

# VHDL Programlama Dili ve Sayısal Elektronik Devrelerin FPGA Tabanlı Uygulaması

S. Uzun<sup>1</sup>, M. R. Canal<sup>2</sup>, M. C. Kaçar<sup>3</sup>

<sup>1</sup>Erzincan Üniversitesi, Erzincan, Türkiye, suzun@erzincan.edu.tr

Gazi Üniversitesi, Ankara, Türkiye, mrcanal@gazi.edu.tr

<sup>3</sup>Bayburt Üniversitesi, Bayburt, Türkiye, mckacar@bayburt.edu.tr

## VHDL Programming Language and Application of FPGA Based Logic Electronics Circuit

**Abstract**—In this paper, digital circuit applications are realized with FPGA (Field Programmable Gate Array) produced by Xilinx Company. At the present day, although to design circuit with either PLD (Programmable Logic Device) or other integrateds are highly simple and inexpensive, there are some disadvantages like huge circuit dimensions and not being developed rapidly. FPGA's accomplish these disadvantages. FPGA's a lot of advantages. These advantages can be ordered as follow. FPGA's dissipated less power, programmable simply, faster, have high performance and have highly small dimensions.

**Key Words**—VHDL, Digital Electronics Circuit Design, FPGA

### I. GİRİŞ

Günümüzde teknolojinin hızla gelişmesine paralel olarak kullanılan elektronik devreler de daha karmaşık hale gelmektedir. Geleneksel olarak kullanılan yöntemler ile bu devreleri tasarlamak oldukça zor hale gelmektedir. Bu ihtiyacı karşılamak için yeni teknolojiler geliştirilmektedir. FPGA' lar da bu teknolojilerden biridir. FPGA' lar programlanabilir sayısal bloklar ve bağlantılarını içeren yarı iletken cihazlardır. Programlanabilen bu sayısal kapılar sayesinde karmaşık tasarımlar kolay bir şekilde geliştirilmektedir.

FPGA' ları programlamak için Xilinx firması tarafından geliştirilen ISEWebPack programı kullanılır. Bu program kullanıcıya üç ayrı programlama seçeneği sunmaktadır. VHDL programlama dili FPGA' ları programlama da kullanılan program seçeneklerinden sadece bir tanesidir. VHDL bir donanım tanımlama dilidir. Diğer bir programlama seçeneği şematik programlamadır. Bu programlama seçeneği bize sayısal kapılar, flip-floplar, mux' lar, vb. elemanları kullanarak devre tasarımlarımızı yapmamıza olanak sağlar. Diğer bir programlama seçeneği akış diyagramı şeklinde programlamadır. Bu çalışmada şematik programlama ve kod ile programlama kullanılacaktır.

Bu çalışmanın ilk bölümünde FPGA' lar hakkında genel bilgiler verilmekte, ikinci bölümde VHDL programlama dili hakkında genel bilgiler verilmekte, üçüncü bölümde üç bitlik yukarı-aşağı sayıcı devresi tasarımı hem kod hem de şematik

olarak tasarlanıp FPGA içerisine nasıl yerleştirileceğinden bahsedilmektedir.

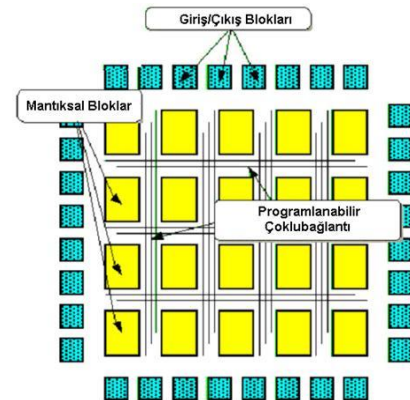
### II. FPGA(FIELD PROGRAMMING GATE ARRAY- APKD- ALAN PROGRAMLAMALI KAPI DİZİLERİ)

