

FPGA için JPEG Donanımsal Hızlandırıcı Tasarımı

JPEG Hardware Accelerator Design for FPGA

¹Kaan Duman, ²Fuat Çoğun, ³Levent Öktem

^{1,2}Elektrik ve Elektronik Mühendisliği Bölümü, Bilkent Üniversitesi, Ankara, TÜRKİYE
³Synplicity Ar-Ge Ltd. Şti., Ankara Teknoloji Geliştirme Bölgesi, Ankara, TÜRKİYE
kduman@ug.bilkent.edu.tr, cogun@ug.bilkent.edu.tr, levent@synplicity.com

Özetçe

FPGA üzerinde çalışan bir tam boruhatlı JPEG donanımsal hızlandırıcısı sunulmaktadır. Hızlandırıcı, benzetim ortamında etkileşimsel olarak, bir DSP donanım tasarlama otomasyon araç zinciri kullanılarak tasarlanmıştır. Hızlandırıcının kodlayıcı kısmı 8x8'lik imge bloklarını kesintisiz bir biçimde alıp, çıkışta zikzak tarama yapılmış ve nicemlenmiş iki boyutlu ayrık kosinüs dönüşümü (2-B AKD) katsayılarını verir. Kodçözücü ise bu katsayıları alıp, çıkışında 8x8'lik imge bloğunu yeniden oluşturur. Kodlayıcı ve kodçözücünün iş çıkarma yetenekleri, her kanal için piksel başına bir sistem saatidir. Kodlayıcı 2-B AKD gerçekleştiriminde hızlı boruhatlı gerçekleştirimi[1] kullanılmıştır. Kodçözücüde ise yeni bir boruhatlı mimarisi kullanarak iki boyutlu ters ayrık kosinüs dönüşümü (2-B TAKD) yapısı geliştirilmiştir. Kullanılan TAKD yapısı, yazılım gerçekleştiriminde kullanılan TAKD çarpanlara ayırma yöntemine dayalı olup[2], kodlayıcıda kullanılan boruhatlı AKD yapısından esinlenilerek tasarlanmıştır. Belirli FPGA hedeflerindeki kaynak kullanımı ve en yüksek frekans figürleri hızlandırıcımızın rekabet edebilecek bir performans sağladığını göstermektedir.

Abstract

A fully pipelined JPEG hardware accelerator that runs on FPGA is presented. The accelerator is designed interactively in a simulation environment, using a DSP hardware design automation tool chain. The encoder part of the accelerator accepts 8x8 image blocks in a streaming fashion, and outputs the zigzag-scanned, quantized 2-D DCT coefficients of the block. The decoder part accepts zigzag-scanned, quantized DCT coefficients, and outputs reconstructed 8x8 image block. Each part has a throughput of one system clock per pixel per channel. The encoder employs a fast pipelined implementation for 2-D DCT [1]. For the decoder, a new pipelined 2-D IDCT structure is developed. Our IDCT structure is based on an IDCT factorization for software implementation [2], and is inspired by the pipelined DCT structure employed in the encoder. The resource utilization and maximum frequency figures for a particular FPGA target suggest that our accelerator has competitive performance.

1. Giriş

JPEG'le kodlanmış imgeler, imge saklama ve gönderme işlemlerinde hayati bir öneme sahiptir. Hafıza açısından büyük tasarruf sağlayan JPEG imgelerinin daha hızlı kodlanabilmesi ve çözülebilmesi ise video sıkıştırma ve gönderme

algoritmalarının yüksek hızda gerçekleştirilmesini sağlar. MPEG video kodlama algoritması iki çerçeve arasında kestirim dışında aynı JPEG algoritmasına benzer teknikler kullanarak çalışır. Bu yüzden bu donanımsal hızlandırıcının bileşenleri MPEG gibi bazı video sıkıştırma algoritmalarının gerçekleştiriminde de kullanılabilir.

