ECE 476 MIDTERM EXAM SOLUTION SET

23 APRIL 2010

(1) Two VHDL codes are given as

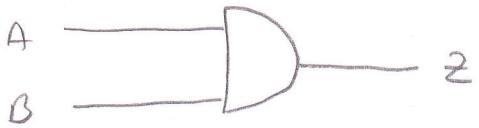
a)
```
SIGNAL Z, A, B, C, D : std_logic;
BEGIN
Z <= A AND B ;
Z <= C OR D ;
END Behavior;
```

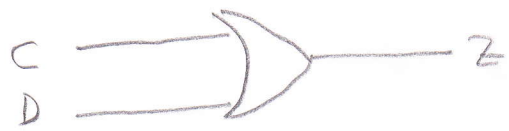
b)
```
SIGNAL Z, A, B, C, D : std_logic;
BEGIN
PROCESS (A, B, C, D)
BEGIN
Z <= A AND B ;
Z <= C OR D ;
END PROCESS
END Behavior;
```

Show the corresponding logic circuits to these VHDL codes

SOL:

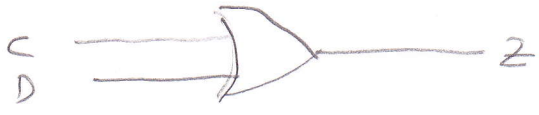
a)





(5p)

b)

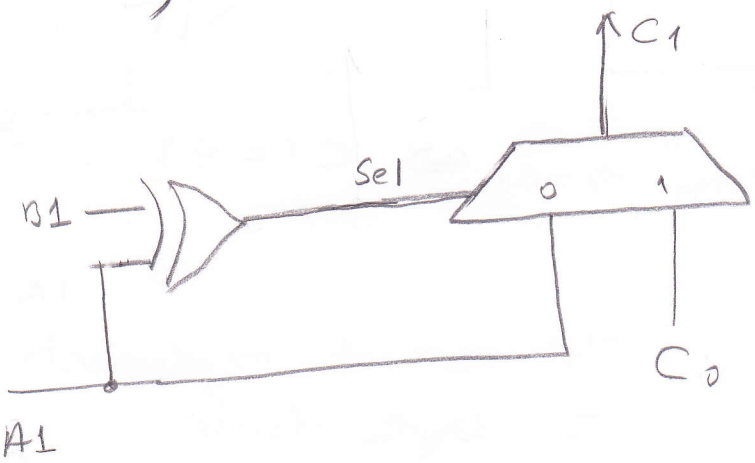


(5p)

② To prevent the ripple in the adders, a design can be made as shown below. The circuit immediately calculates the carry bit.

a) Explain how this circuit calculates the carry bit.

b) Write the VHDL code to implement this circuit (A1, B1, C0 inputs, C1 output) by using the 2-to-1 mux as a component (write the mux code as well by using a conditional statement)



c) Write the VHDL code of the circuit this time by using the 2-to-1 mux as a package (write the mux code as well by using a SELECT statement).

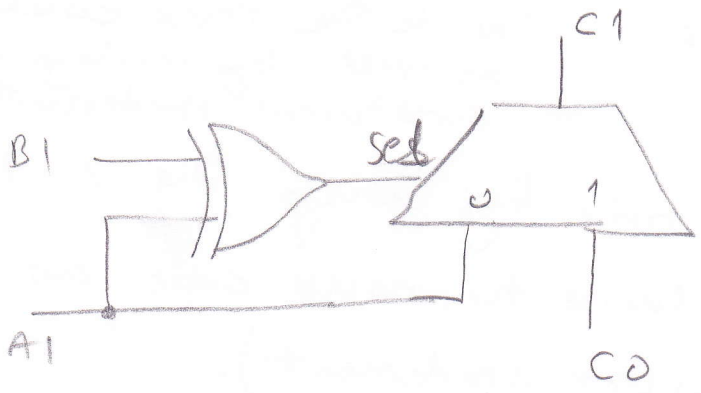
SOL:

a)

A1	B1	Sel	
0	0	0	Sel = 0 when A1 = B1
0	1	1	Sel = 1 when A1 ≠ B1
1	0	1	
1	1	0	Sel = 0 when A1 = B1

A1	B1	C0	C1	
0	0	0	0	When A1=B1 ⇒ C1 = A1
0	0	1	1	
0	1	0	0	When A1≠B1 ⇒ C1 = C0
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	1	When A1=B1 ⇒ C1 = A1
1	1	1	1	

The truth table



This circuit implements the truth table. When A1=B1 Then a 0 is generated at Sel input which selects the 0 input in the multiplexer. This selects A1 and C1 = A1.

