

Hızlı IP Arama ve Eşleştirme için SRAM Temelli Sistolik Dizilim Mimarisi

Oğuzhan Erdem¹

Cüneyt F. Bazlamaçcı²

Elektrik-Elektronik Mühendisliği Bölümü, Orta Doğu Teknik Üniversitesi, Ankara

¹e-posta: ogerdem@metu.edu.tr

²e-posta: cuneytb@metu.edu.tr

Özetçe

Bu makale, SRAM temelli, paralel çoklu boruhattı davranışı sergileyen özgün bir sistolik dizilim mimarisi önerisi sunmaktadır. İki boyutlu sistolik dizilim mimarisi üzerinde farklı uzunluklarda ve birbirini kesen boruhatları döngüsel biçimde inşa edilmektedir. Bu mimari de geleneksel ikili ağaç yapısı yerine herhangi bir öntakı arama ağacı ile birlikte kullanım da olanaklıdır. Dolayısıyla bu çalışma kapsamında ikiterimli (binomial) kapsayan ağaç yapısı kullanılarak daha avantajlı ve alternatif bir öntakı arama ağacı yapısı sunulmuştur. Tasarlanan mimari, paralellığı ve arama yeteneğini artırarak yüksek başarımlar sağlamaktadır. Sonuçlar önerilen sistemle yüksek hızlı yönlendirici tasarımlarında kullanılacak 2-7 Tbps kapasitesinde bir IP arama ve eşleştirme yeteneğine ulaşıldığını göstermektedir. Önerilen mimari, işleme elemanları açısından ölçeklenebilir olması nedeniyle IPV6 adresleme yapısı için de uygundur.

1. Giriş

İnternet kullanıcı sayısında ve bağlantı hızlarında görülen hızlı yükseliş, bu artışı karşılayabilecek daha güçlü mimarilerin geliştirilmesini gerekli kılmaktadır. İnternet kullanımının ilk yıllarında basit sınıf tabanlı adresleme yapısı yeterli iken, sınıfsız bölgelerarası yönlendirmenin (inter domain routing) gelişmesi ile birlikte IP arama, İnternet yönlendiricilerinin en önemli işlevi olmuştur. Bu yeni yaklaşım iki önemli avantajı getirmiştir. İlk olarak bu öntakılar farklı uzunlukta olabilmekte ve bu sayede adres alanı çok daha etkin bir biçimde kullanılabilir. İkinci olarak, ağların gelişigüzel kümelenebilmesi sağlanarak, IP başvuru çizelgesindeki (IP lookup table) kayıt sayısı azaltılmaktadır. Bu yaklaşımda herhangi bir address, öntakı örtüşmesi sebebiyle IP başvuru çizelgesinde bulunan kayıtların birden fazlasıyla eşleşebilir. Bu durumda yönlendirici en doğru karar olarak en özel eşleşmeyi bulmalıdır. En özel eşleşmeyi bulmak tam eşleşmenin bulunmasından daha zordur. Bunun nedeni gelen bir paketin hedef adres alanında en özel eşleşme uzunluğu hakkında herhangi bir bilginin bulunmamasıdır. Literatürde, tablodan IP eşleşmesine yönelik olarak önerilen çözümler yazılım ve donanım tabanlı olmak üzere iki sınıfa ayrılmaktadır [1,2]. Bu çözümlerin başarımlarını ölçebilmek için ise, toplam bellek erişim sayısı, toplam kullanılan bellek miktarı, güncelleme süresi, ölçeklenebilirlik ve uygulamadaki esneklik gibi bazı parametreler önemlidir. Donanım tabanlı çözümlerde içerik adreslenebilir bellek türleri de kullanılmaktadır [3,4]. Bu tür belleklerde bir saat periyodunda tüm kayıtlar ile karşılaştırma yapılabilen ve en özel

eşleşme bulunabilmektedir. Son yıllarda çok ilgi gören bu bellek elemanlarının yüksek fiyat ve yüksek güç gereksinimleri ise dezavantajlardır. Ölçeklenebilir olmamaları ve uzun güncelleme süresi ise diğer dezavantajlardır [5,6,7]. Öte yandan SRAM tabanlı mimariler bellek yoğunluğu, güç kullanımı ve hız açısından içerik adreslenebilir belleklere göre daha avantajlıdır. Ancak en özel eşleşmeyi bulma işlevi sırasında gerçekleşen toplam bellek erişim sayısı kullanılan ağaç yapısının ortalama derinliğine bağlıdır. Tek SRAM'e kayıtlı ağaç içerisinde yapılan arama ve eşleşme işlemi sırasında çok sayıda bellek erişimine gereksimin duyulur. Bu nedenle son yıllarda başarımlarını arttırabilmek için SRAM tabanlı boruhattı davranışı içeren mimariler de geliştirilmektedir [8,9,10,11,12,13,14,15].

