0 (10), 5.0 (10)

1.0 **< 4.3>** For the problems in this exercise, assume that there are no pipeline stalls and that the breakdown of executed instructions is as follows**:**

add	addi	slt	beq	lw	sw	jump	and
25 %	13 %	3 %	10 %	23 %	16 %	5 %	5 %

In what fraction of all cycles is the input of the shift-left circuit that feeds the ADD circuit found just above the AND gate needed? What is the circuit doing in cycles in which the input is not needed?

The shift left circuit is used for branch and jump for effective address calculations and is used 15% of the time. When that input is not needed it is still calculated, the mux that controls whether the PC gets PC + 4 or the other address just doesn't select it.

2.0 **<4.4>** In this exercise we examine in detail how an instruction is executed in a single-cycle datapath. Problems in this exercise refer to a clock cycle in which the processor fetches the following instruction word:

0000 0000 0000 1100 0001 1011 0000 0000

Assume that data memory is all zeros and that the processor's registers have the following values at the beginning of the cycle in which the above instruction word is fetched.

r0	r1	r2	r3	r4	r5	r6	r8	r12	r31
0	197	-2	-35	4	-100	674	-89	222	-16

2.0.1 What are the outputs of the sign-extend and the jump "Shift left 2" unit (near the top of Figure 4.24) for this instruction word?

0000_00|00_0000|0_1100|0001_1011_0000_0000

Sign Extend: 0000_0000_0000_0000_0001_1011_0000_0000
Shift Left 2: 0000_0000_0000_0110_1100_0000_0000

Wasn't sure if you meant following the data line shift left 2 or from the initial instruction so from initial instruction the shift left 2 is: 0000_0011_0000_0110_1100_0000_0000

2.0.2 For the ALU and the two add units, what are their data input values?

ALU: Read Data 1 which is equal to r0 = 0
Read Data 2 or the sign extension which is r12 = 222 or 6912 from the sign extension

Add unit 1 (left most add): gets PC and 4 (output PC + 4)
Add unit2(one with shift left) gets PC+4 and 2768

3.0 **< 4.5>** 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
310 ps	220 ps	300 ps	350 ps	220 ps

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

alu	beq	lw	sw
45%	10%	30%	15%

3.0.1 What is the clock cycle time in a pipelined and non-pipelined processor?
1400 ps non-pipelined
350ps pipeline

3.0.2 Assuming there are no stalls or hazards, what is the utilization of the data memory?
How many times was the data memory doing something useful. Useful/total cycles
Data memory is only utilized during Load word and store word
DM was doing something useful 45% of time

3.0.3 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 instruction 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.
Lw would take 5 cycles
Sw would take 4 cycles
Branch would take 3
Alu would take 4

Pipeline ET is the time of the longest stage to complete: 350ps
Multi Cycles: (5*30%) + (4*(45%+15%)) + (3*10%) = 4.2
Single cycle: NonPipe/Pipe = 1400/350 = 4
Single cycle is 4x faster than pipeline.
x/350 = 4.2 gives ET of 1470 for multi cycle.
ET(Single)/ET(Multi) = 1400/1470 = 0.95
Multi cycle is 4.2 times faster than pipelined but 0.95 slower than single cycle.

4.0 **<4.5>** In this exercise, we examine how data dependences affect execution in the basic 5-stage pipeline described in Section 4.5. Problems in this exercise refer to the following sequence of instructions:

```
(1)     or   $t0, $s2, $t0
        nop
(2)     and  $t0, $t0, $s3 //2 depends on 1 bc of t0
(3)     lw   $t5, 24($t0) //3 depends on 2 bc of t0
        nop
        nop
(4)     sw   $t5, 12($s6) //4 depends on 3 bc of t5
(5)     sub  $t3, $t5, $t1
```

Also, assuming the following cycle times for each of the options related to forwarding:

Without Forwarding	With Full Forwarding	With ALU-ALU Forwarding Only
230 ps	290 ps	270 ps

Assume there is no forwarding in this pipelined processor. Indicate hazards and add nop instructions to eliminate them.

5.0 Consider the following loop.

```
Loop:  lw    $t0, 0($t1)
       and   $s6, $s2, $t0
       lw    $t9, 0($s6)
       lw    $t8, 0($s6)
       beq   $t8, $t9, loop
```

Assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, and that the pipeline has full forwarding support. Also, assume that many iterations of this loop are executed before the loop exits.

Show a pipeline execution diagram for the third iteration of this loop, from the cycle in which we fetch the first instruction of that iteration up to (but not including) the cycle in which we can fetch the first instruction of the next iteration. Show all instructions that are in the pipeline during the cycles (not just the third iteration).

Cycle	IF	ID	EX	MEM	WB
1 (Loop 1)	lw $t0	Ahead1	Ahead2	Ahead3	Ahead4
2	and $s6	lw $t0	Ahead1	Ahead2	Ahead3
3	lw $t9	and $s6	lw $t0	Ahead1	Ahead2
4	lw $t8	lw $t9	and $s6	lw $t0	Ahead1
5	beq $t8	lw $t8	lw $t9	and $s6	lw $t0
6 (Loop 2)	lw $t0	beq $t8	lw $t8	lw $t9	and $s6
7	and $s6	lw $t0	beq $t8	lw $t8	lw $t9
8	lw $t9	and $s6	lw $t0	beq $t8	lw $t8
9	lw $t8	lw $t9	and $s6	lw $t0	beq $t8
10	beq $t8	lw $t8	lw $t9	and $s6	lw $t0
11 (Loop 3)	lw $t0	beq $t8	lw $t8	lw $t9	and $s6

	and $s6	lw $t0	beq $t8	lw $t8	lw $t9
	lw $t9	and $s6	lw $t0	beq $t8	lw $t8
	lw $t8	lw $t9	and $s6	lw $t0	beq $t8
	beq $t8	lw $t8	lw $t9	and $s6	lw $t0
Loop 4	lw $t0	beq $t8	lw $t8	lw $t9	and $s6
			beq $t8	lw $t8	lw $t9
				beq $t8	lw $t8
					beq $t8