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: **Emirhan Topcu** **Student ID:** **21827899**

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)

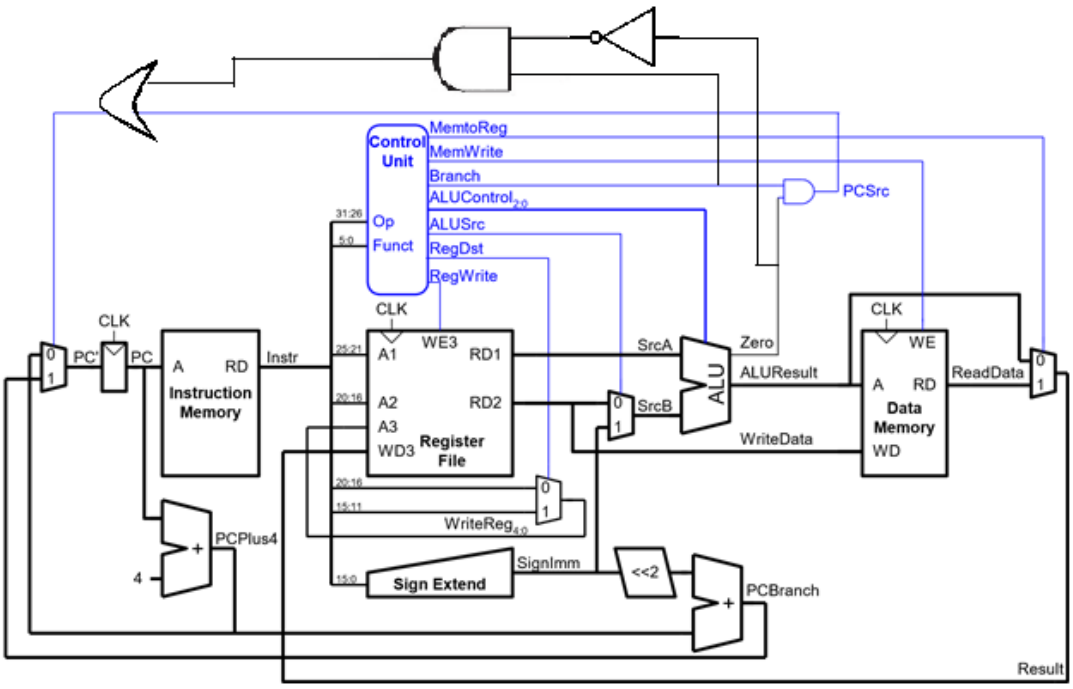


Figure 1: Single cycle processor

Beq and bne instructions are very similar so just a little difference in this circuit will be enough for us. If we need to check bne, we need to see if Zero value is 0. So I connected Zero to an not gate then to an and gate with branch as the other input. Then I connected this and Gates output to an or gate with the beq output as the second input. If one of them is 1 PC will jump to that line.

b)

Table I: **bne** control signals

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

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)
nop
nop
add $t0, $s0, $s1

or $t1, $s2, $s3
nop
nop
and $t1, $t1, $t0
```

There are 9 instructions so it takes 9 cycles to fetch those. The last instruction's execution takes 4 more cycles so its 13 cycles in total.

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)
nop
add $t0, $s0, $s1
or $t1, $s2, $s3
and $t1, $t1, $t0
```

