# Assignment 5

IT<sub>451</sub>: COA LAB

# ANINDYA KUNDU

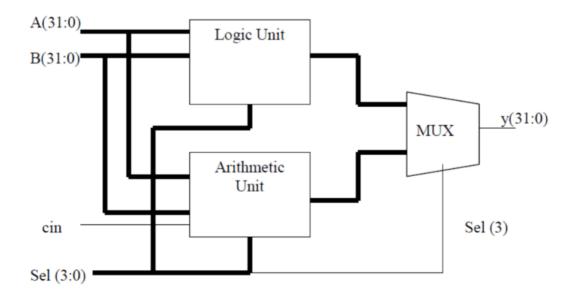
IT, 4<sup>th</sup> Semester ID: 510817020 (Hx-19)

# Question 1

Design and simulate a 32 bit ALU with standard operations as given in Assignment II Q 5.

- a. Synthesize it for Spartan 6. Report device utilization.
- b. Synthesize the previously designed 16 bit ALU for the same device and report device utilization.
- c. Create appropriate test benches to reflect its functional behavior for all possible cases. Include the snapshots in your report.
- d. Prepare a comparative table between the two designs to reflect their difference in hardware complexities.

Unit	Function	Sel
Arithmetic	Transfer a	0000
	Increment a	0001
	Decrement a	0010
	Transfer b	0011
	Increment b	0100
	Decrement b	0101
	Add a and b	0110
	Add a and b with carry	0111
Logical	Complement a	1000
	Complement b	1001
	AND	1010
	OR	1011
	NAND	1100
	NOR	1101
	XOR	1110
	XNOR	1111



#### VHDL Module:

#### arithUnit.vhd

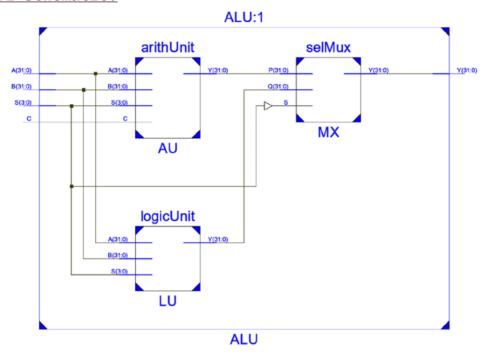
```
library IEEE;
use IEEE.STD LOGIC 1164.ALL;
use IEEE.NUMERIC STD.ALL;
entity arithUnit is
   Port ( S : in STD_LOGIC_VECTOR ( 3 downto 0);
          A : in
                     STD_LOGIC_VECTOR (31 downto 0);
          B : in STD_LOGIC_VECTOR (31 downto 0);
          C : in
                      STD LOGIC:
          Y : inout
                      STD LOGIC VECTOR (31 downto 0));
end arithUnit;
architecture Behavioral of arithUnit is
begin
process(S, A, B, C)
     variable uA, uB: unsigned(31 downto 0) := (others => '0');
begin
     uA := unsigned(A);
     uB := unsigned(B);
     if(S(3) = '0') then
           case S is
                 when "0000" => Y <= A;
                 when "0001" =>
                      Y <= STD LOGIC VECTOR(uA + 1);
                 when "0010" =>
                      Y <= STD_LOGIC_VECTOR(uA - 1);</pre>
                 when "0011" => Y <= B;
                 when "0100" =>
                      Y <= STD LOGIC VECTOR(uB + 1);
                 when "0101" =>
                      Y <= STD_LOGIC_VECTOR(uB - 1);</pre>
                 when "0110" =>
                      Y <= STD LOGIC VECTOR(uA + uB);
                 when "0111" =>
                      if(C = '1') then Y <= STD LOGIC VECTOR(uA + uB + 1);</pre>
                                      Y <= STD LOGIC VECTOR(uA + uB + 0);
                      else
                      end if;
```

```
end case;
    end if:
end process;
end Behavioral;
                            logicUnit.vhd
library IEEE;
use IEEE.STD LOGIC 1164.ALL;
entity logicUnit is
   Port (S: in
               STD LOGIC VECTOR ( 3 downto 0);
               A : in
                             STD_LOGIC_VECTOR (31 downto 0);
         B : in
                STD LOGIC VECTOR (31 downto 0);
        Y : inout STD_LOGIC_VECTOR (31 downto 0));
end logicUnit;
architecture Behavioral of logicUnit is
begin
process(S, A, B, Y)
begin
    if(S(3) = '1') then
         case S is
              when "1000" => Y <= not A;
              when "1001" => Y <= not B;
              when "1010" => Y <= A and B;
              when "1011" => Y <= A or
              when "1100" => Y <= A nand B;
              when "1101" => Y <= A nor B;
              when "1110" => Y <= A xor B;
              when "1111" => Y <= A xnor B;
              end case;
    end if;
end process;
end Behavioral;
```