Bir sistolik dizilim (systolic array) çoklu ve birbirini kesen boruhatları oluşturulmak istendiğinde doğal bir çözüm olarak ortaya çıkar. Paralel çoklu boru hattı davranışı sergileyen SRAM tabanlı özgün bir sistolik dizilim mimarisi ikili ağaç yapısı kullanılarak SAAFFIL adıyla [16]'da, ikiterimli kapsayan ağaç yapısı kullanılarak SAAFFIL-BT adıyla [17]'de geliştirilmiş, ayrıntılar ve aşağıda listelenen özgün katkılar sunulmuştur:

- Farklı uzunluklarda ve birbirini kesen boruhatları, işlem bloklarından oluşan iki boyutlu sistolik dizilim mimarisi üzerinde döngüsel (circular) bir biçimde inşa edilmiştir. Döngüsel yapı boruhattı uzunluğunda herhangi bir kuramsal üst sınırın olmadığı anlamına gelir.
- Herbiri m SRAM'den oluşan n tane paralel boruhattı içeren mimari yerine $\sqrt{nm} \times \sqrt{nm}$ şeklinde düzenlenmiş, birbirini kesen ve döngüsel boruhatları içeren bir mimari kullanılmış ve bu mimariye uygun ve etkin bir eşleme (mapping) yapısı önerilmiştir.
- Birbirini kesen boruhatlarındaki kaynak paylaşım çatışmasını çözmek için her bir işlem bloğunda bir kuyruk düzeneği kullanılmıştır.
- İki terimli kapsayan ağaç yapısı kullanarak IP tablosu oluşturma yöntemi ve sistolik dizilim mimarisi üzerine eşleştirme yöntemi geliştirilmiştir.
- Önerilen mimari için bir IP arama yöntemi ortaya konulmuştur.

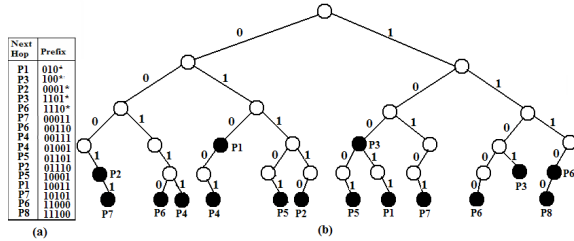
Bölüm 2 ikili ağaç üzerinde IP arama ve eşleştirme işlemi hakkında temel bilgiden sonra hızlı IP arama ve eşleştirme için önerilen sistolik dizilimli özgün mimariyi ve ardından da önerilen mimarinin ikiterimli kapsayan ağaç (binomial spanning tree) yapısı ile kullanımını sunmaktadır. Bölüm 3

temel yapı üzerinde gerçekleştirilecek başarımların artırılmasına yönelik çeşitli mimari uzantı olanaklarını vermektedir. Benzetim yöntemi, sonuçlar ve başarımların düzeyi Bölüm 4'te, genel değerlendirme, sonuçlar ve gelecek çalışmalar ise Bölüm 5'te sunulmaktadır.

2. Hızlı IP Arama ve Eşleştirme için Sistolik Dizilim Mimarisi

2.1. İkili Ağaç Temelli IP Arama ve Eşleştirme İşlemi

IP arama ve eşleştirme için önerilen yazılım ve donanım temelli çözümler için kullanılan en yaygın veri yapısı ikili (binary) ağaçtır. Öntakılar ikili ağaç yapısındaki düğümlerdir. İkili ağaç yapısının kök düğümünden ilgili uç ya da ara düğüm kadar olan bit dizimi öntakıyı oluşturur. Ağaç üzerinde bir düğümden diğerine geçiş kararı öntaki içerisindeki bitlere bakılarak yapılır. Şekil 1'de örnek bir öntaki tablosu ve buna karşılık gelen ikili ağaç yapısı verilmiştir.



Şekil 1: (a) Öntaki tablosu (b) Karşılık gelen ikili ağaç yapısı

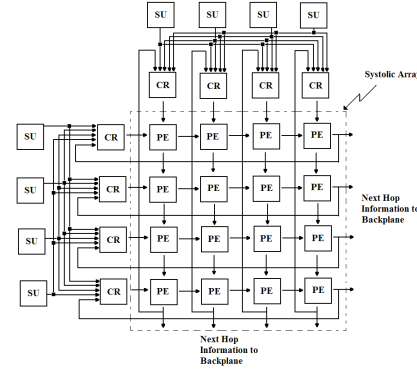
IP arama ve eşleşme işlevi aranan IP adres bitleri kullanılarak ağaç üzerinde ilerleyerek gerçekleşir. Uç düğüme ulaşıldığında ya da boş bir işaretçi (pointer) ile karşılaşıldığında arama ve eşleşme işlemi biter.

