LAB 5 Part 5

Name : Perera G. K. B. H
E No : E/16/276

- Supported Instructions and Relevant OPCODES

INSTRUCTION	OPCODE
loadi	0000 0000
mov	0000 0001
add	0000 0010
sub	0000 0011
and	0000 0100
or	0000 0101
j	0000 0110
beq	0000 0111
bne	0000 1000
srl	0000 1001
sra	0000 1010
ror	0000 1011
sll	0000 1100
mult	0000 1101

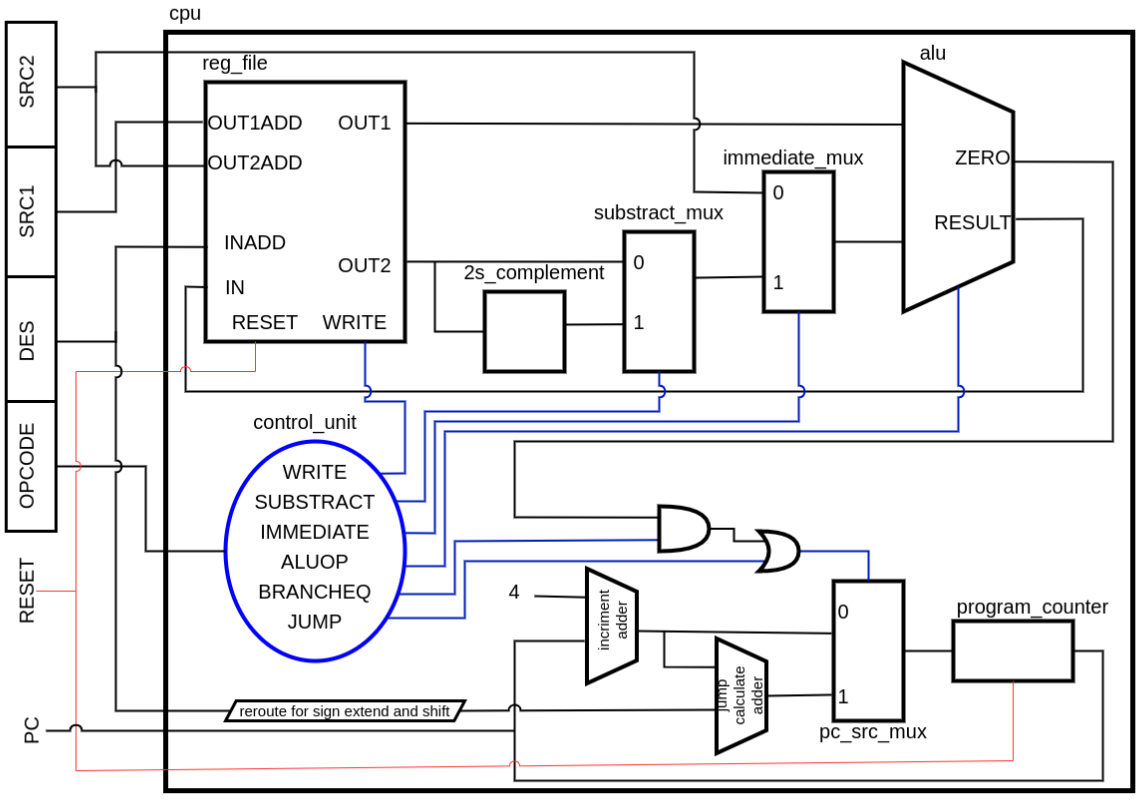
- ALUOP

ALUOP	Delay	DEFINITION	RELATED INSTRUCTIONS
000	#1	Forward	loadi, mov
001	#2	Add	add, sub, beq, bne
010	#1	And	and
011	#1	Or	or
100	#2	Result from barrel shifter	sll, srl, sra, ror
101	#4	Result form multiplier	mult

