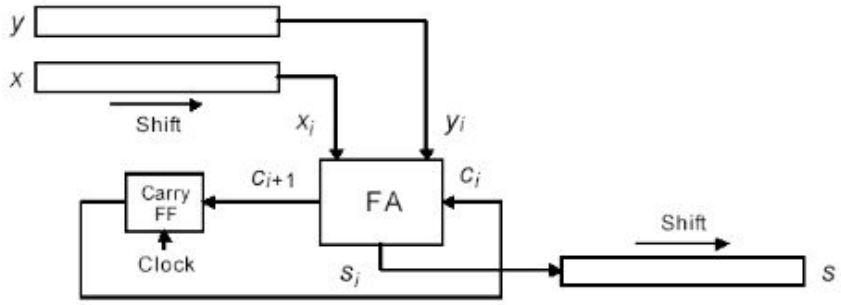
Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 7, 2020
ECE 420

Fall 2020
California State University, Northridge
Department of Electrical and Computer Engineering
Computer Assignment 4:
Serial Adder

The following diagram shows the top level block diagram of a synchronous serial adder. All the inputs and outputs are shown. The circuit consists of two serial data inputs: x and y, system clock, system reset, and a serial output data s, all single bits. Serial data are synchronized with clock and the ordering of bits on serial input streams is from LSB applied first to MSB applied last.



1. Write a structural code for the top level of the above serial adder. The lower level modules such as carry flip flop, shift registers, and full adder must be modeled behaviorally. The following VHDL entity is given for your reference. You should not use any port other than what is given here. **(25 points)**

```
entity sadder
port (x, y, clk, reset:      in std_logic;
      s:                     out std_logic);
end entity;
```

Your structural model must instantiate the lower level modules such as shift register, full adder, and single D flip flop. Write a paramterized VHDL model for serial input serial output register and instantiate it in your design three times for x, y, and s registers. **(10 points)**

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

Serial Adder (Top Level Design Code)

```
1  ----------------------------------------------------------------------------------
2  --
3  -- Engineer: Jose L. Martinez
4  --
5  -- Create Date: 11/07/2020 10:14:54 PM
6  -- Design Name: Serial Adder Design
7  --
8  ----------------------------------------------------------------------------------
9
10
11 library IEEE;
12 use IEEE.STD_LOGIC_1164.ALL;
13 use IEEE.NUMERIC_STD.ALL;
14
15 entity serialAdder is
16   Port (
17     x, y, clk, rst, start: in std_logic;
18     s: out std_logic);
19 end serialAdder;
20
21 architecture Behavioral of serialAdder is
22
23 component fa is
24   Port (
25     A: in std_logic;
26     B: in std_logic;
27     cin: in std_logic;
28     sum: out std_logic;
29     cout: out std_logic);
30 end component fa;
31
32 component shiftRegister is
33   Port (
34     D, clk, rst: in std_logic;
35     Q: out std_logic);
36 end component shiftRegister;
37
38 component dff is
39   Port (
40     D, clk, rst: in std_logic;
41     Q: out std_logic);
42 end component dff;
43
44 signal Yi, Xi, Ci, Co, Si: std_logic;
45 begin
46
47 coutDff: dff port map(D => Co, clk => clk, rst => rst, Q => Ci);
48 Adder: fa port map(A => Xi, B => Yi, cin => Ci, sum => Si, cout => Co);
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
49  XSISO: shiftRegister port map(D => x, clk => clk, rst => rst, Q => Xi);
50  YSISO: shiftRegister port map(D => y, clk => clk, rst => rst, Q => Yi);
51  SSISO: shiftRegister port map(D => Si, clk => clk, rst => rst, Q => s);
52
53  end Behavioral;
54
```