6 instructions instead of 9 so it's 13-3 = 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
add $t0, $s0, $s1
and $t1, $t1, $t0
```

There are 5 instructions so it takes 5 cycles to fetch those. The last instruction's execution takes 4 more cycles so its 9 cycles in total.

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?

CPI of single cycle is always 1 so: 10^9

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

0.4 + (0.2*2) load
0.1 store
0.1 + (0.05*3) extra flush
0.1 + 0.1 jump
0.3 + (0.15*2) r-type

CPI = 0.8 + 0.1+ 0.25+ 0.2 + 0.6 = 1.95

$1.95 * 10^9 *$ clock = $1.95x10^9$

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

0.4 + 0.2*1
0.1
0.1 + 0.05*1
0.1 + 0.1
0.3

CPI = 0.6 + 0.1+ 0.15+ 0.2 + 0.3 = 1.35

$1.35 * 10^9 *$ clock = $1.35x10^9$

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?

Clock cycle = 7+7+6+9+5 = 34 ns

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

MEM stage takes 9ns and it's maximum. We add delay of the pipeline register so its 10ns
Clock cycle = 9+1= 10ns

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?

I would have devided MEM stage since it's the longest one so the new longest stage will take 7ns.
Clock cycle = 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?

Single cycle = 34 * 1000 = 34000ns

Clock time of five stage = (999 * 10) + 50 = 10040 ns

Clock time of six stage = (999 * 8) + 48 = 8040 ns

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

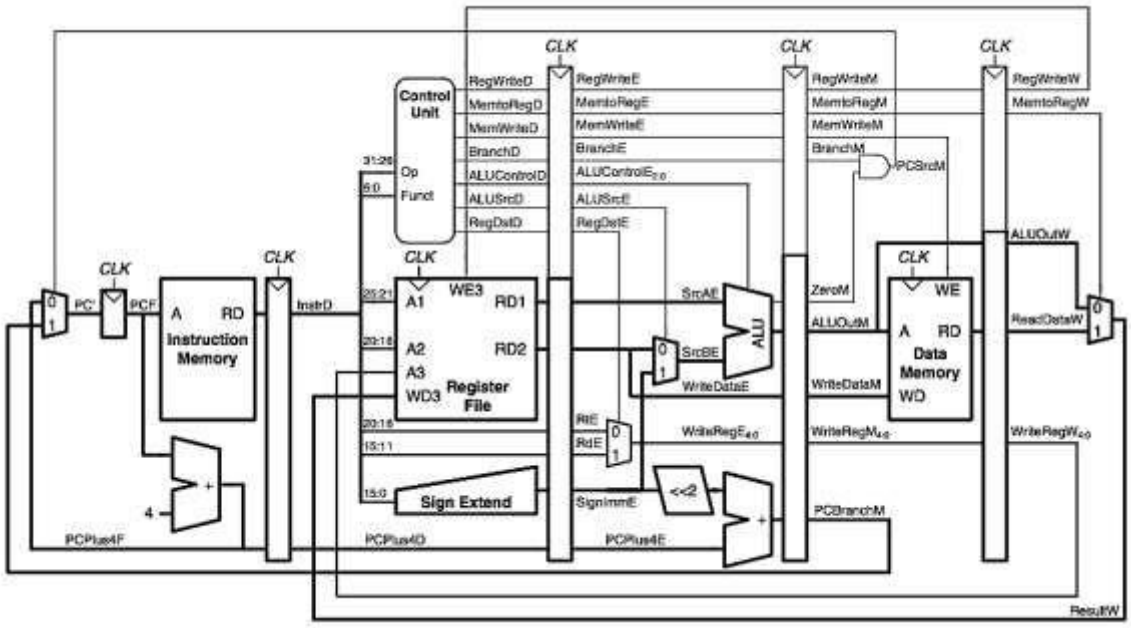
```
MIPS     program:
addi  $t1,  $0,  5
addi  $t2,  $0,  6
addi  $t3,  $0,  4
addi  $t4,  $0,  5
lw    $s2,  40($0)
add $s3, $t1, $t2
sub $s4, $t3, $t4
and $s5, $t2, $t3
sw   $s6,  20($t1)
or  $s7, $t3, $t4
```

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	0x016CA022
SrcBE	0x00000006
ALUOutM	0x00000028
MemWriteM	0
WriteRegW	0xC


CLK
Control Unit
Op
Funct
RegWriteD
MemtoRegD
MemWriteD
BranchD
ALUControlD
ALUSrcD
RegDstD
RegWriteE
MemtoRegE
MemWriteE
BranchE
ALUControlE2:0
ALUSrcE
RegDstE
RegWriteM
MemtoRegM
MemWriteM
BranchM
PCSrcM
RegWriteW
MemtoRegW
ALUOutW
ReadDataW
ResultW
PC'
PCF
A
RD
Instruction Memory
InstrD
31:26
5:0
25:21
20:16
15:11
15:0
A1
A2
A3
WD3
WE3
RD1
RD2
Register File
Sign Extend
SignImmE
SrcAE
SrcBE
ALU
ZeroM
ALUOutM
WriteDataE
WriteDataM
WE
WD
Data Memory
RtE
RdE
WriteRegE4:0
WriteRegM4:0
WriteRegW4:0
<<2
PCBranchM
4
PCPlus4F
PCPlus4D
PCPlus4E