CS2100 Computer Organization
Tutorial 4: Data and Control Path
(Week 6: 18 Sep – 22 Sep 2023)

1. [Past-year's exam question]

   An ISA has 16-bit instructions and 5-bit addresses. There are two classes of instructions: class *A* instructions have one address, while class *B* instructions have two addresses. Both classes exist and the encoding space for opcode is completely utilized.

   (a) What is the minimum total number of instructions?

   (b) What is the maximum total number of instructions?

2. You have seen how **`blt`** ("branch less than") can be implemented in the lecture slides. MIPS assembly also allows for "pseudo-instructions" which the assembler will expand to multiple instructions to implement the functionality.Show how the following pseudo-instructions are implemented using real MIPS instructions:

   (a) **`bgt $r1, $r2, L`** ("bgt = branch greater than")

   (b) **`bge $r1, $r2, L`** ("bge = branch greater than or equal")

   (c) **`ble $r1, $r2, L`** ("ble = branch less than or equal")

   (d) **`li $r, imm`** ("load immediate" where the immediate can be any length up to 32-bits)

   (e) **`nop`** ("No operation", i.e., a null operation)

Questions 3 and 4 refer to the complete datapath and control design covered in lectures #11 and #12. Please use the diagram in Lecture #12 slide 29 or in the COD MIPS 4th edition textbook, Figure 4.17. For your convenience, Lecture #12 slide 29 is also included at the end of this tutorial sheet.

3. Let us perform a complete trace to understand the working of the complete datapath and control implementation. Given the following three hexadecimal representations of MIPS instructions:

   i. `lw $24, 0($15)`
   ii. `beq $1, $3, 12`
   iii. `sub $25, $20, $5`

   For each instruction encoding, do the following:

   (a) Give the binary representation for the instruction.

   (b) Fill in the tables below. The first table concerns with the various data (information) at each of the datapath elements, while the second table records the control signals generated. Use the notation $8 to represent register number 8, [$8] to represent the content of register number 8 and Mem(X) to represent the memory data at address X.

Registers File				ALU		Data Memory	
RR1	RR2	WR	WD	Opr1	Opr2	Address	Write Data
$15	$24	$24	Mem($15)	[$15]	0	[$15] + 0	[$24]
$1	$3	$3 or $0	[$1] - [$3] or random value	[$1]	[$3]	[$1] - [$3]	[$3]
$20	$5	$25	[$20] -[$5]	[$20]	[$5]	[$20] -[$5]	[$5]

[Wr = Write; Rd = Read; M = Mem; R = Reg]

get all these values from lecture 6 table

RegDst	RegWr	ALUSrc	MRd	MWr	MToR	Brch	ALUop	ALUctrl
0	1	1	1	0	1			
X	0	0	0	0	X			
1	1	0	0	0	0			

1 for R-format; 0 for I-format | 1 for write; 0 for no write | 1 for I-format; 0 for R-format

   (c) Indicate the value of the PC after the instruction is executed.

4. We shall now estimate the latency (time needed for a task) for the various type of instructions. Given below are the resource latencies of the various hardware components (ps = picoseconds = 10^{-12} second):

