**DOĞU AKDENİZ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ DEPARTMANI**

**BLGM 224 SAYISAL MANTIK SİSTEMLERİ**

**Hedefler:**

Bu labaratuvar kapsamında EDA (Electronic Design Automation) ismi verilen elektronik çizim ve tasarım program olan Quartus programını kullanmayı ve mantıksal devrelerimizi gerçekleştirmeyi göreceğiz. Quartus program profesyonel dünyada çok büyük bir kullanım alanına sahiptir ve modern bir çok cihazın yapımında kullanılmaktadır.

**Aşama 1:**

Quartus programının tanıtılması.

**Adım 1. Proje oluşturma:** Quartus programında yapacağınız her yeni dizayn bir proje olmak zorundadır. Bu projeyi oluşturmak için program açıldıktan sonra:

[File] -> [New Project Wizard]

Butonlarına tıklayarak yeni bir proje yaratmak için gerekli olan “oluşturucuyu” açın.



Bu aşamadan sonra karşınıza proje oluşturucu ekranı gelecektir.



Bu ekran proje oluşturucusunun giriş ekranıdır. Burada Projenizin hangi aşamalardan oluştuğunu ve neleri düzenleyebileceğinizi göreceksiniz.

Burayı [Next >] butonuna tıklayarak geçiyoruz.

Gelen yeni ekranda bize projenin hangi dizine oluşturulacağı ve isminin ne olacağı ile alakalı bilgileri girmemizi isteyen giriş bölümleri göreceksiniz. Buradaki bilgileri uygun şekilde doldurunuz. (Quartus program için proje isimleri çok önemlidir.)



Proje tipi ekranı size projenizin hangi donanıma göre derleneceği ve uygulanacağı fazladan ayar seçenekleri sunar. Biz projelerimizi program üzerinde simule edeceğimiz ve herhangi bir canlı donanım üzerinde koşturmayacağımız için burada [Empty Project] seçeneğini işaretleyeceğiz ve ardından [Finish] butonuna tıklayarak projemizi oluşturmuş olacağız.



Projeniz oluşturulduktan sonra buna benzer bir ekranla karşılacaksınız:



**Adım 2. Basit şematik dizayn:**

Başarılı şekilde projenizi oluşturduktan sonra basit bir tasarım ve simulasyon işlemi yapabiliriz. Bu aşamada aşağıda göreceğiniz dizaynı örnek olarak kullanacağız.



1. Sıradaki adım boş bir Block Diagram oluşuturmak. Bunun için [File] -> [New] yolunu izleyerek açılan pop-up menu üzerinde **Block Diagram / Schematic File** seçeneğini seçmek ve [OK] butonuna tıklamaktır. Bu aşamada karşımıza boş bir tasarım penceresi açılacaktır.
2. Açılan boş çalışma sayfasına çift tıkladığınız zaman karşınıza Quartus programında dahili olarak gelen kütüphaneleri gösteren bir ekran gelecektir. Bu ekran üzerinde **Primitives (veya advanced)** kütüphanelerin altında **Logic** kapılar kütüphaneleri göreceksiniz. Bunu seçtikten sonra karşınıza tasarımlarınızda kullabileceğiniz mantıksal kapılar gelecektir.
3. Burada örnek devremizde görüldüğü gibi 1 adet 2 girişli and (**and2**), 1 adet 3girişli nand (**nand3**), 2 adet 2 girişli or (**or2**), 2 adet **JK-FF**, 2 adet **input** ve son olarak 1 adet **output** bileşenini projemize dahil edeceğiz.

**Elemanları birbirine bağlmak:**

Quartus şematik sayfasına eklediğimiz elemanların yerşelimlerini fare ile sürükleyerek yada elemanı seçtikten sonra klavyemizde bulunan ok tuşları ile hareket ettirerek düzenliyoruz.

Bu aşamanın ardından devre elemanlarını birbirine bağlamak için fare imlecini devre elemanlarının giriş yada çıkış pinlerinin üzerine getirdiğimiz zaman bize bağlantı ile alakalı sembole dönüşerek bağlantı yapılmaya hazır olduğu bilgisini verecektir. Farenin sol tuşuna basarak bağlantı kablosunu / telini bağlanacak diğer parçaya doğru çektip sürüklüyoruz. Bağlantının yapılacağı giriş bölümüne doğru bir şekilde bağlantı kurulacağını göstermek için fare imlecimizin olduğu yerde tel ve bağlantı noktası arasında küçük bir kare belirecektir. Bu kare göründükten sonra farenin tuşunu bırakabiliriz.

**Not:**

Eğer bağlantınız hatalı olursa bağladığınız kablonun ucunda “x” işaretine benzer bir çarğı olacaktır. Bu işaret size bağlantının doğru yapılmadığını gösterecektir. Bu kısımda bağladığınız kabloya sağ tıklayarak “delete” diyerek kabloyu silmeli ve tekrar bağlamalısınız.

