**DOĞU AKDENİZ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ DEPARTMANI**

**BLGM 224 SAYISAL MANTIK SİSTEMLERİ**

**DENEYSEL ÇALIŞMA #3**

**VeriLog HDL’de REGISTERLAR**

**HEDEFLER:**

Bu laboratuar çalışması, registerların tasarımı üzerine mimari ve davranışsal tanımlardan pratik bir çalışma getirmeyi amaçlamaktadır. Mimari tanım, geleneksel tasarım prosedürü ile tasarlanmış devrelerin hem şematik hem de yazılım uygulamasını kapsar. Davranıssal açıklamaları, durum geçiş diyagramlarını kullanan uygulamaları kapsar.

**Önemli Not:** *Aşağıdaki deneysel görevlerin her biri için (her bir aşamada), aynı proje dosyalarında bileşenlerin çoklu kullanımı nedeniyle derleme hatalarını önlemek için yeni bir proje açın.*

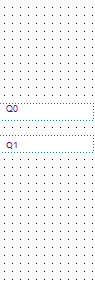
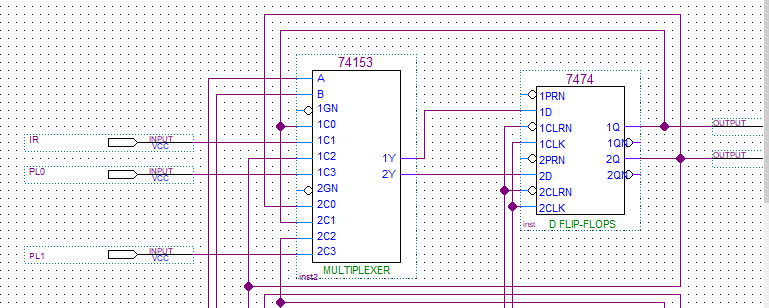
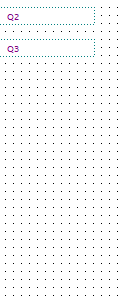
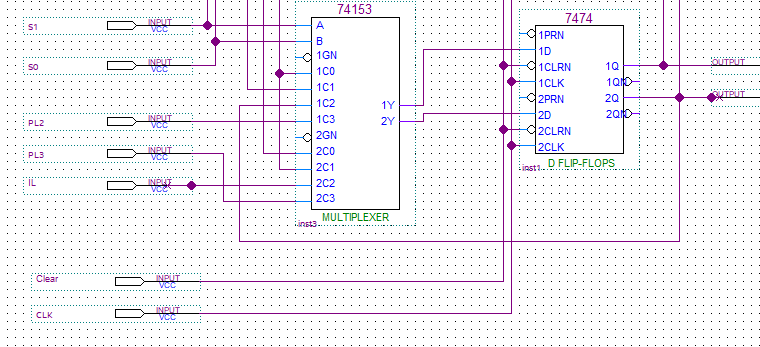
**Aşama 1: Şema**

Aşağıdaki S1 ve S0 kontrol girişleri tarafından kontrol edilen aşağıdaki çok fonksiyonlu registerı tasarlamak istediğimizi varsayalım:

|  |  |  |
| --- | --- | --- |
| **Mod Kontrolü** | | **Register**  **İşlemleri** |
| ***S1*** | ***S0*** |
| 0 | 0 | *Değişiklik yok* |
| 0 | 1 | *Sola kaydır* |
| 1 | 0 | *Sağa kaydır* |
| 1 | 1 | *Paralel Yükleme* |

Bu çok fonksiyonlu registera karşılık gelen şematik devre aşağıda verilmiştir:

1.1. Bu çok işlevli registerın devresini çizin, VeriLog HDL ortamında derleyin ve simülasyonunu gerçekleştirin ve çalışma modunu doğrulayın. Farklı fonksiyon modlarını kolayca gözlemlemek için uygunın dalga formlarını ayarlayın ve sonuçları gözlemleyin.



