

# FPGA ÜZERİNDE HİBRİT SAYISAL SİSTEM MODELEMESİ

Ahmet Turan ÖZDEMİR<sup>1</sup>

Kenan DANIŞMAN<sup>2</sup>

<sup>1,2</sup>Elektrik Elektronik Mühendisliği Bölümü, Erciyes Üniversitesi  
38039, Melikgazi, Kayseri

<sup>1</sup>e-posta: aturan@erciyes.edu.tr

<sup>2</sup>e-posta: danismak@erciyes.edu.tr

## ABSTRACT

*In this work, a MCU (MicroController Unit) with a combinational logic circuit design are implemented on an FPGA (Field Programmable Gate Array) chip. VHDL (Very high speed integrated circuit Hardware Description Language) code can be concurrent (parallel) or sequential. Concurrent code is also called dataflow code. For that reason, VHDL is usually referred to as a code rather than a program. In this work either of two structure that sequential and concurrent are implemented on an FPGA chip. FPGAs have traditionally found use in high-speed custom digital applications where designs tend to be more constrained by performance rather than cost. The explosion of integration and reduction in price has led to the more recent widespread use of FPGAs in common embedded applications.*

**Anahtar sözcükler:** FPGA, SoC, ASIC, DSP, MCU.

## 1. GİRİŞ

Günümüz sayısal sistem tasarım teknolojisinde sıklıkla kullanılan yapılar FPGA (Alan Programlanabilir Kapı Dizileri) yongalarıdır. VHDL (Çok Hızlı Tümeleşik Devre Donanım Tanımlama Dili) dilinin tasarımcıya sunduğu esneklik ve etkili sistem tasarımı yeteneği sayesinde karmaşık sistemleri çok kısa sürelerde gerçekleştirmek mümkündür. FPGA tabanlı tasarımlar, gerek düşük maliyeti gerekse kısa tasarım süreçleri ile günümüz modern tasarım teknolojisinin vazgeçilmezi olmuşlardır. SoC (System on Chip) çözümler hem PCB (Baskılı Devre Kartı, Printed Circuit Board) alanının büyümesini önlemekte hem de devre güç tüketimlerini azaltmaktadır. Bu avantajların bütününe kullanıldığı platform FPGA yongalarıdır [1].

Çok yüksek seviyeli tasarımları oluşturmak için sadece VHDL yeterli olmamakta ve tasarım karmaşıklığı artmaktadır. Bu bakımdan gerek üretici firmalar gerekse üçüncül firmalar (third party vendor) tarafından sağlanan donanımsal ve yazılımsal destekler ile tasarım araçlarının tipi sayısı ve kabiliyetleri artırılmaktadır [2].

Bu tasarım ve yazılım araçlarından bazıları aşağıdaki gibidir. Mathworks Matlab ile Xilinx System Generator, Altium Designer ile Xilinx ISE ve daha birçok yazılım aracı ile tasarımlar oluşturmak mümkündür. Bu tür üreticiler ürünlerinin mevcut özelliklerini de FPGA tasarımı içerisine ilave ederek bu yonga teknolojisinin daha etkili olarak kullanılmasını sağlamaktadırlar [3, 4, 5].

Sistem çözümlerinde isteklerin ve sistem gereksinimlerinin yapısına göre tasarımlar geliştirilir. Gömülü devre tasarımlarında genellikle mikroişlemciler ve bunlara bağlı çevresel sayısal donanımlar kullanılır. Bu sayısal donanımlar, işlemci yükünü hafifletmek, işlem hızını arttırmak veya işlemcinin kapasitesinin yetersiz kaldığı durumları tamamlamak için kullanılırlar. Bir ön veya uç sayısal donanım işlemciye bilgi üretmek veya işlemcinin ürettiği bilgiyi başka bir bilgi tipine dönüştürmek için kullanılabilir. Bu çalışmada FPGA yongası içerisinde bir sayısal donanım ve bir işlemci yongası oluşturulmuştur. Yonga içerisinde oluşturulan sayısal donanımın ürettiği değerler yine yonga içerisindeki İşlemci tarafından yorumlanarak sonuç çıkarımları yapılmıştır.

Bu çalışmada FPGA kullanılarak bir SoC uygulaması yapılmıştır. SoC yaklaşımlarda en iyi performans ASIC (Uygulamaya Özel Entegre Devre, Application Specific Integrated Circuit) yongalara aittir. Fakat gerek üretim maliyeti gerekse üretim süresi bakımından FPGA yongaları avantajlıdır. Bu yüzden ilk örneklerin geliştirilmesi ve deneysel araştırma çalışmalarının yapılması bakımından FPGA yongalarının kullanımı tasarımcıya büyük avantajlar sunmaktadır [6].