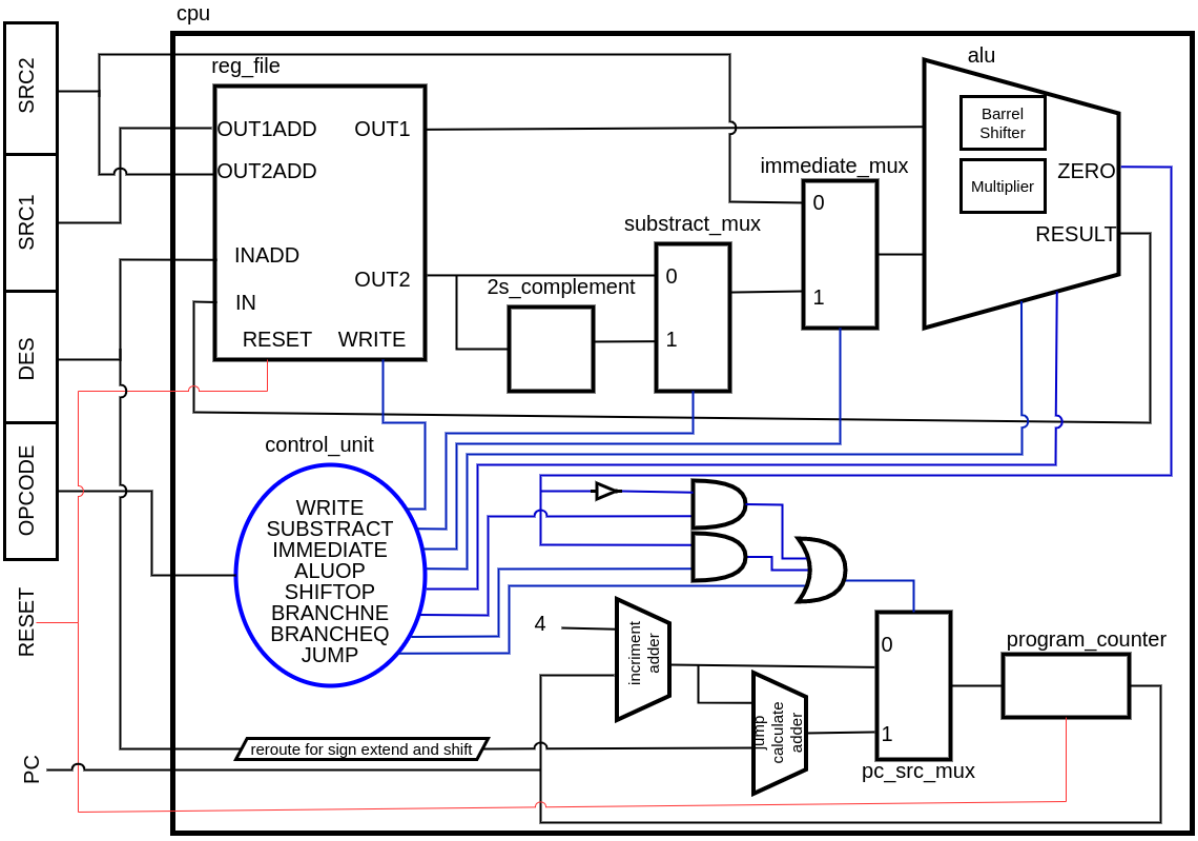
- Control Signals

Control Signal	No if bits	Purpose
ALUOP	3 bits	Perform necessary ALU operation
WRITE	1 bit	Enable write to the register
IMMEDIATE	1 bit	Select the Immediate value
SUBSTRACT	1 bit	Select the 2s complement
JUMP	1 bit	To control the input to the program counter resister
BRANCHEQ	1 bit	To control the input to the program counter resister
BRANCHNE	1 bit	To control the input to the program counter resister
SHIFTOP	2 bits	Control signal for barrel shifter inside the ALU

