

# Sayısal FIR Süzgeç için VHDL Kod Geliştirilmesi

Sabri BİCAKCI<sup>1</sup>, Bahadır ÇETİNKAYA<sup>1</sup>, Nurhan KARABOĞA<sup>1</sup>

<sup>1</sup>Elektrik ve Elektronik Mühendisliği Bölümü, Erciyes Üniversitesi, Kayseri  
1030225197@erciyes.edu.tr, cetinkaya@erciyes.edu.tr, nurhan\_k@erciyes.edu.tr

## Özet

Bir çok elektronik cihazda kullanılan temel elemanlardan birisi sayısal süzgeçlerdir. Sayısal süzgeçlerin donanımsal olarak gerçekleştirilmesinde farklı yaklaşımlar mevcuttur. Örneğin APKD (Field Programmable Gate Array Logic – Alan Programlanabilir Kapı Dizisi)’ler, mikro denetleyiciler veya opamplar vb. kullanılarak sayısal filtreler gerçekleştirilebilmektedir. APKD ile gerçekleştirilen tasarımlarda güç tüketimlerinin düşük ve performanslarının yüksek olduğu bilinmektedir. APKD ile yapılan tasarımlar oldukça esnek olabilmektedir ve APKD’lerin bir işlemi gerçekleştirme hızları oldukça yüksektir. Ayrıca APKD’ler ile aritmetik birimler kolayca gerçekleştirilebilmektedir. Bu çalışmada APKD kullanılarak sayısal FIR süzgeç tasarımı incelenerek, düşük güç tüketimine ve iyi bir performansa sahip olan APKD’lerin sayısal süzgeçlerin donanım olarak gerçekleştirilmesinde kullanılabilecek bir VHDL kodu tasarımı mantığı geliştirilmiştir.

## 1. Giriş

Doğrusal zamanla değişmeyen sistemler dürtü cevaplarına göre FIR (Finite Impulse Response – Sonlu Dürtü Yanıtlı) veya IIR (Infinite Impulse Response – Sonsuz Dürtü Yanıtlı) şeklinde iki grupta sınıflandırılabilirler. Eğer dürtü cevabı sınırlı süreli ise, sistem sonlu dürtü yanıtlı veya kısaca FIR sistem olarak adlandırılır. Bir FIR süzgeç çıkışı şimdiki ve geçmişteki giriş değerlerinin ağırlıklı toplamından oluşan bir süzgeç olarak tanımlanabilir. Bir çok elektronik cihazda sayısal süzgeçler kullanılmaktadır. Sayısal süzgeçler, simülasyonlarla gerçekleştirilebilirler fakat asıl önemli olan süzgeçlerin donanımsal olarak da gerçekleştirilebilmesidir. Literatürde sayısal süzgeç tasarımında APKD’lerin kullanılmasıyla ilgili birçok çalışma vardır[1-10].

APKD ile gerçekleştirilen tasarımlarda güç tüketimlerinin düşük ve performanslarının yüksek olduğu bilinmektedir. APKD ile yapılan tasarımlar oldukça esnek olabilmektedir ve APKD’lerin bir işlemi gerçekleştirme hızları oldukça yüksektir. Ayrıca APKD’ler ile aritmetik birimler kolayca gerçekleştirilebilmektedir.

Sayısal süzgeçler, doğrusal fazlı ve daima kararlı oldukları için aynı zamanda tasarımları IIR süzgeçlere göre daha kolay olduğu için bu çalışmada, FIR süzgeçlerin APKD kullanılarak tasarımı konusu incelenmiştir. Sayısal bir FIR süzgeç aşağıdaki transfer fonksiyonu ile karakterize edilir:

$$H(z) = \sum_{n=0}^N a(n)z^{-n} \quad (1)$$

Burada N süzgeç derecesi, a(n) ise süzgeç katsayılarıdır.

Çalışmanın ikinci bölümünde APKD’ler hakkında bilgi verilerek APKD’lerin sayısal süzgeç tasarımında nasıl kullanıldığı konusu anlatılmıştır. Üçüncü bölümde APKD kullanılarak sayısal FIR süzgeç tasarlayabilmek için geliştirilen VHDL kodu anlatılmış ve sonuçlar yorumlanmıştır.