#### selMux.vhd

```
library IEEE;
use IEEE.STD LOGIC 1164.ALL;
entity selMux is
   Port ( S : in STD LOGIC;
           P : in STD LOGIC VECTOR (31 downto 0);
           Q : in STD LOGIC VECTOR (31 downto 0);
           Y : out STD_LOGIC_VECTOR (31 downto 0));
end selMux;
architecture Behavioral of selMux is
begin
      Y \le P when S = '0' else Q;
end Behavioral:
                                      ALU. vhd
library IEEE;
use IEEE.STD LOGIC 1164.ALL;
entity ALU is
   Port ( A : in STD LOGIC VECTOR (31 downto 0);
           B : in STD_LOGIC_VECTOR (31 downto 0);
           C : in STD_LOGIC;
           S : in STD LOGIC VECTOR ( 3 downto 0);
           Y : out STD_LOGIC_VECTOR (31 downto 0));
end ALU;
architecture Structural of ALU is
      signal M, N, 0 : STD_LOGIC_VECTOR (31 downto 0);
begin
AU : entity work.arithUnit port map(S, A, B, C, M);
LU : entity work.logicUnit port map(S, A, B, N);
MX : entity work.selMux
                        port map(S(3), M, N, Y);
end Structural;
```

## RTL Schematic:



#### VHDL Test Bench: tb\_ALU.vhd

```
LIBRARY ieee;
USE ieee.std logic 1164.ALL;
USE ieee.numeric_std.ALL;
ENTITY tb ALU IS
END tb ALU;
ARCHITECTURE behavior OF tb ALU IS
      COMPONENT ALU
            PORT ( A : in STD LOGIC VECTOR (31 downto 0);
                   B : in STD LOGIC VECTOR (31 downto 0);
                   C : in STD_LOGIC;
                   S : in STD LOGIC VECTOR ( 3 downto 0);
                   Y : out STD_LOGIC_VECTOR (31 downto 0));
      END COMPONENT;
SIGNAL a : STD LOGIC VECTOR (31 downto 0) := (others => '0');
SIGNAL b : STD LOGIC VECTOR (31 downto 0) := (others => '0');
SIGNAL c : STD_LOGIC := '0';
SIGNAL s : STD LOGIC VECTOR ( 3 downto 0) := (others => '0');
SIGNAL y : STD LOGIC VECTOR (31 downto 0);
BEGIN
      uut: ALU PORT MAP(
```

```
A \Rightarrow a
             B \Rightarrow b,
             C => c,
             S => s,
            Y => y
      );
process
begin
      a <= "000000001100000001010010111010";
      b <= "000000000000010101001001011011011";
      c <= '1';
      s <= "0000";
      while s >= "0000" loop
             wait for 100ns;
             s <= STD_LOGIC_VECTOR(unsigned(s) + 1);</pre>
      end loop;
      wait;
end process;
END;
```

