Kısıtlanmış 1-Bit Dönüşümü Temelli Hareket Kestirimi Algoritmasının HVTDD Yaklaşımıyla Tasarımı

MVBLA Based Design of Constrained 1-Bit Transform Based Motion Estimation Algorithm

Anıl ÇELEBİ[1], Oğuzhan URHAN[1], Sarp ERTÜRK[1], İlker HAMZAOĞLU[2], Günhan DÜNDAR[3]

1. Elektronik ve Haberleşme Mühendisliği Bölümü
Kocaeli Üniversitesi, Kocaeli
anilcelebi@kou.edu.tr, urhano@kou.edu.tr, sertur@kou.edu.tr

2. Elektrik-Elektronik Mühendisliği Bölümü
Sabancı Üniversitesi, İstanbul
hamzaoglu@sabanciuniv.edu

3. Elektrik-Elektronik Mühendisliği Bölümü
Boğaziçi Üniversitesi, İstanbul
dundar@boun.edu.tr

Özetçe

Bu çalışmada blok uyumlama temelli hareket kestirimi işleminin gerçek zamanlı yapılabilmesini sağlamak amacıyla özgün bir donanım önerilmiştir. Tasarlanan sistem genel amaçlı bir FPGA yongası üzerinde çok az yer kapladığı için tüm bir video kodlama algoritmasının tek bir FPGA ile gerçekleştirilmesi mümkün olabilmektedir. Tasarlanan sistem 50MHz saat hızında çalışabilmekte ve 2048×1152 piksel boyutundaki imgeler için yaklaşık 20 çerçeve/saniye hızında hareket vektörü üretebilmektedir.

Abstract

In this work a novel hardware proposed for Constrained 1-bit Transform based motion estimation to facilitate real time operation. The designed system occupies a small area in a general purpose FPGA fabric and it is therefore efficient to implement a whole video coding architecture on a single FPGA chip. The designed system can perform ME operation for a 2048×1152 pixel sized image frame at a speed of 20 frames/second.

1. Giriş

Video kodlama uygulamalarında hesapsal işlemlerin yaklaşık %50' si hareket kestirimi işlemi esnasında yapılmaktadır. Özellikle yaygın bir standart haline gelen H.264/AVC etkin video kodlama için hareket kestiriminin alt-piksel doğrulukta ve birden fazla çerçeve üzerinde yapılmasını önermektedir [1]. Böylelikle hareket kestiriminin işlem yükü daha da artmış ve hızlı hareket kestirimi yaklaşımlarının kullanılması kaçınılmaz hale gelmiştir.

Literatürde uyumlama ölçütü olarak mutlak farklar toplamı (SAD) temelli hareket kestiriminin donanımsal olarak yapılmasını öneren yaklaşımlar oldukça yaygındır[2, 3, 4, 5]. Özellikle donanımsal uygulamalar için daha uygun uyumlama ölçütü olan mantıksal EX-OR işlemini kullanan 1-bit dönüşümü [6], 2-bit dönüşümü [7], çarpmasız 1-bit dönüşümü [8] ve kısıtlanmış 1-bit dönüşümü [9] gibi yaklaşımlar yakın geçmişte önerilmiştir. Bu çalışmada donanımsal olarak sistem kaynaklarını daha etkin kullanan ve daha düşük güç tüketimine sahip bir yapı elde edilmesine olanak sağlayan ve bununla birlikte başarımı oldukça yüksek olan kısıtlanmış 1-bit dönüşümü tekniği kullanılmış ve bu yöntem özgün olarak FPGA üzerinde gerçeklenmiştir.

2. Kısıtlanmış 1-Bit Dönüşümü

Hareket kestirimi yöntemleri büyük çoğunlukla blok uyumlaması temelli çalışır. Uyumlama ölçütü olarak mantıksal EX-OR işlemini kullanılmasını öneren ilk çalışmada 1-bit dönüşümü yapılırken blok ortalaması eşik olarak alınmış ve bit-düzlemini elde etmek için piksel değerleri bu eşikle karşılaştırılmıştır [6]. [10]' deki çalışmada ise 1-bit dönüşüm ile hareket kestirimi yapmak için eşik kullanmak yerine süzgeçlenmiş imge çerçevesi ve orijinal imge çerçevesi pikselleri arasında bir karşılaştırma yapılmaktadır ve dönüşüm bu karşılaştırma sonucuna göre gerçekleştirilmektedir. [10]' deki çalışmada süzgeçleme yapılırken gerçekleştirilen çarpma işlemleri hesapsal yükü büyük ölçüde arttırmaktadır, bu nedenle [8]' de yapılan çalışmada bu problemi ortadan kaldırmak için elmas şekilli özgün bir çekirdek yapısı önerilmiştir.