2.2. Hızlı IP Arama ve Eşleştirme Amaçlı Mimari

Bir sistolik dizilimde işlem elemanları iki boyutlu bir matris biçiminde yerleştirilir. Her bir işlem elemanı (processing element - PE) veriyi işledikten sonra komşularına iletir. Dış dünya ile bağlantı sadece sınırdaki işlem elemanları üzerinden gerçekleşir. Sistolik dizilim mimarilerinin eş zamanlılık, düzenlilik, paralel boruhatları içerme ve modüler genişleyebilirlik gibi istenen özellikleri vardır.

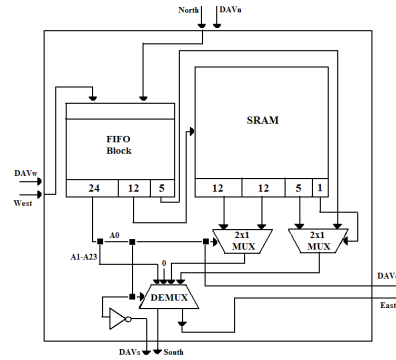
Doğal olarak içerisinde paralel boruhatları içeren sistolik dizilim yapısı hızlı IP arama ve eşleştirme için iyi bir çözümdür. Bu nedenle bu çalışmada arama ve eşleştirme işlevinin başarımlarını artırmak için Şekil 2'de gösterilen sistolik dizilim mimarisi önerilmektedir. Bu yaklaşımla, literatürde yer alan SRAM tabanlı tek boyutlu boruhatları içeren mimarilerin aksine, her bir arama işlemi için çok yönlü bir veri akışı gerçekleşir ve bu eşleşme sonucu mimari içerisinde herhangi bir işlem bloğundan dış dünyaya verilerek, boruhatlarının daha etkin serbest kalma olasılıkları sayesinde daha etkin kullanımlarını sağlar. Genel bir sistolik dizilim yapısı, geliştirilerek her bir satır ve sütundaki uç işlem elemanları arasında bağlantı kurularak dairesel (circular) bir yapı oluşturulmuştur. Eğer birden fazla arama ve eşleştirme işlemi aynı giriş işlem elemanını kullanmak isterse kaynak çatışması ortaya çıkar. Şekil 2'de yer alan çatışma çözücü (contention

resolver - CR) girişteki arama ve eşleştirme istekleri arasındaki bu kaynak çatışmasını çözmek için kullanılmaktadır. CR kaynak talep edenler arasından sadece bir tanesini seçerek ilgilinin isteğini sistolik dizilim içerisine iletir, diğer istekler ise seçici ünite (selector unit - SU)'de bir sonraki saat için bekletilir. SU aynı zamanda giriş işlem elemanını belirlemek için kullanılır.



Şekil 2: 4x4 sistolik IP arama ve eşleştirme mimarisi.

Her bir işlem elemanı, bir FIFO kuyruk, bir SRAM bellek ve ek mantık devrelerinden oluşur (Şekil 3).



Şekil 3: İşlem elemanı (PE)

2.3. Sıkışıklık Kontrol Mekanizması

Birbirini kesen boruhatlarındaki kaynak paylaşım çatışmasını çözmek için her bir işlem bloğunda bir kuyruk düzeneği kullanılmıştır. Sonlu uzunluktaki herhangi bir FIFO kuyruk düzeneğinde meydana gelebilecek taşmanın önlenmesi için bir sıkışıklık kontrol mekanizması geliştirilmiştir. Herhangi bir kuyruk düzeneğindeki yoğunluğun kuyruk eşik değerini aşması halinde mimarinin girişinde yer alan seçici ünitelerin yarısı devre dışı bırakılarak gelen trafiğin kısılması ve o kuyruktaki yoğunluğun azaltılarak taşmadan kaynaklanabilecek paket miktarının önlenmesi hedeflenmektedir. Kuyruk yoğunluk miktarının eşik değerinin üzerinde olduğu her saat diliminde sıkışıklık kontrol mekanizması devreye girerek aktif SU sayısını yarıya düşürdüğü gibi, yoğunluğun eşik değerin altında kaldığı durumlarda ise her bir saat diliminde bir seçici ünite daha aktif hale getirilerek başlangıçtaki SU sayısına yeniden ulaşılması hedeflenmektedir.

sayısından, ikinci altağaçta yer alan adreslerinde ise 1 olan bitlerinin sayısı 0 olan bitlerin sayısından daha fazladır. Bir başka deyişle 00000 adresine yakın adresler birinci bölümde 11111 adresine yakın adresler ise ikinci bölümde yer almaktadır. Hem 1'lerin yoğun olduğu hemde 0'ların yoğun olduğu adresler ile eşleşme olabilecek bazı öntakılar ise her iki bölümde de yer alabilir. Bölüntüleme işlemi sayesinde arama alanı ikiye bölünmekte, bu şekilde paralel arama ve eşleştirme yapma olanağının artması hedeflenmektedir. Bölümlerdeki prefix yoğunluklarının eşitsizliğinin söz konusu olduğu durumlarda bölüntüleme seviyesi değiştirilerek diğer bir deyişle aşağıya ya da yukarıya kaydırılarak prefix yoğunlukları dengelenebilir. Mimarinin başarım ölçümlerinde altağaç prefix yoğunluk dengesi gözetilerek en uygun bölüntülenme seviyeleri belirlenmiştir. Bölümlerden biri batıdan diğeri de kuzeyden başlayarak mimariye eşleme yapılır.

