

GRAFİK SİSTEMLERİ İÇİN FPGA CİHAZLARINDA ÇALIŞMAK ÜZERE TASARLANMIŞ MATRİS ÇARPIM MOTORU

İbrahim ŞAHİN¹ ve İsmail KOYUNCU²

¹D.Ü. Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Eğitimi Bölümü, 81620, Düzce.

²D.Ü. Düzce Meslek Yüksekokulu, Endüstriyel Otomasyon Bölümü, 81100, Düzce.

ÖZET

Bilgisayar grafiklerinde sıklıkla kullanılan üç boyutlu (3B) dönüşümlerde, animasyon sahnesindeki dönüşüme uğrayan nesne sayısı ve bu nesnelere tanımlamada kullanılan nokta sayısı arttıkça dönüşümü hesaplamak için gereken CPU zamanı katlanarak artmaktadır. Sonuçta, kısa bir animasyon filminin hesaplanması uzun zaman almakta ve eğer animasyon gerçek zamanlı bir animasyon ise hesaplanması imkansız hale gelmektedir. Bu çalışmada, FPGA (Alan Programlanabilir Kapı Dizileri) çiplerini kullanarak üç boyutlu dönüşümleri hızlandırmak için kesirli sayı tabanlı donanım modülü tasarlanmıştır. Tasarlanan modül gerçek veri üzerinde işlemler yapılarak test edilmiş ve modülün ürettiği sonuçların doğrulanması yapılmıştır. Modülün veri işleme hızı değişik bilgisayarlarla karşılaştırılmıştır. Karşılaştırma sonuçları göstermiştir ki, tasarlanan modül kullanılarak üç boyutlu grafik dönüşümleri PC'lere göre 4 kata kadar daha hızlı gerçekleştirilebilmektedir.

Anahtar Kelimeler: FPGA, Bilgisayar Grafikleri, 3B Dönüşüm, Donanım Modülü.

A MATRIX MULTIPLICATION ENGINE FOR GRAPHIC SYSTEMS DESIGNED TO RUN ON FPGA DEVICES

ABSTRACT

In Computer Graphics, as the number of objects in an animation scene and the number of vertices used to define each object increase, calculations of the three dimensional (3D) transformations require huge amount of CPU time. As a result, calculation of an animation scene could take a long time. Moreover, if it is a real time animation, it becomes impossible to calculate transformations in real time. In this study, a floating point based hardware module has been designed to speed up three dimensional transformations using FPGA (Field Programmable Gate Array) chips. The module was tested and functional verification of the module is done by comparing of results produced by module and the results generated by general purpose computers (PCs) for the same set of input data. Module's data processing speed was compared to various PCs. The results showed that using the module, 3D graphic transformations can be speeded up by factor of up to 4.

Keywords: FPGA, Computer Graphics, 3D Transformation, Hardware Module.

1. GİRİŞ

Üç boyutlu (3B) grafik dönüşümler, grafik paketlerinin ve grafik uygulama programlarının vazgeçilmez parçalarıdır. Bu dönüşümler binlerce noktadan oluşan karmaşık şekillere uygulandığında, grafik programları işlemci gücünü önemli miktarda kullanmaktadır.

Özellikle animasyon işlemlerinde, animasyon sahnesindeki nesne sayısı ve bu nesnelere tanımlamada kullanılan nokta sayısı arttıkça, animasyonun hesaplanması saatler alabilmektedir. Örneğin tipik bir animasyon filminin bir sahnesinde onlarca grafik nesne bulunabilmekte ve nesnelere tanımlamak için yüz binin üzerinde nokta kullanılabilir [1]. Böyle bir

animasyon sahnesinin DVD standardına göre saniyede 24 defa hesaplanması gerekmektedir. Bunun anlamı, sahneyi tanımlamada kullanılan yüz binin üzerindeki noktanın yeni değerlerinin saniyede 24 defa hesaplanması gerektiğidir. Üç boyutlu animasyonda bir noktanın yeni değerinin hesaplanabilmesi için 4x4 ile 4x1'lik iki matrisin çarpımının gerçekleştirilmesi söz konusudur. Yüz bin noktalı bir sahnenin 1 saniyelik animasyonunda toplam 38.4 milyon çarpma ve 28.8 milyon toplama işleminin yapılması gerekmektedir. Bu durumda genel amaçlı bilgisayarlar yetersiz kalmakta ve animasyonun hesaplanması çok uzun süreler almaktadır. Bu duruma çözüm olarak değişik yaklaşımlar geliştirilmiştir [2]. Bu yaklaşımlardan bazıları; gelişmiş grafik kartlarının, grafik işlemler için tasarlanmış özel amaçlı bilgisayarların [3], daha hızlı işlemcilerin ya da paralel işlemcilerin kullanılması olarak özetlenebilir. Bu sunulan çözümlerin bazı dezavantajları bulunmaktadır. Örneğin süper bilgisayarlar oldukça maliyetlidirler. Normal bilgisayarlar, grafik işlemlerinin gerçekleştirilmesinde yeterli performans gösterememektedirler. Özel olarak tasarlanan kartlarda ise herhangi bir tasarım veya üretim hatası telafi edilememekle birlikte yeniden tasarım süreci oldukça uzun zaman kaybına sebep olmaktadır.

Bu çalışmada yukarıda sayılan yöntemlere bir alternatif olarak üç boyutlu grafik dönüşümleri hızlandırmak amacıyla, FPGA çipleri üzerinde çalışabilecek kesirli sayı tabanlı bir donanım modülü tasarlanmıştır. Böylelikle 3B grafik dönüşümlerin, daha ucuz ve daha hızlı hesaplanması hedeflenmiştir.

