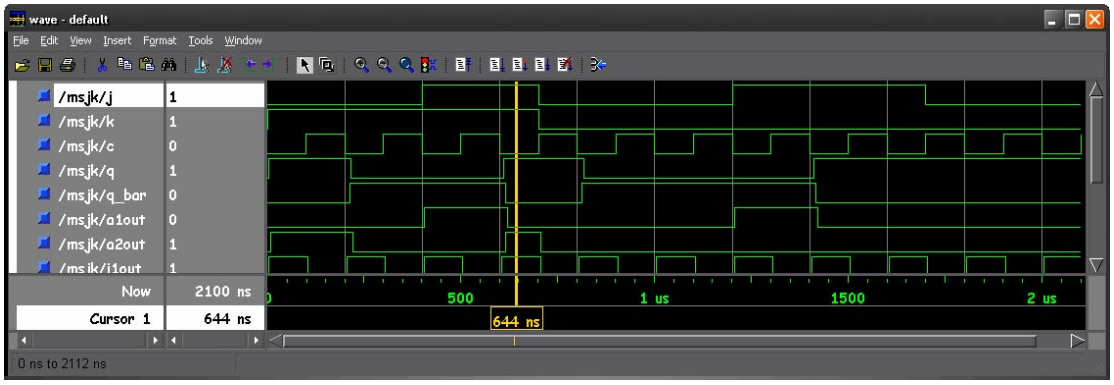
COE 405 **Dr. Aiman El-Maleh**

HW#4 Solution

Q 5.15

```
Package HW4 is
COMPONENT p1 PORT (in1: IN BIT; out1: BUFFER BIT); END COMPONENT ;
COMPONENT p2 PORT (in1,in2: IN BIT;out1: BUFFER BIT);END
COMPONENT;
component counter port (en, clk: in bit; q: buffer bit; co: buffer bit); end component;
component MSJK port (j,k, c: in bit; q : buffer bit ; q_bar: buffer bit); end component;
component myand2 port(in1,in2: in bit; out1: buffer bit); end component;
END HW4;
==========

package mygates is
component mynand2  port (in1, in2 : In BIT; out1 : BUFFER BIT); end component;
component myand2   port (in1, in2 : In BIT; out1 : BUFFER BIT); end component;
component myinv    port (in1  : In BIT; out1 : BUFFER BIT); end component;
END mygates;

==========

Entity mynand2 is
port ( in1, in2 : In Bit; out1 : BUFFER bit);
End mynand2;

Entity myand2 is
port ( in1, in2 : In Bit; out1 : BUFFER bit);
End myand2;

Entity myinv is
port ( in1 : In BIT; out1 : BUFFER BIT);
END myinv;

Architecture fast_single_delay OF mynand2 is
begin
        out1 <= in1 NAND in2 AFTER 3 NS;
END fast_single_delay;

Architecture single_delay OF myinv is
begin
        out1 <= NOT in1 AFTER 5 ns;
END single_delay;

Architecture single_delay OF myand2 is
begin
```

```
        out1 <= in1 AND in2 AFTER 5 NS;
END single_delay;

Architecture single_delay OF mynand2 is
begin
        out1 <= in1 NAND in2 AFTER 5 NS;
END single_delay;

entity MSJK is

   port (J    : in BIT;
         K    : in BIT;
         C  : in BIT;
         Q  : BUFFER BIT;
         Q_BAR : BUFFER BIT);
end MSJK;

Use work.mygates.all;
Use work.HW4.all;

Architecture Struct of MSJK is
        FOR ALL : p1 USE ENTITY WORK.myinv;
        FOR a1,a2 : p2 USE ENTITY WORK.myand2(single_delay);
        FOR n1,n2,n3,n5,n6,n7 : p2 USE ENTITY
WORK.mynand2(fast_single_delay);
        FOR n4,n8 : p2 USE ENTITY WORK.mynand2(single_delay);
        signal a1out, a2out, i1out, n1out, n2out, n3out, n4out, n5out, n6out : BIT;
        begin

        a1 : p2 port map ( Q_BAR, J, a1out);
        a2 : p2 port map ( K, Q, a2out);
        i1 : P1 port map ( C, i1out);
        n1 : p2 port map (a1out, C, n1out);
        n2 : p2 port map (C, a2out, n2out);
        n3 : p2 port map (n1out, n4out, n3out);
        n4 : p2 port map (n3out, n2out, n4out);
        n5 : p2 port map (n3out, i1out, n5out);
        n6 : p2 port map (n4out, i1out, n6out);
        n7 : p2 port map (n5out, Q_BAR, Q);
        n8 : p2 port map (Q, n6out, Q_BAR);
        end struct;
```



