

**BİLGİSAYAR MİMARİSİNDE YENİ
YAKLAŞIMLAR**

DÖNEM PROJESİ

ARM11
MİMARİSİ

Ersin ÖZKAN

704071008

İçindekiler

1. Giriş	3
1.1 ARM Hakkında Genel Bilgi	3
1.2 ARM11 Hakkında Genel Bilgi	4
1.2.1 ARMv6 Hakkında Genel Bilgi	4
2. ARM11 Mimarisinin Ayrıntılı İncelenmesi	5
2.1 Saklayıcılar ve Çalışma Modları	5
2.2 Adresleme Yöntemleri	7
2.3 Güvenlik Uygulamaları	7
2.4 Performans	8
2.5 ARM DSP	8
2.6 Jazelle™	9
2.7 SIMD	9
2.8 VFP	10
2.9 Thumb®	10
2.10 İş Hattı	11
2.10.1 İş Hattında Paralellik	12
2.11 Dallanma	12
2.11.1 Dallanma Tahminleri	12
2.12 Bellek	13
2.12.1 Bellek Mimarisi	13
2.12.2 Bellek Organizasyonu	14
2.12.3 MMU	14
2.12.4 TLB	14
2.12.5 Yazma Tamponu	15
2.12.6 TCM	15
2.13 Veri Yolu	15
3. SONUÇ	16
3.1 ARM11'in Eski Versiyonlarla Karşılaştırılması	16
Referanslar	17

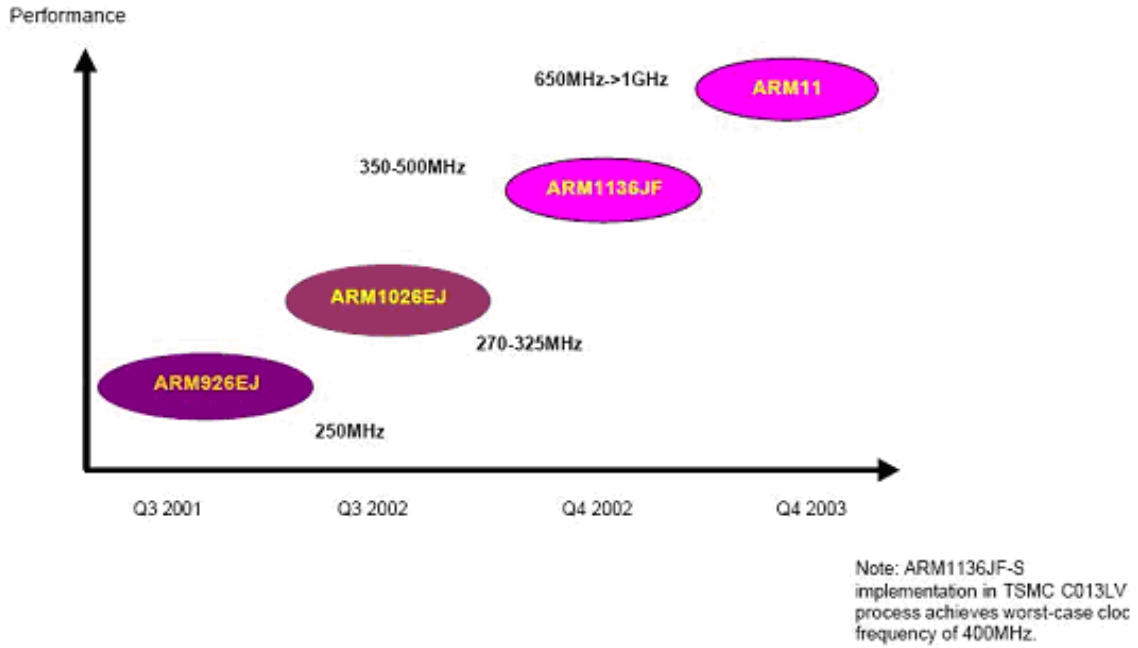
1. Giriş

1.1 ARM Hakkında Genel Bilgi

ARM, İngilizce “*Advanced RISC(reduced instruction set computer) Machines*” in kısaltmasıdır. RISC işlemciler üzerinde performans-fiyat-güç konularında 1983'ten günümüze çalışan dünyanın en önemli işlemci üreticilerinden biridir. Geliştirdiği ve ürettiği ürünler şu şekilde sıralanabilir:

- 16/32 bit RISC işlemciler
- Veri Makineleri (Data Engines)
- 3D İşlemciler
- Sayısal kütüphaneler
- Gömülü Bellekler
- Çevre Birimleri
- Yazılımlar ve geliştirme araçları

Günümüzde ARM mimarisi, dünyanın en yaygın 32 bit mikroişlemcili mimarisidir ve modern SoC (System on Chip) model tasarımlarının en önemli örneğidir.