FPGA' lar diğer teknolojilerden çok farklıdır ve daha yüksek mantık kapasitesi sunarlar. Mantık blokları dizisinden oluşan bir FPGA programlanabilir giriş-çıkış blokları ile çevrelenmiştir ve programlanabilir dahili veri yolları (interconnect) ile bağlantılıdır. Tipik bir FPGA on binlerce mantıksal bloktan ve daha büyük sayıda flip-flop tan oluşur. Çoğu FPGA' lar mantıksal bloklar arasında %100 bağlantı sağlayamazlar. (Bunu yapmak çok masraflıdır.) Bunun yerine PCB otomatik yönlendirici ve gelişmiş yazılımlar kullanılır. Bir FPGA' nın genel tanımı, programlama devresi, dâhili mantık bloğu dizisi, giriş-çıkış blokları ve bunların bağlantı şekli aşağıda gösterilmiştir. Bu grup içinde alt mimari çeşitleri vardır. Devre içindeki yoğunluk ve performansın sırrı mantık bloklarının performansından ve yönlendirme mimarisinin verimliliğinden geçer [1].

#### A. FPGA' ların Mimarisini

FPGA' lar temelde üç bloktan oluşur:

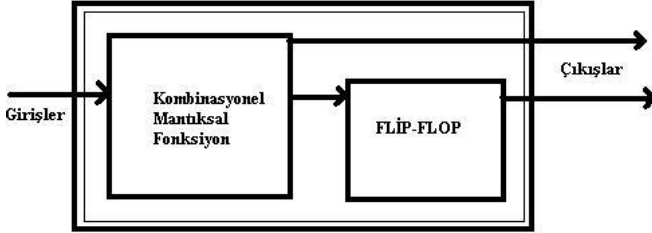
- Sayısal Bloklar(CLB)
- Giriş-Çıkış Blokları(I/O)
- Bağlantı Blokları



Şekil 1.FPGA İç Yapısı

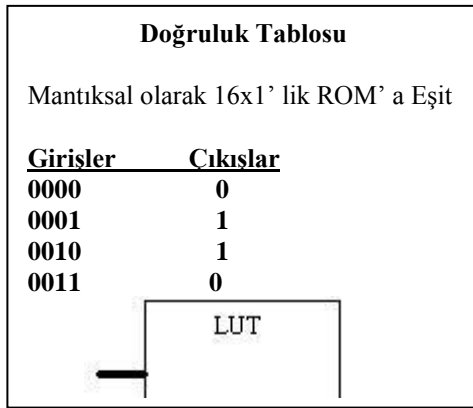
## 1) Sayısal Bloklar

Sayısal bloklar(CLB) Boolean fonksiyonlarının gerçekleştirildiği yapıdır; küçük taneli(fine-grain) ve kaba taneli(coarse-grain) olarak adlandırılan iki sınıfa ayrılırlar. Bu sınıflandırmada; CLB'nin oluşumunda kullanılan transistör sayısı, sayısal bloğun gerçekleyebileceği Boolean fonksiyon sayısı veya sayısal bloğun giriş-çıkış sayısı büyüklük ölçütü olarak kullanılabilir.



Şekil 2 CLB iç Yapısı

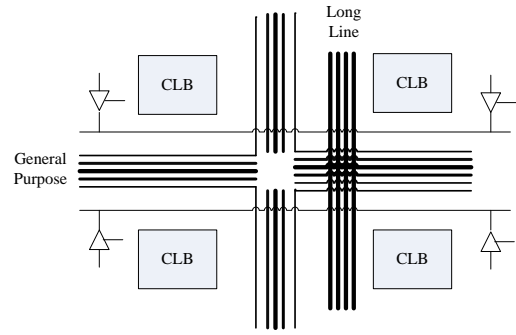
Sekil 2'de iç yapısı verilen CLB, küçük taneli bloklar, genellikle iki girişli bir sayısal kapıya veya bir kaç girişli bir çoklayıcıya eşlik eden saklama elemanından oluşur. Kaba taneli sayısal blokların yapıları çok çeşitlilik göstermelerine karşın yaygın olarak şekil 3 ile gösterilen doğruluk tablosu (LUT) veya çoklayıcı (Multiplexer) gibi daha büyük saklama elemanlarından oluşur.