## **Simulation:**

	0 ns	50 ns	100 ns	150 ns	200 ns	250 ns	300 ns	350 ns
a[31:0]		30.5	100,100		0101001010111010	250,15	500,115	550,15
b[31:0]					10010010111011			
C				000000000000000000000000000000000000000				
		0000	00	001	00	10	C	011
/[31:0]	000000011000	0000101001010111010	0000000011000000	0101001010111011	0000000011000000	0101001010111001	0000000000000101	010010010110
	400 ns	450 ns	500 ns	550 ns	600 ns	650 ns	700 ns	750 ns
a[31:0]	100,100	135,13			0101001010111010			100,100
(31:0)					10010010111011			
C				000000000000000000000000000000000000000	1001001011011011			
s[3:0]	·	0100	01	01	01	10		111
y[31:0]	00000000000000	0101001001011011100		1001001011011010		1110010110010101	000000001100101	
,[ozio]								
b[31:0]	800 ns	850 ns	900 ns		1,000 ns 01010010101111010 1001001011011011	1,050 ns	1,100 ns	1,150 ns
[31:0]	800 ns		900 ns	000000011000000	0101001010111010	1,050 ns	1,100 ns	1,150 ns
b[31:0] c s[3:0]		1000	10	0000000011000000	01010010110111010	10		1,150 ns
b[31:0] c s[3:0]			10	000000011000000	01010010110111010			011
b[31:0] c s[3:0]	(1111111100111	1000	10111111111110101	00000000011000000 0000000000000001010 001	01010010111010 1001001011011011  1000000	0001001010011010	00000001100101	011
b[31:0] : s[3:0] y[31:0]		1000	10	00000000011000000 00000000000001010 001 011011	100100101111010 1001001011011011011 1000000	10	1	011
b[31:0] c s[3:0] y[31:0]	(1111111100111	1000	10111111111110101	00000000011000000 000000000000001010 01 01	0101001010111010 1001001011011011 1001001011011011011 100000000	0001001010011010	00000001100101	011
b[31:0] c s[3:0] y[31:0] a[31:0] b[31:0]	(1111111100111	1000	10111111111110101	00000000011000000 000000000000001010 01 01	100100101111010 1001001011011011011 1000000	0001001010011010	00000001100101	011
b(31:0) c s(3:0) y(31:0) a[31:0] b(31:0) c	(1111111100111	1000 1111010110101000101 11,250 ns	1(111111111111010111111111111111111111	00000000011000000 0010110100100100 0110110100100100 11,350 ns 0000000011000000 00000000011000000	0101001011011010 1001001011011011011 100000000	110 0001001010011010 11,450 ns	00000001100101 11,500 ns	011 011010010111
a[31:0] b[31:0] c s[3:0] y[31:0] a[31:0] b[31:0] c s[3:0] y[31:0]	1111111100111	1000	11,300 ns	00000000011000000 000000000000001010 01 01	0101001011011010 1001001011011011011 100000000	0001001010011010	00000001100101 11,500 ns	011 0110100101111 1,550 ns

# **Device Utilisation Summary:**

# 32-bit ALU

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slice LUTs	384	27288	1%	
Number of fully used LUT-FF pairs	0	384	0%	
Number of bonded IOBs	101	296	34%	

## 16-bit ALU

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slice LUTs	192	27288	0%	
Number of fully used LUT-FF pairs	0	192	0%	
Number of bonded IOBs	53	296	17%	

# **Hardware Complexity Comparison:**

32-bit ALU

16-bit ALU

		5						
Slice Logic Utilization	Used	Available	Utilization	Used	Available	Utilization		
Number of Slice Registers	0	54,576	0%	0	54,576	0%		
Number of Slice LUTs	384	27,288	1%	192	27,288	1%		
Number used as logic	381	27,288	1%	189	27,288	1%		
Number using O6 output only	226			114				
Number using O5 output only	92			44				
Number using O5 and O6	63			31				
Number used as ROM	0			0				
Number used as Memory	0	6,408	0%	0	6,408	0%		
Number used exclusively as route-thrus	3			3				
Number with same-slice register load	0			0				
Number with same-slice carry load	3			3				
Number with other load	0			0				
Number of occupied Slices	119	6,822	1%	67	6,822	1%		
Number of MUXCYs used	192	13,644	1%	96	13,644	1%		
Number of LUT Flip Flop pairs used	384			192				
Number with an unused Flip Flop	384	384	100%	192	192	100%		
Number with an unused LUT	0	384	0%	0	192	0%		
Number of fully used LUT-FF pairs	0	384	0%	0	192	0%		
Number of slice register sites lost to control set restrictions	0	54,576	0%	0	54,576	0%		
Number of bonded <u>IOBs</u>	101	296	34%	53	296	17%		