Şekil 1- ARM mimarisinin gelişim süreci

Şekil 1’de ARM mimarisinin 2001 yılında üretilen ARM9’lardan günümüze performans açısından gelişimi gösterilmiştir.

ARM, cep bilgisayarları için en güçlü ve en iyi mimariye sahip işlemcileri üretmektedir. Elektronik ürün üreticileri tarafından en çok tercih edilen mimari olduğu için çok fazla yazılım desteği de almaktadır.

ARM işlemcileri lisansına aldığı gibi mimarilerini de lisansına almakta ve satmaktadır. Örnek olarak, Intel® Xscale™ ARM mimarisine dayanarak yapılmıştır.

1.2 ARM11 Hakkında Genel Bilgi

Son çıkan ürün olan ARM11; PDA'lar, kameralar ve üst düzey cep telefonları için tasarlanmıştır. Örnek olarak, Apple iPhone'larda da ARM11 işlemci kullanılmıştır.

ARM11 işlemciler, ARmv6 komut setinin kullanıldığı ilk işlemcidir. Hedefi, yüksek performansa az maliyet ve güç harcaması ile ulaşmaktır.

Bu noktada ARM11’i tanıtmak için ARmv6 komut mimarisi hakkında bilgi vermek yerinde olacaktır.

1.2.1 ARmv6 Mimarisi Hakkında Genel Bilgi

ARmv6, Ekim 2001’de yeni nesil kablosuz ağlarla çalışan ürünlerin ve otomotiv ürünlerin ihtiyaçları göz önüne alınarak üretilmiştir.

ARmv6'nın getirdikleri:

- Ortam (ses, video ...) işleme
 - 2 kat hızlı MPEG4 kodlama/kod çözme
 - 2 kat hızlı ses sinyali işleme

- Gelişmiş cep mimarisi
 - Fiziksel adresli cep
 - Cebi boşaltma ve yeniden doldurma işlemlerinde azalma
 - Cep içeriğini değiştirmede daha az masraf
- Gelişmiş kesme ve aykırı durum tespiti
 - Gerçek zamanlı sistemlerde önemlidir
- Sırasız ve karışık(rasgele) veri sağlama
 - Veri paylaşımını kolaylaştırır ve belleğin efektif kullanılmasını sağlar

ARMv6, eski uygulamalarla tamamen uyumludur ve Thumb ile kod sıkıştırılmaya elverişlidir. İş hattı 8 kattan oluşur ve bu eski işlemcilerle göre %40 daha çok veri işlenmesi demektir.

ARMv6 SIMD komut setini içeren ilk mimariye sahip işlemcidir. ARMv6 komut seti Thumb® özel komutları, Jazelle™, ARM DSP (Digital Signal Processor) komutları ve SIMD (Single Instruction Multiple Data) içeren geniş bir komut setidir.

Burada bahsedilen öğelere ileride ayrıntılı değinilecektir.

2. ARM11 Mimarisinin Ayrıntılı İncelemesi

2.1 Saklayıcılar ve Çalışma Modları

Veri boyutları 8 (word), 16 (half word) ve 32 (byte) bitten oluşmaktadır.

Saklayıcılar:

- Unranked Register R0-R7;
- Banked Register R8-R14;
- SP R13

- Link register R14
- PC R15;
- Current Program Status Register R16 (CPSR)

Banked Register: ARM’de işlemciler farklı modlarda çalışır ve bu modlara göre bazı saklayıcılara erişilebilir. Ortak olarak kullanılmayan bu saklayıcılar banked register olarak adlandırılır.

Unranked Register: İşlemci çalışma modundan bağımsız olarak kullanılabilen saklayıcılardır.

Link Register: Alt programlardan geri dönüş adresinin saklandığı saklayıcıdır.

CPSR: İşlemcinin çalışma modunu, bayrakları, kesme bitlerini tutan saklayıcıdır.

PC: Program Sayıcı (Program Counter)

SP: Yığın İşaretçisi (Stack Pointer)

Her çalışma modu bir R13-R14 (banked) saklayıcıya ve SPSR’ye (Subprogram Status Register) sahiptir. Bunlar aslında R13_xyz, R14_xyz, SPSR_xyz olarak adlandırılır. xyz çalışma moduna göre değişir.