Şekil 3. CLB' yi Oluşturan LUT Elemanı

## 2) Bağlantı Blokları

Şekil 4' de verilen bağlantı blokları sayısal bloklarla giriş-çıkış blokları arasındaki bağlantıyı sağlayan yapılardır. Bu yapılar, yönlendirme kanalları ve programlanabilir anahtarlardan oluşur.



Şekil 4. Bağlantı Blokları

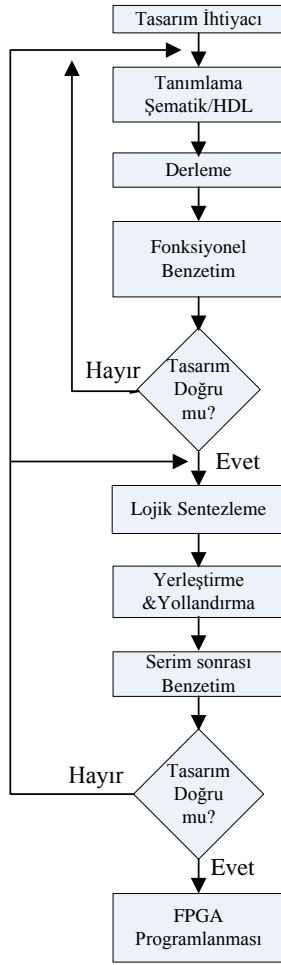
FPGA mimarileri bağlantı kanallarının yapısına göre; Simetrik Dizi Mimarisi (Symmetrical Array), Sıra Tabanlı Mimari(Row-Based Array) ve Kapı Deniz Mimarisi( Sea of Gate) olmak üzere üç ana gruba ayrılır.

## B. FPGA Kullanarak Gerçekleştirilen Devrelerin Tasarım Süreci

Tasarım süreci, gerçekleştirilen devre fonksiyonlarının, sözel veya şematik olarak tanımlanması ile başlar. Sözde tasarımda genellikle yüksek seviyeli donanım tasarımları dilleri (Hardware Description Language, HDL) kullanılır. Şematik tanımlamada ise bir çok firma tarafından gerçekleştirilmiş şematik editör programlarından faydalanılır. Tanımlama ne şekilde olursa olsun, derleme işlemi sonrasında, tüm tanımlamalar, standart bağlantı listesi (netlist) biçimine çevrilir. Yapılan tanımlamaların, istenilen fonksiyonları yerine getirip getirmediği fonksiyonel benzetim( Functional Simulation) yapılarak test edilir. Benzetim sonucuna göre, tanımlamada gerekli değişiklikler yapılır. Şekil 5' de tasarım sürecinin akış şeması verilmiştir [2].

## III. VHDL PROGRAMLAMA DİLİ

VHDL, Very High Speed Integrated Circuits Hardware Description Language İngilizce kelimelerin baş harflerinden oluşan Yüksek Hızlı Entegre Devreleri Donanım Tanımlama Dili anlamına gelen bir programlama dilidir. Günümüzde donanım tanımlama dillerinden en çok kullanılanıdır. Amerika savunma bakanlığı ve IEEE (Amerikan Elektrik ve Elektronik Mühendisleri Enstitüsü) tarafından 1980' lerde geliştirilmiş bir donanım tanımlama dilidir. 1987' de IEEE standardı oluşturulmuş (VHDL-87) daha sonra bu standart 1993' de geliştirilmiştir (VHDL-93) [5]. Xilinx firması tarafından geliştirilen ISEWebPack programı kullanıcılara üç farklı programlama seçeneği sunmaktadır. Şematik programlama, kod ile programlama, akış diyagram şeklinde programlama. Ayrıca şematik olarak yaptığınız tasarımlarınızın VHDL kodlarını ürettilip bundan sonra yolunuza kodları kullanarak da devam etmenize olanak sağlamaktadır. İstedığınız zaman kod istediğiniz zaman şematik olarak programlama yapabilme esnekliğini bizlere sunmaktadır. Bu özellik donanım açıklama dilleri arasında onu ayrıcalıklı kılmaktadır.