- Datapath of the CPU for part 4

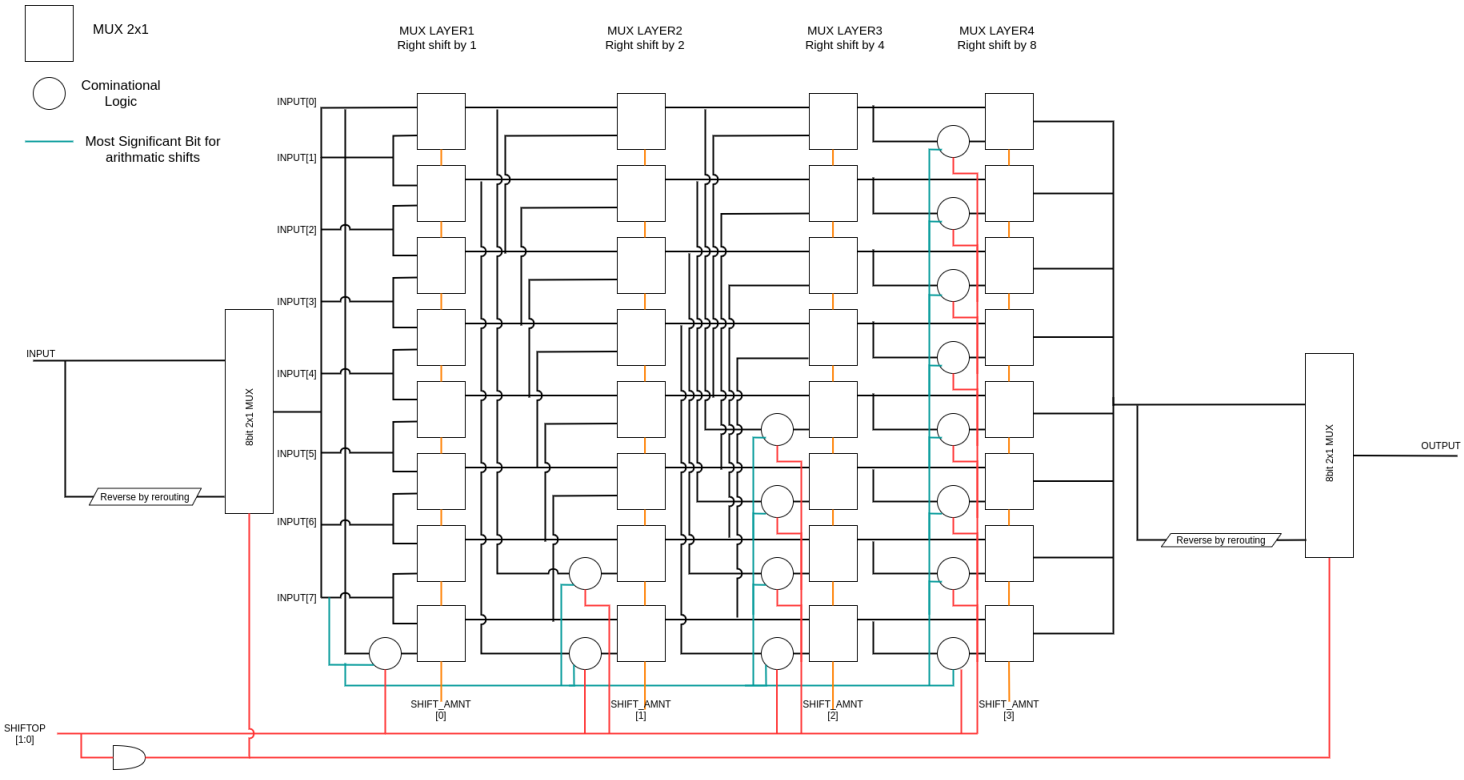


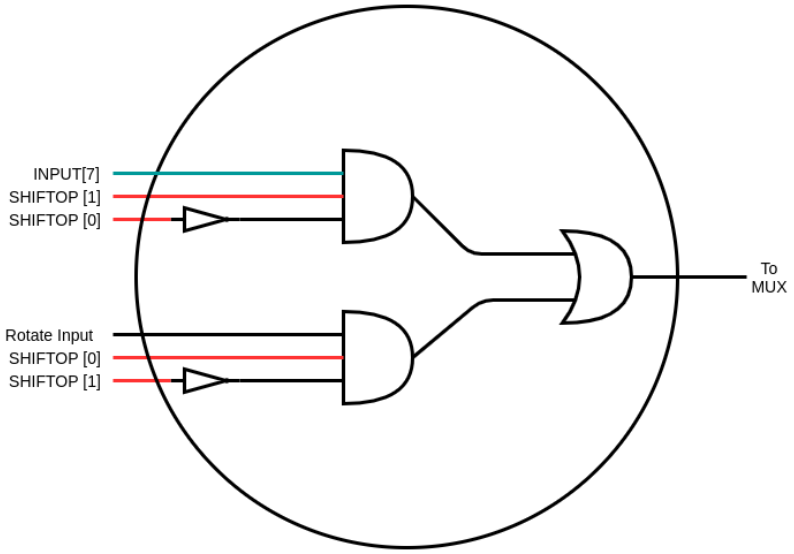
- Changes made in part 5
  - Datapath for Part 5



- Shift Operations (ssl, srl, sra, ror)
  - A barrel shifter is implemented inside the ALU to perform bit shifting.
  - Left shifting is achieved by reversing the input and the output streams of the barrel shifter. Reversing is done by rerouting the wires.
  - A new control signal of 2 bits is generated from the control unit to perform the required shifting in the barrel shifter.
  - Shifter always operates, but the shifted output is connected to the ALU output only when the relevent ALUOP is received.
  - Shifter delay was assumed as #2.