Tasarlanan modül, gerçek veri üzerinde işlemler yapılarak test edilmiş ve modülün ürettiği sonuçların doğrulanması yapılmıştır. Aynı veriler değişik özellikteki bilgisayarlarla da işlenerek modülün veri işleme hızı genel amaçlı bilgisayarlarla karşılaştırılmıştır. Makalenin ikinci bölümünde FPGA çipleri ve RC sistemler, bilgisayar grafikleri ile üç boyutlu dönüşümler kısaca özetlenmiştir. Üçüncü bölümde, tasarlanan modül detaylarıyla anlatılmıştır. Dördüncü bölümde, yapılan test çalışmaları ve bu çalışmalardan elde edilen sonuçlar sunulmuştur. Beşinci bölümde ise sonuçlar değerlendirilmiştir.

2. ÖN BİLGİ

2.1. FPGA Çipleri

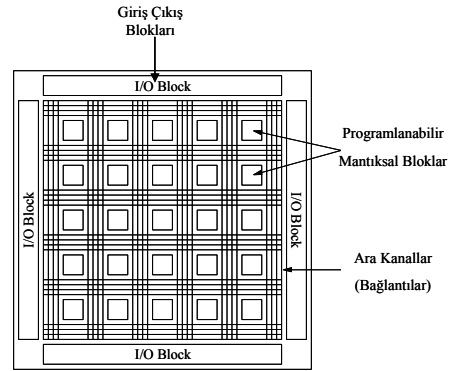
FPGA çipleri programlanabilir tümdevrelerdir. Tasarımcının ihtiyacı olan mantıksal fonksiyonları gerçekleştirebilmesi amacıyla, kullanıldığı yerde programlanabilir olarak üretilirler. Kullanıcının tasarladığı mantıksal devreye göre, mantıksal bloklar, aralarındaki bağlantılar ve giriş/çıkış blokları programlanır. Şekil 1'de genel yapısı verilen FPGA çipi, programlanabilir üç bileşenden oluşur:

a. Yapılandırılabilir Mantıksal Bloklar (Configurable Logic Blocks (CLB)): CLB blokları, mantıksal fonksiyonların oluşturulabildiği Look-up table (LUT) ve

Flip-Flop'lardan oluşmaktadır. CLB, kullanıcının oluşturmak istediği mantıksal devre için fonksiyonel elemanlar sağlar. CLB mimarisinin esnekliği ve simetrisi, uygulamaların kolaylıkla yerleştirilmesine ve yönlendirilmesine olanak tanır [4].

b. Giriş Çıkış Blokları (Input/Output Blocks (IOB)): IOB'ler çipin iç sinyal hatları ile çipin pinleri arasında programlanabilir bir arabirim sağlarlar. IOB'ler sayesinde FPGA pinleri giriş, çıkış ya da çift yönlü olarak programlanabilir. FPGA çipinin türüne göre bir çipteki IOB sayısı (dolayısıyla pin sayısı) 1000'li sayılara ulaşabilmektedir.

c. Ara Bağlantılar (Interconnections): Bu birimler hem CLB'ler arasında hem de CLB'ler ile IOB'ler arasında bağlantıları yapılandırmada kullanılırlar. Programlanabilir olduklarından çok esnek bir yapıya sahiptirler.

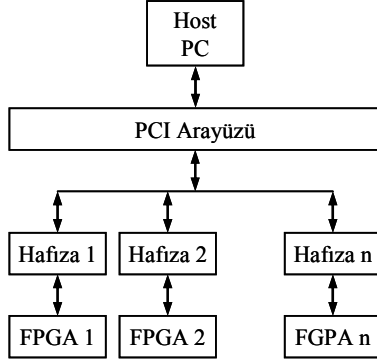


Şekil 1. FPGA genel yapısı [5].

2.2. FPGA Tabanlı Özel Amaçlı Bilgisayarlar

FPGA tabanlı Özel Amaçlı Bilgisayarlar (FPGA-based Custom Computing Machines (F-CCMs)), donanım ve yazılımı bir araya getiren özel veri işleme platformlarıdır [6]. Genellikle bir F-CCM (Reconfigurable Computing (RC) sistem olarak da bilinir) bir adet genel amaçlı bilgisayar ve bu bilgisayara bağlı üzerinde bir ya da daha fazla FPGA ve hafıza çipleri bulunduran elektronik karttan oluşur [7, 8]. Bu sistemler genel amaçlı işlemcilerin sunduğu programlanabilme özelliği ile FPGA çiplerinin sağladığı hız avantajının bir araya getirildiği sistemlerdir [9]. Uygulama programları F-CCM'lere uyarlanırken, programın yüksek işlemci gücü gerektiren bölümleri özel olarak tasarlanmış donanım modülü kullanılarak FPGA çipleri üzerinde çalıştırılır ve bu sayede programın daha hızlı çalışması sağlanır. İyi tasarlanmış bir F-CCM ve donanım modülü sayesinde, uygulama programlarının yoğun CPU gerektiren bölümlerini 100 kata kadar hızlandırmak mümkündür [1, 5]. Hali hazırda çeşitli grafik işleme algoritmaları ve optimizasyon algoritmaları dahil üzere bir çok algoritma F-CCM'lere uygulanmış ve genel amaçlı bilgisayarlarla karşılaştırıldığında, F-CCM'lerin şu ana kadar tespit edilen en düşük çalışma zamanını verdiği görülmüştür [10, 11]. Birçok firma RC sistemlerde

kullanılmak üzere FPGA kartları tasarlamış ve piyasaya sunmuştur. Şekil 2’te RC sistemlerin genel yapısı görülmektedir. Bu yapıda n adet FPGA çipi ve bunlara ait yerel hafızalar bulunur. FPGA kartı bilgisayar ile veri iletişimini bir PCI ara yüzü üzerinden yapar.



