Hacettepe University
Computer Engineering Department
BBM234 Computer Organization
2019-2020 Spring Term

Homework 2

İbrahim Burak Tanrıkulu b21827852

Assigned date: 27.04.2020

Due date: 03.05.2020 (all sections) through submit.cs.hacettepe.edu.tr as a single PDF file.

Q1. You are given 32-bit Program Counter (PC), instruction memory, register file and a data memory. Draw a single-cycle processor that can execute ONLY the following type of instructions. Define necessary control signals and write their values in the table. Note that you do not need some of the control signals in the table. Write NA to them if they are not used in your microarchitecture. First, show the machine code fields (instruction formats) of the given instructions. Then, draw the datapath of the instructions for a single-cycle processor by using the given hardware units and adding new hardware if necessary. Show your extra equipment and what bits are connected to which inputs of the components.

```
and $s0, $s1, $s2
addi $t0, $t1, 1
```

000000 10001 10010 10000 00000 100100

001000 01001 01000 0000000000000001

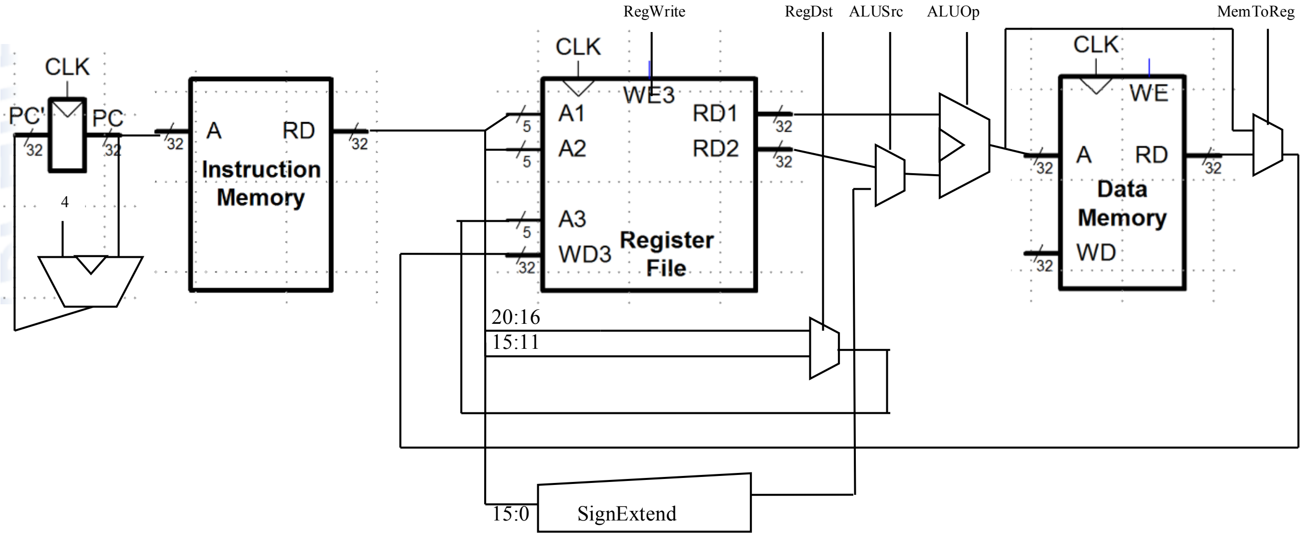

Table I: Control signals for **and** and **addi**

Inst.	**Op**$_{31:26}$	**RegWrite**	**RegDst**	**AluSrc**	**Branch**	**MemWrite**	**MemtoReg**	**ALUOp**
and	000000	1	1	0	NA	NA	1	000
addi	001000	1	0	1	NA	NA	0	010

Q2. 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.

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

a)