Bölüntüleme yanında ikili ağaç yapısından ikiterimli kapsayan ağaç yapısına dönüşümde başka modifikasyonlara da ihtiyaç vardır. Eşleştirme tablosu içinde yer alan tüm öntakıları sabit uzunlukta adreslere dönüştürmek için tümünün sonuna yeterli sayıda 0 eklemek yerine içerisinde 1 bitlerinin sayısı daha fazla olan öntakılara 1 eklemek daha yararlıdır. Bu durumda birinci altağaçta yer alabilecek öntakıların sonuna 0, ikinci altağaçta yer alabilecek öntakıların sonuna ise 1 bitleri eklenmelidir. Örneğin, 10111* ön takısı 10 bitlik adres alanında 1011100000 yerine 1011111111 adresine dönüşmelidir. Dönüşümü yapılan adres ilk alt ağacın dördüncü seviyesi yerine ikinci alt ağacın birinci seviyesine kayıt edilmiş olur. Sonuç olarak ağacın simetrik yapısı sayesinde örnekte verilen öntakıya ulaşmak için gereken bellek erişim sayısı azalmış olur. Şekil 6'da bir ikiterimli kapsayan ağaç yapısının örnek bölüntülenmesi gösterilmektedir

Ağaç yapısını mimariye eşlemek için kullanılan birinci-çocuk-sonraki-kardeş (first-child-next-sibling) gösterimi uyarınca her bir düğüm en sağ bir alt seviye çocuk düğümüne ve en yakın aynı seviye sol düğümüne bağlıdır. Bu yapı ile ikiterimli kapsayan ağaç yapısının kısa arama yolları oluşturma yönündeki avantajı azalmakla birlikte yine de ikili ağaç yapısına göre daha avantajlıdır.

2.5.2. İkiterimli kapsayan ağaç yapısında arama ve eşleştirme

İki terimli kapsayan ağaç yapısı üzerinde arama ve eşleştirme işlemi girişteki seçici ünitelerden başlar. Seçici üniteler basit ikili ağaç yapısındaki ünitelerden farklı olarak giriş işleme elemanını belirlemek için ilgili aranan adresteki ilk r bitin yanında kalan $32-r$ bit adres içerisinde ilk 1 bitinin ya da ilk 0 bitinin pozisyonun bilgisini kullanır. Eğer adres arama işlemi birinci bölümde gerçekleştirilecek ise ilk 1 bitinin aksi halde ilk 0 bitinin yeri kullanılır. Aranan IP adresindeki 0 ve 1 bitlerinin yoğunluğuna bağlı olarak arama işlemi birinci ya da ikinci altağaçta gerçekleştirilir. Birinci altağaçta arama yapılacak ise batı yönündeki aksi halde kuzey yönündeki giriş işleme elemanlarından yapıya giriş sağlanır. Örneğin, adresin ilk r biti dışındaki içeriği 00110000 ise ilk 1 bitinin yerine bakılmalıdır çünkü adres içerisinde 0 bitlerinin sayısı 1 bitlerinin sayısından fazladır ve arama işlemi birinci altağaçta gerçekleşecektir. Bu durumda ilk 1 bitinin yeri de ikidir. Bu değer ilk r bitin değeri ile birlikte kullanılarak mimariye giriş işlem elemanının indeksi belirlenir. Ayrıca soldan ilk 1 ya da 0 bitinin yanı sıra sağdan ilk 1 ya da 0 bitinin yeri de hesaplanmalıdır. Bununla IP adresinin arama ve eşleştirme

işlemi için mimari içerisinde en fazla kaç tane düğümün ziyaret edileceği bilgisini oluştur. İkili ağaç yapısından farklı olarak, bu yeni yaklaşımda sistolik dizilim içinde aramayı yönlendirecek olan bitler soldan ve sağdan ilk 1'ler ya da soldan ve sağdan ilk 0'lar arasında kalan bitlerdir. Eğer içinde bulunulan aşamaya karşılık gelen bitin değeri 1 ise arama ve eşleştirme işlemi doğu yönündeki işleme elemanı, eğer 0 ise güney yönündeki işleme elemanı üzerinden devam eder. Arama ve eşleştirme işlemi en özel eşleşmenin bulunması ile sona erer. Arama ve eşleştirme süresince ziyaret edilen her bir işleme elemanında eğer geçerli öntakı kayıtlı ise çıkış kapı numarası alınır ve sonraki safhaya taşınır. En son ziyaret edilen işlem elemanında geçerli kapı numarası yoksa yani bu düğüm geçerli bir öntakı değilse, adresin birinci ya da ikinci altağaçta yer almış olmasına bağlı olarak en son 1 ya da 0 geçisi öncesinde güncellenen kapı numarası geçerlidir.

