Sajeda Mulla 1160724

BIRZEIT UNIVERSITY

Faculty of Engineering and Technology
Electrical and Computer Engineering Department
ENCS 533 – Advanced Digital Design
2nd Sem. 2018/2019 Time: 80 minutes

Instructor: Dr. Abdellatif Abu-Issa

Q1) 24 points (4+10+10)

Figure 1 shows an unsigned BCD digit adder circuit that will add two BCD digits A = $A_3A_2A_1A_0$ and B = $B_3B_2B_1B_0$. It has a carry input from the previous digit, and will produce 5-bit output S = $S_4S_3S_2S_1S_0$ (S4 is also called Cout, but we will consider it here as S_4).

a. Write the description of the entity called "BCD" with inputs and outputs as described above and shown in Fig. 1.
b. If the system will be built structurally from two 4-bit adders as shown in Fig. 1. Write the structural architecture called "structural". For other gates you can describe them behaviorally.

```
ENTITY adder4 is
PORT(x,y: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
cin: IN STD_LOGIC;
S: OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
cout: OUT STD_LOGIC);
END;
ARCHITECTURE special OF adder4 is…
```

c. Assume we decide to build the BCD Digit Adder behaviorally, write the description of an architecture called "behavioral" for this purpose.

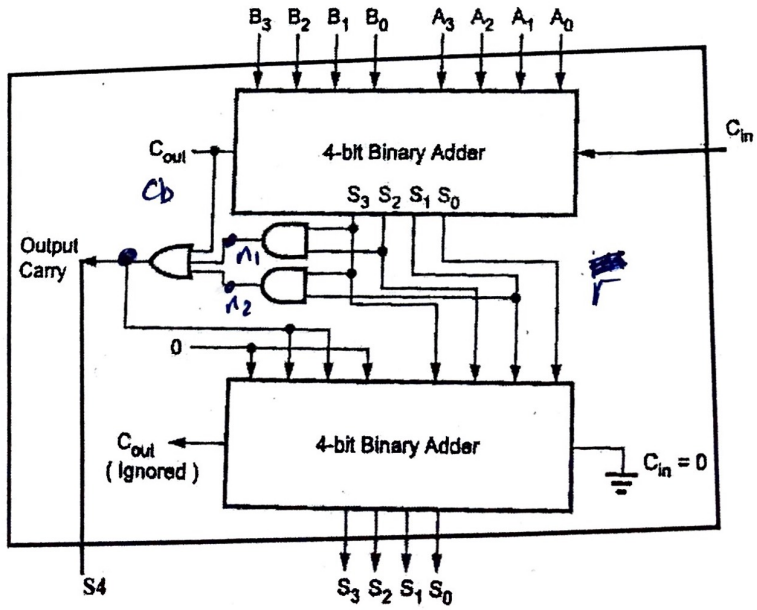


Figure 1: BCD Digit Adder

Note About BCD Adder: The adder adds BCD digit with another digit, each is represented by 4-bit binary code (Binary Coded Decimal). If the result of addition ≤ 9, then the result is a BCD number without any correction. However, if the result > 9 then we should add 6 to the binary result to correct it to be a BCD number.

Q2) (16 points)

Fig.2 shows a generic parallel to serial transmitter. In idle state, Sout = '1'. The system will transmit a frame when input go = '1', at the end of the frame, another frame will be sent if go='1'. Otherwise, the system will be in idle state again if go= '0' (Sout = '0'). The serial frame consists from n+2 bits with the following order: Start bit '0', Pin(0),Pin(1),…,Pin(n-1), Stop bit '1'.

Write a VHDL description for this system.

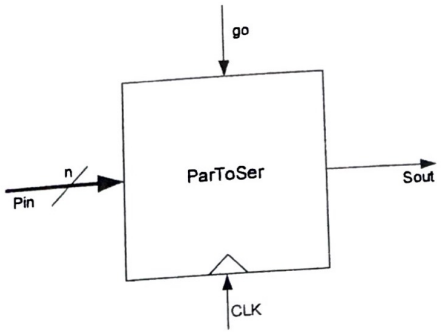


Figure 2: Parallel-to-Serial Transmitter

Q3) (20 points)

For the circuit shown in Figure 3

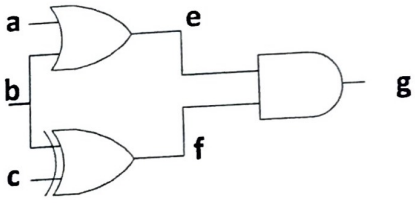


Figure 3: Q3 Circuit

a. Use the Boolean Difference Method to find when output g is sensitive to input b? Then find the test vectors for b sa0. (6 points)
b. Use D-Algorithm to find all test vectors for e sa0. (5 points)
c. Find all the test vectors for g sa1. (2 points)
d. Find all the test vector for g sa0. (2 points)
e. Find all the test vectors for f sa0. (2 points)
f. State all the faults that can be detected by Test Vector abc = 010. (3 points)

