

FPGA Üzerinde Yaklaşık FIR Süzgeç Tasarımı

Approximate Implementation of FIR Filters on FPGA

Y. Fırat Kula, Tuba Ayhan, Mustafa Altun
Elektronik ve Haberleşme Mühendisliği Bölümü
İstanbul Teknik Üniversitesi
34469 - İstanbul, Türkiye
{kulay, tuba.ayhan, altunmus}@itu.edu.tr

Özetçe—İşaret işleme uygulamalarında, sonuçların kesinliğinden feragat ederek, alan ve güç tüketimi azaltılabilir. Bu çalışmada, yaygın sayısal işaret işleme elemanlarından biri olan sonlu impuls yanıtı süzgeçler sahada programlanabilir kapı dizileri üzerinde yaklaşık olarak gerçekleştirilmiştir. Yaklaşık seviyesini belirleyen, ve farklı uygulamalara uyarlanabilir bir frekans yanıtı hatası tanımlanmıştır. Katsayıların bu hataya duyarlılığı sayısal olarak analiz edilmiştir. Bu analizin çıktısıyla bir çoklu sabitle çarpım optimizasyonu çalıştırılmış, böylelikle devrik mimaride gerçekleştirilen süzgeçlerin alanlarının küçültülmesi hedeflenmiştir. Önerilen duyarlılık analizi yöntemi ve geliştirilmiş çoklu sabitle çarpım optimizasyonu farklı derecelerdeki alçak geçiren ve bant geçiren süzgeçlerin gerçeklemede kullanılmıştır. Süzgeç karakteristiğinde 0.01 ve 0.001 mertebelerinde hataya müsamaha edildiğinde, sırasıyla %5-34 ve %3-28 ortalama alan kazancı sağlanmıştır.

Anahtar Kelimeler —Sayısal süzgeç, FIR, yaklaşık hesaplama FPGA.

Abstract—In signal processing applications, accuracy of the results can be traded-off for reducing area and power consumption. In this work, finite impulse response filters are designed approximately on FPGA. A frequency response error, which can be adjusted depending on the application, is defined to determine approximation level. Coefficient sensitivity is numerically analysed for this error. With the result obtained from this analysis, a multiple constant multiplication optimization method is executed, with the aim of reducing transposed filter structure areas. The proposed sensitivity analysis and the modified multiple constant multiplication optimization are used in collaboration, on lowpass and bandpass filter realizations. When filter characteristic error tolerances of 0.01 and 0.001 are introduced, an average area reduction of 5-34% and 3-28% are obtained, respectively.

Keywords —Digital filters, FIR, approximate computing, FPGA.

I. GİRİŞ

Sonlu impuls yanıtı (*Finite Impulse Response* - FIR) süzgeçlerden beklenen performans isterleri uygulamaya ve ortam koşullarına göre farklılıklar gösterir. Sayısal olarak gerçekleştirilen bir süzgecin kapladığı alan da isterlerden biridir. Alanı azaltabilmek için, süzgecin frekans yanıtının arzulan frekans yanıtından uzaklaşması göze alınarak, süzgecin tipi, derecesi ve katsayı büyüklükleri değiştirilebilir. Alan ve doğruluk isterlerini uzlaştırabilmek ilk olarak hedef isterlere

uygun süzgeç tipi belirlenir; çünkü her bir FIR yapısının kendine özgü avantajları vardır [1]. Örneğin, seyrek süzgeç katsayılarının çoğunluğu 0 olduğu için seyrek süzgeçlerin alan ve hız bakımından daha iyi olması beklenir. Ancak, seyrek süzgeçlerin Sahada Programlanabilir Kapı Dizileri (FPGA) gerçeklemede alan kazancının beklenenin altında olması gösteriyor ki, hedef gerçekleştirme platformunun özellikleri göz ardı edilmemelidir [2]. İkinci olarak, katsayı duyarlılıklarını en iyileştirmeye yönelik çalışmalar hem katsayıların saklandığı belleğin alanını küçültmeyi, hem de kesme (*truncation*) yoluyla çarpıcı ve toplayıcıların bit uzunluğunu azaltmayı sağlarlar [3]. Ancak, yakın zamanda oldukça detaylı çalışılmaya başlanan yaklaşık hesaplama teknikleri de alan ve güç tüketimini azaltırken, kesme yöntemine kıyasla daha az hesaplama hatasına sebep olmaktadır [4]. Kısaca, hedef gerçekleştirme platformuna uygun süzgeç tipi ve hesaplama yönteminin seçilmesi, süzgecin alanının küçültülebilmesi için ilk şarttır.