1-bit dönüşüm yöntemlerinin genel bir sorunu 1-bit dönüşümü yapılırken eşik değerine çok yakın fakat eşiğin altında ve üstünde kalan iki pikselin farklı sınıflarda kabul edilmesidir. Bu sorunun ortadan kaldırılması için [9]' de bu çalışmada da baz alınan kısıtlanmış 1-bit dönüşümü yapısı önerilmiştir. Bu yapıda 1-bit dönüşümü hesaplanırken eşik haricinde bir kısıt maskesi kullanılmaktadır. Bu kısıt maskesi de piksellerin eşiğe kabul edilebilir bir uzaklıkta olup olmadığına bakılarak oluşturulmaktadır ve ilgili piksellerin 1-bit dönüşümünde dikkate alınıp alınmayacağına karar verilmesi için kullanılmaktadır. Şekil 1' de 1-bit dönüşümünde karşılaşılan hata ve kısıtlanmış 1-bit dönüşümü yaklaşımının temel mantığı özetlenmektedir.

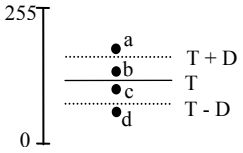
Kısıtlanmış 1-bit dönüşümü yönteminde imgelerin 1-bit dönüşümü (1)' de görüldüğü şekilde hesaplanmaktadır.

978-1-4244-1999-9/08/$25.00 @2008 IEEE

$$B(i,j)=\begin{cases}1, & \text{if } I(i,j)\geq I_F(i,j)\\ 0, & \text{diğer}\end{cases} \quad (1)$$

Burada $I(i,j)$ gerçek imge çerçevesindeki, $I_F(i,j)$ ise süzgeçlenmiş imge çerçevesindeki piksel değerlerini belirtmektedir. Kısıtlanmış 1-bit dönüşümünde kısıt maskesi ise (2)' deki ifadeye göre hesaplanır.

$$CM(i,j)=\begin{cases}1, & \text{if } \left|I(i,j)-I_F(i,j)\right|\geq D\\ 0, & \text{diğer}\end{cases} \quad (2)$$



Şekil 1: 1-bit dönüşümü ve kısıtlanmış 1-bit dönüşümü yönteminin çalışma prensibi

Burada gerçek imge çerçevesi ile süzgeçlenmiş imge çerçevesinin mutlak farkının önceden belirlenmiş bir tolerans (eşik) değerinden büyük olup olmadığına bakılarak karar verilir. Hareket vektörünün bulunması için kullanılan kısıtlanmış uyumsuz nokta sayısı ifadesi de (3)' deki eşitlik ile elde edilir.

$$CNNMP(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left\{\begin{bmatrix}CM^t(i,j)\,||\,CM^{t-1}(i+m,j+n)\end{bmatrix}\right\}\\ \left\{\odot\begin{bmatrix}B^t(i,j)\oplus B^{t-1}(i+m,j+n)\end{bmatrix}\right\} \quad (3)$$

$$-s\leq m,n\leq s-1$$

Burada elde edilen en küçük değeri veren konum değişimi hareket vektörü olarak alınır. Şekil 1' de görülen b ve c pikselleri söz konusu olduğunda iki piksel değeri de toleranslar arasında olduğundan kısıt maskesi ifadesinden "0" geleceği için bu iki değerin kısıtlanmış uyumsuz nokta sayısına etkisi yoktur yani bu iki piksel arasındaki mantıksal EX-OR ilişkisi sonuca etki etmez.

3. Sistem Mimarisi

Geleneksel blok uyumlama temelli hareket kestiriminde işlemler 8-bit derinliğindeki imge çerçevelerinde gerçekleştirilir ve iki blok arasındaki mesafe en küçük mutlak fark veya en küçük ortalama karesel hata ölçütleri ile belirlenir. Bu ölçütlerin elde edilmesi kullanılan arama yönteminden bağımsız olarak donanım karmaşasını artırırlar. Bir-bit dönüşümü yöntemi ile bu karmaşa önemli bir miktarda düşürülebilir.