Bu çalışmada FPGA üzerinde çalışan bir tam boruhatlı JPEG donanımsal hızlandırıcısı gerçekleştirilmiştir. VHDL ile sentezlenen sistemimizin performans ölçütleri (kaynak kullanımı ve en düşük çalışma peryodu) açısından rekabet edebilecek düzeyde olduğu görülmüştür.

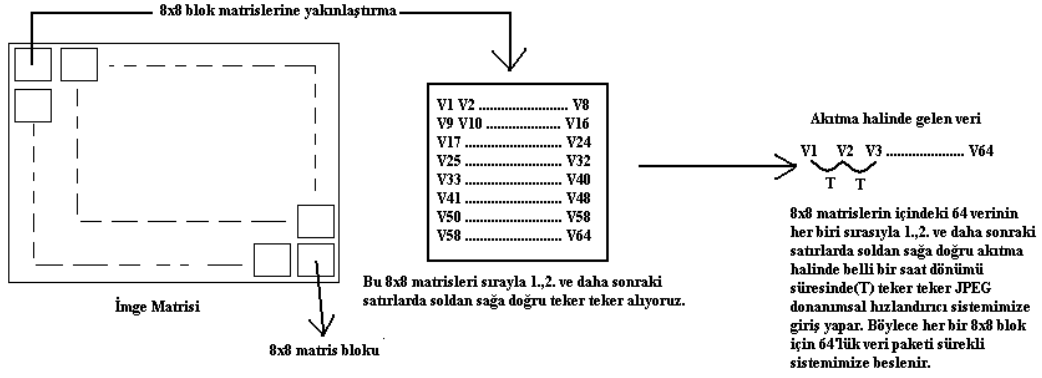
2. Yöntem

Hızlandırıcı Simulink ortamında bir DSP donanım tasarlama otomasyon araç zinciri kullanılarak tasarlanmıştır. Benzetim ortamında etkileşimli tasarım, VHDL kodu ve sinama tezgahı üretimi için Simulink ve Synplify DSP; VHDL kodunun sinama tezgahında sinama benzetimi için Modelsim; mantık sentezleme aşaması için Synplify Pro; yerleştirme ve yönlendirme aşaması için Xilinx ISE kullanılmıştır.

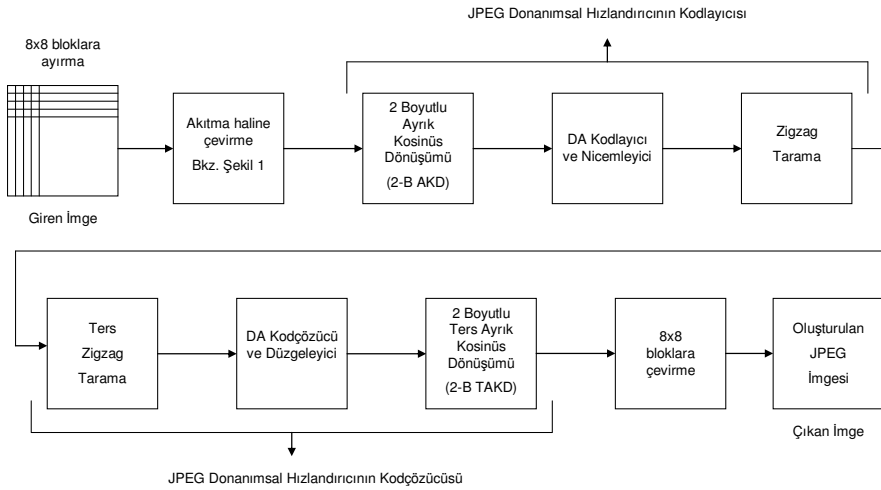
JPEG donanımsal hızlandırıcı tasarımında 1991'de ortaya çıkan JPEG standardı[3] temel alınmıştır. Kodlayıcı ve kodçözücü modüllerinin Simulink benzetim modelleri Simulink'in imge sıkıştırma tanıtımının içindeki kodlayıcı ve kodçözücüyle uyumludur. Benzetim modelinde sisteme giren imge Şekil 1'deki gibi 8x8 bloklara ayrıldıktan sonra hızlandırıcıya akıtmalı 64'lük veri blokları halinde beslenmektedir. Bu blokların içindeki verilerin her biri grintonlu bir imge için bir pikseli temsil eder. Şekil 2'nin ilk bölümünde bileşenleri görünen kodlayıcıya akıtmalı olarak beslenen veriye önce hızlı boruhatlı 2-B AKD uygulanır [1]. 2-B AKD çıktısı nicemlendikten sonra, her 64'lük frekans katsayı blokunun ilk elemanı, yani DA (doğrusal akım) katsayısı JPEG algoritmasına uygun olarak fark-kodlanır. Son olarak, blok elemanları zikzak tarama etkisi gösterecek şekilde yeniden sıralanır.