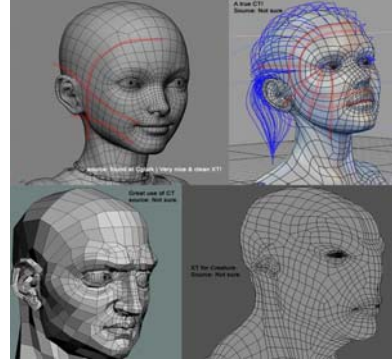
Şekil 2. RC sistem genel yapısı.

FPGA kartlarının kullanımı ise şu şekildedir: İlk olarak, daha önceden tasarlanmış donanım modülünün yapılandırma (configuration) bilgileri PCI yolu üzerinden FPGA çiplerine yüklenir. Ardından, yine PCI üzerinden, işlenecek veri kart üzerindeki yerel hafıza ünitelerine aktarılır. Bu aşamadan sonra FPGA çiplerine özel bir başlat sinyali gönderilerek, çiplerin hafızadaki veriyi işleme sağlanır. Çipler bu verileri işlerken, ana bilgisayar diğer işlerine devam eder. Çipler işlemi bitirince, bilgisayara bir kesme sinyali göndererek işlenmiş verinin hazır olduğunu bildirir. Son olarak bilgisayar yine PCI üzerinden FPGA kartı üzerindeki hafızalara erişerek işlenmiş veriyi alır [7]. FPGA yerel hafızası ile bilgisayar hafızası arasında hızlı veri iletişimini gerçekleştirmek için DMA (Direct Memory Access-Doğrudan Hafıza Erişimi) kullanır.

2.3. Bilgisayar Grafikleri ve Üç Boyutlu Dönüşümler

Grafik tasarımlar ve animasyonlar için çeşitli grafik paketleri (API-Application Programming Interface-Uygulama Programlama Arayüzü) geliştirilmiştir. Bu paketler genellikle grafik işlemleri yerine getirebilmek için birçok fonksiyon içerirler. Bu grafik paketlerinden en yaygın olarak bilinenleri OpenGL (Graphic Library) ve Direct X'tir. Paket ile gelen fonksiyonlara C/C++ başta olmak üzere birçok programlama dilinden erişim mümkündür. Bu paketlerle animasyonların oluşturulabilmesi için nesnelerin 2 veya 3 boyutlu ortamda matematiksel olarak noktalar, çizgiler ve yüzeyler olarak modellenmesi gerekmektedir. Basit bir nesneyi oluşturmak için yüzlerce nokta tanımlanması gerekmektedir. Şekil 3'te noktaların tanımlanarak birleştirilmesiyle oluşturulmuş çeşitli şekiller verilmiştir. Grafik paketlerinde nesnelere tanımlanırken, nesne üzerinde bulunan noktaların kartezyen koordinat sistemindeki üç boyutlu koordinat değerleriyle (x, y, z) tanımlanırlar. Üç boyutlu (3B) grafik dönüşümler, grafik paketlerinin ve grafik uygulama programlarının

vazgeçilmez parçalarıdır. Nesnelere üzerinde genel olarak *öteleme*, *döndürme* ve *ölçeklendirme* olmak üzere temel bir kaç değişik dönüşüm veya bunların kombinasyonu uygulanır.



Şekil 3. Nesnelerin noktalarla oluşturulması [4].

Bu dönüşümlerden üç boyutlu *döndürme* ve *ölçeklendirme* işlemleri 3×3 ile 3×1 matris çarpımı olarak gerçekleştirilir. Fakat *öteleme* işlemi diğer işlemler gibi doğrusal bir işlem olmadığından matris çarpımı şeklinde ifade edilemez. Bu işlemi de doğrusal hale getirebilmek için kartezyen koordinat sisteminde (x, y, z) olarak tanımlanan nokta değerleri homojen koordinat sistemine çevrilerek bütün grafik dönüşümleri standart 4×4 ile 4×1 matris çarpımı haline getirilir. Bu dönüşüm işlemi sırasında nokta x, y, z ile birlikte bir dördüncü değer olarak w ile (x, y, z, w) olarak ifade edilir ($w \neq 0$ olmak kaydıyla). w değeri olarak genellikle 1 alınır.

Aşağıda *öteleme*, *döndürme* ve *ölçeklendirme* işlemleri parametrik ve matris çarpımı notasyonunda verilmiştir. Öteleme işleminde t_x, t_y ve t_z üç boyutlu ortamda nesnenin her boyuttaki öteleme miktarını, döndürme işleminde θ nesnenin dönme açısını, ölçeklendirme işleminde ise s_x, s_y ve s_z nesnenin her bir boyuttaki ölçeklendirme katsayısını belirtir. P dönüşüme uğrayan noktanın orijinal koordinatı, P' ise yeni koordinatıdır [1, 4, 13].

$$P' = T(t_x, t_y, t_z) \cdot P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Öteleme

$$P' = R(\theta) \cdot P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Döndürme

$$P' = S(s_x, s_y, s_z) \cdot P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Ölçeklendirme

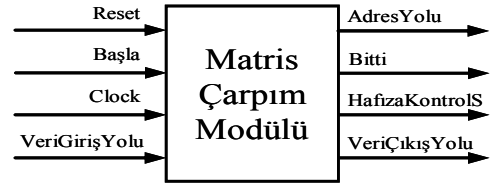
Yukarıdaki eşitliklerden de görüleceği üzere dönüşüm işlemleri 4x4 ile 4x1 matris çarpımı halinde gerçekleştirilmektedir. Bunların dışında grafik paketlerinde, *eğme* (shearing), *gölgeleme* (shadow) gibi dönüşümler yine 4x4 ile 4x1 matris çarpımı şeklinde gerçekleştirilmektedir. Hatta bu dönüşümlerin kombinasyonu da yine aynı formatta olmaktadır. Bu şekillerden bir sahnede yüzlerce olabilir. Grafik paketleri bu matematiksel modeller üzerinde yukarıda açıklanan grafik dönüşümleri uygulayarak animasyonları oluşturur. Yani basit bir kamera hareketiyle ilgili bir animasyonda, bütün nesnelerin sahnedeki yeni görünüşleri (nesnelere tanımlayan noktaların yeni değerleri) bu dönüşümlerde verilen matris çarpımları yoluyla hesaplanır.