Tüm bunları bir örnekle anlatalım: İşlemcimiz kullanıcı modunda olsun. Yazılım R0-R15 saklayıcılarına erişebilmektedir. Bu arada bir kesme olsun. Bu durumda çalışma durumu değişir ve bazı yeni saklayıcılar (banked) görünür hale gelir.

Yapılan işlemler sırayla şu şekilde olur:

1. Şu anki PC (R15) , yeni çalışma modunun R14 saklayıcısına (Banked – Link) geri dönüş adresi olarak saklanır ve bu saklayıcı **R14_kes** olarak adlandırılır. “kes” kesme’den gelmektedir.
2. CPSR, yeni çalışma moduna ait olan SPSR’ye kopyalanır. **SPSR_kes**
3. CPSR, yeni çalışma moduna uygun olarak güncellenir.
4. R15 yani PC’ye ilgili kesme vektörü adresi yüklenerek kesme işlemleri başlatılır.
5. İşlem bitince ilgili ayarlar SPSR_kes ve R14_kes’ten geri yüklenir.

Dikkat edilirse R13 (Yığın İşaretçisi) de banked saklayıcıdır. Yani her modun kendi yığını vardır. Hızlı kesmelerde (FIQ) başka saklayıcılarda “banked” olarak ele alınır. Bunlar R8-R12 saklayıcılarıdır. FIQ, vektör tablosunda en sondadır ve bu sebepten bir dallanma gerekmeden kesme işlemlerine hemen başlanır. FIQ’a ait diğer bir özellik de fazladan 5 saklayıcıyı kullanmasıdır. Bellek erişiminin yavaş olduğunu düşünürsek fazladan kullanılan bu 5 saklayıcı daha hızlı çalışma sağlar.

Bu arada iş hattında aynı anda birden fazla komut olacağı için ana programa geri dönüşte PC’den uygun miktarda geri gidilerek iş hattı doldurulmaya başlanır.

2.2 Adresleme Yöntemleri

ARM, RISC işlemcilerde göre çok geniş bir adresleme alanı sağlar. Bunları örneklerle gösterecek olursak:

ADD r3, r2, r1, LSL #3	$r3 = r2 + (r1 \ll 3)$ “3 bit sola ötele”
ADD r4, r3, r2, LSL r1	$r4 = r3 + (r2 \ll r1)$
LDR r5, [r2]	$r5 = \text{bellek}[r2]$
STR r5, [r2]	$\text{bellek}[r2] = r5$
LDR r4, [r3, #8]	$r4 = \text{bellek}[r3 + 8]$
LDR r3, [r2], #12	$r3 = \text{bellek}[r2],$ $r2 = r2 + 12$
LDR r2, [r3, #8]!	$r2 = \text{bellek}[r3 + 8],$ $r3 = r3 + 8$

2.3 Güvenlik Uygulamaları

TrustZone(TM) olarak bilinir. Güvenli ve güvensiz durumlar olarak adlandırılan iki sanal işlemciden oluşur. Bunun için mimariye SMC komutu eklenmiştir. Bu komut ile komutun yürütüleceği mod değiştirilir. Böylece gelişmiş bir işletim sistemi ile güvenli bir kernel bir arada aynı anda çalışabilir. Kernel tam anlamıyla işletim sisteminden soyutlanmıştır. Dışarıdaki gelişmiş işletim sistemindeki tüm erişim haklarını eline geçirmiş bir saldırı programı bile kernel'a erişemez. Gelişmiş işletim sistemi güvensiz uzayda, güvenliğin önemli olduğu kodlar güvenli uzayda çalışır.

2.4 Performans

ARM11, 350Mhz ile 500Mhz arasında hıza sahiptir ve 0.13µmm mimarisi ile üretilmektedir. Harcadığı enerji ise oldukça düşüktür: 0.13µm için 0.6 mW/MHz enerji tüketimi vardır. 1.2V ile çalışır. Yükselen hız için yüksek kademeli işlemci gerekmektedir ve bundan dolayı iş hattı 8 kata çıkarılmıştır. 0.1µmm mimarisiyle üretilmesi planlanan yeni nesil 1ghz maksimum hıza ulaşması planlanıyor.