Süzgecin alanı küçültülürken, frekans yanıtının bozulması kaçınılmazdır. Bazı süzgeçlerin geçirme bölgesinde işaretin kazancını ve fazını en az etkilemesi beklenirken, bazıları için bastırma bölgesinde ne kadar etkili olduğu önemlidir. Örneğin, gürültü biçimlendirme ve yüksek hızda örnekleme kullanan analog-sayısal dönüştürücülere uygun süzgeç özellikleri [5]'te incelenmiştir. Dolayısıyla, frekans yanıtındaki bozulma hedef uygulamalara yönelik olarak şekillendirilmelidir.

Bu çalışmada, frekans yanıtı uygulamaya göre kritik olan frekans aralığında beklenene en yakın süzgeç en küçük alanı kaplayacak şekilde gerçekleştirilmiştir. FPGA üzerinde devrik FIR mimarisinde süzgeç gerçekleştirilmiş ve süzgecin kullandığı LUT (*Lookup Table*) sayısı raporlanmıştır. Uygulamaya göre, frekans yanıtı farkının ağırlıklandırıldığı bir hata ölçütü tanımlanmıştır. Süzgeç katsayıları belirlenirken, hem katsayıların hata ölçütü üzerindeki etkisi hem de katsayıların alana olan etkileri göz önüne alınmıştır.

Alanı küçültmek için kullanılmış olan yöntemin kısa bir açıklaması Bölüm II'de verilmiştir. Farklı uygulamalar için yeniden üretilebilir hata ölçütü Bölüm III'te tanımlanmıştır. Ardından aynı bölümde anlatılan sayısal analiz ile süzgeç katsayıları tanımlanan hata ölçütüne olan katkılarına göre gruplanmıştır. Farklı tipte süzgeçler çeşitli hata ölçütü tanımları ve büyüklükleri için optimize edilmiş, Xilinx Spartan 3E üzerinde gerçekleştirilen devrelerin büyüklükleri Bölüm IV'te verilmiştir. Son olarak Bölüm V'te sonuçlar değerlendirilmiş ve gelecek çalışmalar tartışılmıştır.

II. FIR SÜZGEÇ İÇİN LUT BAZINDA ALAN OPTİMİZASYONU

Devrik FIR mimarisinde devrenin girişine gelen işaret, farklı sabit sayılar ile çarpılıp toplanır. Bu sebeple bu çalışmada LUT sayısını en azlamak için Çoklu Sabitle Çarpım (*Multiple Constant Multiplication* - MCM) optimizasyonu yapan THETIS [6] algoritması geliştirilerek kullanılmıştır. THETIS'te çok sayıda sabitle çarpım işlemlerinde alanı azaltmak için tutulan yol, katsayıları ikinin kuvvetlerine getirmek ve katsayılar arasında ortak kullanılan çarpanlar elde etmektir. İkinci kuvvetleri ile çarpmak donanımsal olarak bit kaydırmak anlamına geldiğinden bu işlem için gereken alan oldukça küçüktür. Ayrıca çarpan kümesi içindeki sayılar arasında ortak çarpanların sayısı arttıkça, nihai olarak harcanması gereken alan da azaltılmış hale gelir; çünkü herhangi bir ortak çarpanın sonucunu içeren devre bloğunun çıkışının, bu ortak çarpanı içeren tüm çarpma işlemlerine bağlanması hesaplama için yeterli olacaktır.

