Name: Key **Id#**

COE 405, Term 021

Design & Modeling of Digital Systems

Quiz# 5

Date: Saturday, Jan. 4, 2003

Q.1. Rewrite the ***bin2int*** procedure such that it functions properly for any size input, declared with any range and in any direction. Treat the left bit of the input as the most significant bit, and the right-most bit as the least significant. Bit 3 is the MSB for a BIT_VECTOR ranging from 3 TO10 associated with the input of this procedure and the LSB is bit 10.

```
Procedure Bin2Int (Bin : IN BIT_VECTOR; Int: OUT Integer) IS
Variable Result, Index: Integer;
Begin
Result := 0;
  index := Bin'length - 1;
  For I IN Bin'Range Loop
        If Bin(I) = '1' Then
              Result := Result + 2**index;
        End If;
        index := index - 1;
        End Loop;
Int := Result;
End Bin2Int;
```

OR

```
Procedure Bin2Int (Bin : IN BIT_VECTOR; Int: OUT Integer) IS
Variable Result: Integer;
Variable temp: bit_vector(Bin'length-1 downto 0);
Begin
temp := bin;
Result := 0;
  For I IN temp'Range Loop
        If temp(I) = '1' Then
              Result := Result + 2**I;
        End If;
        End Loop;
Int := Result;
End Bin2Int;
```

Q.2. Use an array of BITS as shown below to model master-slave JK flip-flop.
ARRAY (BIT, BIT, BIT) OF BIT
Use J, K, and Q values for the indices of this array, and let the array represent the next state of the flip-flop. In the declarative part of the architecture of the flip-flop declare a constant (for example jk-table) of the type of the array shown above. Initialize this constant to appropriate next values of a JK flip-flop. In the statement part of the architecture of the flip-flop look up next Q values by indexing the JK table using J, K, and present Q values.

```
Entity JKF is
port ( clk, j, k: IN bit; q: buffer bit);
end JKF;
Architecture q2 of JKF is
Type bit3 is ARRAY (BIT, BIT, BIT) OF BIT;
constant jk_table: bit3 :=
-- j k q
(
(('0' ,  -- '0', '0', '0'
'1') ,   -- '0', '0', '1'
('0' ,   -- '0', '1', '0'
'0')) ,  -- '0', '1', '1'
(('1' ,  -- '1', '0', '0'
'1') ,   -- '1', '0', '1'
('1' ,   -- '1', '1', '0'
'0')) ); -- '1', '1', '1'
begin
process(clk)
begin
if (clk = '1') then
  q <= jk_table(j,k,q);
end if;
end process;
end q2;
```