DOĞU AKDENİZ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BLGM223 SAYISAL MANTIK TASARIMI

DENEY VI : QUARTUS II TASARIM ORTAMINDA DEKODER ve MULTİPLEXER ile DEVRE KURULUMU

AMAÇLAR:

QUARTUS II tasarım ortamında N-bit dekoder ve Multiplekser tasarımı ve bu tasarımlarla Bool işlevlerinin gerçekleştirilmesine yönelik uygulamalar yapacağız.

Deney Öncesi Çalışma:

Ders notlarından dekoder ve multiplekser konularını tekrar çalışınız.

1. Aşağıda size 3x8 Dekoder için (a,b,c giriş sinyalleri, d0-d7 çıkış sinyalleridir) VeriLOG HDL kodu verilmiştir. Bu kodu Quartus ortamına kopyalayarak çalıştırınız ve simülasyonunu yapınız.

```
module Decoders(a,b,c,d0,d1,d2,d3,d4,d5,d6,d7);
  input a,b,c;
  output d0,d1,d2,d3,d4,d5,d6,d7;
  assign d0=(~a&~b&~c),
         d1=(~a&~b&c),
         d2=(~a&b&~c),
         d3=(~a&b&c),
         d4=(a&~b&~c),
         d5=(a&~b&c),
         d6=(a&b&~c),
         d7=(a&b&c);
endmodule
```

2. Yukarıda verilen Dekoder ile gerçeklenen $F(A,B,C)=\Sigma(0,2,5,7)$ işlevi için yazılan VeriLog HDL kodu aşağıda verilmiştir.

```
module Decoders(a,b,c,F);
        input a,b,c;
        output F;
        wire d0,d1,d2,d3,d4,d5,d6,d7;

        assign
                d0=(~a&~b&~c),
        d1=(~a&~b&c),
        d2=(~a&b&~c),
        d3=(~a&b&c),
        d4=(a&~b&~c),
        d5=(a&~b&c),
        d6=(a&b&~c),
        d7=(a&b&c);

        assign
        F=d0|d2|d5|d7;
endmodule
```

Bu kodu Quartus ortamına kopyalayarak çalıştırınız ve simülayonunu yapınız.

Yukarıda verilen tasarımları dikkate alarak:

A. Bir adet 4x16 Dekoder tasarlayınız.
B. Bu dekoderi kullanarak F(A,B,C,D)= Σ(0,4,6,7,8,14,15) Bool işkevini gerçekleyiniz ve Quartus ortamında testlerini yapınız.

3. Aşağıda size 8x1 Multiplekser için (Sel: kontrol giriş vektörünü, a,b,c giriş sinyalleri, d0-d7 çıkış sinyalleridir) VeriLOG HDL kodu verilmiştir. Bu kodu Quartus ortamına kopyalayarak çalıştırınız ve simülayonunu yapınız.

```
module Multiplexers(
   input [2:0] Sel,                // Sel = 3 elemanlı kontrol sinyalleri vektörü
   input [7:0] Inp,                // Inp = 8 elemanlı giriş sinyalleri vektörü
   output reg Out);                // Out = Multiplexer çıkış sinyali

always @(Sel or Inp)
  begin
    case (Sel)
      2'b000: Out <= Inp[0];
      2'b001: Out <= Inp[1];
      2'b010: Out <= Inp[2];
      2'b011: Out <= Inp[3];
       2'b100: Out <= Inp[4];
      2'b101: Out <= Inp[5];
      2'b110: Out <= Inp[6];
      2'b111: Out <= Inp[7];
    endcase
  end
endmodule
```

Bu kodu Quartus ortamına kopyalayarak çalıştırınız ve simülasyonunu yapınız.

4. Yukarıdaki kodu değiştirerek bir adet 4x1 MUX tasarlayınız.

5. F(A,B,C,D)= Σ (1,2,4,5,6,8,9,11)+d(0,7,15) işlevininin devresini bir 4x1 MUX ile kurmak istiyoruz ve A,C,D giriş sinyallerini de MUX kontroş girişlerine bağladığımızı varsayıyoruz. Bu devreyi kağıt üzerinde kurunuz.

6. Aşağıda 4x1 MUX ile gerçeklenmiş 4 değişkenli bir Bool işlevinin VeriLog kodu verilmiştir. Beşinci adımda yaptığınız tasarımı bu kodu dğiştirerek gerçekleyiniz ve simülasyonunu yapınız.

```
module Multiplexers(
input [2:0] Sel,                    // Sel = 3 elemanlı kontrol sinyalleri vektörü
input A,B,C,D,
output reg Out);

wire [3:0] MuxInp;

assign
MuxInp[0]=A,
MuxInp[1]=(A|B),
MuxInp[2]=A&B,
MuxInp[3]=~A&~B;

always @(Sel or A or B or C or D)
  begin
    case (Sel)
      2'b00: Out <= MuxInp[0];
      2'b01: Out <= MuxInp[1];
      2'b10: Out <= MuxInp[2];
      2'b11: Out <= MuxInp[3];
    endcase
  end
endmodule
```

BAŞARILAR DİLERİM.

Doç. Dr. Adnan ACAN