3. Mimari Uzantılar

Bu bölümde yukarıda önerilen temel yapı üzerinde uygulanabilecek olası ek özellikler önerilmektedir. Sunulan uzantıların uygulanması sonucunda gözlenen başarım değerleri Bölüm 4'te verilmektedir.

3.1. Önbellek Kullanımı

Paralel mimarilerde trafiğin bir yöne doğru olması eğiliminden kaynaklanacak başarım düşüşünün önüne geçebilmek için önbellek yapıları da kullanılmaktadır. Paralel boruhattı davranışı sergileyen mimaride de trafiğin bir sürede olsa herhangi bir boruhattına yönelmesi sonucu mimari yapı paralel çoklu boruhatları gibi değil tek bir boruhattına sahip yapılar gibi çalışır. Bu tip trafik yönelimlerinin yoğun olduğu durumlarda başarım düşüşünü önlemek için önbellek kullanmak en yaygın çözümdür. Bu amaçla kullanılacak iki çeşit önbellek yapısı önerilmiştir. Seçici ünitelerde küçük önbellek yapıları kullanarak en çok tercih edilen öntakıları kaydetmek ilk yöntemdir. Aranan adres öncelikle bu küçük önbelleklerde aranır ve eğer eşleşme gerçekleşirse doğrudan çıkış aktarılır aksi halde sistolik mimariye giriş yapar. İkinci yöntem ise akış önbellek yapısıdır. Bu yöntemde de önbelleğe yazılacak olan öntakı arama sonucunun oluşmasını beklenmez. Bu yöntemin uygulanması ise maliyetli ve karmaşık bir yapı gerektirmektedir. Bu çalışma kapsamında söz edilen ilk yöntem kullanılmıştır.

3.2. Çift Giriş-Çıkış SRAM Kullanımı

Önerilen mimari çift giriş-çıkışa sahip SRAM bellek elemanları ile de kullanılabilir. Bu durumda bir saat periyodunda herhangi bir SRAM üzerinde iki farklı arama ve eşleştirme işlemi aynı anda gerçekleşir. Bu durumda bellek gereksiniminde herhangi bir değişiklik olmaksızın sadece mantık devrelerinde küçük değişiklikler yeterlidir.

3.3. 4-Yönlü Giriş Kullanımı

Mimarinin yapısı iki taraflı (batı ve kuzey) giriş yanında dört taraflı yani tüm yönlerden girişe de olanak tanımaktadır. Bu uzantı ile birlikte alternatif giriş noktalarının sayısı artmaktadır ve bu artış ile birlikte de başarım iki katına çıkmaktadır. Bu durumda birinci altağaçta gerçekleşecek arama işleminin hangi yönde (batı yada doğu) aynı şekilde ikinci altağaçta gerçekleşecek arama işleminin hangi yönde (kuzey yada güney) olacağı belirlenmesi için IP adresinin ilk 8 biti

kullanılabilir. IP adresinin ilk 8 bitinin değeri belirli bir seviyenin altında ise batı (kuzey) yönünde aksi halde doğu (güney) yönünde girişi sağlanabilir. Başarım ölçümlerinde mimariye giriş yük dengesi gözetilerek en uygun seviyeler belirlenmiştir. Bu yöntemin uygulanabilmesi için işlem elemanları arasında iki yönlü bağlantılara ve ek mantık devrelerinde küçük değişikliklere ihtiyaç vardır. Bu yöntem ile işlem elemanlarına düşen yük de daha dengeli dağılmaktadır.

3.4. İkiterimli Kapsayan Ağaç Yapısının Üç İşaretçili Kullanımı

İki işaretçili birinci-çocuk-sonraki-kardeş gösterimi yerine üç işaretçili kullanım ile arama derinliği daha da azaltılabilmektedir. Bu yöntem ile kullanılan bellek miktarı azaltılabılır ancak her bir işlem elemanında kullanılan mantık devresi bir miktar değişmektedir. Üç işaretçili gösterimde üçüncü işaretçinin hangi çocuk düğümüne işaret edeceği kullanıcının kontrolindedir. Bu yöntem ile arama ve eşleştirme işleminin uzunluğu anlamında çok önemli bir avantaj kazanılmakta ve genel başarım düzeyi yükselmektedir.