Örnek olarak $K = \{k_1, k_2, k_3\} = \{6, 25, 34\}$ şeklinde bir sabit çarpanlar kümesi ele alınsın. Katsayılar için verilen müsamaha edilebilir hata da $E = 1$ olarak verilsin. Başlangıç durumunda K kümesindeki elemanların içlerindeki ikinin kuvvetlerine ayrılmış halleri sırasıyla 3, 25 ve 17 olup bu sayıların oluşturduğu birbirinden farklı çarpanlar kümesinin $K_A = \{3, 25, 17\}$ olduğu görülmektedir. Dolayısıyla bu devreyi gerçeklemek için 3, 25 ve 17 sayılarının girişe gelen belirli uzunlukta bir sayı ile çarpımını hesaplayan devre bloklarına ihtiyaç duyulacaktır. K tamsayı kümesindeki her bir farklı eleman için katsayı adayları kümeleri $k_{nA} = [k_n - E, k_n + E]$ şeklinde ifade edilir. Örnek K kümesindeki elemanlar için katsayı adayları; $k_{1y} = \{5, 6, 7\}$, $k_{2y} = \{24, 25, 26\}$, $k_{3y} = \{33, 34, 35\}$ olarak bulunacaktır ve bu üç küme içinden de özgün katsayıların yerine koymak için birer katsayı seçilecektir. Yöntem, bu katsayı adayları kümelerinin içinden hangi elemanların seçildiği takdirde azami alan kazancı sağlanacağını bulmaktadır. Bu durumda aday kümeler içinden sırasıyla 6, 34 ve 24'ün seçildiği görülmüştür. Bu çarpanlar için alternatif katsayılar kümesi $K' = \{k'_1, k'_2, k'_3\} = \{6, 24, 34\}$ için de her elemanın ikinin kuvvetlerinden ayrılmış halde yazımı sırasıyla 3, 3 ve 17 olup sadece girişin 3 ve 17 ile çarpımını üreten devre bloklarını bulundurması yeterli olacaktır. Burada 3'ün ortak çarpan haline getirildiği görülmektedir. Ayrıca 17 sayısı ile çarpım için gereken LUT sayısı, diğer katsayı adaylarının kullanacağı çarpanlara ait devrelerden daha küçük olmaktadır [7]; bu nedenle THETIS tarafından öncelikle tercih edilir.

Özgün algoritmada her katsayı için geçerli olan tek bir E hatası belirlenebiliyorken, bu çalışmada söz konusu FIR süzgecin katsayıları üzerinde bir hata duyarlılığı analizi yapılarak bu analiz sonucunda elde edilen veriler ışığında her katsayıya farklı miktarda hata değerleri verebilmek amaçlanmıştır. Bununla birlikte algoritmanın süzgecin önemli karakteristiklerini bozmadan daha da yüksek alan kazancı sağlayabileceği öngörülmüştür.

III. SÜZGEÇ KATSAYILARI İÇİN HATA ANALİZİ

Herhangi bir n . dereceden FIR süzgece ait katsayılar $R \subset H$ olmak üzere $H = \{h_1, h_2 \dots h_{n+1}\}$ şeklinde bir H sıralı kümesi ile ifade edilebilir. Bu kümenin eleman sayısı $n+1$ 'dir ve kümenin ilk yarısındaki elemanlar ile ikinci yarısındaki elemanlar her

zaman birbirinin simetridir. Algoritmaya eklenecek bu hata analizi bölümünde, her süzgeç katsayısının nihai frekans yanıtı üzerindeki etkisinin incelenmesi hedeflenip her katsayı sırasıyla bireysel olarak bir miktar hataya maruz bırakılmış, bu hatanın süzgecin frekans yanıtındaki önemli karakteristikleri ne miktarda etkilediği incelenmiştir. Bunun için öncelikle frekans yanıtının önemi yüksek bölümlerindeki bozulmaları ön plana çıkaracak bir hata ölçütü tanımlanması gerekmektedir. Ardından, bu hata ölçütü esas alınarak yapılacak tekrarlamalar sonucunda elde edilen hata toleransı bilgisine göre H kümesi, H_x şeklinde ($x \in Z^+$) hata duyarlılığına göre oluşturulmuş alt kümelerle ayrılmalıdır. Her H_x alt kümesi için farklı birer hata sınırı seçilmelidir.

Analiz sonunda H katsayılar kümesine ait tüm elemanlar H_x alt kümelerine dağıtılmış olmalıdır ve bir katsayıyı temsil eden H kümesi elemanı birden fazla H_x kümesinde bulunamaz. Bu katsayı gruplandırmasının sonucunda, süzgeç devresi yaklaşık hale getirilirken frekans yanıtında yüksek önem verilen bölümü hatırı sayılır ölçüde etkileyen katsayı gruplarına verilecek hata sınırı düşük tutulabilecek, nispeten düşük miktarda etkileyen gruplara daha esnek hatalar verilebilecektir. Devrenin yaklaşık sentezlenmesinde böyle bir yöntem izlemenin, süzgeç karakteristiğinde ciddi bozulmalar yaşamadan harcanan alanı daha da düşüreceği öngörülmüştür. Eklenen hata analizi algoritması aşağıdaki adımlarla detaylandırılabilir.