## 2. Alan Programlanabilir Kapı Dizisi (APKD)

Programlanabilir mantıksal elemanlar, bağımsız olarak oluşturulabilen mantıksal kapılar, kaydediciler ve bunları birbirine bağlayan programlanabilir bağlantılardan oluşmaktadır.

Bu elemanların ilki, 1975 yılında Monolithic Memories Inc. (MMI) tarafından geliştirilmiş olan PAL (Programmable Array Logic – Programlanabilir Dizi Lojik) elemanıdır[11]. PAL’ler de, Boolean fonksiyonları, De Morgan kuralları kullanılarak çarpımların toplamı yada toplamların çarpımı şeklinde gerçekleştirilebilirler. PAL yapısında girişlerin kendisi ve terslenmiş halleri AND kapılarına girmekte ve bu kapıların çıkışları da OR kapılarına uygulanmaktadır. Bu yapılar kullanılarak fonksiyon gerçekleştirilmektedir. PAL yapılarında OR kapıları sabit iken AND kapıları programlanabilir yapıya sahiptir.

Diğer bir eleman ise PLA (Programmable Logic Array – Programlanabilir Mantıksal Dizi)’dir. Bu elemanın PAL’lerden farkı ise çıkıştaki OR kapılarının da programlanabilir olması ve böylece daha büyük fonksiyonların gerçekleştirilebilmesidir. Bu yapılarda programlama işlemi bağlantı hatlarının koparılması yada bağlı kalması sağlanarak gerçekleştirilmektedir.

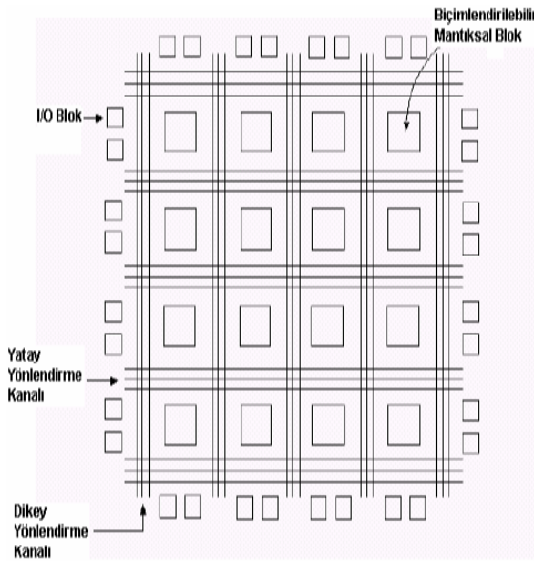
Daha karmaşık fonksiyonların gerçekleştirilebilmesi için PAL ve PLA’ların birleşiminden oluşan CPLD (Complex Programmable Logic Device – Karmaşık Programlanabilir Mantıksal Eleman)’ler geliştirilmiştir. PAL’ler bir programlayıcı ile programlanabilirken, CPLD’ler üreticinin geliştirdiği bir metotla veya bilgisayara bağlı JTAG kablo ile bir sistem yardımıyla programlanabilmektedirler. CPLD’ler de yüksek performanslı elemanlardır.

Geliştirilen alternatif bir programlanabilir mantıksal elemanda APKD’lerdir. Bu yapıyı Xilinx firması 1984 yılında geliştirmiştir[12]. Bu eleman daha esnek bir yapıya sahiptir. APKD’nin temel parçası LUT (Look Up Table – Hafıza Tablosu)’dür.

LUT bir fonksiyon jeneratörü gibi davranmakta yada bir ROM (Read Only Memory – Yalnızca Okunabilir Hafıza) veya RAM (Random Access Memory – Rasgele Erişilebilir Hafıza) olarak da ayarlanabilmektedir. Birbiri ile hızlı mantıksal bağlantısı olan hücreler içerdikleri için APKD’ler aritmetik fonksiyonları ve daha ileri DSP (Digital Signal Processing – Sayısal İşaret İşleme) uygulamaları için oldukça kullanışlıdır.