4. Benzetim Çalışması ve Başarım

Önerilen mimarinin simülasyonu Visual C++ kullanılarak yazılan program ile gerçekleştirilmiştir. Ticari, etik ve güvenlik kaygıları nedeniyle gerçek paket trafiği ve bunlara uygun öntakı tabloları açık kaynaklı olarak bulunmamaktadır. Ya gerçek IP trafik dizgisi bulunabilmekte ya da gerçek öntakı tabloları elde edilebilmektedir. Bu nedenle literatürde genellikle, gerçek IP trafik dizgisi kullanılarak öntakı tablosu ya da gerçek öntakı tabloları kullanılarak IP trafik dizgisi oluşturulmaktadır. Bu çalışma kapsamında her iki yöntemde kullanılarak ölçümler gerçekleştirilmiştir. Birinci yöntem doğrultusunda gerçek öntakı tabloları [20] kullanılarak içerisindeki çoğuşma (burst) uzunlukları farklı üç ayrı paket trafiği oluşturulmuş ve bu trafik örnekleri kullanılarak başarım ölçümü gerçekleştirilmiştir. Bu durumda oluşturduğumuz örnek T1, T2 ve T3 paket trafiklerinde çoğuşma miktarı $[1, k]$ $k=2,5,10$ aralığında birbiciimli dağılıma (uniform distribution) göre seçilmiştir. Bir diğer yöntem ile gerçek IP trafik dizgileri [21] örnek olarak alınmış, gerçek öntakı uzunluk dağılımları kullanılarak uygun öntakı tabloları oluşturulmuş ve bu örnekler (T4 ve T5) üzerinde başarım ölçümleri gerçekleştirilmiştir.

Benzetim çalışmamızda 16×16 sistolik dizilim mimarisi, 32 adet seçici ünite ve her bir girişte bulunan 32 adet CR elemanı kullanılmıştır. Birden fazla paket aynı giriş elemanına yönlendirilse de bunların yalnızca bir tanesi sisteme giriş yapabilmekte diğerleri seçici ünitelerde bir sonraki saat periyodu (clock cycle) için bekletilmektedirler. Benzetim çalışması hem önbellek hem de önbellek olmaksızın gerçekleştirilmiştir. Her bir işlem bloğunda bulunan FIFO kuyruk düzeneğinin sıkışıklık kontrolü eşik değeri 10 ve uzunluğu 50 olarak varsayılmıştır. Başlangıç bölüntülenmesi için r değeri 8 olarak seçilmiştir. Bölümlerdeki prefix yoğunluk dengesinin eşitlenmesi amacıyla bölüntülenme seviyesi en uygun değer olarak ağacın 7. seviyesi 4-yönlü uzantılı mimaride ise ağacın 5. seviyesi seçilmiştir. Mimarinin 4-yönlü giriş uzantısı ile gerçekleştirilen başarım ölçümlerinde arama işleminin yönü en uygun yük dağılımı gözetilerek IP adresinin ilk 8 bitinin karşılaştırılacağı seviye olarak 85 değeri seçilmiştir. Öntakı tablosu öncelikle MIPS algoritmasına uygulanmış ve birbirinden bağımsız öntakı seti elde edilmiştir.

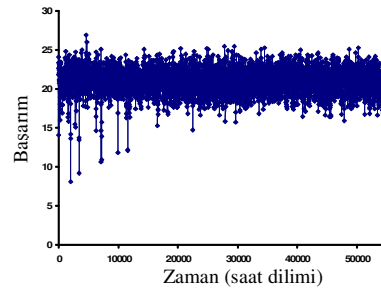
Bu algoritma ile öntakı tablosunda yaklaşık %50 oranında sıkıştırma sağlanmıştır. Elde edilen bağımsız öntakılar kullanılarak ikiterimli kapsayan ağaç yapısı elde edilmiş ve mimariye eşleşmiştir. Tablo 1'de farklı paket trafik örnekleri altında önerilen mimarinin (SAAFFIL-BT) başarım sonuçları sunulmaktadır. Tabloda verilen çıkan iş oranı (throughput) değeri bir saat periyodunda yapılabilen toplam arama ve eşleştirme işlemi sayısını vermektedir.

Tablo 1: Başarım (birim zamanda iş çıktısı) (arama eşleştirme miktarı/zaman)

Mimari Uygulama Yöntemi	T1	T2	T3	T4	T5
SAAFFIL-BT	8,95	6,56	4,61	7,83	7,12
SAAFFIL-BT- öb	8,96	6,61	4,97	10,16	9,79
SAAFFIL-BT- ç	17,66	14,07	10,07	16,04	14,42
SAAFFIL-BT- öb/ç	17,72	14,03	10,34	19,00	19,86
SAAFFIL-BT- ç/d	23,60	19,64	13,08	19,38	16,19
SAAFFIL-BT- öb/ç/d	23,60	19,77	13,42	19,83	22,68
SAAFFIL-BT- ç/d/ü	29,95	21,40	14,17	20,95	16,44
SAAFFIL-BT- öb/ç/d/ü	29,95	21,57	14,84	26,22	27,25

