**DOĞU AKDENİZ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ DEPARTMANI**

**BLGM 224 SAYISAL MANTIK SİSTEMLERİ**

**DENEYSEL ÇALIŞMA #4**

**VeriLog HDL’de SAYAÇLAR**

**Hedefler:**

Bu laboratuvar çalışmasında, Sayaçların tasarımı üzerine mimari ve davranışsal tanımlardan deneysel bir çalışma yapılması amaçlanmaktadır. Mimari tanım, geleneksel tasarım prosedürü ile tasarlanmış devrelerin hem şematik hem de yazılım uygulamasını kapsar. Davranış açıklamaları, durum geçiş diyagramlarını kullanarak uygulamayı kapsar.

**Önemli Not:** Proje dosyalarında bileşenlerin çoklu kullanımı nedeniyle meydana gelebilecek derleme hatalarını önlemek için, aşağıdaki deneysel görevlerin her biri için (her bir aşamada), yeni bir proje açılması gerekmektedir.

**Aşama 1: Şema**

Aşağıdaki tabloda belirtilen S1 ve S0 girişleri tarafından kontrol edilen aşağıdaki çok fonksiyonlu sayacı tasarlamak istediğimizi varsayalım:

|  |  |
| --- | --- |
| **Mode Control** | **Yazmaç İşlemi** |
| ***S1*** | ***S0*** |
| 0 | 0 | *İleri Sayım* |
| 0 | 1 | *Geri Sayım* |
| 1 | 0 | *Paralel Yükleme* |
| 1 | 1 | *Değişiklik yok* |

Bu çok işlevli sayacın devre şeması aşağıda verilmiştir.

* 1. Bu çok işlevli sayacın devresini çizin, VeriLog HDL ortamında derleyin ve simüle edin ve çalışma modunu doğrulayın. Farklı fonksiyon modlarını kolayca gözlemlemek için uygun dalga formlarını ayarlayın ve uygulayın.



**Aşama 2: Verilog HDL'de Çok İşlevli Sayacın Mimari Tasarımının Uygulanması**

Aşağıdaki çok işlevli sayaç tasarımının mimari VeriLog kodunu Quartus Lite programına girin. Doğruluğunu görmek için kodunuzu derleyin ve simüle edin.

/\* S1 ve S0 kontrol girişleri tarafından kontrol edilen çok işlevli sayaç:

 S1 S0 İşlem Modu

 0 0 İleri Sayım

 0 1 Geri Sayım

 1 0 Parallel Yükleme

 1 1 Değişiklik Yok

\*/

module MultiFuncCounter\_Arch (CLK,Clear,S,PL,Q);

 input CLK, Clear;

 input [1:0] S;

 input [3:0] PL; // Paralel Load

 output reg [3:0]Q; // Sayaç Çıktısı

 wire [3:0] W;

 wire [3:0] QT;

 MUX\_4\_1 m1(W[0],S[1],S[0],I31,I21,I11,I01);

 MUX\_4\_1 m2(W[1],S[1],S[0],I32,I22,I12,I02);

 MUX\_4\_1 m3(W[2],S[1],S[0],I33,I23,I13,I03);

 MUX\_4\_1 m4(W[3],S[1],S[0],I34,I24,I14,I04);

 T\_FF t1(QT[0],W[0],CLK,Clear);

 T\_FF t2(QT[1],W[1],CLK,Clear);

 T\_FF t3(QT[2],W[2],CLK,Clear);

 T\_FF t4(QT[3],W[3],CLK,Clear);

 assign

 I31=1'b0,

 I21=QT[0]^PL[0],

 I11=1'b1,

 I01=1'b1,

 I32=1'b0,

 I22=QT[1]^PL[1],

 I12=QT[0],

 I02=~QT[0],

 I33=1'b0,

 I23=QT[2]^PL[2],

 I13=QT[1]&QT[0],

 I03=~QT[1] & ~QT[0],

 I34=1'b0,

 I24=QT[3]^PL[3],

 I14=QT[2]&QT[1]&QT[0],

 I04=~QT[2] & ~QT[1] & ~QT[0];

 always

 Q <= QT;

endmodule

module MUX\_4\_1(Y,S1,S0,I3,I2,I1,I0);

 input S1,S0,I3,I2,I1,I0;

 output reg Y;

 always @(S1,S0,I3,I2,I1,I0)

 begin

 if (S1==0 & S0==0)

 Y=I0;

 else if (S1==0 & S0==1)

 Y=I1;

 else if (S1==1 & S0==0)

 Y=I2;

 else if (S1==1 & S0==1)

 Y=I3;

 end

endmodule

module T\_FF(QT,T,CLK,CLR);

 input T,CLK,CLR;

 output reg QT;

 always @(posedge CLK)

 if (CLR == 1'b1)

 QT<= 1'b0;

 else

 QT<= T^QT;

endmodule

**ÖDEV #4 : (Çok İşlevli Sayacın Davranışsal Tasarımı)**

Yukarıda tarif edilen çok işlevli sayacın davranışsal VeriLog kodu aşağıda verilmiştir:

/\* VeriLog HDL'deki çok işlevli bir sayacın davranışsal açıklaması

 s1 s0 =00 İleri Sayım

 S1 S0 =01 Geri Sayım

 S1 S0 =10 Paralel Yükleme

 S1 S0 =11 Değişiklik Yok

\*/

module MultiFunctCounter\_Behav(CLK,Clear,S,PL,Q);

 input CLK, Clear;

 input [1:0] S;

 input [3:0] PL; // Parallel load

 output reg [3:0]Q; // Counter outputs

 reg[3:0] QT;

always @(posedge CLK)

begin

 if (Clear == 1)

 QT <= 4'b0000;

 else if (S[1]==0 & S[0]==0) // Count up

 QT <= QT+1;

 else if (S[1]==0 & S[0]==1) // Count down

 QT <= QT-1;

 else if (S[1]==1 & S[0]==0) // Parallel load

 begin

 QT[3] <= PL[3]; QT[2] <= PL[2];

 QT[1] <= PL[1]; QT[0] <= PL[0];

 end

 else if (S[1]==1 & S[0]==1)

 begin

 QT=QT;

 end

end

always

 Q <= QT;

Endmodule

Yukarıda verilen davranış kodunu modifiye ederek aşağıda davranışları belirtilen 4 bitlik çok işlevli bir sayaç tasarlayın:

**Enable S1 S0 İşlem Modu**

*0 x x Değişiklik Yok*

*1 0 0 Tümleyen*

*1 0 1 İki ileri sayım*

*1 1 0 İki geri sayım*

*1 1 1 Sola kaydır*

Ödev beşindi laboratuvar çalışmasının başında teslim edilmelidir.

*Hazırlayan: Doç. Dr. Adnan ACAN*