

EL323//EI324 Digital System Laboratory II

LAB 8: Arithmetic Logic Unit

Objective:

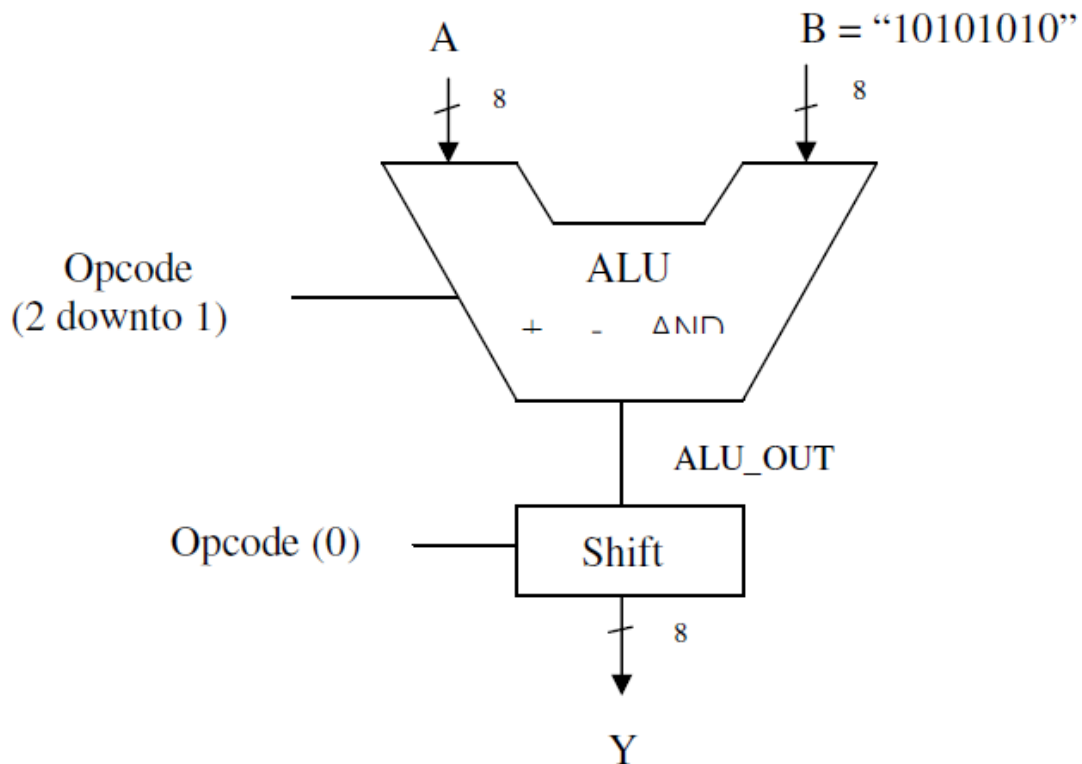
1. Can use the Quartus II and create the new project using VHDL
2. To know the assign pin of DE-0 in Quartus.
3. Design using VHDL
4. To know the ALU and Create Block diagram.

LAB 8.1 ALU8BITS VHDL CODE

1. Create new Project name "ALU8bit".
2. Create new entity for ALU in this below.

Construct an 8-bit arithmetic logic unit (ALU), that adds, subtracts, bitwise ANDs, or bitwise ORs, two operands and then performs an operational shift on the output.

Opcode	Operation	Function
00x	$ALU_OUT \leq A + B$	Add
01x	$ALU_OUT \leq A - B$	Subtract
10x	$ALU_OUT \leq A \text{ AND } B$	And
11x	$ALU_OUT \leq A \text{ OR } B$	Or
xx0	$Y \leq ALU_OUT$	ALU_OUT
xx1	$Y \leq \text{SHL}(ALU_OUT)$	Shift_Left



3. The SW₇₋₀ are the input of A, the SW₉₋₈ are the Opcode(2 downto 1) and the Button₀ is the Opcode(0). The LED₇₋₀ are the output of Y.
4. Use the DE0 User Manual to define the DE0 pin.
5. Compile the code and program to DE0.