3. ÜÇ BOYUTLU DÖNÜŞÜMLER İÇİN FPGA MODÜL TASARIMI

Bu çalışmada kesirli sayı tabanlı, üç boyutlu homojen dönüşümler için FPGA sistemleri ile birlikte kullanılacak bir donanım modülü tasarlanmıştır. Modül sabit bir 4x4 matris ile bir dizi 4x1 matris serisini çarparak yeni bir dizi 4x1 matris serisi oluşturacak şekilde düzenlenmiştir. Modül 32-bit IEEE 754-1985 kesirli sayı standardına uygun olarak bir donanım tanımlama dili olan VHDL'de tasarlanmış ve Xilinx ISE 9.2 aracı kullanılarak değişik FPGA çipleri için sentezlenmiştir.

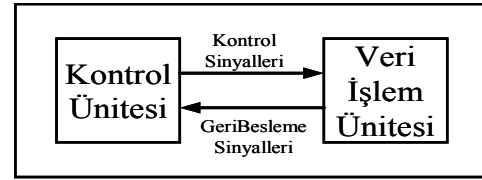
3.1. Modülün Genel Yapısı

Şekil 4'te tasarlanan modülün en üst seviye blok diyagramı görülmektedir. Modül üzerinde 32-bitlik veri girişi ve çıkışı ayrı ayrı verilmiştir. Bunun sebebi modülün herhangi bir özel FPGA kartına göre tasarlanmamış olmasıdır. Modül kullanıcıları, kullanılacağı karta uygun bir hafıza ara yüzü aracılığıyla veri girişi ve çıkışı yollarını birleştirerek çift yönlü tek bir veri yoluyla hafıza ünitesine bağlantı yapmalıdırlar. Modül 32-bitlik bir adres yoluna sahiptir. Bu sayede 4 GB'lık bir hafıza alanını adresleme imkanı sunulmuştur. Reset, Basla ve Bitti sinyalleri modül zamanlaması ve modül ile modülün bağlı olduğu bilgisayar arasında senkronizasyonu (hand-shaking) sağlamak için kullanılır. Hafıza ile senkronizasyonu sağlamak ve veri okuma yazma işlemlerini yapabilmek için ayrıca hafıza kontrol sinyalleri bulunmaktadır.



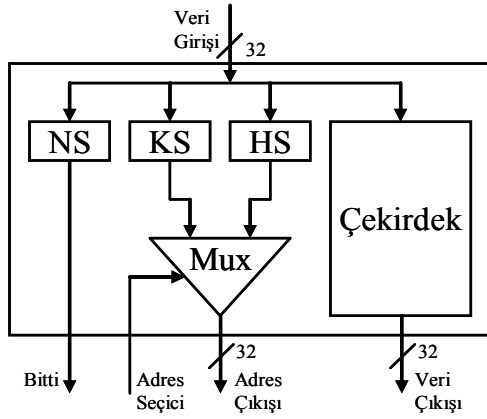
Şekil 4. Matris Çarpım modülü blok diyagramı.

Şekil 5'te matris çarpım modülünün ikinci seviye blok diyagramı görülmektedir. *Kontrol Ünitesi* ve *Veri İşlem Ünitesi* olmak üzere, modül iki bölümde tasarlanmıştır. *Kontrol Ünitesi* gerekli zamanlarda ihtiyaç duyulan kontrol sinyallerini üretirken, hafızadan verilerin alınmasını, verilerin *Veri İşlem Ünitesi* üzerinde işlenerek sonuçların hafızaya geri yazılmasını sağlar. *Veri İşlem Ünitesi* ise kontrolörden aldığı sinyallerle 4x4 ile 4x1 matris çarpımını yapan donanımı içerir.

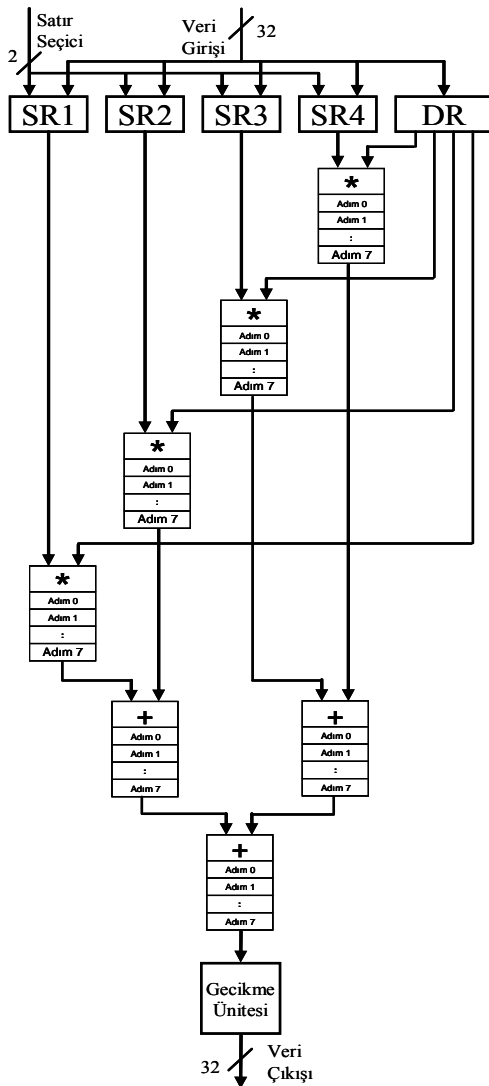


Şekil 5. Modülün ikinci seviye blok diyagramı.