Şekil 5. FPGA Kullanılarak tasarım sürecinin genel akış diyagramı

Bir tasarımın yapısını tamamen VHDL programlama dilini kullanarak yapmak mümkündür. Bu tasarımı parçalarına ayırıp bu parçaları tek tek oluşturup birbirleri ile bağlamak mümkündür. Yapılan tasarımları hayat geçirmeden önce sanal ortamda benzetimini yapmakta mümkündür. VHDL programlama dili ile SPLD' leri de programlamak mümkündür.

#### IV. FPGA TABANLI TASARIM

FPGA ile gerçekleştirilen örnek bir tasarım, Xilinx firmasının geliştirdiği ISE WebPack programı kullanılarak geliştirilmektedir. Öncelikle tasarım şematik olarak gerçekleştirilmekte ve daha sonra aynı tasarım VDL kodları ile gerçekleştirilmektedir.

##### A. 3 Bitlik Senkron Yukarı-Aşağı sayıcının VHDL kodları ile Tasarımı

3- bitlik yukarı-aşağı sayıcı devresi ileri ve geri sayma işlemi yapabilen senkron sayıcısı tasarlanmaktadır. Bizim yapacağımız tasarımda yukarı sayma ve aşağı sayma diye iki adet kontrol ucu bulunmaktadır. Yukarı sayma kontrol ucu sayıcı devresinin yukarıya doğru yani "000" dan "111" e kadar

saymasını sağlamakta, aşağı sayma kontrol ucu ise sayıcı devresinin aşağıya doğru yani "111" dan "000" a kadar saymasını sağlamaktadır. Sadece bizler sayıcı devresini hangi yöne saymasını istiyorsak ilgili kontrol ucuna lojik 1 uygulamamız yetecektir. Örneğin sayıcı devresinin yukarıya doğru saymasını istiyorsak yukarı sayma kontrol ucunu lojik 1 yapmamız yeterlidir. Aşağıda 3 bitlik senkron yukarı-aşağı sayıcının VHDL kodları görülmektedir.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity UP_DOWN is
port(CLK, CNT_UP, CNT_DOWN: in Bit
      Q0, Q1, Q2: buffer Bit);
end UP_DOWN;
architecture COUNTER of UP_DOWN is
Component JK_FF
port(J, K, CK: in Bit
      Q, QN: buffer Bit);
end Component;
Signal S1, S2, S3, S4, S5, S6, S7, S8: BIT;
begin
JK1: JK_FF port map ('1', '1', CLK, Q0, S1);
A1: AND2 port map (CNT_UP, Q0, S2);
A2: AND2 port map (S1, CNT_DOWN, S3);
O1: OR2 port map(S2, S3, S4);
JK2: JK_FF port map ('1', '1', S4, Q1, S5);

Component AND2
port(A, B: in Bit;
      C: out Bit);
end Component;
    
```

Öncelikle VHDL kodlarının bulunduğu kütüphaneler programa dahil edilerek bu kütüphane içerisindeki kodlar kullanılmaktadır. Bu kütüphane çağırma işlemi;

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.AL
    
```

Şeklinde yapılmaktadır. Daha sonra Entity(Varlık) tanımlaması yapılmaktadır. Bu tanımlama tasarımımızda bulunan bütün giriş ve çıkışlarımızı ve bu giriş ve çıkışlarını veri tipleri tanımlamaktadır. Bu tanımlama işlemi aşağıda görüldüğü gibi yapılmaktadır.

```

entity UP_DOWN is
port(CLK, CNT_UP, CNT_DOWN: in
      Bit
      Q0, Q1, Q2: buffer Bit);
    
```

Bu aşamadan sonra Architecture (mimari) tanımlaması yapılmaktadır. Bu aşamada tasarımınızın ne iş yapmasını istiyorsanız bu tanımlama altında kodlarla belirtilmelidir. Bizler öncelikle tasarımımızda kullanacağımız sayısal elemanların her birinin tanımlamasını Component (Eleman) tanımlaması ile ayrı ayrı tanımlayacağız. JK flip-flop' u aşağıda olduğu gibi tanımlanmaktadır;

```
Component JK_FF
port(J, K, CK: in Bit
      Q, QN: buffer Bit);
end Component;
```

