B122/B222 VHDL Tutorial Solutions

Please Try the Questions BEFORE looking at these solutions

(Error in Question 8 corrected - reset now active low.... 16th May 2002)

Question 1

```
entity Q1 is
   Port ( K : in bit;
          L : in bit;
          M : in bit;
          T : out bit);
end Q1;

architecture Q1A of Q1 is
begin
T <=    '0' when K='0' and L='0' and M='0' else
          '0' when K='0' and L='1' and M='1' else
          '0' when K='1' and L='0' and M='1' else
          '0' when K='1' and L='1' and M='0' else
          '1';
end Q1A;
```

Question 2

```
Entity Q2 is
port (A, B, C, D : in bit; R, S : out bit);
end Q2;
architecture Q2A of Q2 is
begin
R <= ( A or B ) and ( C or D );          -- use STANDARD functions 'or' & 'and'
S <= ( A or C ) and ( B or D );
end Q2A;
```

Question 3

```
Entity Q3 is
port (A, B : in bit; sum, carry : out bit);
end Q3;
architecture Q3A of Q3 is
begin
sum  <= ( A xor B );                     -- use STANDARD functions 'or' & 'and'
carry <= ( A and B );
end Q3A;
```

Question 4

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity Q4 is
   Port ( V : in std_logic_vector(3 downto 0);
          P : out std_logic);
end Q4;

architecture Q4A of Q4 is
begin
-- assume when V has equal number of ones & zeros
-- then P=0 therefore....
P <=    '1' when V="0111" else          -- only include cases where P= '1'
        '1' when V="1011" else
        '1' when V="1101" else
        '1' when V="1110" else
        '1' when V="1111" else          -- (alternative is to list when P='0'
        '0';                            -- the default else would be '1')
end Q4A;
```

Question 5

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity Q5 is
   Port ( clk : in std_logic;
          rst : in std_logic;
          Q : out std_logic_vector(4 downto 0));  -- width of 5
end Q5;

architecture Q5A of Q5 is
signal count : std_logic_vector(4 downto 0);
begin
process (clk,rst)
begin
        if rst='1' then   -- active high reset
                count <= "00000";
        elsif clk'event and clk='1' then          -- +ve clock edge
                count <= count + "00001";     -- count up
        end if;
end process;
Q <= count;     -- output latest count
end Q5A;
```

Question 6

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity Q6 is
    Port ( clk : in std_logic;
           rst : in std_logic;
           Q : out std_logic_vector(4 downto 0));
end Q6;

architecture Q6A of Q6 is
signal count : std_logic_vector(4 downto 0);
begin
process (clk,rst)
begin
           if rst='1' then   -- active high reset
                     count <= "11111";
           elsif clk'event and clk='1' then          -- +ve clock edge
                     count <= count - "00001";      -- count down
           end if;
end process;
Q <= count;     -- output latest count
end Q6A;

-- the count is reset to a value of all ones (Hex 1F) so that when it counts down
-- the values would go "11111", "11110", "11101"..... etc.
```

Question 7

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity Q7 is
    Port ( clk : in std_logic;
           rst : in std_logic;
           Q : out std_logic_vector(3 downto 0));
end Q7;

architecture Q7A of Q7 is
signal count : std_logic_vector(3 downto 0);
begin
process (clk,rst)
begin
          if rst='1' then                          -- active high reset
                    count <= "0111";               -- Hex 7
          elsif clk'event then                     -- both clock edges
                    count <= count + "0001";       -- count up
          end if;
end process;
Q <= count;     -- output latest count
end Q7A;
```

Question 8

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Q8 is
    Port ( clk : in std_logic;
           rst : in std_logic;
           Q : out std_logic_vector(3 downto 0));
end Q8;

architecture Q8A of Q8 is
signal count : std_logic_vector(3 downto 0);
begin
process (clk,rst)
begin
          if rst='0' then   -- active low reset
                    count <= "0000";
          elsif clk'event and clk='0'then          -- -ve clock edge
                    if count = "1001" then         -- test for largest allowed value
                              count <= "0000";     -- roll over
                    else
                              count <= count + "0001";      -- count up
                    end if;
          end if;
end process;
Q <= count;    -- output latest count
end Q8A;


-- the solution above is correct, but in the hardware there is a remote chance
-- that a glitch could set the counter to above 9 (ie. between 10 to 15)
-- the solution above would count up to 15 before then the roll over
-- would take it back into range. This could be avoided by the roll over
-- test being replaced with.....
                    if count >="1001" then         -- test if last value at or above 9
                              count<="0000";       -- roll over
```

Question 9

This question combines most of the concepts seen in the VHDL lectures... it has

- **a process to handle the counting, and**
- **three other concurrent statements to set the output values.**

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity Q9 is
    Port ( clk, rst : in std_logic;
           LOW, MID, HIGH : out std_logic);
end Q9;

architecture Q9A of Q9 is
signal count : std_logic_vector(3 downto 0);
begin
process (clk,rst)
begin
        if rst='1' then   -- active high reset
                count <= "0000";
        elsif clk'event and clk='1'then          -- +ve clock edge
                if count = "1010" then
                        count <= "0000";       -- roll over
                else
                        count <= count + "0001";      -- count up
                end if;
        end if;
end process;
LOW   <= '1' when count < "0110" else '0';
MID   <= '1' when count = "0110" else '0';
HIGH  <= '1' when count > "0110" else '0';
end Q9A;
```