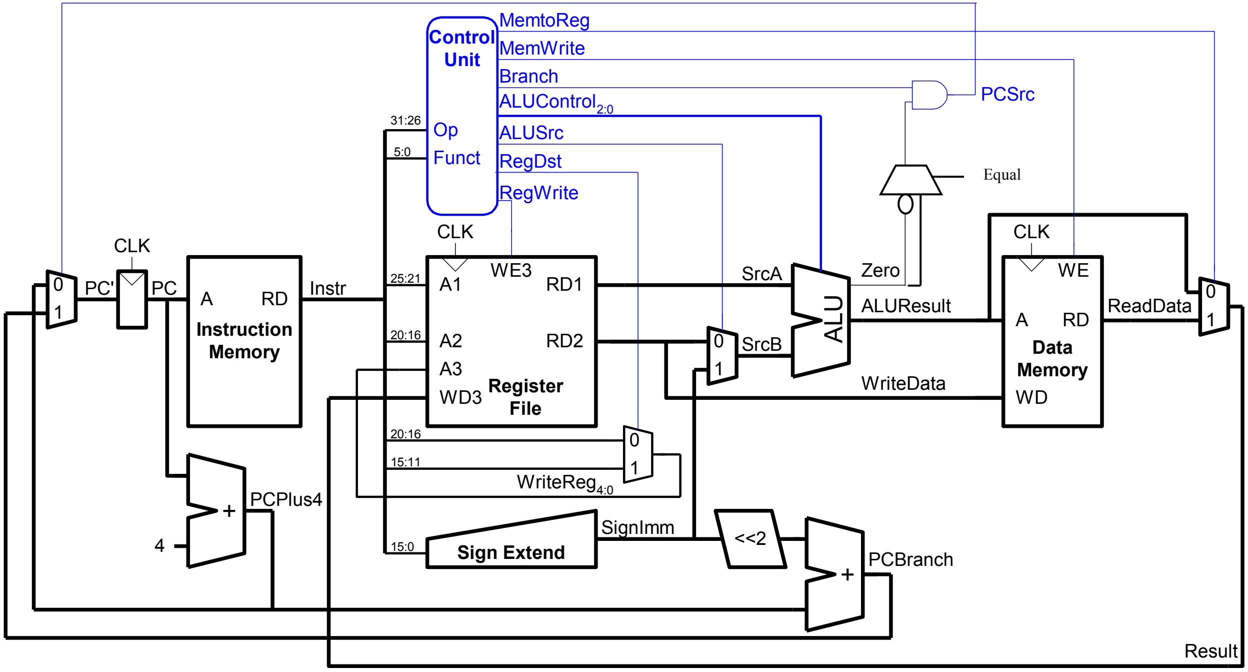

Figure 1: Single cycle processor

We can add a MUX, NOT and a control signal.
Input of this MUX is “Zero” and “NOT Zero” and control signal.
Output of this MUX, goes to AND gate with Branch signal.
Control signal is 1 when beq and 0 when bne.
If beq, then (Branch && Zero), else (Branch && ! Zero)

b)

Table I: **bne** control signals

Inst.	Op$_{31:26}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	Equal	MemtoReg	ALUOp$_{1:0}$
bne	000101	0	NA	0	1	0	0	NA	010

Q3. 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)          # $s0 at decode       $s1 at fetch
nop                     # $s0 at execution    $s1 at decode
nop                     # $s0 at memory       $s1 at execution
add $t0, $s0, $s1       # $t0 at fetch        $s0 at writeback     $s1 at memory


or $t1, $s2, $s3        # $t0 at decode       $t1 at fetch
nop                     # $t0 at execution    $t1 at decode
nop                     # $t0 at memory       $t1 at execution
and $t1, $t1, $t0       # $t0 at writeback    $t1 at memory
```

13 <-- lw+lw+nop+nop+add+or+nop+nop+and+and(decode)+and(exe)+and(memory)+and(wb)

b) If there is a 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.

```
lw $s0, 0($0)
lw $s1, 4($0)
nop                     # $s0 stalled
add $t0, $s0, $s1       # $s0 forwarded       $s1 forwarded
or $t1, $s2, $s3
nop                     # $t0 stalled
and $t1, $t1, $t0       # $t0 forwarded       $t1 forwarded
```

11

c) If there is a forwarding unit in the MIPS processor, rearrange the 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        # $s0 stalled
add $t0, $s0, $s1       # $s0 forwarded       $s1 forwarded        $t1 stalled
and $t1, $t1, $t0       # $t0 forwarded       $t1 forwarded
```

9

Q4. a) The pipelined 5-stage MIPS processor is running the following program. Which registers are being written, and which are being read in the fifth cycle? Assume the processor has a hazard unit with forwarding capability. Show the stages (IF, ID, EX, M, WB) for each instruction in the pipeline diagram. Register names without filling the diagram will not be accepted.

	Clock Cycles								
Instructions	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
add $s0, $t0, $t1	IF	ID	EX	M	WB				
sub $s1, $t2, $t3		IF	ID	EX	M	WB			
and $s2, $s0, $s1			IF	ID	ID	EX	M	WB	
or $s3, $t4, $t5				IF	IF	ID	EX	M	WB
slt $s4, $s2, $s3						IF	IF	ID	EX

Written register/s	**Read register/s**
$s0	$s2

b) The pipelined 5-stage MIPS processor is running the following program. Which registers are being written, and which are being read in the fifth cycle? Assume the processor has a hazard unit with forwarding capability. Show the stages (IF, ID, EX, M, WB) for each instruction in the pipeline diagram. Register names without filling the diagram will not be accepted.

	Clock Cycles								
Instructions	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
sw $t5, 72($t0)	IF	ID	EX	M	WB				
addi $s1, $s2, 5		IF	ID	EX	M	WB			
sub $t0, $t1, $t2			IF	ID	EX	M	WB		
lw $t3, 15($s1)				IF	ID	ID	EX	M	WB
or $t2, $s4, $s5					IF	IF	ID	EX	M

Written register/s	**Read register/s**
$t5	$s1