1) Uygulamaya göre, süzgeç frekans karakteristiğinin önemli bölgeleri için aşağıda görüldüğü gibi bir ϵ ağırlık fonksiyonu belirlenir.

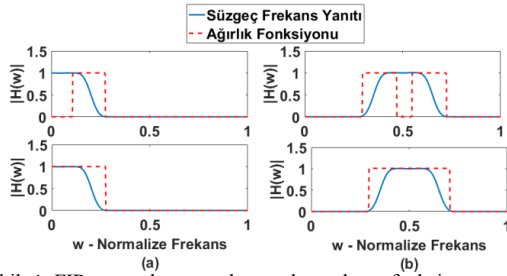
$$\epsilon(f) = \begin{cases} 0, & f < f_{k1} \\ 1, & f_{k1} \leq f \leq f_{k2} \\ 0, & f > f_{k2} \end{cases} \quad (1)$$

Burada f frekansı, f_{k1} ve f_{k2} ise süzgecin frekans yanıtında yüksek önem verilen bölümün sırasıyla başlangıcını ve sonunu gösteren frekans değerleridir. Bu hata ölçütü fonksiyonu, (1)'de verilen yapıdaki birkaç farklı fonksiyonun üstüdüümü şeklinde de seçilebilir. Duyarlılık analizinde kullanılacak hata ölçütü ifadesi (2)'deki gibi alınmıştır.

$$N = \frac{\sum |H(f) - \hat{H}(f)| \epsilon(f)}{\sum \epsilon(f)} \quad (2)$$

Burada $H(f)$ ve $\hat{H}(f)$ sırasıyla, süzgecin beklenen ve yaklaşık katsayılarla hesaplanan frekans yanıtını vermektedir. Bu iki fonksiyon arasındaki fark $\epsilon(f)$ ile çarpılmış, önemli frekans bandının hata ölçütünü doğrudan etkilemesi sağlanmıştır. Ayrıca, hata ölçütü $\epsilon(f)$ üzerinden normalize edilmiştir. Böylece, ağırlık fonksiyonu değiştirildiği zaman, hata ölçütü N 'nin sayısal aralığı merteye değiştirmez.

Örneğin Şekil 1'a'da keyfi seçilmiş bir alçak geçiren süzgecin sadece geçiş bandındaki bozulmayı dikkate alacak bir ağırlık fonksiyonu görülmektedir. Şekil 1b'de ise yine keyfi seçilmiş bir bant geçiren süzgecin hem geçiş bantlarını hem de geçirme bandını ön planda tutup söndürme bandında olacak hataları önemsiz tutan bir ağırlık fonksiyonu tercih edilmiştir. Hata ölçütünde kullanılacak olan bu fonksiyon, süzgeç türüne ve uygulamaya göre, (1) için anlatılan biçime uyacak şekilde belirlenebilir.



Şekil 1. FIR süzgeçlere uygulanmış hata ölçütü fonksiyonu örnekleri, (a) alçak geçiren süzgeç (b) bant geçiren süzgeç için birinci (üstte) ve ikinci (altta) ağırlık fonksiyonları.

2) Bu aşamada, bir eklenecek hata değeri aralığı ve bu aralığa ait adım seçilerek bir hata değeri kümesi oluşturulur. Süzgeç katsayılarından sadece biri hatalı, diğerleri gerçek değerinde bulunacak şekilde tüm katsayılara sırayla bu hata kümesindeki elemanlar tek tek eklenir. Bunun sonucunda birinci adımda tanımlanan hata ölçütünün değerinin her bir iterasyon ve katsayı için nasıl etkilendiğine dair eğriler elde edilir. Şekil 2'de, bulunan bu eğrilere ait bir örnek grafik görülmektedir. Grafik anahtarında belirtilmemiş olan tüm eğriler, süzgece ait her bir katsayının verilen hata miktarına göre hata ölçütünü hangi değere getirdiğini göstermektedir.

3) Bir hedef nihai hata seçilir. İkinci adımda elde edilmiş olan, katsayı duyarlılığını gösteren eğriler bu nihai hata değeri civarında doğrusallaştırılır ve buradan nihai hata ölçütünü sağlayan katsayı hataları bulunur. Bunun sonucunda her bir katsayıya verilebilecek uygun hatalar kümesi elde edilir.