**Aşama 2:** **Verilog HDL'de Çok Fonksiyonlu Registerların** **Mimari Tasarımının Uygulanması**

Aşağıdaki çok fonksiyonlu register tasarımına ait mimari VeriLog kodunu Quartus Lite geliştirme paketine girin. Tasarımı doğrulamak için kodunuzu derleyin ve simüle edin. Sonuçları gözlemleyin.

/\* S1 ve S0 iki kontrol girişi tarafından kontrol edilen 4-bitlik Çok Fonksiyonlu Register aşağıdaki gibidir:

s1 s0=00 *Değişiklik yok*

01 *Sola kaydır*

10 *Sağa kaydır*

11 *Paralel Yükle*

\*/

module MultiFuncRegister\_Arch(Clear,CLK,S,PL,IL,IR,Q);

input CLK;

input Clear;

input IL, IR; // Serial load from left and right

input [1:0] S; // Vector of control inputs S1 and S0

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

output [3:0] Q; // Register ouputs Q3, Q2, Q1, Q0

wire [3:0] W; // Internal signals among components

MUX\_4\_1 m1(W[0],S[1],S[0],PL[0],IL,Q[1],Q[0]);

MUX\_4\_1 m2(W[1],S[1],S[0],PL[1],Q[0],Q[2],Q[1]);

MUX\_4\_1 m3(W[2],S[1],S[0],PL[2],Q[1],Q[3],Q[2]);

MUX\_4\_1 m4(W[3],S[1],S[0],PL[3],Q[2],IR,Q[3]);

D\_FF d1(Q[0],W[0],CLK,Clear);

D\_FF d2(Q[1],W[1],CLK,Clear);

D\_FF d3(Q[2],W[2],CLK,Clear);

D\_FF d4(Q[3],W[3],CLK,Clear);

endmodule

module D\_FF(Q,D,CLK,CLR);

input D,CLK,CLR;

output reg Q;

always @(posedge CLK)

if (CLR == 1'b1)

Q<= 1'b0;

else

Q<= D;

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

**ÖDEV # 3: (Çok Fonksiyonlu Bir Registerın Davranışsal Açıklaması)**

Yukarıda tarif edilen çok fonksiyonlu registerın davranışsal VeriLog kodu aşağıda verilmiştir:

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

s1 s0 = 00 Degişiklik yok

S1 S0 = 01 Sola kaydır

S1 S0 = 10 Sağa kaydır

S1 S0 = 11 Paralel yükle

\* /

module MultiFunctRegister\_Behav(Clear,CLK,S,PL,IL,IR,Q);

input Clear, CLK;

input [3:0] PL;

input [1:0] S;

input IL,IR;

output [3:0] Q;

reg [3:0] R;

always @(posedge CLK)

begin

if (Clear == 1)

R <= 4'b0000;

else if (S[1]==0 & S[0]==0) // No change

R <= R;

else if (S[1]==0 & S[0]==1) // Shift left

begin

R[0] <= IR; R[1] <= R[0];

R[2] <= R[1]; R[3] <= R[2];

end

else if (S[1]==1 & S[0]==0) // Shift right

begin

R[3] <= IL; R[2] <= R[3];

R[1] <= R[2]; R[0] <= R[1];

end

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

begin

R=PL;

end

end

assign Q = R;

endmodule

Aşağıdaki şekilde çalışan 4 bitlik çok fonksiyonlu bir register tasarlamak için yukarıda verilen davranış kodunu değiştirin:

**Etkinlik S1 S0 İşlem Modu**

*0 x x Degişiklik yok*

*1 0 0 Sola döndür*

*1 0 1 İçerigini XOR’la (0101 )*

*1 1 0 Sağa döndür*

*1 1 1 Paralele yükle*

Ödeviniz bir sonraki laboratuarda teslim edilmelidir.

*Doç.Dr. Adnan ACAN tarafından hazırlanmıştır.*