Full Adder (Design Code)

```
1  ----------------------------------------------------------------------------------
2  --
3  -- Engineer: Jose L. Martinez
4  --
5  -- Create Date: 11/07/2020 10:14:54 PM
6  -- Design Name: Full Adder Design
7  --
8  ----------------------------------------------------------------------------------
9
10
11 library IEEE;
12 use IEEE.STD_LOGIC_1164.ALL;
13 use IEEE.NUMERIC_STD.ALL;
14
15 entity fa is
16   Port (
17     A: in std_logic;
18     B: in std_logic;
19     cin: in std_logic;
20     sum: out std_logic;
21     cout: out std_logic);
22 end fa;
23
24 architecture Behavioral of fa is
25
26 begin
27
28 sum <= A xor B xor cin;
29 cout <= (A or B) and (A or cin) and (B or cin);
30
31 end Behavioral;
32
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

D-FF (Design Code)

```
----------------------------------------------------------------------------------
--
-- Engineer: Jose L. Martinez
--
-- Create Date: 11/07/2020 10:14:54 PM
-- Design Name: D-Flip-Flop Design
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity dff is
  Port (
    D, clk, rst: in std_logic;
    Q: out std_logic);
end dff;

architecture Behavioral of dff is

begin

process(clk, rst)
begin
    if rst = '1' then
        Q <= '0';
    elsif rising_edge(clk) then
        Q <= D;
    end if;
end process;

end Behavioral;

```

Shift Register (Design Code)

```
----------------------------------------------------------------------------------
--
-- Engineer: Jose L. Martinez
--
-- Create Date: 11/07/2020 10:14:54 PM
-- Design Name: Shift Register Design
--
----------------------------------------------------------------------------------


library IEEE;
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
12  use IEEE.STD_LOGIC_1164.ALL;
13  use IEEE.NUMERIC_STD.ALL;
14
15  entity shiftRegister is
16    generic (WIDTH: integer := 16);
17    Port (
18      D, clk, rst: in std_logic;
19      Q: out std_logic);
20  end shiftRegister;
21
22  architecture Behavioral of shiftRegister is
23
24  signal tmp_Signal: std_logic_vector(WIDTH-1 downto 0);
25  begin
26
27  process(clk, rst)
28  begin
29
30      if (rst = '1') then
31          tmp_signal <= (others => '0');
32      elsif rising_edge(clk) then
33          for i in 0 to WIDTH-2 loop
34              tmp_signal(i+1) <= tmp_signal(i);
35          end loop;
36          tmp_signal(0) <= D;
37      end if;
38  end process;
39
40  Q <= tmp_signal(WIDTH-1);
41
42  end Behavioral;
43
```

Serial Adder (Test Bench)