Kodlayıcı çıktısı kodçözücüye beslendiğinde ise, sırasıyla ters zikzak tarama, DA kodçözme ve düzgeleme evrelerinden geçtikten sonra, gerçekleştirimi kendi tasarımımız olan tam boruhatlı 2-B TAKD birimine beslenir. 2-B TAKD tasarımımızda, 2-B AKD için kullanılan [1]'deki gerçekleştirim yapısından esinlenilmiş ve hızlı yazılım gerçekleştirimi için geliştirilmiş olan [2]'deki çarpanlara ayırma yöntemi temel alınmıştır.

Son olarak hızlandırıcıdan çıkan veriler yine benzetim blokları aracılığıyla ilk olarak 8x8 bloklara ve sonra da bütün bir imgeye dönüştürülür.



Şekil 1: İmgenin 8x8 akıtmalı bloklara dönüştürülmesi



Şekil 2: Kodlayıcı ve kodçözücü donanımsal hızlandırıcılarının bileşenleri

3. Donanımsal hızlandırıcının modülleri

3.1 2-B Ayrık kosinüs dönüşümü (2-B AKD)

Bu dönüşümdeki amaç imgedeki her 8x8 bloğun 2-B AKD'nü almaktır. Donanımsal uygulamada dikkat edilmesi gereken nokta her bloktaki 64 verinin hızlandırıcıya kesintisiz olarak akıtma halinde gelmesidir. Buna göre dönüşümün uygulaması [1]'de belirtildiği gibi şu şekildedir:

- 8x8 bloklar 64 verilik akıtma bloklarına dönüştükten sonra ilk AKD'nden geçer ve böylece her 1x8'lik satır vektörünün dönüşümü alınmış olur.
- 8x8'lik blokların ana köşegenleri boyunca devrilecekleri şekilde 64'lük blokun içindeki verilerin yerleri değiştirilir.
- Yeri değiştirilen dizi ilk maddedekinin aynısı olan ayrık kosinüs dönüşümünden bir daha geçer, bu sefer 8x1'lik sütun vektörleri dönüşümünü tamamlamış olur ve sonuç olarak baştaki 8x8'lik bloğun akıtma halinde gelen verilerinin AKD alınmış olur.

Önemli bir not ise ortaya çıkan AKD'nün gerçek dönüşümün ölçeklenmiş hali olmasıdır. Bu ölçekleme nicemleyici kısmında dengelenmiştir.