Çalışmada kullanılan FPGA yongası Xilinx Spartan-3 ailesine ait XC3S200'dür. Bu çalışmada Digilent firmasına ait "Xilinx Spartan 3 starter kit" uygulama geliştirme kartı kullanılmıştır.

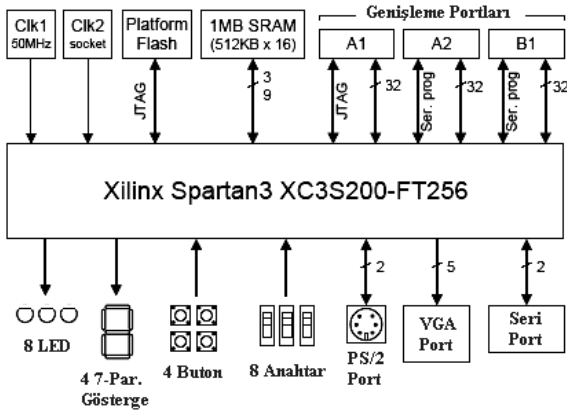
Kullanılan geliştirme kartı, sistem mimarisi, uygulama gerçekleştirilmesi ve sonuçlar bölümler şeklinde detaylı olarak aşağıda verilmiştir.

## 2. UYGULAMA GELİŞTİRME KARTI

Bu çalışmada Digilent firmasına ait “Xilinx Spartan 3 Starter Kit” uygulama geliştirme kartı kullanılmıştır. Bu kart üzerinde;

- 200.000 kapı elemanına sahip, Xilinx Spartan-3 FPGA yongası;
- 2Mbit flash hafıza
- 8 kaydırmalı anahtar, 4 buton, 8 LED ve 4 adet 7 parçalı göstergeler;
- Seri, VGA, PS/2 ve Klavye portları;
- 3 adet 40 bacaklı genişleme portu;
- 1MB SRAM bulunmaktadır.

Kartın blok şeması Şekil-1’de verilmiştir.



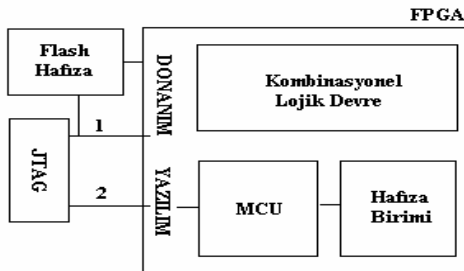
Şekil 1: Digilent Spartan 3 Starter Kit Blok Şeması

Yapılan çalışmada kart üzerindeki ledler, 7 parçalı göstergeler, sistem saat girişi ve butonlar sistem kontrol ve göstergeleri olarak kullanılmıştır. Sistem saat girişi 50 MHz’dir [7].

Spartan3 XC3S200 FPGA yongası 3.3V besleme altında, I/O (Giriş/Çıkış, Input/Output) iletişim standardı LVCMOS33 (Düşük Voltaj CMOS 3.3V, Low Voltage CMOS 3.3V) kullanılmıştır [8].

## 3. SİSTEM MİMARİSİ

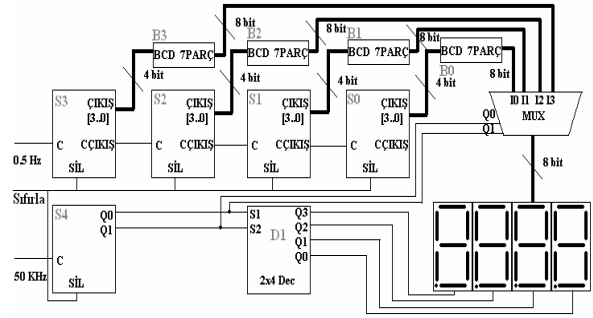
Bu çalışmada FPGA içerisinde bir MCU (Mikro Denetleyici Birimi, Micro Control Unit) ve kombinasyonel lojik devre oluşturulmuştur. Bu yapı ile ilgili blok diyagram Şekil-2’de verilmiştir.



Şekil 2: Sistem Mimarisi

Sistem gerçekleştirilmesi için 2 adet JTAG (Birleşik Test İşlemi Grubu, Joint Test Action Group) Programlama portu kullanılmıştır. Bu portlardan JTAG-1 donanımı tanımlayan, konfigürasyon bit serisinin kart üzerindeki Flash hafızaya veya FPGA’ye yüklenmesini sağlamaktadır. JTAG-2 ise FPGA içerisinde oluşturulan işlemcinin programını yani derlenmiş hex dosyasını program hafızasına yüklemek için kullanılmıştır. Bu yüzden JTAG-1 donanımı, JTAG-2 ise yazılımı tanımlayan yükleme portu olarak ifade edilebilir.