Şekil 6'da *Veri İşlem Ünitesi*'nin blok diyagramı görülmektedir. *Veri İşlem Ünitesi*, *Çekirdek* ve *Veri Erişim Sayaçları* olmak üzere iki bölüm olarak tasarlanmıştır. Veri erişim kısmında üç adet sayaç kullanılmıştır. *NoktaSayacı (NS)* dönüşüme uğrayacak nokta sayısının kontrolü içindir. Modül dönüşüm işlemine başlamadan önce, dönüşüme uğrayacak nokta adedi bu yazaca yüklenir. Herbir nokta işlendikten sonra sayacın değeri bir azaltılır. Sayaca eklenen özel bir tasarım sayesinde sayaç içindeki değer sıfıra ulaştığında kontrol ünitesine *Bitti* sinyali gönderilerek dönüşümün sonlanması sağlanır. *KaynakSayacı (KS)* dönüşüme uğrayacak noktaların hafıza adresini, *HedefSayacı (HS)* ise dönüşüm sonrasında oluşan yeni nokta değerlerinin hafızada yazılacağı adresi takip etmek için kullanılmıştır. Bu sayaçların değeri dönüşüm işlemine başlamadan önce hafızadan okunur. Her bir nokta için x , y , z ve w olmak üzere hafızada dört değer bulunduğundan, her bir noktanın işlenmesi sırasında *KS* ve *HS*'nin değerleri dörder defa arttırılır. Bu bölümde yer alan multiplexer, adres seçicidir. Kontrolörden gelen *AdresSecici* sinyaline göre *KS* ya da *HS*'nin adres yoluna yazılmasını sağlar. Modül, dönüşüme uğrayacak nokta değerlerini hafızadan alarak işledikten sonra uygun zamanlama ile ürettiği sonuçları yine aynı veri yolu üzerinden hafıza ünitesine göndermektedir.



Şekil 6. Veri İşlem Ünitesi blok diyagramı.

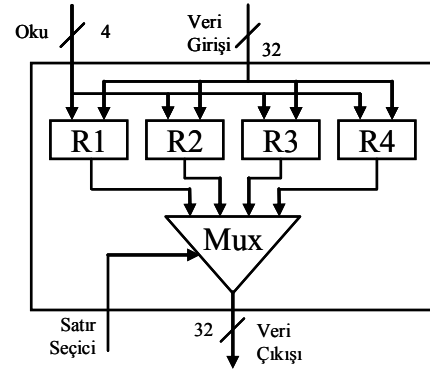


Şekil 7. Modül çekirdek blok diyagramı.

Şekil 7'de çekirdeğin blok diyagramı görülmektedir. Çekirdek, dört adet 4x32-bitlik *SabitYazac* (*SR1...SR4*), bir adet 4x32-bitlik *DonusumYazac*'ı (*DR*), dört adet 32-bitlik kesirli sayı çarpıcı ve üç adet 32-bitlik kesirli sayı toplayıcıdan oluşmaktadır. 32-bitlik kesirli sayı

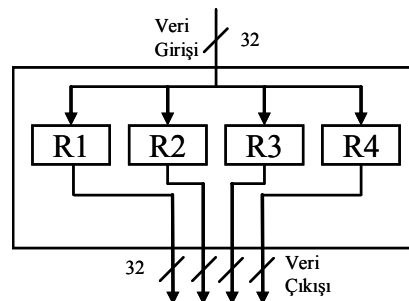
toplama ve çarpma üniteleri daha önceden tasarlanmış IEEE 754-1985 standardında işlem yapan ünitelerdir. [5, 7, 8]. Bir dönüşüm işlemi 4 saat periyodu boyunca gerçekleşir. Dönüşüm işlemi sırasında her saat periyodunda 4x4 dönüşüm matrisinin bir satırı ile 4x1 dönüşüme uğrayan nokta matrisi çarpma ünitelerine gönderilir. 4 çarpma işlemi aynı anda yapılır ve çarpma sonuçları aynı anda toplama ünitelerine ulaşır. Toplama ünitelerinde sayılar toplandıktan sonra sıra ile hafızaya yazılmak üzere çıkışa gönderilirler. Sondaki gecikme ünitesi veri giriş çıkışının senkronizasyonu için kullanılmıştır.

4x32-bitlik *SabitYazac*'lardan her birinin içinde 4 adet yazac bulunmakta ve herbiri içerisinde dönüşüm matrisinin bir sütunu tutulmaktadır. Şekil 8'de blok diyagramını verilen *SabitYazac*'ın çalışmasını her bir yazac için ayrı ayrı tasarlanmış 4-bitlik Oku ve SatırSecici sinyalleri sağlamaktadır. Gelen verinin hangi yazaca kaydedileceğini Oku sinyali, hangi yazacın içindeki değer çıkışa aktarılacağını ise SatırSecici sinyali belirler.



Şekil 8. *SabitYazac* blok diyagramı.