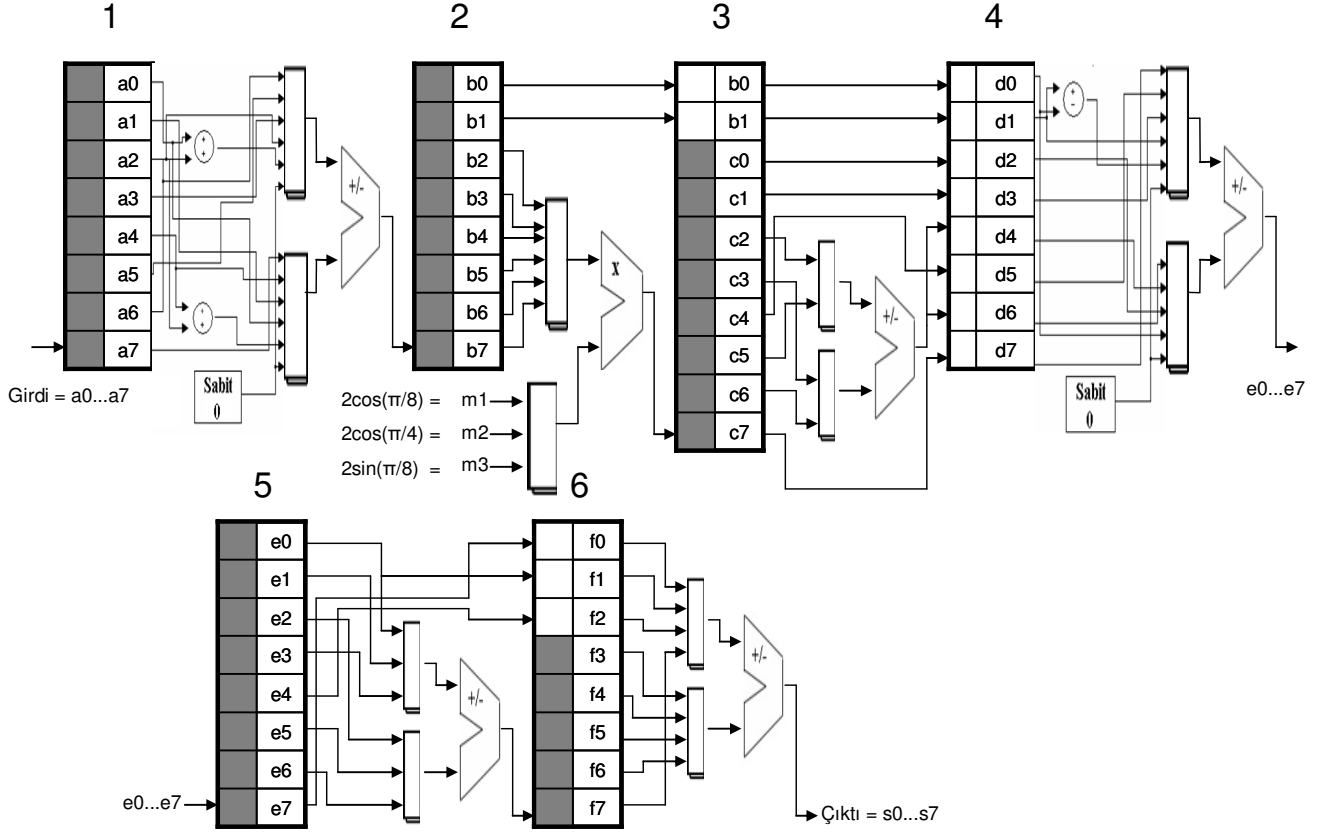
3.2 DA fark kodlayıcı ve nicemleyici

Frekans katsayılarına dönüşmüş olan giriş verilerinin genellikle en yüksek değerlisi en düşük frekanslıdır. En düşük frekanslı veri 64'lük blokların başında yer alır ve DA sembolüyle analogi kurulmak suretiyle bu değer DA değeri olarak belirtilir. Sisteme ilk imge girişi yapıldığında gelen ilk 64'lük bloktan DA fark kodlayıcı ilk değeri olduğu gibi alır ve daha sonra gelen blokların ilk değerlerinin önceki ilk değerlerle olan farkını kodlar. Böylece yüksek DA değerleri yerine sadece iki ardışık DA değeri arasındaki fark kodlanacağından istatistiksel olarak sıkıştırma avantajı sağlanır.