☺ GOOD LUCL ☺

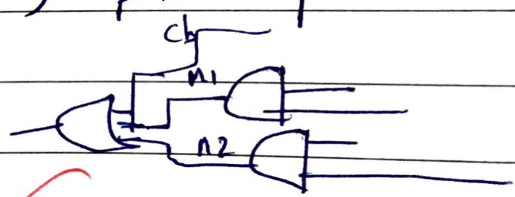
Q1)-

a)

```
Library ieee;
use ieee.std_logic_1164.All;
use ieee.std_logic_unsigned.All;

ENTITY bcd is
port ( A, B : IN std_logic_vector (3 downto 0);
       cin : IN std_logic;
       S : OUT std_logic_vector (4 downto 0));
End entity bcd;
```

b)

```
library ieee;
use ieee.std_logic_1164.All;
use ieee.std_logic_unsigned.All;

Architecture struct of bcd is
  signal r,d,s1: std_logic_vector (3 downto 0);
  signal cb,s4,n1,n2: std_logic;
Begin
  g1: entity work.adder4(special) port map
   (A, B, cin, r, cb);
   n1 <= r(3) and r(2);
   n2 <= r(3) and r(1);
   s4 <= cb or n1 or n2;
   d <= '0' & s4 & s4 & '0';
  g2: entity work.adder4(special) port map
   (r, d, '0', s1, open);

   S <= s4 & s1(3 downto 0);
End architecture struct;
```



10

- r is first add outp

Library ieee;
use ieee.std_logic_1164.ALL;
use ieee.std_logic_unsigned.ALL;

c) Architecture beh of bcd is
signal sumstd, sum2s: std_logic_vector(4 downto 0);
signal input2 : std_logic_vector(4 downto 0);
signal s4 : std_logic;
Begin
c1 <= conv_integer(cin);
A1 <= conv_integer(A);
A2 <= conv_integer(B);
sum1 <= A1 + A2 + c1;
n <= conv_INTEGER(sum1);
input2 <= "0000" when n < 9,
ELSE "0110" when n >= 9;
sum2 <= sum1(3 downto 0) + input2;
s4 <= '1' when n >= 9 or sum(4) = '1'
ELSE '0';
s <= s4 & sum2(3 downto 0);
END architecture beh;

signal A1, A2, c1, sum1, input, input2 : integer;
Begin
c1 <= conv_integer(cin);
A1 <= conv_integer(A);
A2 <= conv_integer(B)
sum1 <= A1 + A2 + c1;
sumstd <= conv_std_Logic_vector(sum
input <= 0 when sum <= 9;
else 6 when sum > 9;
sum2 <= sumstd(3 downto 0) +

sum1 := ('0' & A) + ('0' &B)
+ ("000" &c

std_logic

Why ??

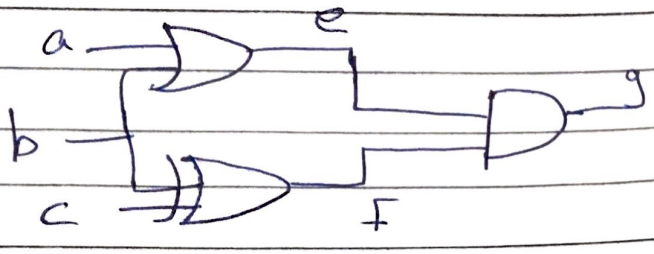
sumstd <= conv_std_logic_vector(sum1,5);
input2 <= 0 when sum1 <= 9,
else 6 when sum1 > 9;
sum2 <= input2 + input;
s4 <= '1' when sum1 > 9,
else '0';
sum2s <= conv_std_logic_vector(sum2,5);
s <= s4 & sum2s(3 downto 0);
end architecture beh;

input2 <= conv_integer(sumstd(3 downto 0));

Q2)

```
library ieee;
use ieee.std_logic_1164.ALL;
Entity partoser is
    (generic n : positive)
    port ( pins : std_Logic_vector(n-1 downto 0);
           go,clk: std_Logic; sout:out std_Logic);
end entity partoser;

Architecture beh of partoser is
    signal count:integer range 0 to n+2;
    signal savepins : std_Logic_vector(n-1 downto 0);

Begin

    if (rising_edge(clk)) then

        if (go = '1' and count=0 ) then
            sout <= '0';
            count <= count+1;
            savepins <= pins;
        elsif (g = '0' and count = 0) then
            sout = '1';
        end if;
    if (count >0) then -started moving
        if (count >0 and count <n+1 ) then
            sout <= savepins(count-1);
            count <= count+1;
        elsif (count = n+1) then
            sout <= '1';
            count = count+1;
        end if;
        if (count = n+2) then
            count <= 0;
        end if;
    end if;
end if;
end architecture beh;
```