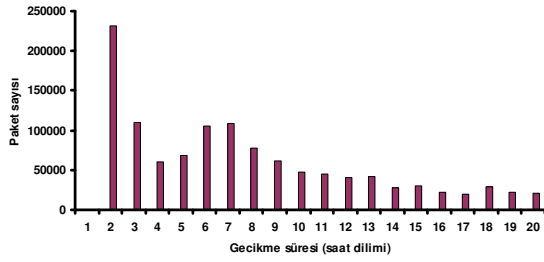
öb: önbellekli ç: çift giriş-çıkışlı d: 4-yönlü ü: 3 işaretçili

Sonuçlar, ilk yöntem kullanılarak oluşturulan trafik örneklerinde mimarinin başarımının daha düşük olduğunu gerçek trafik örneklerinde ise çok daha yüksek olduğunu göstermektedir. Bu durum gerçek trafik örneklerinin mimari üzerine dağılımlarının paralellikten daha fazla yararlanılmasına sebep olduğuna işaret etmektedir. Ayrıca kullanılan önbellek yapısının gerçek trafik örneklerinde başarımı oldukça artırdığı diğer örneklerde ise çok fazla etkili olmadığı gözlenmektedir. Gerçek trafik örneğinde çoğuşmanın belirli zamanlarda olması diğer bir deyişle ilk yöntem ile oluşturulan trafik örneklerinde olduğu gibi sürekli olmaması önbellek yapısından daha etkili bir biçimde faydalandığımızı göstermektedir. 3-ışaretçili kullanımda hoplama sayısının azalması ile sonucun önbelleğe yazılması için geçen süre kısılacığından önbellek kullanımının etkisi daha da artmıştır. Şekil 7'de 4-yönlü çift giriş çıkışlı SRAM kullanılarak 3-ışaretçili uygulama ile önbelleksiz mimari için birim zamandaki iş çıktısı dalgalanması verilmektedir. Örnek benzetim çalışmamızda 57.416 saat dilimi süresince eşik değerini aşma sayısı 55 olarak gözlenmiştir.



Şekil 7: Başarım dalgalanması

Şekil 8'de 4-yönlü çift giriş çıkışlı SRAM kullanılarak 3-ışaretçili uygulama ile önbelleksiz mimari için paket gecikme dağılımı verilmiştir ve yaklaşık ortalama gecikme 6 saat dilimi olarak ölçülmüştür.



Şekil 8: Gecikme histogramı

Deney sonuçlarına göre toplam 256 işlem biriminin herbiri için en çok 4.000 kayıt tutabilecek SRAM bellek elemanına ihtiyaç vardır. Her bir kayıt iki işaretçi tutacağından toplam $12+12=24$ bit işaretçi alanı yeterlidir. Buna ek olarak port numarası tutmak için de 5 bit kullanılabilir. Böylece her bir kayıt 30 bit uzunluğunda olmalıdır. Bu durumda toplam kullanılacak bellek miktarı $256 \times 2^{12} \times 30 = 3.93$ Mbit'tir. Kullanılan her bir kuyruk düzeneği için de 256 byte bellek yeterlidir. Mimaride kullanılan saat hızı tamamen bellek elemanlarına yapılan erişim zamanı ile ilintilidir. Mimaride bir saat periyodu içerisinde kuyruk düzeneğinden okuma ve SRAM elemanından da okuma yapılır. Bir saat periyodu 65nm teknoloji altında 1,2ns olarak literatürde yer almaktadır [22]. Mimarinin genel başarımlı düzeyinin ise, ikinci yöntemle gerçek trafik örneği kullanıldığında en basit önbelleksiz yapıda ortalama iş çıkışı değeri olarak 8 alınarak, saniyede 6,67 paket işleme kapasitesi olduğu görülür. Bu seviye en küçük IP paket uzunluğu olarak 40 byte alındığında yaklaşık 2 Tbps kapasitesinde bir yönlendirici için IP arama ve eşleştirme yeteneğine ulaşıldığı göstermektedir. Önbellek kullanımı, çift giriş/çıkışlı SRAM kullanımı, 4-yönlü giriş ve 3-işaretçili kullanım gibi mimari uzantılarının kombinasyonları ile ortalama iş çıkışı artmaktadır. Benzetim çalışmalarında trafik örnekleri için en yüksek değer olarak 27 gözlenmiştir [Tablo 1]. Çift ve tek giriş/çıkışlı SRAM erişim hızlarının aynı olduğu varsayımıyla mimari 7 Tbps hızlarına ulaşmaktadır.