```
1   ----------------------------------------------------------------------------------
2   --
3   -- Engineer: Jose L. Martinez
4   --
5   -- Create Date: 11/07/2020 10:14:54 PM
6   -- Design Name: Serial Adder Test Bench
7   --
8   ----------------------------------------------------------------------------------
9
10
11  library IEEE;
12  use IEEE.STD_LOGIC_1164.ALL;
13  use IEEE.NUMERIC_STD.ALL;
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
14
15  entity serialAdder_tb is
16  end serialAdder_tb;
17
18  architecture Behavioral of serialAdder_tb is
19
20  component serialAdder is
21      Port (
22        x, y, clk, rst: in std_logic;
23        s: out std_logic);
24  end component serialAdder;
25
26      signal x_tb, y_tb, clk_tb, rst_tb: std_logic;
27      signal s_tb: std_logic;
28      constant CP: time := 10 ns;
29
30  begin
31
32  testSA: serialAdder port map(x => x_tb, y => y_tb, clk => clk_tb, rst => rst_tb, s => s_tb);
33
34  process
35  begin
36  clk_tb <= '1';
37  wait for CP/2;
38  clk_tb <= '0';
39  wait for CP/2;
40  end process;
41
42  process
43  begin
44  rst_tb <= '1';
45  wait for CP;
46  rst_tb <= '0';
47  wait;
48  end process;
49
50  process
51  begin
52  x_tb <= '0';
53  y_tb <= '0';
54  wait for CP;
55  x_tb <= '0';
56  y_tb <= '1';
57  wait for CP;
58  x_tb <= '1';
59  y_tb <= '1';
60  wait for CP;
61  x_tb <= '1';
62  y_tb <= '1';
63  wait for CP;
64  x_tb <= '0';
65  y_tb <= '1'; --
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
66      wait for CP;
67      x_tb <= '1';
68      y_tb <= '1';
69      wait for CP;
70      x_tb <= '1';
71      y_tb <= '1';
72      wait for CP;
73      x_tb <= '1';
74      y_tb <= '0';
75      wait for CP;
76      x_tb <= '1';
77      y_tb <= '0'; --
78      wait for CP;
79      x_tb <= '0';
80      y_tb <= '0';
81      wait for CP;
82      x_tb <= '0';
83      y_tb <= '0';
84      wait for CP;
85      x_tb <= '0';
86      y_tb <= '0';
87      wait for CP;
88      x_tb <= '0';
89      y_tb <= '0'; --
90      wait for CP;
91      x_tb <= '0';
92      y_tb <= '0';
93      wait for CP;
94      x_tb <= '1';
95      y_tb <= '1';
96      wait for CP;
97      x_tb <= '0';
98      y_tb <= '0';
99      wait for CP;
100     x_tb <= '1';
101     y_tb <= '1'; --
102     wait for CP;
103     x_tb <= '0';
104     y_tb <= '0';
105     wait;
106   end process;
107
108   end Behavioral;
109
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

Serial Adder (Waveforms)

2. The block diagram shown above does not indicate the start/end of serial data on the input/output. Assume this serial adder is used to add 16 bits of serial data at a time. Another input called START for the duration of 16 clock pulses indicates the LSB bit of the serial data. Modify your design to start adding the bits serially when START pulse is active. START bit is active for only one clock pulse. When START is enabled for one clock cycle, your circuit should start adding the next 16 bits serially and then stop until the next pulse. **(25 points)**

Serial Adder With Start (Waveforms)

```
1  ----------------------------------------------------------------------------------
2  --
3  -- Engineer: Jose L. Martinez
4  --
5  -- Create Date: 11/07/2020 10:14:54 PM
6  -- Design Name: Serial Adder Design
7  --
8  ----------------------------------------------------------------------------------
9
10
11 library IEEE;
12 use IEEE.STD_LOGIC_1164.ALL;
13 use IEEE.NUMERIC_STD.ALL;
14
15 entity serialAdder is
16     Port (
17       x, y, clk, rst, start: in std_logic;
18       s, max: out std_logic);
19 end serialAdder;
20
21 architecture Behavioral of serialAdder is
22
23 component fa is
24   Port (
25     A: in std_logic;
26     B: in std_logic;
27     cin: in std_logic;
28     sum: out std_logic;
29     cout: out std_logic);
30 end component fa;
31
32 component shiftRegister is
33   Port (
34     D, clk, rst: in std_logic;
35     Q: out std_logic);
36 end component shiftRegister;
37
38 component dff is
39   Port (
40     D, clk, rst: in std_logic;
41     Q: out std_logic);
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
42      end component dff;
43
44      signal Yi, Xi, Ci, Co, Si: std_logic;
45      signal enabled, clkEnabled: std_logic;
46      signal count: unsigned(5 downto 0);
47
48      begin
49
50      clkEnabled <= enabled and clk;
51
52      coutDff: dff port map(D => Co, clk => clkEnabled, rst => rst, Q => Ci);
53      Adder: fa port map(A => Xi, B => Yi, cin => Ci, sum => Si, cout => Co);
54      XSISO: shiftRegister port map(D => x, clk => clkEnabled, rst => rst, Q => Xi);
55      YSISO: shiftRegister port map(D => y, clk => clkEnabled, rst => rst, Q => Yi);
56      SSISO: shiftRegister port map(D => Si, clk => clkEnabled, rst => rst, Q => s);
57
58      process(clk)
59      begin
60          if start = '1' then
61              count <= (others => '0');
62              enabled <= '1';
63              max <= '0';
64          elsif count = to_unsigned(48, 6) then
65              max <= '1';
66              enabled <= '0';
67          elsif rising_edge(clk) then
68              count <= count + "0001";
69          end if;
70      end process;
71
72      end Behavioral;
73
```