Q3) -



a) $g = e.f = (a+b)(b \oplus c)$

$$\frac{dg}{db} = g(B=0) \oplus g(B=1)$$

$$= (a+0)(0 \oplus c) \oplus (a+1)(1 \oplus c)$$
$$= a.c \oplus 1.c'$$
$$= (ac)'c' + (ac)c$$
$$= (a'+c')c' + ac$$
$$= a'c' + c' + ac$$
$$= c'(a'+1) + ac$$
$$= c' + ac$$

a	c	a.c	c'	a.c ⊕ c'
0	0	0	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

6

g is sensitive to b when $\frac{dg}{db} = 1$

ac = 00 or 10 or 11

b sa at ϕ b=1

$\frac{dg}{db} = 1$

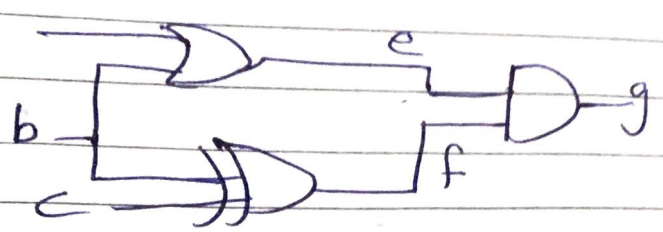
$b(c' + ac) = 1$

a	b	c
0	1	0
1	1	0
1	1	1

test vectors

b)

e sa ∅



but D on e

a e g path

back trace $a + b = D$

propagation

$b \oplus c = 1$

$g = D \Rightarrow f = 1$

a b

01 , 10 , 11

b c

01 , 10

$g = D \cdot 1$

$g = D$

if g=1 not faulty

if g=0 faulty

test vectors		
a	b	c
0	1	0
1	0	1
1	1	0

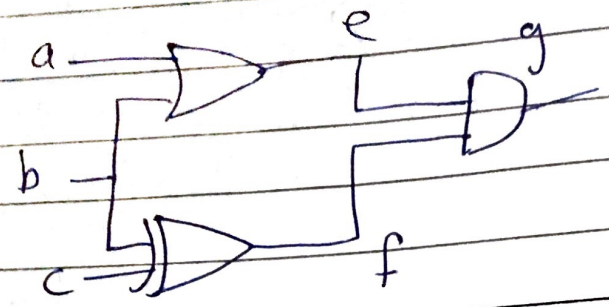
5

~~but D on e~~

~~f g path~~

~~a + b D~~

c)



~~using boolean difference~~
using fault model.

g sa 1 ⇒ g=0

back trace ef = 00 or 01 or 10

a	b	c
0	0	0

a	b	c
0	0	1

a	b	c
1	1	1
1	0	0
0	1	1

~~test~~ test vectors

a	b	c
0	0	0
0	0	1
1	0	0
0	1	1
1	1	1

if g = 0 ~~faulty~~ not faulty
∴ if 1 faulty

2

d) g sa ϕ

g = 1

back trace ef = 11

a	b	c
1	1	0
1	0	1
0	1	0

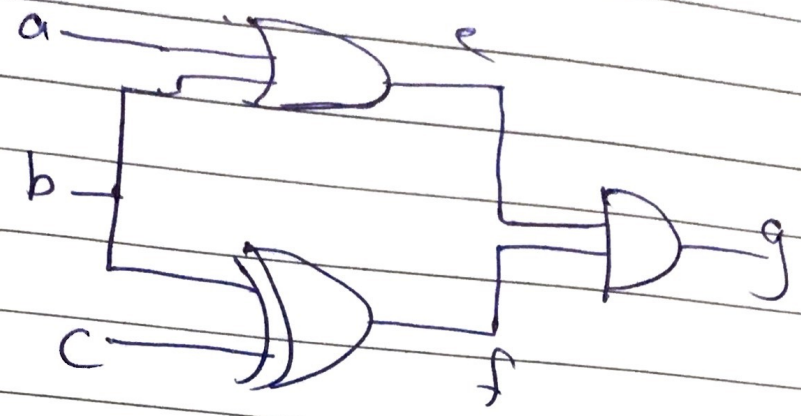
a	b	c
ϕ	0	0
ϕ	0	1
0	1	0
0	1	1
1	0	0
1	0	1

test vectors

a	b	c
1	1	0
1	0	1
0	1	0

2

f sa ϕ



f = 1

back trace $b \oplus c = 1$

b	c
0	1
1	0

propagation $a + b =$ 1

a	b
0	1
1	0
1	1

test vectors

a	b	c
0	1	0
1	0	1
1	1	0

f)

a b c
0 1 0 ⇒

e = 1
f = 1
g = 1

if no faults

① g sa ϕ — g=0
② e sa ϕ — g=0
③ f sa ϕ — g=0
④ c sa 1 — g=0
⑤ b sa ϕ — g=0