Nicemleyici ise 64'lük blok içindeki her değişik frekans değerini belirli bir katsayıya böler. Bu katsayılar değişebilir. Hızlandırıcımızda [3]'te örnek tablo olarak verilen nicemleyici tablosunu kullanılmıştır.

3.3 Zikzak ve ters zikzak tarama

Zikzak tarama JPEG standardında 8x8 bloklar üzerinde yapılan, Huffman veya aritmetik kodlamasının verimini artıran bir işlemdir [3]. Hızlandırıcımızda yol alan bloklardaki 64 verinin her birinin çaprazlama tampon devresinden geçerek yer değiştirmesi sonucunda elde edilir. Ters zikzak tarama da aynı tekniği kullanarak verilerin nicemleyici çıkışındaki sıralamasına geri dönüşünü sağlar.



Şekil 3: Tam boru hatlı 1-B TAKD gerçekleştirimi

3.4 DA çözücü ve düzgeleyici

DA çözücü kodlanan DA farklarını toplayarak, fark kodlaması öncesindeki DA değerlerini bulur.

Düzeleme aşamasında ise yine 64'lük bloktaki her bir veri nicemleyicideki ölçeklendirmeyi dengeleyecek bir katsayı ile çarpılır.

3.5 2-B Ters ayrık kosinüs dönüşümü (2-B TAKD)

TAKD gerçekleştirimini AKD'den ayıran temel fark, bir boyutlu dönüşümlerde ortaya çıkar. Dönüşüm çarpanlarının farklı olması nedeniyle, çarpanlara ayırma işleminin sonucu da farklıdır, ve dolayısıyla AKD için kullanılan aşamalar ve aşamalar arası bağlantılar birebir kullanılamaz. 1-B TAKD'nin çarpanlara ayrılma sonucuna göre, aşamalar ve aşamalar arası bağlantılar TAKD için yeniden düzenlenmiştir.

2-B TAKD'de de veriler Bölüm 3.1'de maddelendirilen işlemlerin benzerinden geçer. Bu dönüşüm konusunda daha detaylı bilgi bir sonraki bölümde verilmiştir.

4. İki Boyutlu tam boruhatlı ters ayrık kosinüs dönüşümü (2-B TAKD)

2-B TAKD, imge bilgisini 2-B frekans bölgesinden 2-B uzam bölgesine 8x8'lik piksel veri birimleri olarak geri çevirir. 2-B TAKD modülü, kodlanmış JPEG imgesinin çözümünde kilit bir rol oynayarak, gelen kodlanmış giriş verisinin üzerinde gerekli işlemleri yaparak, çıkışında JPEG donanımsal hızlandırıcısına sunulan imgenin geri çatılmasını sağlar.

4.1 2-B TAKD algoritması

2-B TAKD modülü çıkışında imge haline geri dönüştürülmeye hazır bir halde bulunan kesintisiz çıktı verisini oluşturur. İki boyutlu Ters Ayrık Kosinüs Dönüşümünün hesaplanması yüksek hesaplama karmaşıklığı gerektirdiğinden dolayı JPEG imge sıkıştırılmasındaki 2-B TAKD'nün gerçekleştiriminde boruhatlı mimarisi kullanılmıştır. İki boyutlu ters ayrık kosinüs dönüşümünün gerçekleştiriminde, 2-B TAKD'nün ayrılabilirlik özelliğinden yararlanılarak; 2-B TAKD, iki tane 1-B TAKD devresi ve bir tane çaprazlayıcı tampon devresi kullanılarak gerçekleştirilmiştir. Bu algorithmada kullanılan 2-B TAKD mimarisi Şekil 4'teki gibidir. 1-B TAKD devreleri her algoritma adımı bir aşamadan oluşmak üzere, altı aşamalı bir boruhatlı mimarisinden oluşur.