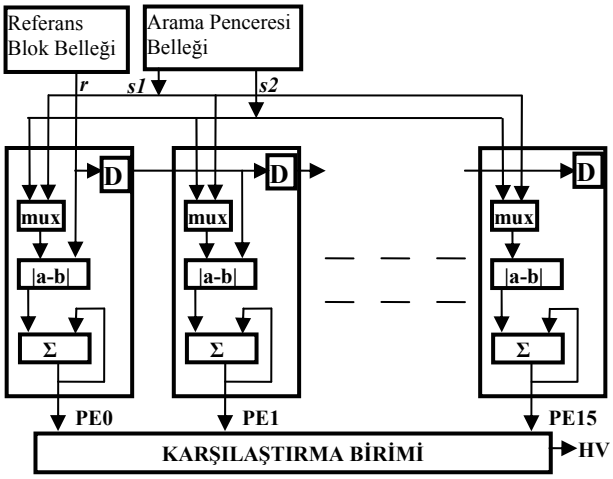
3.1. Doğrusal Diziler kullanarak blok uyumlama

16x16 boyutunda bir blok için [-8,7] aralığında bir arama işlemini ele alalım. Blok uyumlama işlemi sırasında arama penceresinin dört köşesinde bulunan pikseller sadece bir kez kullanılır ancak diğer pikseller farklı arama bölgelerinde birden fazla sayıda kullanılabilirler. Bu özellik kullanılarak paralel işlem yapma olanağı yakalanabilir.

Geçmişte bu işlemi yapmak için iki farklı yaklaşım benimsenmiştir [11]:

1. Hareket vektörü temelli doğrusal dizilerin kullanılması yöntemi.
2. Kaynak piksel temelli doğrusal dizilerin kullanılması yöntemi.

[12]' de hareket vektörü temelli doğrusal diziler kullanılarak hareket kestirimi yapılabileceği önerilmiştir. Şekil 2' de [12]' de önerilen, 16 işlem biriminden oluşan bir boyutlu bir dizi işlemcisi görülmektedir. Yapıdaki her bir işlem bloğu tek bir mutlak fark alma işlemini gerçekleştirebilir böylelikle aynı anda 16 ayrı mutlak fark alma işlemi gerçekleştirilebilir. 256 adet mutlak fark alma işleminin tamamlanabilmesi için referans blok ardı ardına 16 kez taranmalı ve işlenmelidir. Bu işlemin sonunda 16 farklı blok konumu için hareket vektörü değerleri hesaplanmış ve bu 16 vektörü içindeki en olası değer bulunmuş olur. Bu yapıda verilerin akışının nasıl olduğu Tablo 1' de açıkça görülmektedir.



Şekil 2. Hareket kestirimi sisteminin on altı adet işlem bloğundan oluşan mimarisi

Tablo 1'e bakıldığında her bir saat çevriminde *r* girişinden gelen verinin işlemciler arasında gezdiği kolaylıkla görülebilir. Bundan biraz farklı olarak *s* girişinden gelen veri her saat çevriminde değişmektedir fakat işlemcilere farklı şekilde dağıtılmaktadır. Bu dağıtım şeklinin sebebi Şekil 2'deki devreye bakıldığında görülebilir. Şekilde, *referans çerçeve* bloğundan tek bir çıkış alınmakta ve buradan alınan veri *D* tipi yaz-bozlar yardımı ile işlemciler arasında tek yönde ilerlemektedir. Bu sayede her işlemci 256 saat darbesi boyunca arama penceresindeki farklı konumlara ait *mutlak farklar toplamı* değerini bulur. Bir *mutlak farklar toplamı* değeri için toplamda 256 çevrim gerektiği halde on altı tane paralel çalışan işlem öbeği sayesinde aynı çevrim süresinde 16 tane *mutlak farklar toplamı* değeri hesaplanabilmektedir.