ARM11'in yapısında bulunan Intelligent Energy Manager (IEM) dinamik güç yönetimi sağlamaktadır. IEM %25'ten %50'ye varan enerji tasarrufu sağlamakta ve pil ömrünü uzatmaktadır. ARM11 bellek alt sistemi görev değiştirme işlemlerinde gelişim sağlar. Bu veri yolu erişimlerini düşür, böylece güç gereksinimleri de azalır. Yeni load / store komutları daha etkin semafor tanımlamaları yapılmasını sağlar ve ARM11'in çok işlemcili uygulamalara uyumluluğunu artırır.

ARM11 sırasız verilere toleranslıdır. Ayrıca, bir durum biti big-endian (en anlamlı bit en solda) / little-endian (en anlamlı bit en sağda) kontrolü yapar ve böylece ARM'nin diğer işlemcilerle ve DSP'ler ile uyumlu çalışmasını sağlar.

2.5 ARM DSP

ARM DSP, az güç harcanmasını beklediğimiz pille çalışan cihazlarda kullanılan

- Ses kodlama/ kod çözme (MP3: AAC, WMA)
- Yardımcı motor kontrolü (HDD/DVD)
- MPEG4 kod çözme
- Ses ve el yazısı tanıma
- Gömülü kontrol sistemleri
- Bit zor algoritmalar (GSM-AMR)

uygulamalarını sağlayan işlemcidir.

2.6 Jazelle™



Jazelle™, ARM mimarisinde yürütme uygulamalarında(Java™) düşük maliyet ve güç ile yüksek performans ve hızlanma sağlamak için geliştirilmiştir.

Jazelle teknolojisi hem donanım hem de yazılım olarak ARM tarafından üretilmiş bir çözümdür. ARM Jazelle teknoloji yazılımı tamamen çok-görevli özellikli JVM (Java Virtual Machine)'dir. Diğer ARM çekirdeklerinde de getirdiği avantajları kullanabilmek için optimize edilmiştir. ARM Jazelle teknoloji donanım uzantılarına ARM'nin 50'nin üzerinde çözüm ortağından erişilebilir.

2.7 SIMD

SIMD, Single instruction multiple data ifadesinin kısaltmasıdır. Özellikle çoklu ortam (multimedia) işlemlerinde önemli bir kullanımı vardır. ARM 11'de "Advanced SIMD" kullanılmıştır. "Advanced SIMD" piyasada NEON teknolojisi olarak bilinir. 64 ve 128bitli komut kümesi içeren işlemcilerde sinyal işleme ve ses/görüntü işleme işlemlerinde hızlanma sağlamaktadır. Kapsamlı bir komut seti ve ayrı bir komut yürütme donanımı sağlamaktadır. Mp3 kod çözme işlemini 10mhz'de ve ses kodlama-kod çözme işlemini en fazla 13mhz'de yapabilir. NEON aynı anda 16 işlem yapabilir.

SIMD'in getirdiği avantajlara değinirsek, örnek olarak birden çok veri ile aynı sayıyı toplayacağımızı düşünelim. Bu veriler ekrandaki piksellerin her birinin tuttuğu R-G-B değerleri olabilir. Ekran parlaklığını değiştirmek için bütün piksellerin ilgili R-G-B değerlerini aynı değerle toplamak gerekecektir. Bu değerlerin hepsi için komut kullanmak yerine tek komutla bütün verileri işlemek SIMD'nin mümkün kıldığı bir çalışma biçimidir. SIMD veriyi tek başına değil bir küme olarak algılar ve işlemek için tek veri değil birden çok veri alır. Tek komutla çalışmak döngü içerisinde çalışmaktan da iyi ve hızlı olacaktır. Bu şekilde çalışmak sonuç olarak bir hızlanma sağlamaktadır.

SIMD işlemciler sadece birçok veriyi aynı anda işleyen komutları içerir. SIMD sistemlerdeki paralelliğin derecesi benzer işlemi yapan superscalar işlemcilere göre daha fazladır.

2.8 VFP

ARM mimarisinde yardımcı işlemcidir. Tek ve çift duyarlılıklı kayan noktalı sayılarla işlem yapmaktadır. Kullanıldığı uygulamaların sayısı çok fazladır: PDA'lar, ses kodlama ve çözme, smartphone'lar, 3D grafikler, dijital ses, yazıcılar... VFP ayrıca SIMD'yi de desteklemektedir. Bu avantaj grafik ve sinyal işleme uygulamalarında kodlamayı kısaltıp yapılan işi arttırmaktadır.

