Hacettepe University
Department of Computer Engineering
BBM234 Computer Organization – Spring 2021
Homework 2

Assigned date: 04.05.2021 Due: 16.5.2021 at 23:59:59

Submit a single (zipped) PDF file via submit.cs.hacettepe.edu.tr

Student Name: Şevval Atmaca **Student ID:** 21827115

Q1. We would like to add **bne** instruction to the single cycle architecture given below. **bne** instruction is a branch instruction and it loads branch target address (BTA) to the PC (PC=BTA) if [rs] != [rt].

a) Show the necessary changes on the data-path in Figure 1 and explain your changes. Your new architecture should be able to execute both **beq** and **bne** instructions together.

b) Fill the control signals in Table I. Add new control signal/signals to the table if necessary.

a).

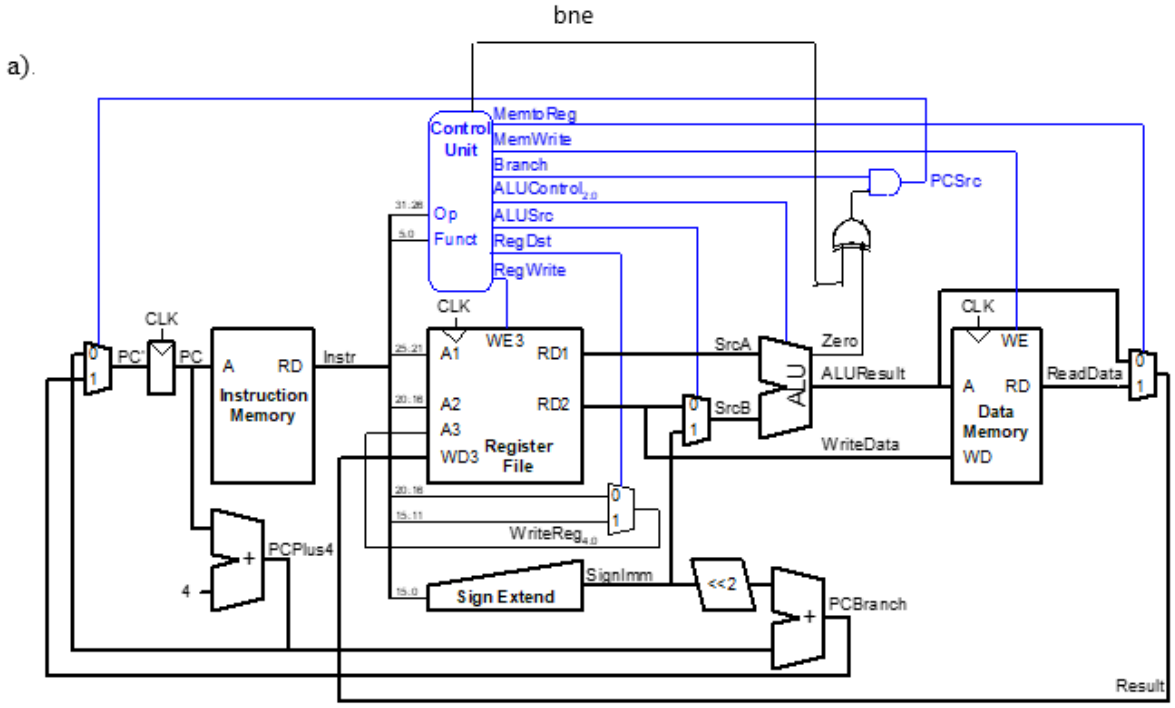

Figure 1: Single cycle processor

PCsrc=(Zero XOR bne) AND Branch

I added a new control signal called “bne”. For bne instruction, we will set branch and bne to 1. We calculate zero (if zero= 0, that means registers are not equal and bne branch will be taken, otherwise zero = 1 branch will not taken). If zero = 0, then branch is taken and from formula PCSrc will be 1. If PCsrc is 1, then it loads branch target address (BTA) to the PC.

Inst.	$Op_{31:26}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUOp_{1:0}$	bne
bne	000101	0	X	0	1	0	X	01	1

Q2. You are given the following MIPS code. You have 5 stage pipelined MIPS processor running the code.

a) If there is no forwarding unit in the MIPS processor, **insert enough nop's between the instructions** to have correct execution of the code. How many cycles does it take to execute all instructions? Show your calculations or explain how you found the cycle time.

 # MIPS assembly code

lw $s0, 0($0)
lw $s1, 4($0) (add instruction use s1 and s0 as a source register but s1 is a destination register in lw instructions so, we need 2 nops.)
nop
nop
add $t0, $s0, $s1
or $t1, $s2, $s3(and instruction use st0 and st1 as a source register but they are destination register in or and add instruction so, we need 2 nops)
nop
nop
and $t1, $t1, $t0

There are 9 instructions, then we need 13 cycles.

b) If there is a forwarding unit in the MIPS processor, insert enough nop's between the given instructions to have correct execution of the code. How many cycles does it take to execute all instructions? Show your calculations or explain how you found the cycle time.

lw $s0, 0($0)
lw $s1, 4($0) (if we use forwarding, lw can receive to add in writeback stage, then we need 1 nops.)
nop
add $t0, $s0, $s1
or $t1, $s2, $s3
and $t1, $t1, $t0

There are 6 instructions, then we need 10 cycles.

c) If there is data hazard unit (data forwarding and stalling hardware) in the MIPS processor, rearrange the given code if possible, to minimize the clock cycles and your code still executes correctly. How many cycles does it take to execute all instructions? Show your calculations or explain how you found the cycle time.

lw $s0, 0($0)
lw $s1, 4($0)
or $t1, $s2, $s3 (or instruction don't use any source register than a destination register in a instruction. Then if we move "or" up then we don't need to use nops.)
add $t0, $s0, $s1
and $t1, $t1, $t0

There are 5 instructions, then we need 9 cycles.