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_ARITH.all;
USE IEEE.STD_LOGIC_UNSIGNED.all;
ENTITY ALU8BITS IS
PORT( -- Input Signals
      Opcode : in std_logic_vector(2 DOWNTO 0);
      A : in std_logic_vector(7 DOWNTO 0);
      -- Output Signals
      Y : out std_logic_vector(7 DOWNTO 0));
END ALU8BITS;
ARCHITECTURE behavior OF ALU8BITS IS
  -- declare signal(s) internal to module here
  SIGNAL ALU_OUT, B : std_logic_vector(7 DOWNTO 0);
BEGIN
  B <= "10101010";
  PROCESS (Opcode, A)
  BEGIN
    -- Select Arithmetic/Logical Operation
    CASE Opcode (2 DOWNTO 1) IS
      WHEN "00" =>
        ALU_OUT <= _____;
      WHEN "01" =>
        ALU_OUT <= _____;
      WHEN "10" =>
        ALU_OUT <= _____;
      WHEN "11" =>
        ALU_OUT <= _____;
      WHEN OTHERS =>
        ALU_OUT <= "00000000";
    END CASE;
    -- Select Shift Operation
    IF Opcode(0) = '1' THEN
      -- Shift bits left with zero fill using concatenation operator
      Y <= _____;
    ELSE
      Y <= _____;
    END IF;
  END PROCESS;
END behavior;

```

Conclusion

.....

.....

.....

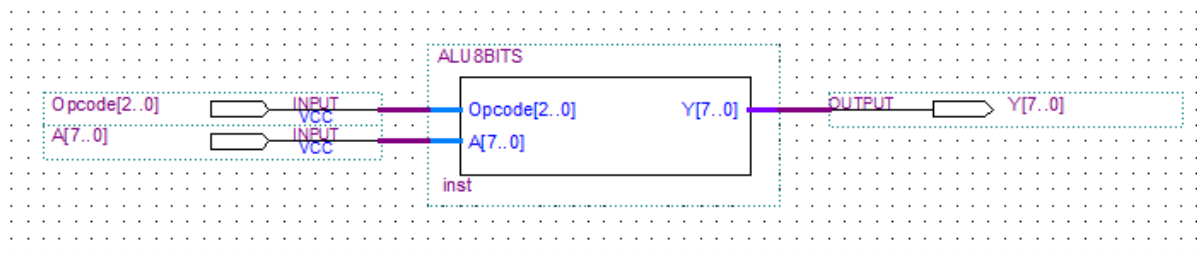
.....

.....

.....

LAB8.2 ALU8BITS BLOCK DIAGRAM

1. Create new Project name "ALU8BITBLOCK".
2. Create new Block Diagram like below.



Block Diagram of ALU8BITS.

3. Use the DE0 User Manual to define the DE0 pin.
4. Compile the code and program to DE0.

Result

A	B	Opcode	Y
00001111	10101010	000	
11110000	10101010	010	
11001100	10101010	100	
00110011	10101010	110	
10101010	10101010	111	
11100011	10101010	101	

Result Simulation

in	▷ A	B 00000000
in	▷ Opcode	B 000
out	▷ Y	B XXXXXXXX

Conclusion

.....

.....

.....

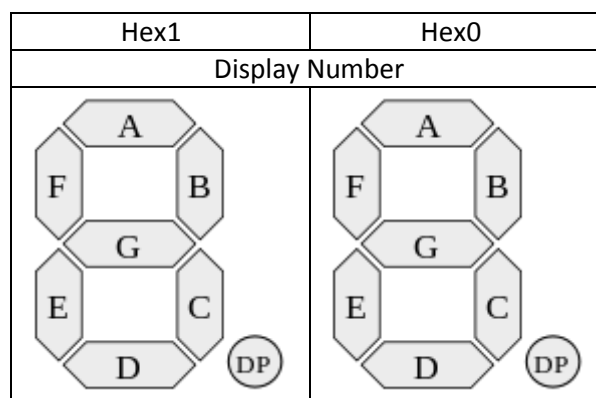
.....

LAB8.3 ALU4BIT with seg7decoder

1. Create new Project name "ALUseg7decoder".
2. Create new entity for shift register in this below.

Opcode	Operation	Function
00x	ALU_OUT <= A + B	Add
01x	ALU_OUT <= A - B	Subtract
10x	ALU_OUT <= A XNor B	Multiply
11x	ALU_OUT <= A Xor B	Divide
xx0	Y <= ALU_OUT	ALU_OUT
xx1	Y <= SHL(ALU_OUT)	Shift_LEFT

THE Output Y Show in the 7-segment



- Use the SW₃₋₀ are the input of A, SW₇₋₄ are the input of B. the SW₉₋₈ are the Opcode(2 down to 1) and the Button₀ is the Opcode(0). and Y is Shown in seg7decoder.
- Use the DE0 User Manual to define the DE0 pin.
- Compile the code and program to DE0.

Conclusion

.....

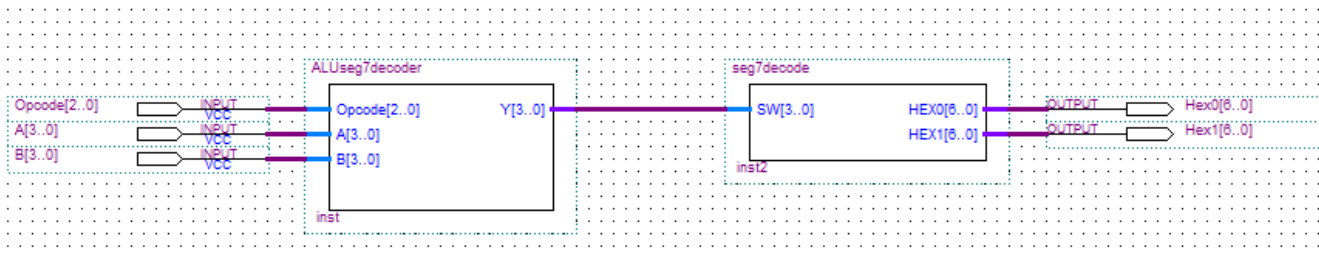
.....

.....

.....

LAB8.4 ALUseg7decoder BLOCK DIAGRAM

- Create new Project name "ALU4bitBlock".
- Create new Block Diagram like below.



Block Diagram of ALU4ITS with 7 segment.

- Use the DE0 User Manual to define the DE0 pin.
- Compile the code and program to DE0.

Result

A	B	Opcode	Y
7	8	000	
4	2	010	
5	5	100	
0	9	110	
1	1	111	
1	1	101	

Result

.....

.....

.....

.....

Conclusion

.....

.....

.....

.....