Bu tip hareket kestirimi donanımları hareket kestirimi işlemini tam piksel derinliğinde gerçekleştirir. Bir-bit dönüşümü temelli hareket kestiriminde ise bu işlem imge çerçevesi bir-bit derinliğinde ifade edildikten sonra gerçekleştirilir. Bu yöntemi kullanarak hareket kestirimi yapmanın düşük işlemsel karmaşa, düşük bellek veri aktarım oranı, düşük güç tüketimi gibi sayısız avantajı vardır.

Tablo 1. On altı işlemcili dizi ile on altı ayrı mutlak farklar toplamının bulunması sırasında piksellerin izlediği yol

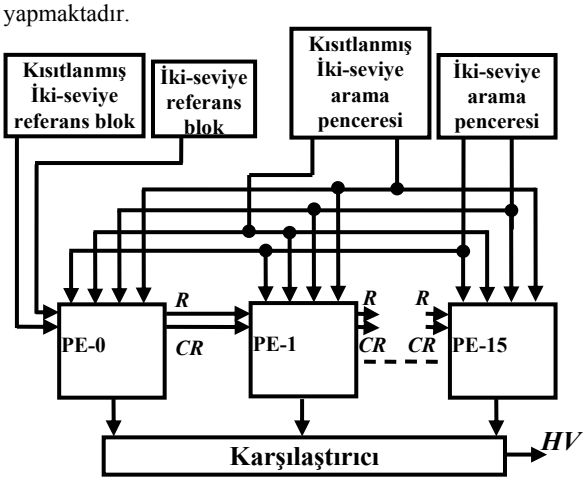
Çevrim zamanı	Giriş Verisi			İşlemci Girişleri			
	r	*s1*	*s2*	PE-0	PE-1		PE-15
0	*r0,0*	*s0,0*		*r0,0-s0,0*			
1	*r0,1*	*s0,1*		*r0,1-s0,1*	*r0,0-s0,1*		
2	*r0,2*	*s0,2*		*r0,2-s0,2*	*r0,1-s0,2*		
.		.		.	.		.
.		.		.	.		.
.		.		.	.		.
14	*r0,14*	*s0,14*		*r0,14-s0,14*	*r0,13-s0,14*		
15	*r0,15*	*s0,15*		*r0,15-s0,15*	*r0,14-s0,15*		*r0,0-s0,15*
16	*r1,0*	*s1,0*	*s0,16*	*r1,0-s1,0*	*r0,15-s0,16*		*r0,1-s0,16*
17	*r1,1*	*s1,1*	*s0,17*	*r1,1-s1,1*	*r1,0-s1,1*		*r0,2-s0,17*
.		.		.	.		.
.		.		.	.		.
.		.		.	.		.
30	*r1,14*	*s1,14*	*s0,30*	*r1,14-s1,14*	*r1,13-s1,14*		*r0,15-s0,30*
31	*r1,15*	*s1,15*	*s0,31*	*r1,15-s1,15*	*r1,14-s1,15*		*r1,0-s1,15*
.		.		.	.		.
.		.		.	.		.
240	*r15,0*	*s15,0*	*s14,16*	*r15,0-s15,0*	*r14,15-s14,16*		*r14,1-s14-16*
.		.		.	.		.
.		.		.	.		.
255	*r15,15*	*s15,15*	*s14,31*	*r15,15-s15,15*	*r15,14-s15,15*		*r15,0-s15,15*
256			*s15,16*		*r15,15-s15,16*		*r15,1-s15,16*
257			*s15,17*				*r15,2-s15,17*
.			.	.	.		.
.			.	.	.		.
.			.	.	.		.
270			*s15,31*				*r15,15-s15,30*

3.2. Kısıtlanmış 1-Bit Dönüşümü Temelli Hareket Kestirimi Donanımının Mimarisi

[10]' de 1-bit dönüşümü yöntemi için işlemci dizilerine dayanan bir donanım mimarisi önerilmiştir. Burada [12]' da önerilenden farklı olarak her bir çevrimde referans çerçeveden tek bir piksel yerine bir satır işlenmektedir böylelikle *mutlak farklar toplamı* yöntemini kullanan bir hareket kestirimi donanımından on altı kat daha hızlı hesaplama yapılabilmektedir. Şekil 3'de [10]' de önerilen bir-bit dönüşümü temelli hareket kestirimi donanımının mimarisinin kısıtlanmış 1-bit dönüşümü algoritmasına uygulanmış hali görülmektedir.