3. Write a VHDL test bench for the serial adder design and show the correct functionality of your design for part 2. Your design must test a real case of x = 0xA0F6 and y = 0xA03F. What is the result in hex format? Does this operation create overflow if these inputs are signed? Does it create overflow if these inputs are unsigned? **(40 points)**

D-Flip-Flop (Test Bench)

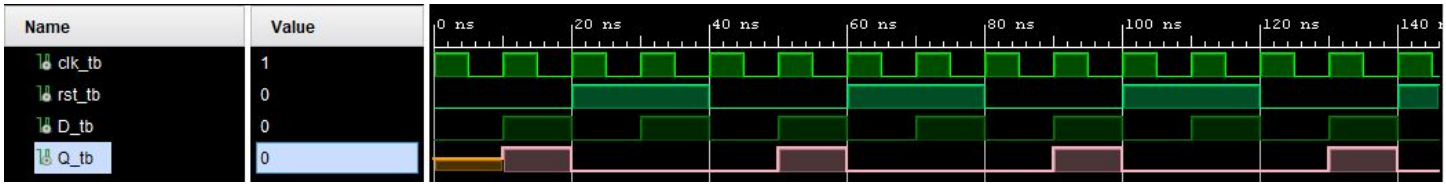
Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
----------------------------------------------------------------------------------
--
-- Engineer: Jose L. Martinez
--
-- Create Date: 11/07/2020 10:14:54 PM
-- Design Name: D-Flip-Flop Test Bench
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;


entity dff_tb is
--  Port ( );
end dff_tb;

architecture Behavioral of dff_tb is

component dff is
  Port (
    D, clk, rst: in std_logic;
    Q: out std_logic);
end component dff;

signal D_tb, clk_tb, rst_tb, Q_tb: std_logic;
constant CP: time := 10 ns;

begin

```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
33      testDff: dff port map(D => D_tb, clk => clk_tb, rst => rst_tb, Q => Q_tb);
34
35      process
36      begin
37      clk_tb <= '1';
38      wait for CP/2;
39      clk_tb <= '0';
40      wait for CP/2;
41      end process;
42
43      process
44      begin
45      D_tb <= '0';
46      rst_tb <= '0';
47      wait for CP;
48      D_tb <= '1';
49      rst_tb <= '0';
50      wait for CP;
51      D_tb <= '0';
52      rst_tb <= '1';
53      wait for CP;
54      D_tb <= '1';
55      rst_tb <= '1';
56      wait for CP;
57      end process;
58
59      end Behavioral;
60
```

D-Flip-Flop (Waveforms)



Full Adder (Test Bench)