APKD’lerin çoğunluğu SRAM (Static Random Access Memory – Statik Rasgele Erişilebilir Hafıza) tabanlı olduğu için standart bir SRAM kadar kolay programlanabilirler. APKD’ler birçok yöntem ile biçimlendirilebilirler. Bu yöntemler şematik tasarım, HDL (Hardware Description Language – Donanım Tanımlama Dili)’ler ve daha üst seviyeli dil derleyicileridir. Her üretici firmanın bir yazılımı vardır ve bu yazılım ile hangi yöntem kullanılarak tasarım yapılmış ise o tasarımı APKD’yi programlayacak, yani APKD’ye yüklenebilecek hale dönüştürür. Şematik tasarım en kolay biçimlendirme metodudur. Yazılımda hazır bulunan elemanlar ve makrolar kullanılarak devre öncelikle şematik olarak oluşturulmaktadır.

APKD’lerin programlanması için kullanılan programlama dilleri HDL türü Verilog ve VHDL (Very High Speed Integrated Circuit Hardware Description Language - Yüksek Hızlı Tümlşik Devre Donanım Tanımlama Dili)’dir. Her ikisi de sayısal devrenin yazı tabanlı tanımlanmasıdır. Ayrıca C gibi üst seviyeli dillerle de biçimlendirme yapılabilir.



Şekil 1: APKD'nin Temsili İç Yapısı

Şekil 1’de bir APKD’nin iç yapısı verilmektedir[2]. Her bir biçimlendirilebilir mantık bloğu, kendi aralarında ve her bir I/O (Input/Output – Giriş/Çıkış) blok arasında yönlendirme kanalları aracılığı ile irtibatlandırılabilir.

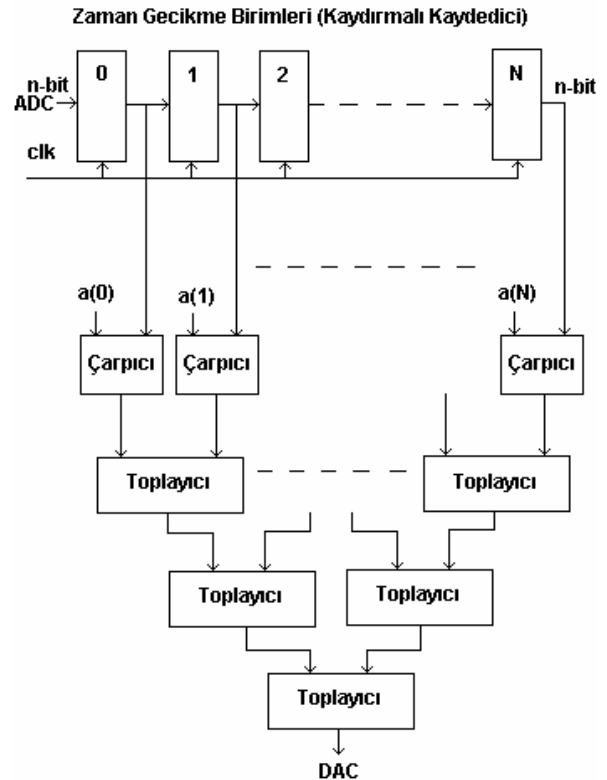
LUT yapıları mantıksal bloklar içinde bulunmaktadır. LUT, temel olarak hafıza birimleri ve bu hafıza birimlerinden hangisinin çıkışa aktarılacağını belirleyen bir çoğullayıcı yapıdan meydana gelmektedir.

APKD yapılarında birçok mantıksal eleman, temel hafıza birimleri ve kontrol edilebilir bağlantılı iletim yolları mevcuttur. Ayrıca sinyal işlemenin en temel fonksiyonları olan toplama, çıkarma ve gecikme fonksiyonları da kolayca gerçekleştirilebilir.