Şekil 3' de görülen mimari ile yapılan hareket kestirimi işleminde bir makro-bloğun hareket vektörünü bulmak için $16 \times 16 + 15$ yani 271 çevrim yeterlidir. Bu sayı doğrusal dizi mimarisini kullanan fakat piksel tabanlı işlem yapan hareket kestirimi donanımı mimarilerinde *16* kat artmaktadır. Paralel olmayan yapılarda ise bu rakam bu çalışma da kullanılan mimari göz önüne alındığın da 16 kat daha artmaktadır. Tüm blokların özdeş hızla çalıştığı düşünülürse, ki bu mutlak fark işlemi ve yüksek bit derinliğindeki diğer aritmetik işlemler düşünüldüğünde imkansızdır, 1-bit dönüşümü temelli hareket kestirimi donanımı klasik ortalama mutlak fark (MAD) algoritmasını en az 256 kat hızlandırmaktadır. Bir-bit dönüşümü mimarisindeki verilerin her bir çevirimde nasıl aktığı Tablo 2' de görülmektedir. Kısıtlanmış 1-bit dönüşümü temelli hareket kestirimi donanımda her bir çevrimde verinin ilerleyişi Tablo 2' de görülen veri akışı ile birebir aynıdır ancak bu tabloda kısıt maskesinden gelen veriler görülmemektedir.

Genelde 8-bit derinliğindeki video dizilerinde hareket kestirimi yapan yöntemlerin başarımı yüksek olsa da son yapılan çalışmalarda başarım oranları arasındaki farklar çok aşağılara çekilebilmiştir. [9]' da yapılan çalışmada *ortalama mutlak fark* ölçütünün elde ettiği PSNR değeri ile önerilen yöntemin elde ettiği PSNR değeri arasındaki fark ortalama 0.6-0.7 olarak bildirilmiştir. Bu çalışma geçmişte önerilen diğer bir-bit temelli hareket kestirimi çalışmalarından daha yüksek bir başarım elde etmiştir. Buna ek olarak [8]' de önerilen, 1-bit dönüşümünün temel aşamalarından süzgeçleme adımındaki bölme işlemini ortadan kaldıran süzgeç çekirdeği de hesapsal maliyetini azaltıcı etki yapmaktadır.



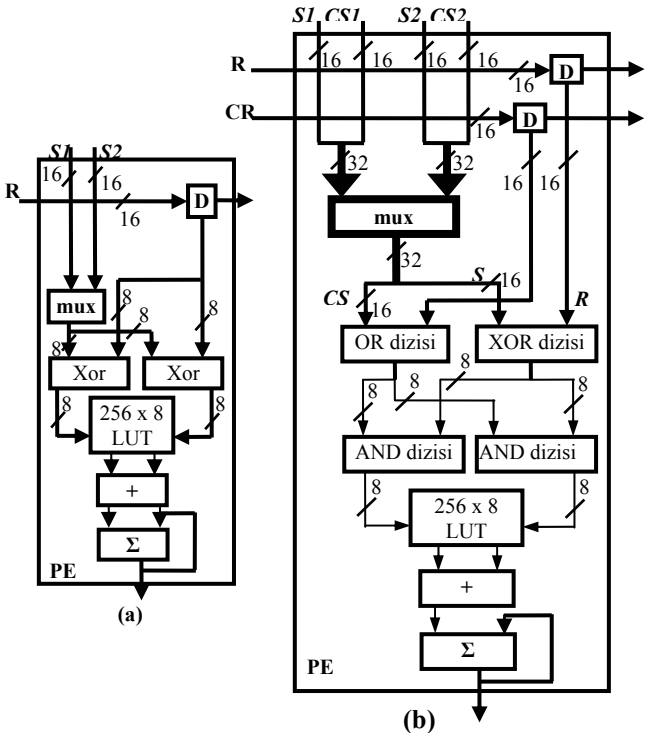
Şekil 3. Kısıtlanmış bir-bit dönüşümü temelli hareket kestirimi donanımının mimarisi.