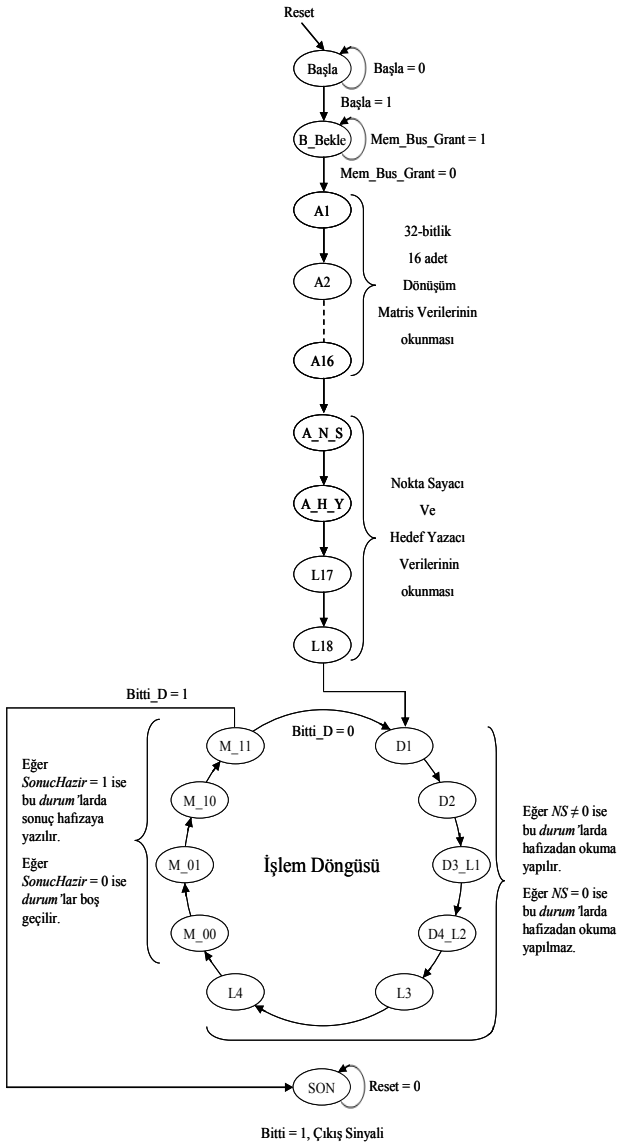
Dönüşüme uğrayacak noktaların değerleri (x, y, z, w) her seferinde bir nokta olmak üzere *DeğişkenYazac*'nda tutulur. Şekil 9'da *DegiskenYazac*'ın blok diyagramı görülmektedir. Veri yolundan gelen veriler 4-bitlik Oku sinyali sayesinde sıra ile R1...R4 yazaclarına aktarılırlar. *DegiskenYazac* 4x32 = 128-bitlik veri çıkışına sahiptir. Bu sayede yazaclarda tutulan 4 değer aynı anda çıkışa aktarılabilir.



Şekil 9. *DegiskenYazac* blok diyagramı.

3.2. Modül Kontrolörü ve Çalışması

Şekil 10'da modül kontrolörünün çalışmasını gösteren *durum* diyagramı verilmiştir. Kontrolör toplam 28 *durum*'dan oluşan bir sonlu durum makinesi olarak tasarlanmıştır. Modül ilk çalışmaya başladığında *Reset durum*'unda bekler. Bilgisayardan *Başla* sinyalini aldığı anda, *Reset durum*'undan çıkarak, hafızaya erişmek için sistem yoluna istek sinyali gönderir ve yolun kendine tahsis edilmesini bekler. Yol tahsis edildikten



Şekil 10. Modül çekirdek blok diyagramı.

sonra, kontrolör hafızada 0...0h adresine giderek ilk önce dönüşüm matrisinin değerlerini hafızadan *SabitYazac*'ta uygun yerlere okur. Bu işlem, kontrolörde toplam 18 *durum* boyunca yapılır. Okunan bu değerler bütün noktaların dönüşümü tamamlanıncaya kadar bu yazıcılar içinde tutulur. Modül, dönüşüm matrisini okuduktan sonra, hafızadan kaç tane noktanın okunacağı bilgisini ve hesaplanan yeni değerlerin hafızada kaydedileceği adres

bilgisini okur ve bunları, *NS* ve *HS* veri erişim sayaçlarına kaydeder. Bu işlemler toplam iki *durum* boyunca yapılır. Bu aşamadan sonra, modül bir döngüye girer ve döngünün her tekrarlanışında bir noktanın değerlerini hafızadan *DegiskenYazacı*'na okur, *SabitYazac* ve *DegiskenYazacı*'ndaki uygun elemanları uygun zamanda çarpıp toplayarak yeni değerleri hesaplar. Hesaplanan yeni değerler, hafızada *HedefSayacı*'nın gösterdiği yere yazılır. Döngünün her tekrarlanışından sonra *NoktaSayacı* bir azaltılır. *NoktaSayacı* sıfıra ulaştığında ise modül döngüden çıkar. Kontrolör döngünün her bir tekrarlanışını on *durum* boyunca gerçekleştirir. Döngüden çıkıldığında, kontrolör bilgisayara işlemin bittiğini belirtmek için bir kesme sinyali gönderir ve bekleme konumuna geçer. Modül bekleme konumunda tekrar *Reset* sinyali gelinceye kadar kalır. Modül içinde kullanılan toplama ve çarpma üniteleri 8 periyot kanallı olduğundan hafızadan okunan ilk değerlerin sonucu gecikme ünitesi ile birlikte toplam 30 saat darbesi sonra çıkışta görülmektedir. Gecikme ünitesi veri giriş çıkış işlemleri arasındaki senkronizasyonun sağlanması için sonuçları 6 saat darbesi boyunca bekletmekte kullanılır. Kontrolör işlem başladığı ilk 30 periyot boyunca çıkışa sonuç yazmasa da her 10 periyotta bir veri setini hafızadan okur, okunan değerleri çarpma ve toplama ünitelerine işlenmek üzere gönderir. Üniteler kanallı (pipelined) olarak tasarlandığından içlerinde aynı anda 3 noktanın değerleri sıra ile işlerler. İlk 30 periyottan sonra her 120 periyotta bir noktanın değerleri hafızadan okunur, daha önceden okunmuş ve sonucu hazır hale gelmiş başka bir noktanın değerleri hafızaya yazılır. İşlenecek noktaların okunması bittiğinde (son noktanın değerleri hafızadan okunduktan sonra) kontrol ünitesi çarpma ve toplama üniteleri içinde kalan değerler hafızaya yazılana kadar işleme devam eder.