APKD ile sayısal süzgeç tasarımında, sayısal süzgecin giriş, çıkış ve katsayılarını belirleyen bitlerin sayısı arttıkça, kullanılması gereken elemanın boyutları da üstel olarak artmaktadır. Bu sorunu aşmak için, bit seri tasarım mantığı kullanılmaktadır. Böylece istenen süzgeç tek bir APKD üzerinde gerçekleştirilebilmektedir. Bu yöntemin paralel tasarıma göre olumsuz yönü; n-bit paralel tasarımın bir çevrimde yaptığı işi seri tasarım n çevrimde gerçekleştirebilmektedir.

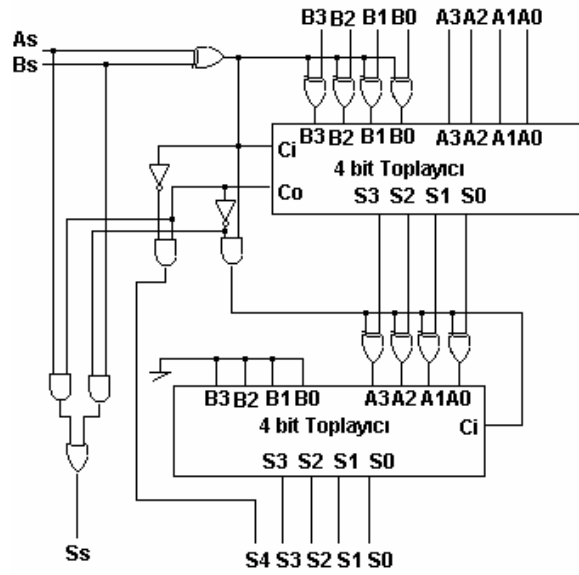
### 3. Sayısal Süzgeç Tasarımında VHDL Kodunun Kullanılması

Tasarımda sistemin hızı ön planda tutulduğu için paralel mantık kullanılmaktadır. Yapının temel parçaları aritmetik birimlerden meydana geldiği için, APKD’nin biçimlendirilmesinde şematik tasarım mantığı kullanılmıştır. FIR süzgecin oluşturulmasında kullanılan birimlerin genel bağlantısı Şekil 2’de görülmektedir.



Şekil 2: Sistemin Genel Yapısı

Zaman gecikme birimlerinde her bir blok n adet D tipi flip-floptan meydana gelmektedir. a(n) katsayıları da n-bittir ve sabittir. n-bitin en yüksek değerlikli olanı işaret bitidir. Bu nedenle çarpıcı ve toplayıcı birimleri bu dikkate alınarak tasarlanmıştır. Çarpıcının çıkışı  $2n-1$  bittir ve her bir toplayıcı katından sonra sistem çıkışı bir bit artmaktadır. Çarpıcı tasarlanırken şematik tasarım içinde hazır olarak bulunan dört bitlik çarpıcı modüllerden faydalanılarak n bitlik çarpıcılar elde edilmiştir. Süzgeç katsayılarının kesirli olmasının oluşturacağı problem de katsayıların, ADC (Analog Digital Convertor – Analog Dijital Çevirici)’nin adım büyüklüğü ile sınırlı ondalıkta seçilmesi ile aşılmıştır. Eğer hassasiyet ön planda tutulmakta ise o büyüklükte çözünürlüğe sahip bir DAC (Digital Analog Convertor – Dijital Analog Çevirici) kullanılmalıdır. Toplayıcı birimler, işaret bitlerine göre hem toplama hem de çıkartma işlemi gerçekleştirmektedir ve sonucun da gerçek değerini üretmektedirler. Her kademede bulunan toplayıcının bit sayısı bir üst kademeden bir bit fazladır. Şekil 3’de bu çalışmada kullanılan toplayıcı birimin yapısı görülmektedir.