SHIFTOP	Related Instruction
00	srl
01	sra
10	ror
11	sll



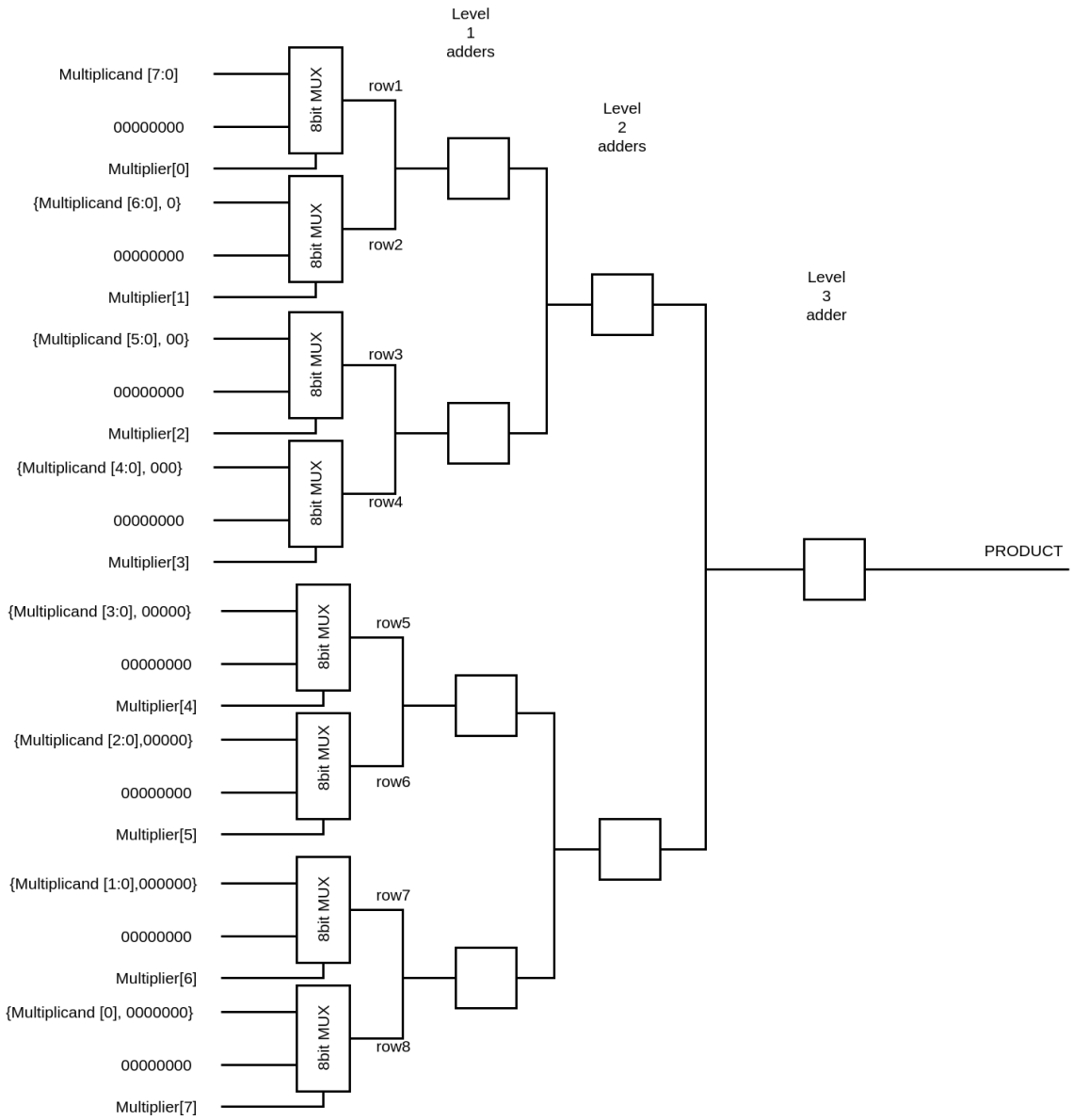


- Branch if not equal
    - ZERO from the ALU is inverted and it is connect to an AND gate (other input is BRANCHNE), the output from the AND gate is passed through an OR gate and the output of OR gate is connected to the PC_SRC_MUX.

- Multiplication of unsigned 8bits (Product <256)
    - Partial products are summed up with 7 adders which are in 3 levels.
        - 1st Level - parallel 4 adders
        - 2nd Level - parallel 2 adders
        - 3rd Level - 1 adder
    - Since unsigned and the result is of 8 bits, overflows were neglected.
    - The delay of a 8bit adder is assumed as #1 (relative to 32bit adder which is #2).Therefore the delay for multiplying is assumed as #4.
    - Multiplier always operates, but the product is connected to the ALU output only when the relevent ALUOP is recived.

A7	A6	A5	A4	A3	A2	A1	A0						