Kodlarda görüldüğü gibi JK flip-flopun girişleri ve çıkışları bit olarak tanımlanmış ayrıca girişleri belirtmek için "in", çıkışları belirtmek için "out" kullanılmaktadır. Aynı şekilde AND(VE) ve OR(VEYA) sayısal kapıları da tanımlanmaktadır.

```
Component AND2
port(A, B: in Bit;
      C: out Bit);
end Component;

Component OR2
port(A, B: in Bit;
      C: out Bit);
end Component;
```

Bu tanımlamalardan sonra begin.....end blokları arasında kodlar yazılmaktadır. Yazılan kodlar xilinx firmasına ait ISE WebPack programında derlenerek JTAG kablo yardımıyla entegre içerisine gömülmektedir.

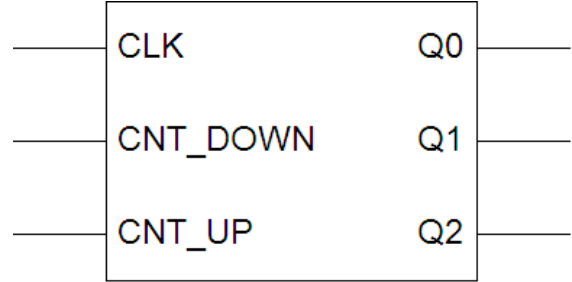
### B. 3 Bitlik Senkron Yukarı-Aşağı Sayıcının Şematik Tasarımı

Bu tasarım Xilinx firmasına ait ISE WebPack programının Schematic (şematik) programlama seçeneği kullanılarak gerçekleştirilmektedir.

Program içerisinde bulunan sayısal elemanlar çalışma ekranına kütüphaneden seçilerek yerleştirilir ve gerekli bağlantılar kolay bir şekilde yapılmaktadır. Bu tasarımın VHDL kodlarına bakacak olursak aşağıda yapılan tasarımı aşama aşama kodlara nasıl döküldüğünü ve ne kadar kolay olduğu görülmektedir. Şematik programlama basit tasarımlarda kullanmak avantajlı olabilir fakat karmaşık devrelerin tasarımlarında kullanmak oldukça zordur. Bunun yerine VHDL kodlarını kullanmak daha avantajlıdır. ISE WebPack programı kullanıcılarına her iki programlama seçeneğini aynı anda kullanma şansını da vermektedir. Örneğin kullanıcı tasarımının bir kısmını VHDL kodlarını kullanmakta. Bir kısmında ise şematik programlamayı kullanmaktadır. İşte bu durumda şematik olarak tasarladığı kısmını program içinde derleyerek programa VHDL kodlarını üretir ve bu kodları kopyalayıp

diğer kodların devamına yapıştırarak tasarımını tamamlayabilir.

Aynı şekilde VHDL kodlarını program içinde derleyerek şematığını çıkarttırarak diğer şematiğe ekleyebilir. Aşağıda VHDL kodlarının şematik haline dönüştürülmüş kılıf şeklindeki hali görülmektedir.