When A1≠B1 the EX-OR generates a 1 at the output. Sel becomes 1. This selects input 1 in the mux and C1 = C0.

(10p)

b) First the mux as a component:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all
COMPONENT Mux2to1
```

```
ENTITY mux 2 to 1 IS
PORT ( W0, W1 : IN STD_LOGIC;
       S      : IN STD_LOGIC
       f      : OUT STD_LOGIC );
END mux2TO1 ;
ARCHITECTURE Behavior OF mux2TO1 IS
BEGIN
f <= W0 WHEN S='0' ELSE W1;
END Behavior;
END COMPONENT;
```

MAIN CODE for the CIRCUIT:

15P

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY CarryRipple IS
PORT (A1, B1, C0 : IN STD_LOGIC;
      C1         : OUT STD_LOGIC);
END CarryRipple;
ARCHITECTURE Behavior OF CarryRipple IS
COMPONENT MUX2 TO 1
PORT (W0, W1 : IN  STD_LOGIC;
      S      : IN  STD_LOGIC;
      f      : OUT STD_LOGIC);
SIGNAL SEL : STD_LOGIC;
BEGIN

SEL = A1 EXOR B1 ;
```

sel=0 Sel=1

```
MUX1: MUX2TO1 PORTMAP (A1, C0, SEL, C1) ;

END Behavior;
```

c) → First the VHDL code of MUX2TO1 as a COMPONENT with SELECT statement:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
COMPONENT MUX2TO1
    :
    :   (entity is the same as previous)
    :
ARCHITECTURE Behavior of MUX2TO1 IS
BEGIN
WITH S SELECT
f <= W0 WHEN 0 ;
f <= W1 WHEN OTHERS ;
END Behavior ;
```

→ Then we declare this component as Package:

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all;
PACKAGE MUX2TO1_PACKAGE IS
COMPONENT MUX2TO1
PORT ( W0,W1: IN STDD-LOGIC ;
       S : IN STD-LOGIC ;
       f ; OUT STD-LOGIC);
END COMPONENT ;
END MUX2TO1_PACKAGE;
```

(ASP)

→ USE the Package in the main code

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE work.mux2to1_package.all; ← to use Package
ENTITY CarryRipple IS
    :
```

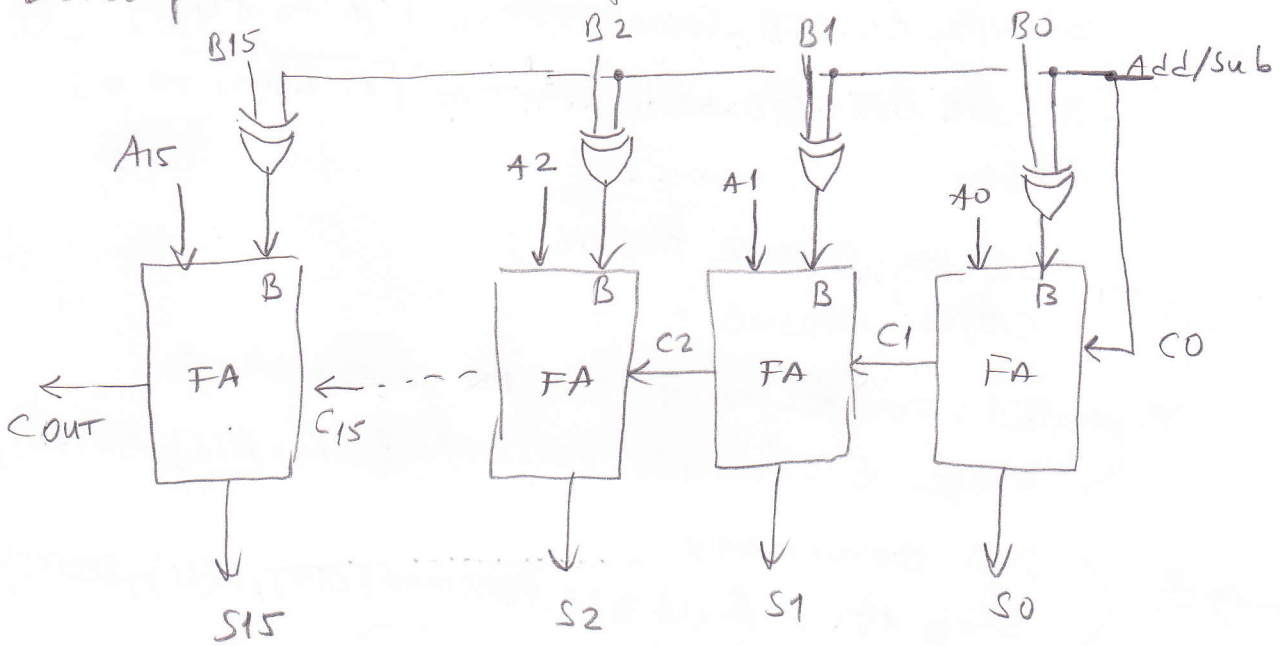
4/2