						B7	B6	B5	B4	B3	B2	B1	B0
				row1		A7B0	A6B0	A5B0	A4B0	A3B0	A2B0	A1B0	A0B0
				row2		A6B1	A5B1	A4B1	A3B1	A2B1	A1B1	A0B1	
				row3		A5B2	A4B2	A3B2	A2B2	A1B2	A0B2		
				row4		A4B3	A3B3	A2B3	A1B3	A0B3			
				row5		A3B4	A2B4	A1B4	A0B4				
				row6		A2B5	A1B5	A0B5					
				row7		A1B6	A0B6						
				row8		A0B7							

- Product is obtained by adding the partial products.


Level
1
adders
Level
2
adders
Level
3
adder
Multiplicand [7:0]
00000000
Multiplier[0]
{Multiplicand [6:0], 0}
00000000
Multiplier[1]
{Multiplicand [5:0], 00}
00000000
Multiplier[2]
{Multiplicand [4:0], 000}
00000000
Multiplier[3]
{Multiplicand [3:0], 00000}
00000000
Multiplier[4]
{Multiplicand [2:0],00000}
00000000
Multiplier[5]
{Multiplicand [1:0],000000}
00000000
Multiplier[6]
{Multiplicand [0], 0000000}
00000000
Multiplier[7]
8bit MUX
8bit MUX
8bit MUX
8bit MUX
8bit MUX
8bit MUX
8bit MUX
8bit MUX
row1
row2
row3
row4
row5
row6
row7
row8
PRODUCT