Tablo 2. [10]' daki mimarinin önerdiği veri akış yapısı

Çevrim Zamanı	Giriş Verisi					İşlemci Girişleri			
	R	*CR*	*S1*	*CS1*	*S2*	*CS2*	PE-0	…	PE-15
0	R0	CR0	S0,0	CS0,0			R0-S0,0		
1	R1	CR1	S1,0	CS1,0			R1-S1,0		
2	R2	CR2	S2,0	CS2,0			R2-S2,0		
.			.	.			.		
.			.	.			.		
.			.	.			.		
14	R14	CR14	S14,0	CS14,0			R14,S14,0		
15	R15	CR15	S15,0	CS15,0			R15-S15,0	…	R0-S15,0
16	R0	CR0	S16,0	CS16,0	CS0,1	CS0,1	R0-S0,1	…	R1-S16,0
17	R1	CR1	S17,0	CS17,0	CS1,1	CS1,1	R1-S1,1	…	R2-S17,0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
30	R14	CR14	S30,0	CS30,0	CS14,1	CS14,1	R14-S14,1	…	R15-S30,0
31	R15	CR15			CS15,1	CS15,1	R15-S15,1	…	R0-S15,1

Hareket kestirimi işleminde aritmetik işlemleri gerçekleştiren şekil 3'de görüldüğü gibi 16 adet PE öbeğidir. 1-bit dönüşümü temelli hareket kestirimi algoritmasının hesapsal bloğu (PE) için önerilen donanım mimarisinde kısıtlanmış bir dönüşümü algoritmasının gerektirdiği işlemleri gerçekleştirecek değişiklikler yapıldığında kısıtlanmış bir-bit dönüşümü algoritması için kullanılacak işlem bloğu mimarisini elde etmek mümkündür. Bu iki yapı Şekil 4'de görülmektedir. Şekil 4a' da görülen yapı [10] de önerilen mimarinin sahip olduğu işlem bloğudur ve bir-bit dönüşüm işleminde kullanılan mantıksal EX-OR işlemi ile hareket bilgisini bulmaya çalışır. Yapının içindeki *D* öbeği referans pencereden gelen verinin bir sonraki saat darbesinde komşu PE öbeğine gitmesini sağlar. *Mux* öbeği ise Tablo 2' de görülen veri akış çizelgesindeki yapının gerçekleştirmesine olanak verir. Şekil 4a' da görülen taramalı tablo ise her bir veri içerisindeki *1* sayısını çıkışına veren küçük bir ROM bellektir ve bu belleğin çıkışında toplam uyumsuz nokta sayısının hesaplanmasını sağlayan bir adet toplayıcı öbeği bulunmaktadır. Bir makro-blok için toplam uyumsuz nokta sayısının hesaplanması 16 çevrim sürdüğü için bu süre boyunca her bir PE öbeğinin içindeki toplayıcı bloğunun çıkışı şekilde de görülen akümülatör aracılığı ile üst üste eklenir ve 16 çevrimin sonunda elde edilen değerler bir birleri ile karşılaştırılmak üzere karşılaştırıcı öbeğine gönderilir.

Bu çalışmada tasarlanan işlem bloğunda ise [10]' deki PE yapısından farklı olarak kısıtlanmış bir-bit dönüşümünün kullandığı uyumlama ölçütünün hesaplanmasını sağlayacak ek öbekler bulunmaktadır. Bunlar referans makro-blok için üretilen kısıt maskesi bilgisi için konulmuş fazladan bir *D* yaz-bozu, uyumlama kriterinde bulunan *AND* işlemini gerçekleştiren bir mantık kapısı dizisi ve yine benzer şekilde *OR* işlemini gerçekleştiren bir mantık kapısı dizisidir. Klasik bir-bit dönüşüme göre iki katı veri bulunduğu için bu çalışmada tasarlanan PE öbeğinin veri yolu 16 bit değil 32 bit genişliğindedir.



Şekil 4. a) Bir-bit dönüşümü temelli hareket kestirimi donanımında kullanılan PE öbeği. ***b)*** Kısıtlanmış bir-bit dönüşümü temelli hareket kestirimi algoritması için önerilen donanım mimarisinde kullanılan PE öbeği.