```
PORT ( A1, B1, C0 : IN  STD_LOGIC;
       C1         : OUT STD_LOGIC);

END CarryRipple;
ARCHITECTURE Behavior OF carryRipple IS
SIGNAL S2 : STD_LOGIC;
BEGIN
S2 = A1 EXOR B1;
MUX1 : MUX2TO1 PORTMAP (A1, C0, S2, C1);
END Behavior;
```

---

(3) Develop the structural VHDL code for the circuit given below



(Note: Use For-Generate Statement)
use FA as a package: full add-package)

SOL

B	Add/Sub	f
0	0	0
0	1	1
1	0	1
1	1	0

When Add/Sub = 0
$f = B$

When Add/Sub = 1
$f = \overline{B}$

This circuit either add or subtract B from A (in 2's complement)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
(*) USE work.fulladd_package.all;
```

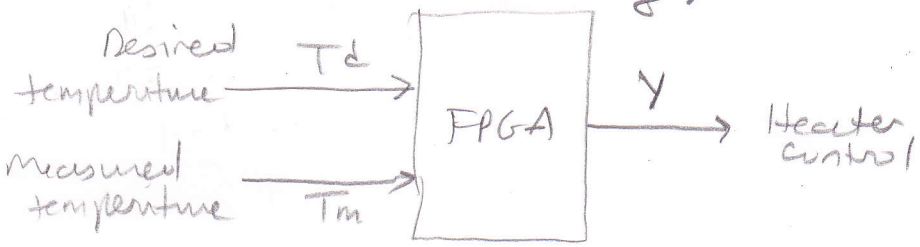
(* NOTE: It is not signed arithmetic. It is UNSIGNED. 2's complement is done by the circuit)

```
ENTITY ADDER16BIT IS
PORT ( A,B : IN  STD_LOGIC_VECTOR (15 DOWN TO 0);
       ADDSUB : IN  STD_LOGIC;
       COUT : OUT STD_LOGIC;
       S : OUT STD_LOGIC_VECTOR (15 DOWN TO 0));
END ADDER16BIT;
ARCHITECTURE structure OF ADDER16BIT IS
SIGNAL C: STD_LOGIC_VECTOR (0 TO 14);   (*)
SIGNAL BEX: STD_LOGIC_VECTOR (15 DOWN TO 0);
BEGIN
  BEX <= B EXOR ADDSUB;
  C(0) <= ADDSUB;
  G1: FOR i IN 0 TO 14 GENERATE
  Stage i : fulladd PORTMAP (C(i), A(i), BEX(i), S(i), C(i+1));
  END GENERATE;
  Stage 15 : full add PORTMAP (C(15), A(15), BEX(15), S(15), COUT);
END Structure;
```

(*) (*)

25p

(4) Consider an FPGA-operated heater shown below. This circuit is required to enable the ON/OFF control of the heater to adjust the temperature to a predefined temperature Td. The desired control is such that the desired (Td) and the measured (Tm) temperatures are converted into unsigned digital data of 4-bit and fed into the FPGA module. The FPGA outputs a control signal Y to control the heater such that the heater is turned on (Y=1) when the measured temperature is lower than the desired, and turned off (Y=0) otherwise. Provide the VHDL code (use Process Statement in the coding).



SOL:

(*)
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY Compare IS
PORT ( TD, TM : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
       Y : OUT STD_LOGIC);
END compare;
ARCHITECTURE Behavior OF compare IS
BEGIN
PROCESS (TD, TM)
IF TM<TD THEN
   Y<='1'
ELSE
   Y<='0'
END IF;
END PROCESS;
END Behavior;
```

25p