CO224- DIGITAL DESIGN

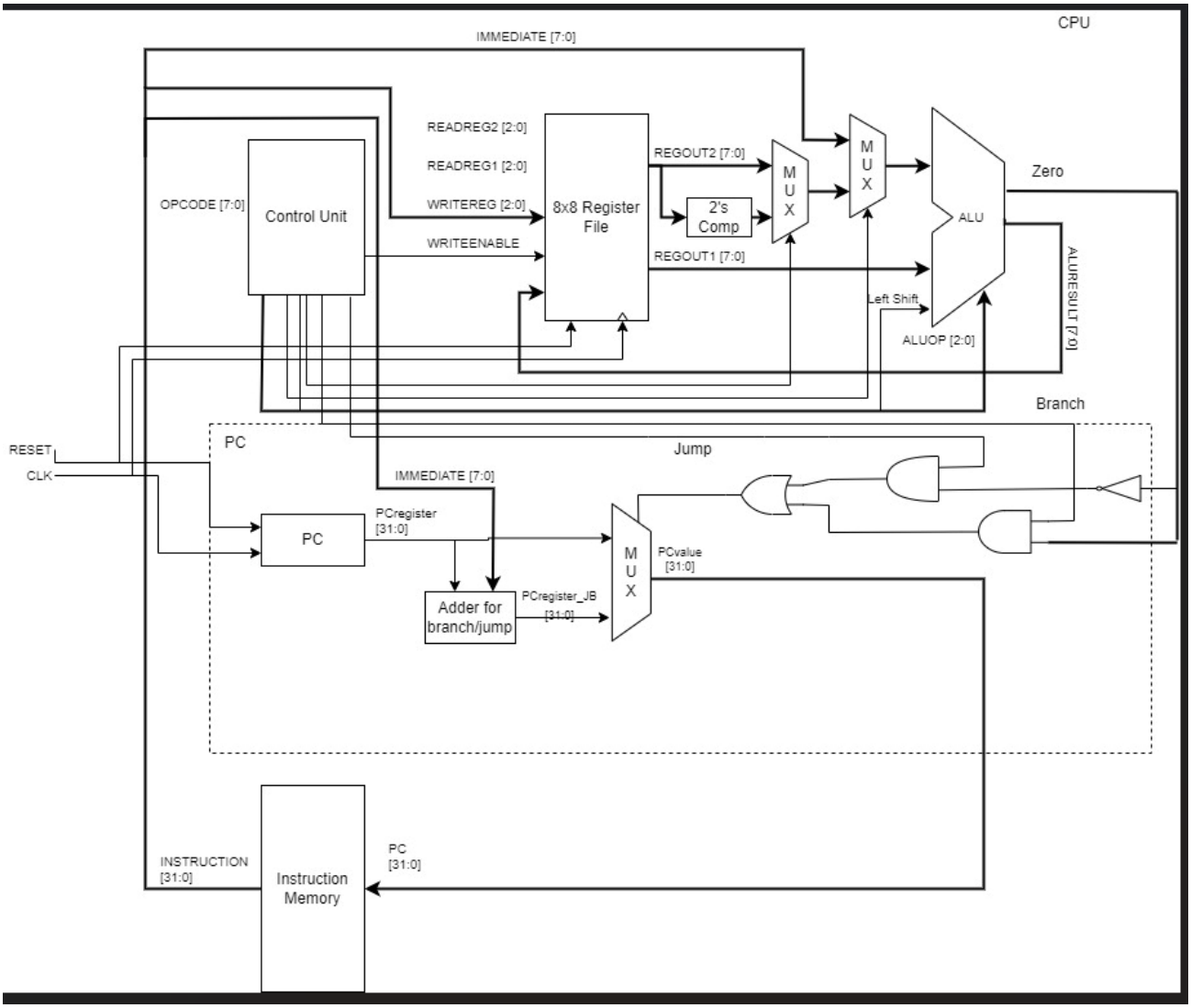
LAB 5 – PART5

BONUS

GROUP MEMBERS: E/18/379

E/18/147

Data path


CPU
IMMEDIATE [7:0]
READREG2 [2:0]
READREG1 [2:0]
WRITEREG [2:0]
WRITEENABLE
OPCODE [7:0]
Control Unit
8x8 Register File
REGOUT2 [7:0]
REGOUT1 [7:0]
2's Comp
M U X
M U X
ALU
Zero
ALURESULT [7:0]
Left Shift
ALUOP [2:0]
Branch
Jump
PC
RESET
CLK
IMMEDIATE [7:0]
PC
PCregister [31:0]
Adder for branch/jump
PCregister_JB [31:0]
M U X
PCvalue [31:0]
INSTRUCTION [31:0]
Instruction Memory
PC [31:0]