Sistem içerisinde oluşturulan kombinasyonel lojik devre 0-9999 arası sayan bir BCD (İkili Kodlanmış Ondalık sayı, Binary Coded Decimal) sayıcı devresidir. Sayıcı devre giriş saat frekansı 50MHz temel saat frekansı bölünerek elde edilmiştir ve 0.5 Hz’dir. Sayıcı her 2 saniye için 1 değer arttırarak saymaktadır. Sistemde kullanılan MCU ise sistem saat frekansının yarısı, 25 MHz olarak ayarlanmıştır. İşlemci BCD sayıcının ürettiği sayma değerlerini alarak ikili koda dönüştürmektedir. Sayıcı devresi blok diyagramı Şekil-3’de, işlemci ile bağlantısı ise Şekil-4’de verilmiştir.

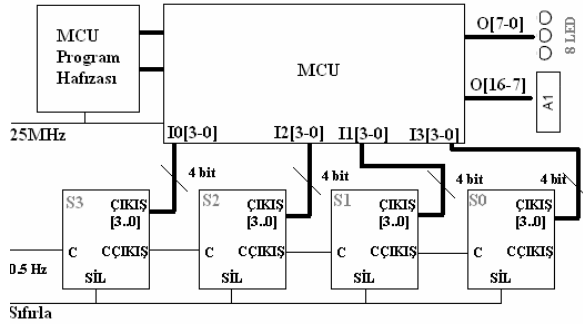


Şekil 3: Sayıcı Devresi Tasarımı

Şekil-3’de verilen sayıcı devresi girişlerine uygulanan iki ayrı saat frekansı bulunmaktadır. 0.5Hz saat frekansı BCD sayıcı devresinin 2 saniyede 1 yukarı doğru sayma değeri üretir. Her bir sayıcı grubunun ürettiği değer Bx işlem grupları ile 7 parçalı göstergeler için uygun değerlere çevrilir ve ortak veri hatlı 7 parçalı göstergeler grubuna uygulanmaktadır. Göstergeler ortak katotludur ve 2x4 kodçözücü çıkışı ile birim zamanda 1 tanesi seçilmektedir. Veri hattına uygulanacak bilgi ise MUX seçici elemanı ile belirlenmektedir. MUX ve DEC seçici girişleri 50KHz saat frekanslı 2 bit sayıcı çıkışı ile kontrol edilmektedir. Ortak seçicili bu iki yapı bu sayede senkron hale getirilmiştir.

Sayma bilgisi üreten yapı yazılımdan bağımsız bir devredir. Sayma değerleri S0, S1, S2 ve S3 sayıcılarından alınarak sistemdeki MCU’ya uygulanacaktır. Burada kullanılan işlemci aldığı BCD sayma bilgilerini binary forma sokarak 14 bitlik bir bilgi üretecektir. Bu bilgilerin 8 bitlik LSB bölgesi kart üzerindeki ledler, 6 bitlik MSB bölgesi ise A1

genişleme portu üzerinden verilmektedir. İşlemci ile sayıcı bağlantısı Şekil-4’de verilmiştir.

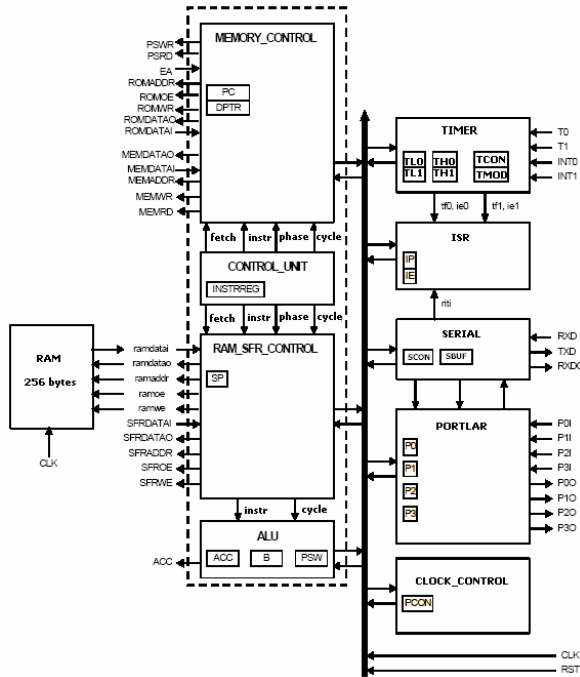


Şekil 4: MCU ve Sayıcı Devresi Bağlantısı

MCU sayıcı devresinden üretilen BCD sayma değerlerini yorumlayarak bunları ikili sisteme çevirmektedir. MCU'nun özellikleri aşağıdaki gibidir;