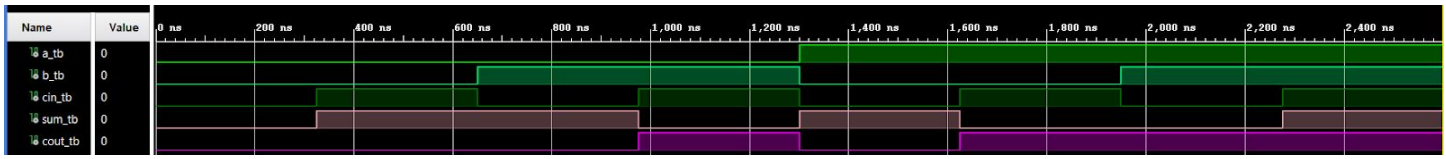
```
1   ----------------------------------------------------------------------------------
2   -- Engineer: Jose Luis Martinez
3   --
4   -- Create Date: 09/14/2020 09:33:16 AM
5   -- Module Name: fa_tb - Behavioral
6   -- Project Name: Computer Assignment 1
7   -- Revision 0.01 - File Created
8   ----------------------------------------------------------------------------------
9
10
11  library IEEE;
12  use IEEE.STD_LOGIC_1164.ALL;
13  use IEEE.NUMERIC_STD.ALL;
14
15  entity fas_tb is
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
16  --  Port ( );
17  end fas_tb;
18
19  architecture Behavioral of fas_tb is
20
21  signal a_tb: std_logic;
22  signal b_tb: std_logic;
23  signal cin_tb: std_logic;
24  signal sum_tb: std_logic;
25  signal cout_tb: std_logic;
26
27  component fa is
28      Port ( a : in STD_LOGIC;
29             b : in STD_LOGIC;
30             cin : in STD_LOGIC;
31             sum : out STD_LOGIC;
32             cout : out STD_LOGIC);
33  end component fa;
34
35  begin
36
37  uutt: fa port map ( a => a_tb,
38                      b => b_tb,
39                      cin => cin_tb,
40                      sum => sum_tb,
41                      cout => cout_tb);
42
43  process
44  begin
45
46  for aFor in 0 to 1 loop
47     for bFor in 0 to 1 loop
48        for cinFor in 0 to 1 loop
49
50           if aFor = 1 then a_tb <= '1';
51           else a_tb <= '0';
52           end if;
53
54           if bFor = 1 then b_tb <= '1';
55           else b_tb <= '0';
56           end if;
57
58           if cinFor = 1 then cin_tb <= '1';
59           else cin_tb <= '0';
60           end if;
61           wait for 325ns;
62
63        end loop;
64     end loop;
65  end loop;
66
67  end process;
68  end Behavioral;
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

Full Adder (Waveforms)



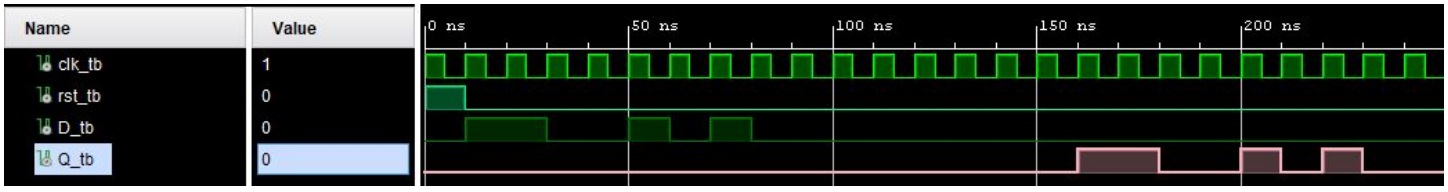
SISO (Test Bench)