8x8'lik bir giriş matrisi için öncelikle ilk 1-B TAKD bloğu giriş matrisini sütun boyunca alır. İkinci 1B-TAKD bloğuna, ilk 1B-TAKD bloğunun çıkışı çaprazlanarak satır boyunca verilir. Çaprazlamada kullanılan çaprazlama tampon devresi sütun sonuçlarını alır ve satır girişi olarak ikinci 1B-TAKD bloğuna verir. İkinci 1-B TAKD bloğu satır girişini alıp dışarıya satır çıkışı verir.

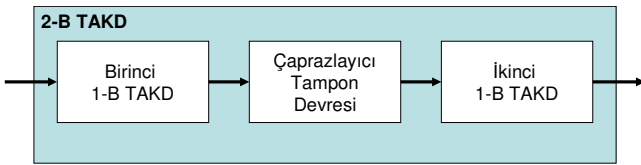
Kullanılan algorithmada her 1-B TAKD bloğunun 58 saat çevrimi bekleme süresi, çaprazlayıcı tampon devresinin ise 53 saat çevrimi bekleme süresi vardır. Sonuç olarak 2-B TAKD modülünün toplamda 169 saat çevrimi bekleme süresi vardır. Gelen giriş verisi kesintisiz olarak geldiğinden dolayı bu bekleme süresi her imge için bir kez geçerlidir.

4.2 Hızlı 1-B TAKD algoritması

[2]'de önerilen ve bu bildiriye tasarlanan Hızlı 1-B TAKD mimarisi Şekil 3'te gösterilmiştir. Hızlandırıcının 1-B TAKD' de uygulanan algoritma da göz önünde bulundurulduğunda boru hattı mimarisinin kullanımı doğaldır.

Hızlı 1-B TAKD, her algoritma adımı bir aşamadan oluşmak üzere, altı aşamalı bir boru hattı mimarisiyle sağlanır. Bu altı aşamadan 5 tanesinde toplama ve çıkarma işlemleri, 1 tanesinde ise çarpma işlemleri gerçekleştirilmiştir. Birinci ve üçüncü aşamalarda sadece o aşamada kullanılmak üzere geçici iki sinyal tanımlanmıştır.

Her 1-B TAKD devresi 58 saat çevrimi bekleme süresi gerektirir ve bütün 1-B TAKD hesaplaması 64 saat çevriminde tamamlanır.



Şekil 4: 2-B TAKD'nin yapısı.

5. Sonuçlar

Hızlandırıcının FPGA için üretilen VHDL kodunun doğruluğu Modelsim ile VHDL benzetim ortamında, Synplify DSP tarafından üretilen sına tezgahı ile sınanmıştır. Daha sonra mantık sentezleme (Synplify Pro) ve yerleştirme-yönlendirme (Xilinx ISE) aşamalarından geçirilerek kaynak kullanımı ve çalıştığı en düşük peryot belirlenmiştir.

Altera EPF10K100EQC208 türü FPGA için hızlandırıcının kaynak kullanım sonuçları Tablo 1'de gösterilmiştir. Bu tahmini sonuçlara kodçözücü çıkışında tatmin edici kalitede imge elde edilmesi esas alınmak suretiyle ulaşılmıştır. Tabloda ayrıca [1]'deki 2-B AKD uygulamasının da sonuçlarına yer verilmiştir. Buna göre mantık hücre sayısının ve hafıza kullanımının hızlandırıcıda göreceli olarak yüksek çıkmasının nedeni bit genişliği optimizasyonunun yapılmamış olmasına dayandırılabilir. TAKD uygulamasında hafıza kullanımının AKD'ye göre fazla çıkmasının nedenlerinden biri de kullanılan bit genişliğinin artmasıdır.