- 8 bit instruction decoder;
- 8 bit aritmetik/lojik işlem birimi;
- 32 Giriş/Çıkış portu;
- 16 bit Timer/Counter;
- 8/9 bit UART full duplex port;
- 5 interrupt pini;
- 256 B Yazılabilir/Okunabilir veri hafızası;
- 64 KB dahili program hafızası;
- 64 KB harici program hafızası adresleme;
- 64 KB harici veri hafızası adresleme.

Oluşturulan MCU'nun blok diyagramı Şekil-5’de verilmiştir.



Şekil 5: MCU Blok Şeması

FPGA kaynaklarının kullanım yüzdeleri ve adetleri Şekil-6’da verilmiştir. Bu şekilden de görüldüğü üzere FPGA içerisindeki toplam kaynakların sadece %52’si kullanılarak çalışma tamamlanmıştır. Şekilde her bir kaynağa ait kullanım bilgisi detaylı olarak verilmiştir.

Sonuç Özeti		
Devre Kaynakları - Kullanım Özeti		
4-Input LUTs - Logic	2,374 / 3,840	61%
4-Input LUTs - Total	2,447 / 3,840	63%
Block RAMs	2 / 12	16%
Global Clocks	1 / 8	12%
I/O Pins	34 / 173	19%
Slice Flip Flops	879 / 3,840	22%
Slices with only related logic	1,425 / 1,425	100%
Slices with unrelated logic	0 / 1,425	0%
Slices	1,425 / 1,920	74%

Şekil 6: FPGA Kaynaklarının Kullanım Yüzdeleri

Sistem iki aşamada oluşturulmuştur, ilk aşama donanım mimarisinin oluşturulması, ikinci aşama ise donanım üzerinde oluşturulacak yazılım mimarisinin oluşturulmasıdır. Donanım sentezlendikten sonra oluşturulan .mcs uzantılı dosya uygulama geliştirme kartı üzerindeki ayırık Flash hafızaya yüklenmektedir. Kart enerjilendiğinde, FPGA konfigürasyon verisini bu hafızadan almaktadır.

## 4. UYGULUMA GERÇEKLEMESİ

Bu çalışma Altium Designer 6.0 ile tasarlanmış ve Xilinx ISE editörü tarafından sentezlenmiştir. Tasarım içerisinde şematik, core ve işlemci C kodu birleştirilmiştir. Uygulamaya özel donanım ihtiyaçları tek bir FPGA yongası üzerinde gerçekleştirilmiştir. Donanım bilgi dizisi ve soft core işlemci yazılım dosyaları sırası ile Flash hafıza ve FPGA içerisinde Block RAM’ler ile oluşturulmuş MCU program hafızalarına yüklenmiştir.

Sistem donanım mimarisinin sentezlenmesi zaman alırken, mevcut donanım içerisindeki MCU’nun programının derlenmesi ve değiştirilmesi çok kısadır. Sistem C derleyicisi MCU için .hex dosyası üretmektedir. Bu sayede oluşturulmuş mevcut donanım tıpkı bir mikrodenetleyici gibi kullanılabilir ve sistem üzerinde denemeler rahatlıkla ve kısa sürelerde yapılabilmektedir.

Mevcut yapı Digilent Xilinx Spartan-3 Starter Kit uygulama geliştirme devresi üzerinde gerçekleştirilmiştir. BCD sayıcı devresinin ürettiği BCD sayma değerleri sistemdeki MCU tarafından yorumlanmış ve kart üzerindeki ledler ile genişleme portu A1 üzerinden gözlemlenmiştir.

FPGA içerisindeki veri hareketliliği ve lojik değerler Nexus Debugger ile JTAG-2 portundan gözlemlenmiştir. Bu sayede sentez ve derleyiciler ile donanımın cevaplarının doğruluğu kontrol edilmiştir.