1. Devremizi problemsiz bir şekilde bir birine bağladıktan sonra **input** ve **output** pinlerinde görülen “PIN\_NAME” yazılarına çift tıklayarak pinlerimize örnekteki gibi isimlerini veriyoruz.
2. Bütün aşamalar bittikten sonra projemizi “ctrl + s” tuş kombinasyonu yada [File] -> [Save] kombinasyonu ile kayıt ediyoruz.

**Adım 3. Projenin derlenmesi:**

1. [Processign -> [Start Compilation] menüsü ile projenin derlenmesini sağlıyoruz.
2. Eğer projenizde her hangi bir hata yoksa “Full compilation was successful” yazılı bir pop-up menu gelecektir.
3. Compiler ekranını kapatıyoruz.

**Adım 4. Projenin Simulasyonu:**

Bir devre iki şekilde simüle edilebilir. En basit yöntem, mantıksal elemanların ve ara bağlantı tellerinin mükemmel olduğunu varsaymaktır, bu nedenle sinyallerin devridaim boyunca yayılmasında gecikmeye neden olmaz. Buna işlevsel simülasyon denir. Daha karmaşık bir alternatif, tüm yayılma gecikmelerini hesaba katarak zamanlama simülasyonuna yol açar. Tipik olarak, işlevsel simülasyon tasarlandığı gibi bir devrenin işlevsel doğruluğunu onaylamak için kullanılır. Bu çok daha az zaman alır, çünkü simülasyon basitçe devreyi tanımlayan mantıksal ifadeler kullanılarak gerçekleştirilebilir. Bu derste sadece fonksiyonel simülasyonu kullanacağız.

1. [File] -> [New] -> [Verification / Debugging Files] -> [University Program VWF] seçin ve “OK” butonuna basin.
2. Açılan pencereden [Edit] -> [Insert Node or Bus] seçin.
3. “Node Finder” butonuna tıklayın.
4. Pinlerinizin olduğu listede “>>” hepsini içe aktar butonuna basin ve “Start” butonuna tıklayın.
5. Ekranınızda gördüğünüz pinlerin sırasını değiştirebilirsiniz.
6. [Edit] -> [End Time] menüsüne girerek “set end time” bölümünü 1 microsaniye olarak ayarlayın hemen yanındaki zaman tipi bölümünü “ns” -> “us” olarak değiştirin.
7. Saat sinyalini periyodik, X sinyalini ise rasgele ayarlıyarak aşağıdaki gibi şekillendiriniz.



1. [Assignments] -> [Settings] menüsü altından “Simulation Settings” bölümünden “Mode” seçeneğini “Functional” olarak değiştirin.
2. [File] -> [Save] diyerek simulasyon modelinizi kayıt edin.
3. [Processing] -> [Start Simulation] diyerek projenizi simule edin.

**Aşama 2:**

Yukarıda şematik dizaynı verilen devrenin Verilog HDL ile uygulanması.

/ \* Bu devre bir X girişine ve bir de Z çıkışına sahiptir.

X üzerindeki son dört bit 1001'e eşitse, Z, 1'e eşittir,

ardışık kodların üst üste binmesine izin verilir.

\* /

module SeqDetect1001(CLK, X, Z);

input CLK;

input X;

output Z;

reg Q0, Q1;

wire J0, K0, J1, K1;

wire Z;

assign J0 = X | Q1;

assign K0 = ~X;

assign J1 = ~X & Q0;

assign K1 = X | Q0;

assign Z = X & Q1 & Q0;

always @(posedge CLK)

 begin

 Q0 <= (J0 & ~Q0) | (~K0 & Q0);

 Q1 <= (J1 & ~Q1) | (~K1 & Q1);

 end

endmodule

1. VeriLog HDL'de yeni bir proje açın ve yukarıda verilen kodu editörüne yazın.
2. Kodu derleyin.
3. Aşama 1'de açıklanan prosedürleri izleyerek simülasyonunu gerçekleştirin.

**ÖDEV:** (Bir sonraki laboratuarda teslim edilmelidir.)

X girişinin 4 bit diziliminin 0110 olduğunun tespiti için bir X girişi ve bir Z çıkış ile saat vuruşuna duyarlı bir devre tasarlayın. Bu dizi algılandığında Z = 1, aksi takdirde Z = 0 olur. 4 bitlik ardışık kodların üst üste gelmesine izin verilir. MSB'nin önce geldiğini varsayın.

1. Tasarımınızın şematik dizaynını çıkarın ve simüle edin.
2. Tasarımınızın VeriLog HDL kodunu yazın, derleyin ve simüle edin.
3. Ödev raporu hazırlayın. Raporunuzda devre şemaları, HDL kodu ve simülasyon sonuçları bulunmalıdır..