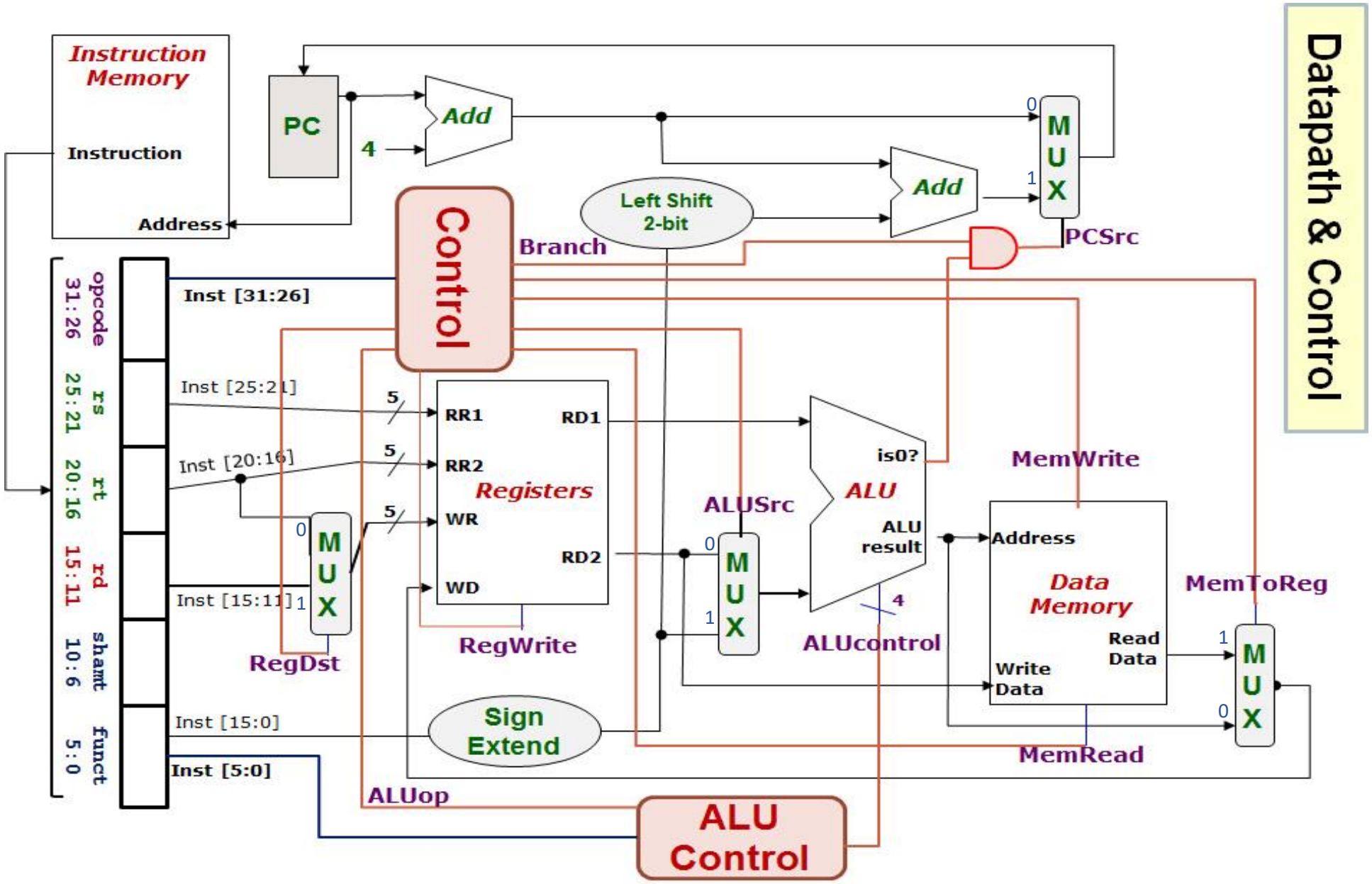
Inst-Mem	Adder	MUX	ALU	Reg-File	Data-Mem	Control/ ALUControl	Left-shift/ Sign-Extend/ AND
400ps	100ps	30ps	120ps	200ps	350ps	100ps	20ps

   Give the estimated latencies for the following MIPS instructions:

   (a) "SUB" instruction (e.g. `sub $25, $20, $5`)
   (b) "LW" instruction (e.g. `lw $24, 0($15)`)

   What do you think the **cycle time** should be for this particular processor implementation?

   Hint: First, you need to find out the **critical path** of an instruction, i.e. the path that takes the longest time to complete. Note that there could be several parallel paths that work more or less simultaneously


Datapath & Control
Instruction Memory
Instruction
Address
PC
4
Add
Add
Left Shift 2-bit
0
1
M U X
PCSrc
Branch
Control
opcode 31:26
rs 25:21
rt 20:16
rd 15:11
shamt 10:6
funct 5:0
Inst [31:26]
Inst [25:21]
Inst [20:16]
Inst [15:11]
Inst [15:0]
Inst [5:0]
5
RR1
RR2
WR
WD
RD1
RD2
Registers
RegWrite
RegDst
ALUSrc
is0?
ALU
ALU result
4
ALUcontrol
MemWrite
Address
Data Memory
Read Data
Write Data
MemRead
MemToReg
Sign Extend
ALUop
ALU Control