Tablo 1'de görüldüğü üzere hızı -1 olan aynı tür FPGA için elde edilen en düşük çalışma peryodu değerlerimiz [1]'deki uygulamadan neredeyse iki kat daha azdır. Hız performansının bu derece yüksekliği karşılaştırılan mimariye göre daha fazla kaynak kullanımını mazur gösterecek niteliktedir.

JPEG uygulamaları için çeşitli AKD ve TAKD uygulamaları [4], [5] ve [6]'da mevcuttur. Buna rağmen, elde edilen sonuçlara dayanarak, hızlandırıcımızın rekabet edebilecek bir performans sağladığını söyleyebiliriz. Sözgelimi, hızlandırıcının kodçözücü birimi MPEG görüntülerinin 2-B TAKD'sini hesaplamak için kullanıldığında, HDTV çözünürlüğünde bir videonun gerçek zamanlı olarak kodçözümü mümkün olmaktadır.

6. Teşekkür

Bu çalışma TÜBİTAK tarafından 105E065 BTT-Türkiye Projesi kapsamında desteklenmiştir. Bu bildiri, Bilkent Üniversitesi'nin bitirme projesi çerçevesinde ortaya çıktı. Bize

bitirme projemizi tamamlamamız yolunda yön gösteren ve yardımcı olan saygıdeğer hocalarımız Erdal Arıkan, Defne Aktaş ve Tarık Reyhan'a teşekkürlerimizi sunuyoruz.

7. Kaynakça

[1] L. C. Agostini, I. S. Silva, S. Bampi. "Pipelined Fast 2-D DCT Architecture for JPEG Image Compression", *SBCCI '01 Proceedings of the 14th symposium on Integrated Circuits and Systems Design*, Eylül 2001, s. 226-231.

[2] Technische Universität Chemnitz – Fakultät für Informatik, "The Implementation of the 2-D IDCT", *Internet çevrimiçi kaynak*: <http://rnvs.informatik.tu-chemnitz.de/~jan/MPEG/HTML/IDCT.html>

[3] G. K. Wallace, "The JPEG Still Pictures Compression Standard". *IEEE Trans. Consumer Electronics*, Cilt 38, No 1, Şubat 1992.

[4] M. Kovac, N. Rangathan. "JAGUAR: A fully Pipelined VLSI Architecture for JPEG Image Compression Standard". *Proceedings of the IEEE*, Cilt 83, No 2, 1995, s. 247-258.

[5] A. Madiseti, A Willson Jr. "A 100 MHz 2-D 8x8 DCT/IDCT Processor for HDTV Applications". *IEEE Transactions on Circuits and Systems for Video Technology*, Cilt 5, No 2, Nisan 1995, s. 158-165.

[6] C. Wang, C. Chen. "High-Throughput VLSI Architectures for the 1-D and 2-D Discrete Cosine Transforms". *IEEE Transactions on Circuits and Systems for Video Technology*, Cilt 5, No 1, Şubat 1995, s. 31-40.

Tablo 1: Kullanılan kaynaklar ve en düşük çalışma peryotları

	Mantık Hücresi Sayısı	Peryot (ns)	Hafıza Kullanımı
Kaynakça [1]'deki mimari için:			
Birinci 1-B AKD	2051	73.2	
İkinci 1-B AKD	2473	80.8	
Çaprazlayıcı Tampon Devresi	274	36.5	1536 bit
2-B AKD	4792	82.1	1536 bit
Bu bildiriye sözü edilen hızlandırıcı için:			
Birinci 1-B AKD	2269	40.0	
İkinci 1-B AKD	3577	42.1	
Çaprazlayıcı Tampon Devresi	113	21.9	2EAB + 71 bit
2-B AKD	5924	42.2	2EAB + 71 bit
Birinci 1-B TAKD	3880	46.4	
İkinci 1-B TAKD	4576	46.3	
Çaprazlayıcı Tampon Devresi	132	21.9	3EAB + 74 bit
2-B TAKD	8834	46.6	3EAB + 74 bit