5. Sonuç

Bu çalışmada paralel çoklu boruhattı davranışı sergileyen SRAM tabanlı özgün bir sistolik dizilim mimarisi önerilmiştir. Bu yapıda farklı uzunluklarda ve birbirini kesen boruhatları, işlem bloklarından oluşan iki boyutlu sistolik dizilim mimarisine döngüsel bir biçimde inşa edilmektedir. Mimarinin ikiterimli kapsayan ağaç yapısı ile kullanımına yönelik tasarımlar da sunulmuş ve gerekli eşleme yaklaşımı verilmiştir. Başarım artırımına yönelik temel yapı üzerinde gerçekleştirilecek çeşitli mimari uzantılar da olanaklıdır. Sonuçlar önerilen sistemle 2-7 Tbps kapasitesinde bir yönlendirici gerçekleştirilecek düzeyde bir IP arama ve eşleştirme yeteneğine ulaşıldığı göstermektedir.

6. Kaynakça

[1] M.A.R.Sanchez, E.W.Biersack ve W.Dabbous, "Survey and taxonomy of IP address lookup algorithms," *IEEE Network*, c.15 s.8-23, 2001.

[2] H.J.Chao, "Next generation routers," *Proc. of IEEE*, c.90, s.1518-1558, 2002.

[3] M.J.Akhbarizadeh, M.Nourani ve C.D.Cantrell, "Prefix segregation scheme for a TCAM based IP forwarding engine," *IEEE Micro*, c.25, s.48-63, Ağustos 2005.

[4] M.Kobayashi, T.Murase, ve A.Kuriyama, "A longest prefix match search engine for multi-gigabit IP processing," in *Proc. of IEEE International Conference on Communications*, c.3, s.1360-1364, Haziran 2000.

[5] V.C.Ravikumar, R.N.Mahapatra ve L.N.Bhuyan, "Ease CAM: An energy and storage efficient TCAM-based router architecture for IP lookup," *IEEE Transactions on Computers*, c.54, s.521-533, 2005.

[6] K.Pagiamtzis ve A.Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE Journal of Solid-State Circuits*, c.41, s.712-727, Mart 2006.

[7] V.C.Ravikumar ve N.Mahapatra, "TCAM architecture for IP lookup using prefix properties," *IEEE Micro*, c.24, s.60-68, Nisan 2004.

[8] S.Kumar, M.Becchi, P.Crowley ve J.Turner, "Camp: fast and efficient IP lookup architecture," in *Proc. of ANCS'06*, s.51-60, 2006.

[9] W. Jiang, Q. Wang ve V.K. Prasanna, "Beyond TCAMs: an SRAM-based parallel multi-pipeline architecture for terabit IP lookup" in *Proc. of IEEE INFOCOM'08*, s.2458-2466, 2008.

[10] W. Jiang ve V.K. Prasanna, "Multi-terabit IP lookup using parallel bidirectional pipelines" in *Proc. of Computing Frontiers*, s.241-250, 2008.

[11] W. Jiang ve V.K. Prasanna, "A memory-balanced linear pipeline architecture for trie-based IP lookup", in *Proc. of IEEE Symposium on High-Performance Interconnects HOTI'07*, s.83-90, Ağustos 2007.

[12] W. Jiang ve V.K. Prasanna, "Parallel IP lookup using multiple SRAM based pipelines," in *Proc. of IEEE IPDPS'08*, s.1-14, Nisan 2008.

[13] H.Le, W.Jiang ve V.K. Prasanna, "Scalable High throughput SRAM-based architecture for IP lookup using FPGA", in *Proc. of FPL'08*, s.137-142, Eylül 2008.

[14] W.Jiang ve V.K. Prasanna, "Multi-way pipelining for power efficient IP lookup," in *Proc. of IEEE GLOBECOM'08*, s.1-5, Aralık 2008.

[15] W. Jiang ve V.K. Prasanna, "Towards green routers: Depth bounded multi pipeline architecture for power efficient IP lookup", in *Proc. of IPCCC'08*, s.185-192, Aralık 2008.

[16] O. Erdem ve C.F. Bazlamaçcı, "A novel multi-pipeline IP lookup architecture", *IEEE Communications Letters*, (değerlendirme aşamasında), Kasım 2009.

[17] O. Erdem ve C.F. Bazlamaçcı, "Systolic array architecture for fast IP lookup", *IEEE Transactions on Computers* (değerlendirme aşamasında), Kasım 2009.

[18] Y.K.Chang, "Simple and fast ip lookups using binomial spanning trees," *Computer Communications*, c.28, s.529-539, 2005.

[19] Erdem, O ve Bazlamaçcı, C.F., "MIPS extension for a TCAM based parallel architecture for fast IP lookup", in *Proc. of ISCIS'09*, s.185-192, Eylül 2009.

[20] "BGP Routing Table Analysis Reports" <http://bgp.potaroo.net>

[21] "AMPATH-I Traces" <http://pma.nlanr.net>

[22] "CACTI 5.3" <http://quid.hpl.hp.com:9081/cacti>