Şekil 6. VHDL Kodlarının şematik haline dönüşmüş RTL şeması

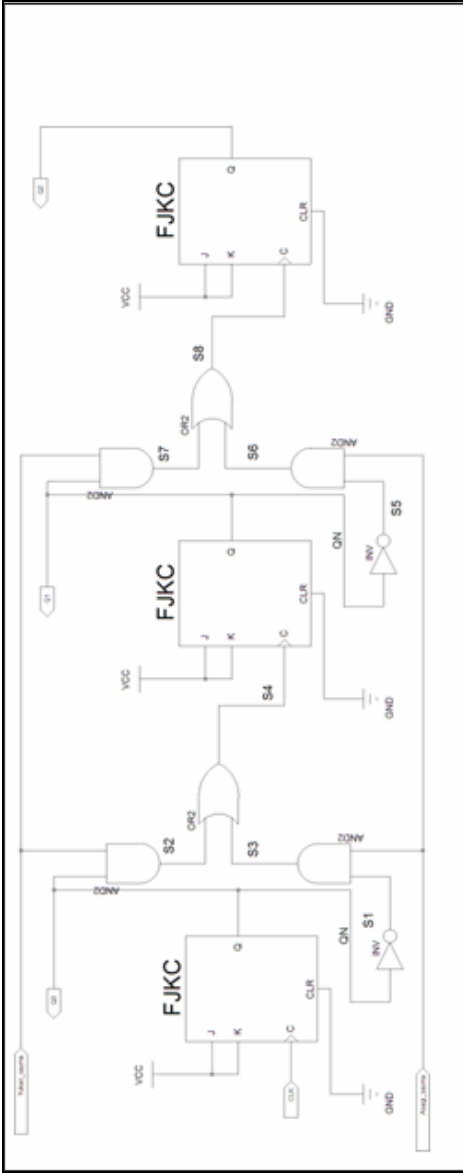
Bu kılıf şeklinin içerisinde senkron 3 Bitlik aşağı yukarı sayıcı devresi bulunmaktadır. Bu kılıf ISE WebPac programının kütüphanesine kaydedilmekte ve istenilen herhangi bir tasarımda kullanılabilir. Bu RTL şemasının içyapısı aşağıda görülmektedir. ISE WebPack programını sağlamış olduğu bu esneklik sayesinde tasarımlar ne kadar karmaşık olursa olsun çok kolay bir şekilde üstesinden gelinmektedir.

### V. SONUÇ

Bu çalışmada Xilinx firması tarafından üretilen FPGA' lar, bu FPGA'ların programlanmasında kullanılan VHDL programlama dili ve örnek olarak da 3 bitlik senkron yukarı-aşağı sayıcı devresi anlatılmaktadır. Yapılan tasarım üzerinden VHDL programlama seçenekleri anlatılmakta ve kullanıcıya sağladığı faydalara değinilmektedir.

FPGA, sayısal devrelerinin hayata geçirilmesi için oldukça kullanışlı ve basittir. Kullanıcılar hem sayısal devre tasarımı hem de bu tasarımlarının VHDL kodlarını eş zamanda yapabilmekte ve uygulamaya geçirebilmektedir. Sadece sayısal devrelerde değil her türlü tasarımları uygulamaya geçirmekte de oldukça kullanışlıdır. Ayrıca gerçekleştirilen tasarımların geliştirilmesi VHDL programlama dili ve FPGA sayesinde oldukça kolay hale gelmektedir.

Burada gerçekleştirilen tasarımlar xilinx firmasına ait XC3S700A entegresinde, Spartan 3A Starter Kit deneme kartına uygulanmıştır.



Şekil 7. 3-Bitlik Senkron Yukarı-Aşağı Sayıcının ISE WebPack programında Schematic Programlama seçeneği kullanılarak oluşturulmuş devre şeması [6]

#### KAYNAKLAR

- [1] UZUN, S., "XC95XX CPLD' sinin İncelenmesi ve Programlama Kartı Tasarımı" , Lisans Tezi, Gazi Üniversitesi Teknik Eğitim Fakültesi, Ankara, 32, 34,2008.
- [2] ÇAVUŞLU, M. A., "FPGA ile Yapay Sinir Ağı Eğitiminin Donanımsal Gerçeklenmesi", Lisans Tezi, Kocaeli Üniversitesi Mühendislik Fakültesi, Kocaeli, 33-35, 38, 2006
- [3] PARNELL, K., METHA, N., "Programmable Logic Design Quick Start Handbook", Xilinx, USA, 1-190, 2003.
- [4] WANG, F., ZHANG, W., YU, S., "Implementation of Hdtv Pes Combiner Based on Horizontal Six-Block Sementation", IEEE Transactions on Broadcasting, 49(2): 217-220, 2003.
- [5] www.cs.ucr.edu, 12.03.2009.
- [6] www.async.elen.utah.edu, 18.08.2009.
- [7] www.xilinx.com, 22.06.2009.
- [8] www.altera.com, 21.05.2009.
- [9] J.BHASKER , "A VHDL Primer Third Edition", Editör: Kathleen M. Caren, Pearson Education, 07458, New Jersey, A.B.D., 132-136, 1999.