4. DENEY DÜZENEKLERİ VE TEST SONUÇLARI

Bu çalışmada tasarlanan modül, değişik FPGA çipleri için sentezlenerek FPGA çip istatistikleri ve modülün azami saat frekansları incelenmiştir. Modülün belirlenen veriyi işleme süresi, ISE simülasyon programı [13] kullanılarak bulunmuş ve bu süre farklı PC'lerin aynı veriyi işleme süreleri ile karşılaştırılmıştır.

Tablo 1. Deneylerde kullanılan PC'lerin özellikleri.

PC Adı	CPU Hızı (GHz)	CPU Cache Bellek (KB)	RAM (MB)	Hafıza Tipi	CPU Türü	FSB BUS Hızı (MHz)	BUS Boyutu (Bit)
PC1	2.60	128	256	DDR 1	Intel Cel.	400	32
PC2	2.00	2048	512	DDR 2	AMD Athl.	667	64
PC3	1.73	2048	512	DDR 2	Intel Pent.	533	32

Modülün çalışma performansını karşılaştırmak için konfigürasyonları Tablo 1'de görülen üç adet bilgisayar seçilmiştir. Seçilen bu bilgisayarlardan PC-1 ve PC-3 tek

çekirdekli olup PC-2 çift çekirdekli AMD işlemciye sahiptir.

Tablo 2’de verilen sentezleme sonuçları göstermiştir ki, kullanılan *slice register*, *LUTs* sayılarına göre seçilen FPGA çiplerine teorik olarak modülün beş adet kopyasının aynı anda sığması mümkündür. Fakat modülde bulunan *IOB* (giriş/çıkış pinleri) yönünden düşünüldüğünde çipe iki adet modülün rahat bir şekilde yerleştirilebileceği görülmektedir.

Tablo 2. Modül FPGA çip istatistikleri.

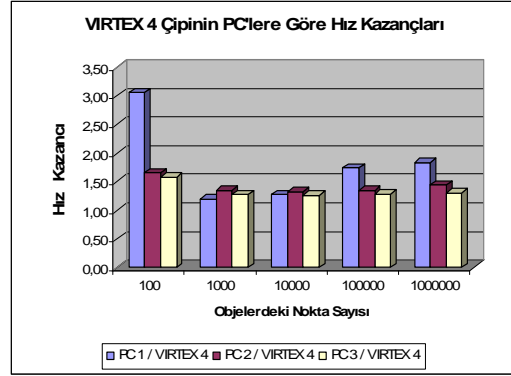
FPGA Çip Türü	Slice Reg. Sayısı / %	LUTs Sayısı / %	Slice FFs Sayısı / %	Bounded IOBs Sayısı / %	Max Saat Hızı (MHz)
<i>Virtex 4</i>	4192 / 38	9085 / 83	4422 / 40	152 / 47	200.49
<i>Virtex 5</i>	4188 / 14	5005 / 17	6288 / 21	152 / 31	288.38

Modül veri işlemlerinin kontrolü için 100, 1000, 10000, 100000 ve 1000000 noktası (*vertex*) olan nesne verileri oluşturulmuştur. Bu veriler kesirli sayı formatında olup her bir nokta için *x*, *y*, *z* ve *w* olmak üzere dört adet rastgele kesirli sayı belirlenmiştir.

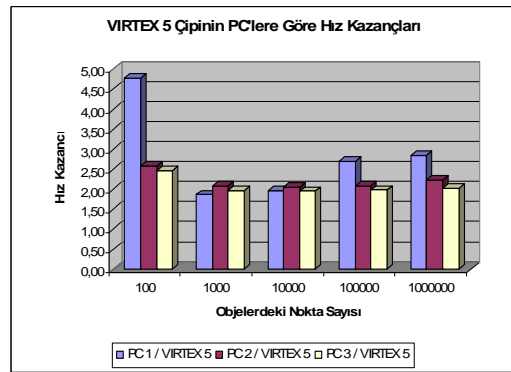
Tablo 3. PC’lerin kesirli sayı türünde oluşturulan nesne nokta verilerini işleme süreleri.

Nokta Sayısı	PC-1 (µs)	PC-2 (µs)	PC-3 (µs)
100	16.87	9.15	8.76
1000	64.99	72.87	68.96
10000	684.65	714.87	679.03
100000	9374.62	7271.61	6920.45
1000000	98939.94	77589.75	70163.72

Tablo 3’te PC’lerin deney verilerini işleme süreleri verilmiştir. Burada verilen süreler verilerin hard diskten okunması için gerekli süreyi değil, noktaların hafızadan okunması, dönüştürülmesi (matris çarpım işlemlerini) ve sonuçların hafızaya yazılması için geçen süreyi göstermektedir. Şekil 11 ve 12’de *Virtex 4* ve *Virtex 5* çiplerine göre sentezlenmiş modülün çalışma süreleri, seçilen PC’lerle karşılaştırılmıştır. Sonuçlar yorumlanacak olursa *Virtex 4* çipinde modül, PC-1’e göre yaklaşık 1.2 ile 3 kat arasında, PC-2’ye göre 1.3 ile 1.5 ve PC-3’e göre ise yaklaşık 1.2 ile 1.5 kat arasında hız kazancı sağlandığı görülmektedir. *Virtex 5* çipinde ise, PC-1’e göre yaklaşık 1.8 ile 4.7 kat arasında, PC-2’ye göre 2.0 ile 2.5 ve PC-3 göre ise yaklaşık 1.9 ile 2.4 kat arasında hız kazancı sağlandığı görülmektedir. Daha önce de belirtildiği gibi verilen hız kazançları tek bir çip üzerinde, tek bir modül çalıştırıldığında elde edilen sonuçlardır. Birden fazla çip üzerinde birden fazla modül paralel olarak çalıştırıldığında hız kazançları katlanarak artacaktır.