Uygulama sonuçlarının kısa sürede alınması ve güçlü tasarım araçları ile FPGA çözümleri tasarımcıya büyük avantajlar sunmaktadır. Uygulamaya özel çözümlerde mevcut mimarilerin sınırlamalarına maruz kalmadan çözüm üretebilmek açısından en iyi çözüm ASIC yongalarıdır. ASIC yongalar gerek sayısal gerek analog gerekse hibrit sistemlerin modellenmesinde en iyi sonuçları üretmektedir. Fakat FPGA çözümleri sadece sayısal sistemlerin modellenmesinde kullanılabilir. Analog ön ve sonlandırıcı gereksinimler harici devreler ile giderilebilmektedir. Fakat gerek tasarım sürelerinin uzun ve maliyetlerinin yüksek olması sebebi ile ilk örnek devrelerin hızlıca yapılmasında ASIC iyi bir seçenek değildir. Bununla birlikte DSP (Sayısal İşaret İşleme, Digital Signal Processing) işlemciler hızlı ilk örnekleme devrelerinin oluşturulmasında iyi bir seçenektir. Fakat DSP yongalar mevcut mimarilerine bağımlı kaldıkları için donanımsal esnekliğe müsait yapılar değildir. FPGA ise hem esnek mimarisin hem de kısa tasarım süresinin avantajlarının birleşimi ile hızlı ilk örneklemede en iyi çözümdür.

## 5. SONUÇLAR

Bu çalışma ile SoC bir çözüm tasarlanmıştır. Tasarlanan sistem, hibrit bir sayısal yapıya sahiptir. Sistem, kombinasyonel lojik ve MCU biriminden oluşmaktadır. Sistemdeki sayıcı yazılımdan bağımsız çalışan bir donanım iken MCU yazılıma bağımlı bir donanım olarak yonga içerisine gömülmüştür. Yapılan çalışmadaki bu örnek paralel çalışan algoritmaların FPGA içerisinde nasıl oluşturulduğuna bir örnektir. Bir mikrodenetleyici veya DSP gerçek anlamda paralel işlem yapamazken veya sınırlı seviyede pipeline çalışabilirken FPGA içerisinde gerçek paralel işlem birimlerinin oluşturulması mümkündür. Bu örnek ile birbirinden bağımsız ve eşzamanlı çalışan işlem birimlerinin SoC olarak nasıl elde edileceği örneklenmiştir. Paralel işlem birimlerinin bir yonga üzerinde gerçekleştirilmesi hem devre elemanı sayısını azaltarak maliyetleri azaltmakta hem de tasarım karışıklığını ortadan kaldırmaktadır. Bu ise maliyet ve performans açısından tasarımcıya avantaj sunmaktadır.

Bir çok alanda eş zamanlı işlem birimlerinin oluşturulması ve bu sonuçların değerlendirilmesine ihtiyaç duyulmaktadır. FPGA bu ihtiyaca cevap vermektedir. Ayrıca ihtiyaç duyulan sistemin Bir yonga içerisinde gerçekleştirilmesi ise sistemin uygulanabilirliği ve tercih edilirlğini önemli ölçüde arttırmaktadır.

## KAYNAKLAR

- [1] Dong Lin, Shiyuan Yang, “An Implementation of Rapid Prototyping Platform of Embedded Systems”, ISCE '06 IEEE Tenth International Symposium On Consumer Electronics, (2006), 1-4,.
- [2] Wigley G., Kearney D., “The first real operating system for reconfigurable computers”, ACSAC 6th Computer Systems Architecture Conference, (2001), 130-137.
- [3] L. Fanucci, A. Giambastiani, F. Iozzi, C. Marino, A. Rocchi, “Platform based design for automotive sensor conditioning”, Design, Automation And Test In Europe, Proceedings, vol. 3, pp. 186-191, 2005
- [4] Haldar M., Nayak A., Shenoy N., Choudhary A., Banerjee P, “FPGA hardware synthesis from MATLAB”, Fourteenth International Conference On VLSI Design, (2001), 299-304.
- [5] Altium Inc., “Why embedded developers”, [www.altium.com/files/pdfs/Why-embedded-developers-EN.pdf](http://www.altium.com/files/pdfs/Why-embedded-developers-EN.pdf), 2006.
- [6] Ebeling C., Fisher C., Guanbin Xing, Manyuan Shen, Hui Liu, “Implementing an OFDM receiver on the RaPiD reconfigurable architecture”, IEEE Transactions On Computers, vol. 53, pp. 1436-1448, 2004.
- [7] Digilent Inc., “Xilinx Spartan-3 Starter Kit” <http://www.digilentinc.com/Products/Detail.cfm?Nav1=Products&Nav2=Programmable&Prod=S3BOARD>, WA, USA, 2006.
- [8] Xilinx Inc., “Spartan-3 Complete Data Sheet (All four modules)”, [http://www.xilinx.com/xlnx/xweb/xil\\_publications\\_display.jsp?category=Publications/FPGA+Device+Families/Spartan-3](http://www.xilinx.com/xlnx/xweb/xil_publications_display.jsp?category=Publications/FPGA+Device+Families/Spartan-3), 2006.