R8_12-08-2021

12 August 2021 09:12

⊛ SPPU Assignments (PART-A)

(A.1) 4-bit ALU with Nand, Nor, And, or, ALU Pass → send i/p → o/p

⇒ A.L.U = Arithmetic Logic Unit

Arithmetic operations

④
- ⊛ Addition ⑦ ✓
- ⊛ Subtraction ⑧ ✓
- ⊛ Multiplication
- ⊛ Division

Logical operations

⑧
- 1 ⊛ nand ✓
- ⊛ nor
- 2 ⊛ not ✓
- 3 ⊛ and ✓
- 4 ⊛ or ✓
- 5 ⊛ exor ✓
- ⊛ exnor
- 6 ⊛ ALU pass ✓

⑧ operation → 6 Logical, 2 Arith.

↓

3-bit

⇒ 4-bit ALU = ALU which can process 4-bits simultaneously

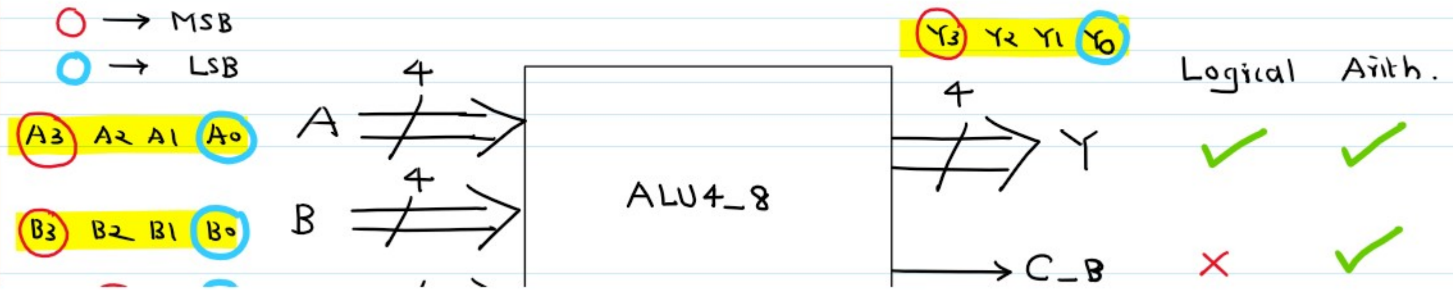
= Accept 4-bit Binary Number as an i/p

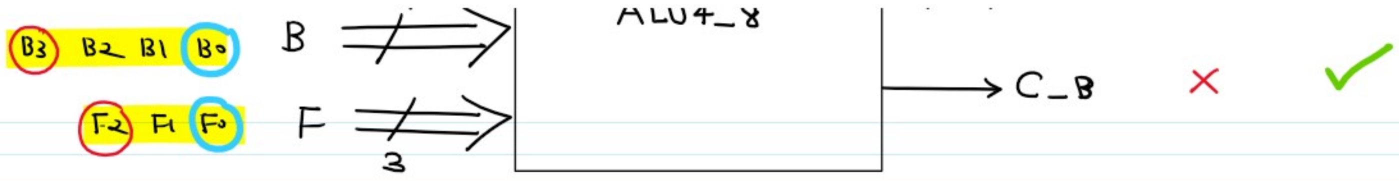
= Accept $\underbrace{\text{4-bit}}_{\text{NIBBLE}}$ OPERANDS

⇒ n- operations can be performed by an ALU on OPERANDS.

⇒ The FUNCTION LINES (BUS) decides which operation is performed.

⊛ BLOCK DIAGRAM OF 4-bit ALU :-





(*) The Result of Logical operation bet'n 2 n-bit Binary nos. will be ____n-____ bit Long

(*) ______ ‖ – Arithmetic ______ ‖ ______ be (n+1) bit Long

$$A = A_3\,A_2\,A_1\,A_0 = 1010 \qquad \overset{1\,1}{1010}$$

$$B = B_3\,B_2\,B_1\,B_0 = 0110 \qquad +\;1110$$

$$\underline{1101} \qquad \boxed{1}\,1000$$

$$4 \qquad 5$$

⇒ Addition operation → SUM
→ CARRY

⇒ Subtraction operation → DIFFERENCE
→ BORROW

⇒ For Logical operations:-
C_B : Neglected
Y : Result

⇒ For Addition operation:-
C_B : CARRY
Y : SUM

⇒ For Subtraction operation:-
C_B : BORROW
Y : DIFFERENCE

(*) (*) FUNCTION-TABLE :- A = 1010 B = 0110

F			Y	C_B	Y
F(2)	F(1)	F(0)			↓
x	$x/2$	$x/4$			
0	0	0	$\overline{A \cdot B}$	x	1101
0	0	1	$\overline{A}$	x	0101
0	1	0	$A \cdot B$	x	0010
0	1	1	$A + B$	x	1110
1	0	0	$A \oplus B$	x	1100
1	0	1	B	x	0110
1	1	0	$\overline{A+B}$	x	0001
1	1	1	$A \odot B$	x	0011

⊛ Modeling Style :-

⇒ Dataflow MS (WHEN - ELSE)

⇒ Behavioral MS → IF - THEN ----
└→ CASE .

⊛ Referring to the Function Table :-

⇒ The Time-periods of bits F(2), F(1), F(0) :-

⇒ If T2, T1 and To are Time-periods of above 3 bits

⇒ If T2 = x ns Then

$T1 = (x/2)$ ns

$To = (x/4)$ ns

⇒ Do the following in TVM :-

ⓐ Initialise Signal A to some Nibble - combination

ⓑ Initialise signal B to Nibble combination other than A

ⓒ Initialise signal F to "111"

ⓓ Remove :-

⇒ CONSTANT <clk-period> ---- Statement

⇒ Clock - generation process

ⓔ Add following to LS :-

```
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
```

ⓕ In Logic - section of AB write the following code :-

```
Stim_procF : PROCESS
             BEGIN
```

```
            F ⇐ F + 1;
             WAIT FOR 25 ns;
          END PROCESS;
```

(9) Run Simulation from ISE Tool.