```
1  ----------------------------------------------------------------------------------
2  --
3  -- Engineer: Jose L. Martinez
4  --
5  -- Create Date: 11/07/2020 10:14:54 PM
6  -- Design Name: Shift Register Test Bench
7  --
8  ----------------------------------------------------------------------------------
9
10
11 library IEEE;
12 use IEEE.STD_LOGIC_1164.ALL;
13 use IEEE.NUMERIC_STD.ALL;
14
15 entity shiftRegister_tb is
16 --  Port ( );
17 end shiftRegister_tb;
18
19 architecture Behavioral of shiftRegister_tb is
20
21 component shiftRegister is
22   generic (WIDTH: integer := 16);
23   Port (
24     D, clk, rst: in std_logic;
25     Q: out std_logic);
26 end component shiftRegister;
27
28 signal D_tb, clk_tb, rst_tb, Q_tb: std_logic;
29 constant CP: time := 10 ns;
30
31 begin
32
33 SISOTEST: shiftRegister port map(D => D_tb, clk => clk_tb, rst => rst_tb, Q => Q_tb);
34
35 process
36 begin
37 clk_tb <= '1';
38 wait for CP/2;
39 clk_tb <= '0';
40 wait for CP/2;
41 end process;
42
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
42
43      process
44      begin
45      rst_tb <= '1';
46      wait for CP;
47      rst_tb <= '0';
48      wait;
49      end process;
50
51      process
52      begin
53      D_tb <= '0';
54      wait for CP;
55      D_tb <= '1';
56      wait for CP;
57      D_tb <= '1';
58      wait for CP;
59      D_tb <= '0';
60      wait for CP;
61      D_tb <= '0';
62      wait for CP;
63      D_tb <= '1';
64      wait for CP;
65      D_tb <= '0';
66      wait for CP;
67      D_tb <= '1';
68      wait for CP;
69      D_tb <= '0';
70      wait;
71      end process;
72
73      end Behavioral;
74
```

SISO (Waveforms)



Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

Serial Adder With Start/End (Test Bench)

```
----------------------------------------------------------------------------------
--
-- Engineer: Jose L. Martinez
--
-- Create Date: 11/07/2020 10:14:54 PM
-- Design Name: Serial Adder Test Bench
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity serialAdder_tb is
end serialAdder_tb;

architecture Behavioral of serialAdder_tb is

component serialAdder is
    Port (
      x, y, clk, rst, start: in std_logic;
      s, max: out std_logic);
end component serialAdder;

    signal x_tb, y_tb, clk_tb, rst_tb, start_tb: std_logic;
    signal s_tb, max_tb: std_logic;
    constant CP: time := 10 ns;

begin

testSA: serialAdder port map(x => x_tb, y => y_tb, clk => clk_tb, rst => rst_tb, start => start_tb, s => s_tb, max => max_tb);

process
begin
clk_tb <= '1';
wait for CP/2;
clk_tb <= '0';
wait for CP/2;
end process;

process
begin
rst_tb <= '1';
wait for CP;
rst_tb <= '0';
wait;
end process;

process
begin
x_tb <= '0';
y_tb <= '0';
y_tb <= '0';
start_tb <= '1';
wait for CP;
start_tb <= '0';
x_tb <= '0';
y_tb <= '1';
wait for CP;
x_tb <= '1';
y_tb <= '1';
wait for CP;
x_tb <= '1';
y_tb <= '1';
wait for CP;
x_tb <= '0';
y_tb <= '1'; --
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

```
68      wait for CP;
69      x_tb <= '1';
70      y_tb <= '1';
71      wait for CP;
72      x_tb <= '1';
73      y_tb <= '1';
74      wait for CP;
75      x_tb <= '1';
76      y_tb <= '0';
77      wait for CP;
78      x_tb <= '1';
79      y_tb <= '0'; --
80      wait for CP;
81      x_tb <= '0';
82      y_tb <= '0';
83      wait for CP;
84      x_tb <= '0';
85      y_tb <= '0';
86      wait for CP;
87      x_tb <= '0';
88      y_tb <= '0';
89      wait for CP;
90      x_tb <= '0';
91      y_tb <= '0'; --
92      wait for CP;
93      x_tb <= '0';
94      y_tb <= '0';
95      wait for CP;
96      x_tb <= '1';
97      y_tb <= '1';
98      wait for CP;
99      x_tb <= '0';
100     y_tb <= '0';
101     wait for CP;
102     x_tb <= '1';
103     y_tb <= '1'; --
104     wait for CP;
105     x_tb <= '0';
106     y_tb <= '0';
107     wait;
108     end process;
109
110     end Behavioral;
111
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 8, 2020
ECE 420

Serial Adder With Start/End (Waveforms)