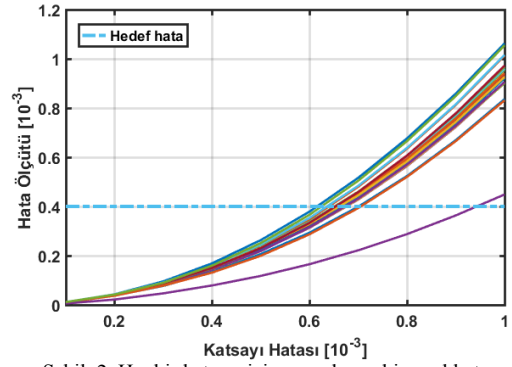
4) Önceki adımda elde edilen her katsayının müsamaha edebileceği nihai hata bilgisinden yola çıkarak katsayılar kümelendirir. Kaç katsayı grubu oluşturulmasının ideal olacağını anlamak için, k-ortalımalı kümelemeler denenmiştir. Küme sayısı 2'den $\lceil (n + 1)/2 \rceil$ değerine kadar arttırılarak, her bir elemanın küme merkezine olan mesafesini gösteren grafik elde edilir. Küme sayısının bu $\lceil (n + 1)/2 \rceil$ değerine kadar arttırılma nedeni, bu değerden sonraki katsayıların bu değerden önceki süzgeç katsayılarının simetrisi olmasıdır.

5) Dördüncü adımda elde edilen grafikteki (Şekil 3) kritik nokta, başka bir deyişle grafiğin kırılma noktası, en uygun kümeleme sayısını verecektir. Bu nokta, grafikteki her bir noktanın ikişer komşuluğu ile türev hesaplanması ve türev sonucunun -1 veya -1'e çok yakın olup olmadığına bakılması ile bulunur. İlgili kümenin ortalaması da, genel algoritma içinde bu katsayı grubuna uygulanacak hata sınırını vermektedir. Şekil 3'te grafik anahtarında belirtilenin haricindeki eğriler, belirli kümeleme sayısına denk gelen noktadaki türevleri belirtir.

6) Önceki adımda bulunan en uygun küme sayısı bilgisine göre yapılan gruplama ve bunlara ait hata miktarları, optimizasyonda kullanılmak üzere seçilir ve bu verilere göre en uygun yaklaşık katsayıları bularak bunlara ait FIR süzgeç devrelerini Verilog dilinde otomatik olarak üretilir.

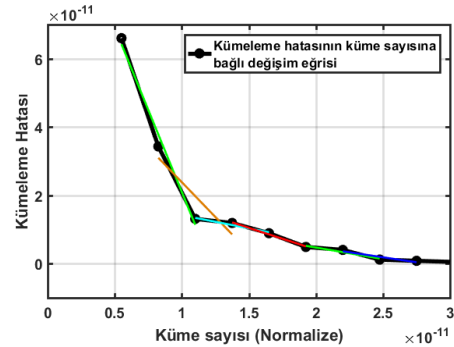
IV. DENEYSEL VERİLER

Deneysel verilerin elde edilişi için öncelikle iki adet farklı türde FIR süzgeç, farklı süzgeç dereceleri için tasarlanıp, her biri için elde edilen süzgeç katsayıları önerilmiş olan duyarlılık analizinden geçirilmiştir. Ardından, analiz sonuçları ve süzgeç



Şekil 2. Her bir katsayı için, uygulanan bireysel katsayı hatası ile hata ölçütü değerinin ilişkisini gösteren eğrilere örnek.