Şekil 11. *Virtex 4* FPGA çipine göre sentezlenmiş modülün PC’lerle karşılaştırılması.



Şekil 12. *Virtex 5* FPGA çipine göre sentezlenmiş modülün PC’lerle karşılaştırılması.

5. SONUÇ

Grafik uygulamaları ve animasyonları oluşturulurken yoğun olarak matris çarpımı işlemleri ve dönüşümleri yapılmaktadır. Bu işlemler ve dönüşümler binlerce noktadan oluşan karmaşık şekillere uygulandığında kısa bir animasyon filminin hesaplanması saatler almaktadır. Eğer animasyon gerçek zamanlı bir animasyon ise hesaplanması imkansız hale gelmektedir. Bu problemin çözümü için gelişmiş grafik kartları, grafik işlemler için tasarlanmış özel amaçlı tasarlanmış bilgisayarlar veya paralel işlemcili bilgisayarlar gibi değişik yaklaşımlar geliştirilmiştir [2, 3]. Bu çalışma da yukarıda belirtilen yaklaşımlara bir alternatif olarak, üç boyutlu grafik dönüşümleri hızlandırmak amacıyla, FPGA çipleri üzerinde çalışabilecek bir donanım modülü tasarlanmıştır. Bu yaklaşımla 3B grafik dönüşümlerinde, paralel bilgisayarlara, süper bilgisayarlara hatta özel tasarlanmış grafik kartlarına göre daha düşük maliyete sahip, normal bilgisayarlarla karşılaştırıldığında daha yüksek performans gösteren ve çok kısa sürelerde defalarca tekrar programlanabilen daha esnek bir yapı ortaya konulmuştur. Tasarlanan modül, gerçek veri üzerinde işlemler yapılarak test edilmiş ve ürettiği sonuçların doğrulanması yapılmıştır. Modülün veri işleme hızı değişik bilgisayarlarla karşılaştırılmıştır. Karşılaştırmada kullanılan PC’ler ortalama 2 GHz

civarında çalışmasına rağmen modul en fazla 288 MHz'e çıkabilmektedir. Buna rağmen karşılaştırma sonuçlarına göre modul kullanılarak kesirli sayı tabanlı üç boyutlu grafik dönüşümleri PC'lere göre 2 ile 4 kat arasında hızlandırmanın mümkün olacağı görülmüştür. Hız kazançları tek bir FPGA üzerinde tek bir modul çalıştırılarak tespit edilmiştir. Bir FPGA çipine birden fazla modülün uygun şekilde yerleştirilmesiyle ya da birden fazla FPGA çipinde modülün birden fazla kopyasını çalıştırılmasıyla hız kazancının katlanarak artacağı düşünülmektedir. Bu modul FPGA çipleri için tasarlanmıştır. FPGA çiplerinin şu anki saat hızları yaklaşık olarak 500MHz civarındadır. Modüller ASIC olarak gerçekleştirildiğinde çok daha yüksek saat hızlarına ulaşılacağı için PC'lere göre çok daha yüksek hız kazançlarına ulaşmak mümkün olacaktır.

KAYNAKÇA

1. Hearn, D. and Baker M.P. 2003. "Computer Graphics with OpenGL 3E". Prentice Hall.
2. Bensaali, F., Amira, A., Uzun, I.S. and Ahmedsaid, A. "An FPGA Implementation of 3D Affine Transformations", IEEE International Conference on Electronics, Circuits and Systems: Proceedings. Sharjah, UAE.
3. <http://www.sgi.com/>. 2007.
4. <http://www.bilesim.com/>. 2004.
5. Sahin, I. 2002. "A Compilation Tool for Automated Mapping of Algorithms onto FPGA Based Custom Computing Machines". NC State University, Doktora Tezi, Raleigh-USA.
6. Şahin, İ. ve Gloster, C.S. 2005. "Evaluation of Ic Physical Design Optimization Algorithms For Acceleration Using Fpga-Based Custom Computing Machines". İleri Teknolojiler Sempozyumu, Konya.
7. Gloster, C. and Sahin, I. 2001. "Floating-Point Modules Targeted for Use with RC Compilation Tools", NASA Earth Science Technology Conference (ESTC), College Park, MD.
8. Sahin, I., Gloster, C. and Doss, C. 2000. "Feasibility of Floating-Point Arithmetic in Reconfigurable Computing Systems", NASA Military and Aerospace Applications of Programmable Devices and Technology Conference, Washington, DC.
9. Rincon, F. and Teres, L. 1998. "Reconfigurable Hardware Systems". International Semiconductor Conference, vol. 1, pp. 45-54.
10. Sridharan, K. and Priya, T.K. 2007. "A Hardware Accelerator and FPGA Realization for Reduced Visibility Graph Construction Using Efficient Bit Representations". IEEE Transactions on Industrial Electronics, Vol. 54, No. 3.
11. Koyuncu, İ. and Şahin, İ. 2007. "Generic Fpga Modules for Integer 2D and 3D Transformations". 12th. Conference for Computer Aided Engineering and System Modeling with Exhibition. Antalya, Turkey.
12. Çetin, A. 2003. "Bilgisayar Grafikleri". Seçkin Yayıncılık.
13. <http://www.xilinx.com/>. 2008.