4. Sonuçlar

Bu çalışmada kullanılan mimari hareket vektörü temelli doğrusal dizi yöntemine dayanarak oluşturulmuştur. İleriki çalışmalarda aynı algoritmanın kaynak piksel tabanlı doğrusal dizi mimarisine uygulanmasına çalışılacaktır. Bu sayede bellekten okuma oranı düşürülecek, kontrol yapısı sadeleştirilecek işlem blokları ile karşılaştırıcı arasındaki geniş veri yolu sadeleştirilerek ek güç tasarrufu ve işlem verimliliği sağlanmaya çalışılacaktır. Bu çalışmada blok uyumlama temelli hareket kestirimi işleminin gerçek zamanlı yapılabilmesini sağlamak amacıyla FPGA tabanlı özgün bir donanım önerilmiştir. Tasarlanan sistem genel amaçlı bir FPGA yongası üzerinde çok az yer kapladığı için tüm bir video kodlama algoritmasının tek bir ASIC yonga veya FPGA ile gerçekleştirilmesi mümkün olabilmektedir. Tasarlanan sistem 50MHz hızında test edilmiştir. Bu sonuçlara göre [-8,7] arama aralığında 2048*1152 [-16,15] aralığında ise 1024*756 boyutlu bir video dizisi için 20 çerçeve/saniye hızında çalışabilmektedir. 30 çerçeve/ saniye için bu çözünürlük değeri yaklaşık 1600*900 ve 800*450 olmaktadır. Düşük çözünürlüklü video dizileri için çalışma frekansı aşağıya çekilerek büyük miktarlarda güç tasarrufu sağlanabilir. Kaba ölçümlerde sistemin harcadığı güç yaklaşık 350 mW olarak gözlemlenmiştir. FPGA' lar ve ASIC uygulaması arasında bir karşılaştırma yapılırsa bu değerin en az 8 kat azalması beklenmektedir [5].

5. Kaynakça

[1] ISO/IEC 14496-10:2003, "Coding of Audiovisual Objects-Part 10:Advanced Video Coding", 2003 also ITU-T Recomendation H.264 "Advanced video coding for generic audiovisual services."

[2] Yao, S., Guo, H.-J., Yu, L., Zhang, K., "A Hardware Implementation of Full-search Motion Estimation of AVS with Search Center Prediction", IEEE Transactions on Consumer Electronics, 52(4): 1356-1361, 2006

[3] Deng, L., Gao, W., Hu, M. Z. , Ji, Z. Z., "An Efficient Hardware Implementation for Motion Estimation of AVC Standard", IEE Transactions on Consumer Electronics, 51(4): 1360-1366, 2005

[4] Luo, J.-H., Wang, C.-N., Chiang, T., "A Novell All-Binary Motion Estimation (ABME) With Optimized Hardware Architectures", IEEE Trans. Circuits Syst. Video Technol., 12(8): 700-712, 2002.

[5] Gao, R. Xu, D., Bentley, J.P., "Reconfigurable Hardware Implementation of an Improved Paralel Architecture for MPEG-4 Motion Estimation in Mobile Applications", IEEE Transactions on Consumer Electronics, 49(4):1384-1390, 2003

[6] J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, "Adaptive block matching motion estimation algorithm using bit plane matching," in Proc. ICIP', 1995, pp. 496–499.

[7] Ertürk, A. and Ertürk, S. "Two-Bit Transform for Binary Block Motion Estimation," IEEE Trans. Circuit Syst. Video Technol., (15)7: 938-946, 2005.

[8] Ertürk, S., "Multiplication-free one-bit transform for low-complexity block-based motion estimation", IEEE Signal Processing Letters, 14(2):109-112. 2007

[9] O. Urhan and S. Ertürk, "Constrained One-Bit Transform for Low Complexity Block Motion Estimation," *IEEE Trans. on Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 478-482, Apr. 2007.

[10] Natarajan, B., Bhaskaran, V. and Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms", IEEE Trans. Circuit Syst. Video Technol., 7(4): 702 - 706, 1997.

[11] V. Bhaskaran and K. Konstantinides, Image and Video Compression Standards: Algorithms and Architectures. 2nd. Kluwer Academic Publishers, 1997.

[12] K.-M.Yang, M.-T.Sun, and L.Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," *IEEE Trans. Circuits Syst*, vol. 36, pp. 1317-1325, Oct. 1989.

[13] I. Kuon, J. Rose, "Measuring the gap between FPGA and ASICS," IEEE Trans. on. Computer-Aided Design of Integrated Circuits and Sys., vol.26, no.2, pp. 203-215, Feb. 2007