katsayıları sırayla, düzenlenmiş THETIS tabanlı uygulamaya girdi olarak verilmiştir. Ayrıca her süzgeç türüne ait dereceler için değişik ağırlık fonksiyonları kullanan ikişer farklı hata ölçütü tanımı ele alınmıştır. Bu ağırlık fonksiyonları biçim olarak Şekil 1'de verilmiştir: Birinci hata ölçütü sadece süzgeçlerin geçiş bandında 1 değerini almakta, ikincisi ise hem geçiş hem geçirme bantlarında 1 değerini almaktadır. Çıktı olarak elde edilen süzgeç devreleri Xilinx Spartan3E XA3S500E cihazında dört girişli LUTlar kullanılarak sentezlenmiştir. Sentezlenen test devrelerine ait sonuçlar Tablo 1'de verilmiştir. Bu tabloda nihai hata sütunlarında görülen "Hatasız", "Düşük" ve "Yüksek" ibareleri sırasıyla gerçek katsayılı (8-bit) süzgeç ile 10^{-3} ve 10^{-2} mertebelerinde nihai hataya sahip süzgeçleri belirtmektedir. Kesim frekansları ve örnekleme frekansı da sırasıyla f_c ve f_s ile verilmiştir. Elde edilen alan kazancı incelendiğinde özellikle düşük derecelerde LUT sayısında hatırı sayılır ölçüde azalma elde edildiği görülmektedir. Yüksek derecelerde gerçek filtre tarafından kullanılan LUT sayısına oranla yüzdesel olarak düşük derecelere göre daha az LUT sayısı azalışı görülmektedir. Bunun nedeni, süzgecin derecesi yani devrede kullanılan katsayı sayısı arttıkça hata sınırının altında kalarak çok sayıda katsayıyı aynı ortak çarpanlar kullanacak şekilde yaklaşılaştırmanın zorlaşmasıdır. Alan sonuçlarına ek olarak, Şekil 4'te elde edilen yaklaşık süzgeç katsayıları sonucunda gerçek hatanın hedef hata ve süzgeç derecesi ile nasıl değiştiğine dair yapılan deneylerden bir örnek gösterilmiştir.



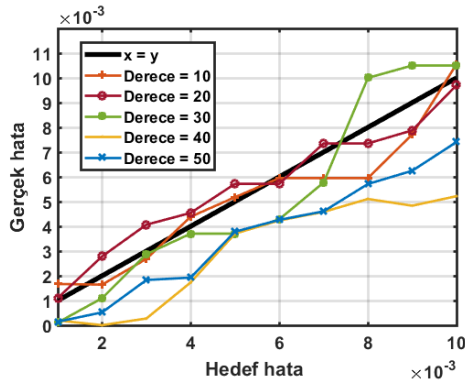
Şekil 3. Kümeleme hatasının küme sayısına bağlı değişim eğrisinin küme sayılarına göre türevleri.

TABLO I. FPGA ORTAMINDA DENEY SONUÇLARI

Süzgeç Türü 1: Alçak Geçiren, $f_c = 1$ kHz, $f_s = 10$ kHz						Süzgeç Türü 2: Bant Geçiren, $f_{c1} = 2$ kHz, $f_{c2} = 3$ kHz, $f_s = 10$ kHz							
Süzgeç Derecesi	1. Hata Ölçütü			2. Hata Ölçütü			Süzgeç Derecesi	1. Hata Ölçütü			2. Hata Ölçütü		
	Nihai Hata	LUT Sayısı	LUT Kazancı	Nihai Hata	LUT Sayısı	LUT Kazancı		Nihai Hata	LUT Sayısı	LUT Kazancı	Nihai Hata	LUT Sayısı	LUT Kazancı
10	Hatasız	299	-	Hatasız	299	-	20	Hatasız	572	-	Hatasız	572	-
	Düşük	168	%43.8	Düşük	296	%1.0		Düşük	483	%15.5	Düşük	483	%15.5
	Yüksek	164	%45.1	Yüksek	193	%35.4		Yüksek	436	%23.7	Yüksek	455	%20.4
20	Hatasız	583	-	Hatasız	583	-	40	Hatasız	1339	-	Hatasız	1339	-
	Düşük	462	%20.7	Düşük	581	%0.3		Düşük	1306	%2.4	Düşük	1297	%3.13
	Yüksek	396	%32.0	Yüksek	521	%10.6		Yüksek	1217	%9.11	Yüksek	1204	%10.1
30	Hatasız	971	-	Hatasız	971	-	60	Hatasız	2573	-	Hatasız	2573	-
	Düşük	776	%20.0	Düşük	966	%0.5		Düşük	2461	%4.3	Düşük	2562	%0.4
	Yüksek	736	%24.2	Yüksek	954	%1.7		Yüksek	2425	%5.7	Yüksek	2392	%7.0
40	Hatasız	1323	-	Hatasız	1323	-	80	Hatasız	4083	-	Hatasız	4083	-
	Düşük	1231	%6.9	Düşük	1314	%0.6		Düşük	4040	%1.1	Düşük	4045	%0.9
	Yüksek	1189	%10.12	Yüksek	1288	%2.6		Yüksek	3990	%2.2	Yüksek	3980	%2.5
50	Hatasız	1933	-	Hatasız	1933	-	100	Hatasız	6146	-	Hatasız	6146	-
	Düşük	1775	%8.1	Düşük	1907	%1.3		Düşük	6102	%0.7	Düşük	6102	%0.7
	Yüksek	1763	%8.7	Yüksek	1880	%2.7		Yüksek	5986	%2.6	Yüksek	5951	%3.1