Şekil 3: Toplayıcı Birim

Bu çalışmada ADC'nin çözünürlüğü  $n = 8$  bit olarak alınmış ve her birimin tasarımı da ona göre yapılmıştır.  $n = 8$  olarak alındığı için çarpıcı birimin çıkışı ( $2n-1=15$ ) 15 bit olmaktadır. Bundan dolayı çarpıcıdan hemen sonra gelen toplayıcı birim 15 bit olarak düzenlenmiştir ve bu birimi takip eden toplayıcılar 16, 17 ve 18 bitlidir. Çünkü toplayıcı birimlerin çıkışları giriş sayısının bir bit fazlası olmaktadır. DAC'ın çözünürlüğü 8-bit seçildiği için de 4 adet toplayıcı birimi gerekmektedir.

Sistem girişi n-bit iken sistem çıkışı  $2n$  den fazla olmaktadır. Bu sistemde ondalıklı ve tam kısmın daha az önemdeki bitleri atılarak bit sayısı n-bit'e indirilebilmektedir.

## 4. Sonuç

Bu çalışmada FIR tipi sayısal süzgecin APKD ile donanımsal olarak gerçekleştirilmesinde kullanılacak bir VHDL kodu geliştirilmiştir. Yani oluşturulan şematik tasarımdan her firmanın ürettiği APKD'yi programlamada kullanılabilecek VHDL kodu elde edilmiştir. Elde edilen kodlar Xilinx XC4000 ve bu APKD'nin alan kapasitesine sahip diğer APKD'leri programlamada kullanılabilmektedir. Tasarımda esnek bir yapı kullanıldığı için bit sayısı kolayca artırılabilir.

## 5. Kaynakça

- [1] Louzao J, Paz S, Tejera D, Bellora G, Langwagen G; Architectural desing of a programmable cell for the implementation of a filter bank on FPGA. *Microelectronics Reliability*; September 2003
- [2] R.e M, Cardarilli G. C, Re A. D, Lojacona R; FPGA Implementation of a Demux Based on a Multirate Filter Bank. *ISCAS 2000 – IEEE International Symposium on Circuits and Systems*; May 28-31, 2000, Geneva, Switzerland.
- [3] Lee H, Sobelman G. E; Performance evaluation and optimal design for FPGA-based digit-serial DSP functions. *Computers and Electrical Engineering*; January 1999.
- [4] Liu K. C, W. C. Lin and C. K. Wang; A Pipelined Digital Differential Matched Filter FPGA Implementation & VLSI Desing. *IEEE Custom Integrated Circuits Conference*, 1996.
- [5] Bates G. L, Nooshabadi S; FPGA Implementation of a Madian Filter. *IEEE TENCON – Speech and Image Technologies for Computing Telecommunications*, 1997.
- [6] Dick C, Haris F; High-Performance FPGA Filter Using Sigma-Delta Modulation Encoding. *IEEE International Conference On Acoustics, Speech, and Signal Processing*, 1999, Vol. 4, pp 2123-2126
- [7] Delma J. G. R., Reza A. M., Turney R. D.; FPGA Implementation of a Nonlinear Two Dimensional Fuzzy Filter. *IEEE International Conference On Acoustics, Speech, and Signal Processing*, 1999, Vol 4, pp 2143-2146
- [8] Delma J. G. R, Reza A. M., Turney R. D; FPGA Implementation of Adaptif Temporal Kalman Filter for Real Video Filtering. *IEEE International Conference On Acoustics, Speech, and Signal Processing*, 1999, Vol. 4, pp 2231-2234
- [9] Demirsoy, S. S., A. G. Dempster and I. Kale, "Transition analysis in multiplier-block based FIR filter structures", *IEEE International Conference on Electronic Circuits and Systems (ICECS 2000)*, Kaslik, Lebanon, December 2000.
- [10] Yamada M, Nishihara A; High-Speed FIR Digital Filter with CSD Coefficients Implemented on FPGA. *IEICE Trans. Fundamentals.*, E84-A, 8, pp.1997-2003, Aug. 2001
- [11] "Hardware Description Language." *Wikipedia: The Free Encyclopedia*. 18 May 2004

<http://en.wikipedia.org/>

[wiki/Hardware\\_description\\_language](http://en.wikipedia.org/wiki/Hardware_description_language)

- [12] Xilinx Inc., "Virtex-II Pro™ Platform FPGAs: Functional Description," DS083-2(v3.0), December 10, 2003.