Q 5.17

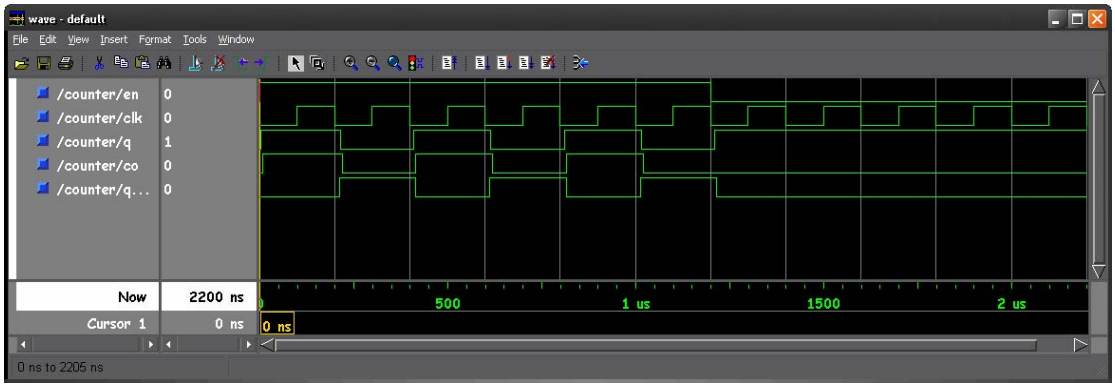
```
entity function_f is port (a, b, c, d, e: in bit ; f:out bit);
end function_f;
architecture configurable of function_f is
component n1 port (w: in bit; z: out bit); end component;
component n2 port (w, x: in bit; z: out bit); end component;
component n3 port (w, x, y: in bit; z: out bit); end component;
```

for all: n1 use entity work.nand3(single_delay) port map (w,w,w,z);
for all: n2 use entity work.nand3(single_delay) port map (w,w,x,z);
for all: n3 use entity work.nand3(single_delay) port map (w,x,y,z);

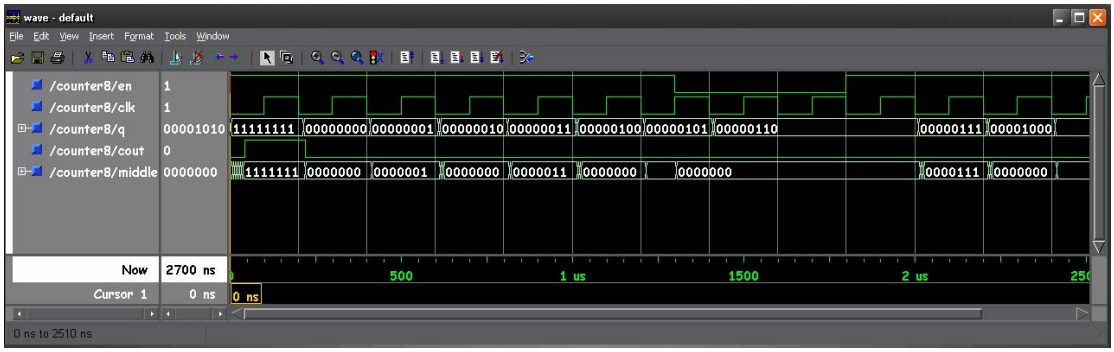
```
signal i1, i2, i3, i4: bit;
begin
g0: n1 port map (d, i1);
g1: n1 port map (e, i2);
g2: n2 port map (i1, c, i3);
g3: n2 port map (a, b, i4);
g4: n3 port map (i2, i3, i4, f);
end configurable;
```

Q 5.20

```
Package HW4 is
COMPONENT p1 PORT (in1: IN BIT; out1: BUFFER BIT); END COMPONENT ;
COMPONENT p2 PORT (in1,in2: IN BIT;out1: BUFFER BIT);END
COMPONENT;
component counter port (en, clk: in bit; q: buffer bit; co: buffer bit); end component;
component MSJK port (j,k, c: in bit; q : buffer bit ; q_bar: buffer bit); end component;
component myand2 port(in1,in2: in bit; out1: buffer bit); end component;
END HW4;
==========

entity counter is
port (en, clk: in bit; q: buffer bit; co: buffer bit);
end counter;

USE WORK.HW4.all;
architecture struct of counter is

FOR ALL : MSJK USE ENTITY WORK.MSJK;
FOR ALL : Myand2 USE ENTITY WORK.myand2(single_delay);
signal q_bar: bit;
begin
jk: MSJK port map (en, en, clk, q, q_bar);
anding: myand2 port map (q, en, co);
end struct;
```




```
entity counter8 is
port (en, clk: in bit; q : buffer bit_vector(7 downto 0); cout: buffer bit);
end counter8;

USE WORK.HW4.all;
architecture struct of counter8 is
For all : counter USE ENTITY WORK.counter;
signal middle : bit_vector(6 downto 0);
constant n: INTEGER := 8;

begin

ccount: for i in 0 to 7 generate
        one: if i = 0 generate
                first: counter port map (en, clk, q(i), middle(i));
        end generate;

        last: if i = n-1 generate
                most: counter port map(middle(i-1), clk, q(i), cout);
        end generate;

        mid: if i<n-1 and i>0 generate
                midl: counter port map(middle(i-1), clk, q(i), middle(i));
        end generate;

end generate;
end struct;
```


wave - default
File Edit View Insert Format Tools Window
/counter8/en
/counter8/clk
/counter8/q
/counter8/cout
/counter8/middle
1
1
00001010
0
0000000
11111111 00000000 00000001 00000010 00000011 00000100 00000101 00000110 00000111 00001000
1111111 0000000 0000001 0000000 0000011 0000000 0000000 0000111 0000000
Now
2700 ns
Cursor 1
0 ns
500
1 us
1500
2 us
0 ns
0 ns to 2510 ns