V. SONUÇ

Bu çalışmada MCM işlemi içeren sayısal süzgeç yapıları için belirlenen miktarda hataya tabi olarak devre alanını küçültme konusuna dikkat çekilmiştir. Bu işlem için yapılmış olan çalışmalardan birinden yola çıkılarak alan optimizasyonu için yeni bir bakış açısı olarak, katsayı hassaslıklarını da çözümlenerek alan indirilmesi yapan bir geliştirme sunulmuştur. Sonuçta, düzenlenmiş olan yeni algoritmanın devrelerde kullanılan LUT sayılarını küçük ve orta dereceli süzgeçler için önemli ölçüde azaltabildiği; bununla birlikte süzgeç frekans yanıtındaki bozulmanın uygulamada önem verilen bölgeleri korumaya öncelik de vererek, yaklaşık süzgeç hatasını istenilen hata sınırının altında veya komşuluğunda tuttuğu görülmüştür (Şekil 4). Alçak geçiren süzgeçte alçak ve orta süzgeç dereceleri için %20 - %45 aralığında, yüksek dereceleri için %2 - %10 aralığında; bant geçiren süzgeç örneklerinde düşük ve orta dereceli süzgeçlerde %5 - %23, yüksek dereceleri için %1 - %3 oranlarına alan kazancı elde edildiği sonuçlardan görülmektedir. İkinci hata ölçütü tanımında hem geçirme hem de geçiş bantları önemli addedildiği için,



Şekil 4. Tablo I'de kullanılan süzgeç türü 1 ve hata ölçütü 1 için hedeflenen hata ve yaklaşık katsayılar sonucunda bulunan hatanın kıyaslanması

özellikle alçak geçiren süzgeç için alan kazancının sınırlandırıldığı görülmektedir.

Gelecekteki çalışmalarda, önerilmiş olan hassaslık analizini sayısal halden tam analitik hale geçirmek hedeflenmektedir. Ayrıca hız, boyut ve güç tüketiminin yüksek doğruluklu sonuçlardan daha önemli olduğu konvolüsyonel sınır ağları gibi uygulamalarda test etmek de olası yapılabilecek çalışmalar arasındadır.

BİLGİLENDİRME

Bu çalışma 117E078 numaralı TÜBİTAK projesi tarafından desteklenmiştir.

KAYNAKLAR

- [1] Schlichthärle D., *Digital Filters: Basics and Design*, Springer Publishing, 2011.
- [2] Patronis S.G. and L. S. DeBrunner, "Sparse FIR filters and the impact on FPGA area usage," *42nd Asilomar Conference on Signals, Systems and Computers*, pp. 1862-1866, 2008.
- [3] Constantinides G. A., Cheung P. Y. K. and W. Luk, "Wordlength optimization for linear digital signal processing," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 10, pp. 1432-1442, Oct. 2003.
- [4] Han J. and Orshansky M., "Approximate computing: An emerging paradigm for energy-efficient design," 2013 18th IEEE European Test Symposium (ETS), pp. 1-6, 2013.
- [5] Brian Pisani, Application Report, SBAA230–May 2017, Digital Filter Types in Delta-Sigma ADCs, Texas Instruments, 2017, <http://www.ti.com/lit/an/sbaa230/sbaa230.pdf>, erişim: 13.02.2018
- [6] Aksoy L., Flores P., Monteiro J., "Approximation of Multiple Constant Multiplications Using Minimum Look-Up Tables on FPGA", *ISCAS, IEEE International Symposium*, pp 2884-288, 2015.
- [7] Wirthlin M., "Constant Coefficient Multiplication Using Look-Up Tables," *J. of VLSI Signal Processing*, vol. 36, no. 1, pp. 7–15, 2004.