Q3. The distribution of the instructions for the program with one billion (10^9) instructions is given below.

40% load, 10% store, 10% branch, 10% jump, 30% R-type

Suppose

- 50% load instruction results are used by the next instruction.
- 50% R-type instruction results are used by the next instruction.
- 50% branches are taken (i.e., mispredicted).
- All jumps flush the next instruction.

a) How many clock cycles does this program take in a single cycle MIPS processor?

CPU time = Instructions executed * CPI* Clock cycle time = 10^9 * 1 * Clock Cycle

b) How many clock cycles does it take on a pipelined MIPS processor with no hazard unit (no forwarding and no early branch resolution)?

If there is no hazard unit, lw and r take 3 clock cycles, otherwise 1.
If the branch taken, Branch will take 3 flushed and branch instruction otherwise, it will take only one cc.

CPI (lw)=0.50 * 1 + 0.50 * 3=2
CPI (R)==0.50 * 1 + 0.50 * 3=2
CPI (branch)=0.50 * 1 + 0.50 * 4=2.5
CPI (J)=2
CPI (average)= 0.40 * 2 + 0.10 * 1 + 0.10 * 2.5 + 0.10 * 2 + 0.30 * 2=1.95
CPU Time = 10^9 * 1.95 * clock cycles

c) How many clock cycles does it take on a pipelined MIPS processor with hazard unit (with forwarding and early branch resolution)?

If there is forwarding and early branch resolution, lw instructions will take 2 cycles. R-type instructions take 1 cycles. If the branch taken. Branch will take branch instruction plus one flushed instruction, 2 instruction. Otherwise, it will take 1 clock cycles.

CPI (lw)=0.50 * 1 + 0.50 * 2 = 1.5
CPI (R)==0.50 * 1 + 0.50 * 1 = 1
CPI (branch)=0.50 * 1 + 0.50 * 2 = 1.5
CPI (J) = 2
CPI (average)= 0.40 * 1.5 + 0.10 * 1 + 0.10 * 1.5 + 0.10 * 2 + 0.30 * 1=1.35
CPU Time = 10^9 * 1.35 * clock cycles

Q4. The propogation delay of each stage for a CPU architecture is determined as shown in the table below. The delay of a pipeline register is found as 1ns. ($ns=10^{-9}$)

IF	ID	EX	MEM	WB
7ns	7ns	6ns	9ns	5ns

a) If we design a single-cycle processor, what would be the minimum clock cycle in ns?

Minimum clock cycle = 7 + 7 + 6 + 9 + 5 = 34 ns

b) What would be the clock cycle of a 5 stage pipelined processor?

Clock cycle = the longest stage + pipeline register = 9 + 1 = 10 ns

c) We would like to decrease the clock cycle of pipelined processor and we decided to divide one of the stages into two stages. Thus, the new pipelined processor will have 6 stages. Which stage would you divide to two? What would be the new clock cycle?

We need to divide MEM stage because it is the longest stage, then the longest stage will be IF and ID as a 7 ns.

Clock cycle = the longest stage + pipeline register = 7 + 1 = 8 ns

d) We would like run 1000 instrcutions on above specified processors. Assume there is no data and control hazards. What are the CPU times of these 1000 instruction on these three processors?

CPU time = Instructions executed * CPI (Cycles per instruction) * Clock cycle time

CPU time for a = 1000 * 1 * 34*10^-9 = 34 * 10^-6 s

(For b there are 1000 instructions and 5 stage, then it will be 1004 cycles)
CPU time for b = 1000 * (1004 / 1000) * 10 * 10^-9 = 10,04 * 10^-6 s

(For b there are 1000 instructions and 6 stage, then it will be 1005 cycles)
CPU time for c = 1000 * (1005 / 1000) * 8* 10^-9 = 8,04 * 10^-6 s

Q5. The MIPS program below is executed on the pipelined architecture given below. At a time instant, we observe that SrcAE=0x0000 0005.

```
MIPS program:
addi $t1, $0, 5
addi $t2, $0, 6
addi $t3, $0, 4
addi $t4, $0, 5
lw  $s2, 40($0)
add $s3, $t1, $t2
sub $s4, $t3, $t4
and $s5, $t2, $t3
sw  $s6, 20($t1)
or  $s7, $t3, $t4
```

a) At the same moment, write down the instructions in the following phases:

Stage	Instruction
Fetch	`and $s5, $t2, $t3`
Decode	`sub $s4, $t3, $t4`
Execute	`add $s3, $t1, $t2`
Memory	`lw $s2, 40($0)`
Writeback	`addi $t4, $0, 5`

b) At the same moment, what are the values of the following signals?

Signal	Value
InstrD	`sub $s4, $t3, $t4`
SrcBE	6
ALUOutM	40
MemWriteM	0
WriteRegW	t4