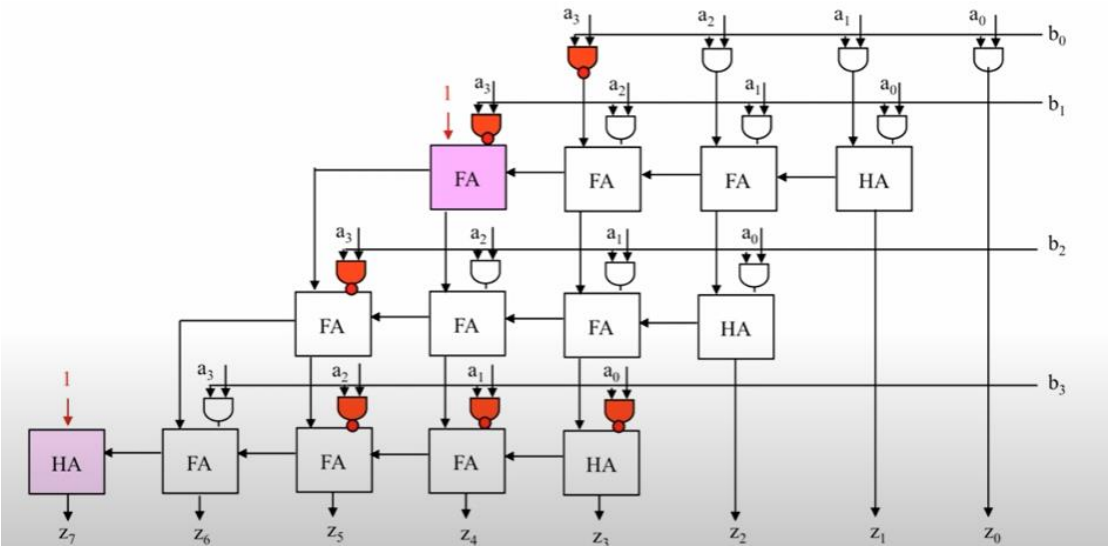
Assumptions for latencies

OPCODE	Select	Function	Unit's Delay
0001 0001	111	mult	#3
0000 1101	100	sll	#2
0000 1110	100	srl	#2
0000 1111	101	sra	#2
0001 0000	110	ror	#2
0000 1100	001	bne	#1

1. mult

The following circuit was used to get the signed multiplication of two 4-bit binary numbers. This operation is assumed to take a latency of #3-unit delays.

The circuit was implemented using full and half adders.



a: DATA1 (Multiplicand)

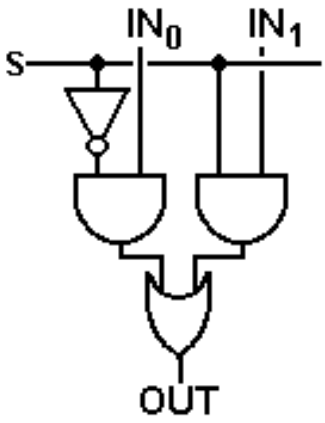
b: DATA2 (Multiplier)

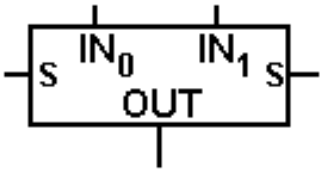
z: OUTPUT

2. sll, srl, sra, ror

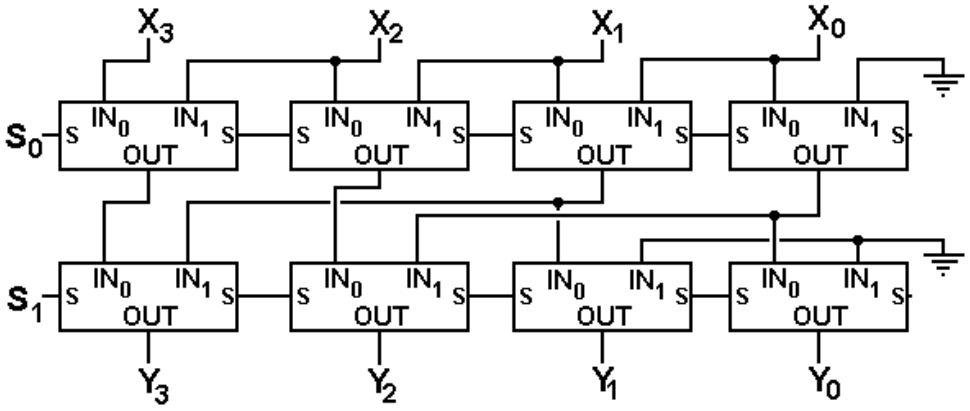
An additional control signal was generated to accommodate both sll and srl in the same functional unit.
The sra and ror instructions have their own functional units.

The following circuit logic was used to implement all the shift and rotate functions in gate level. The basic unit is a multiplexer/ shift unit shown below.





Using appropriate inputs and select signals shifting was done as follows.



X: DATA1
S: Shift amount
Y: Output

3. bne

No additional control signals were generated here.
The same functional unit used for the add instruction was used for this purpose.
To generate the correct signals, the output signal zero of the ALU was used in its ACTIVE LOW state.