2.9 Thumb®

Özetle 8/16 bit sistem maliyetine 2 bit performans sağlar. Tasarımcılar maliyetin önemli olduğu gömülü kontrol sistemler kullanılan cep telefonu, modemler... gibi uygulamalarda 32bit performansına ve adres alanına 32 bit maliyetine ulaşmadan erişmek istemektedir.

Thumb'ın tasarımcılara sundukları:

- En düşük sistem maliyeti ve fiyata kod yoğunluğu
- Düşük maliyetle 8 veya 16 bit veri yolu ve bellekten 32-bit performansı
- MIPS/Watt maksimum pil ömrü ve RISC performansı
- Minimum yonga maliyeti
- Küresel anlamda çözüm ortakları bulma kolaylığı

Thumb 32-bit ARM mimarisine bir uzantıdır. Thumb komut seti en çok kullanılan 32 bit komutların 16 bitlik işlem kodlarına sıkıştırılmasıyla oluşturulmuş bir alt kümedir. Yürütmeye bu işlem kodları maliyet kaybı olmadan 32 bite geri çözülür. Böylece bir programın harcadığı bellek alanı dolayısıyla maliyet küçülür.

Tasarımcılar Thumb veya 32 bit ARM arasından istedikleri komut setini kullanabilir. Böylece uygulamanın gerektirdiği performans veya kod alanı kısıtlamalarında esneklik sağlanmış olur.

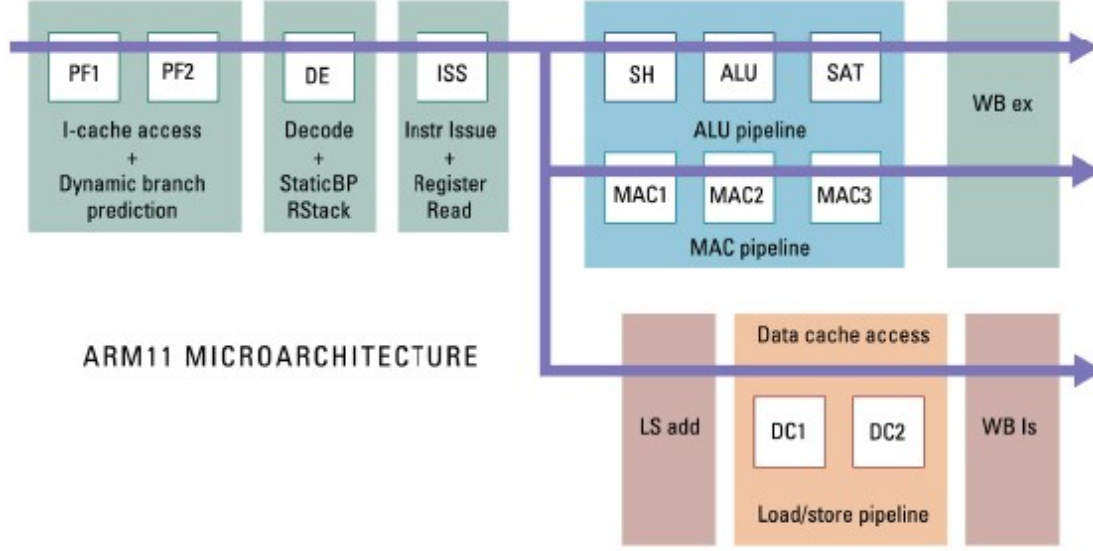
"Thumb-aware" çekirdeği Thumb komutlarını çözen bir komut iş hattına sahip standart bir ARM işlemcisidir. Tasarımcı 32 bit ARM mimarisinin ve Thumb komut setinin gücünü 8 bitlik sistem maliyetiyle elde eder.

Thumb, genel 8 ve 16-bit CISC/RISC'lerden daha çok kod yoğunluğuna sahiptir. Thumb mimarisi Windows yazılım geliştirme ortamları tarafından desteklenmektedir.

2.10 İş Hattı

ARM11, sekiz katlı iş hattına sahiptir. Kat sayısının artması normalde efektif çalışmayı olumsuz etkileyecektir. ARM, "forwarding" (iş hattının sonuna gelmeden elde edilen sonucu yazma işlemi yapmadan sonraki komuta iletebilme) ve dallanma tahminleri ile oluşacak zaman kayıplarını ortadan kaldırmaya çalışmaktadır. Böylece ARM9'daki beş katlı iş hattındaki aynı gecikme ile daha fazla işlem yapılır.

ARM'de aynı anda tek komut yayınlanır(issue) ve bu sıralıdır (i-o / in-order). o-o (out of order) komut sonlandırmaya izin vardır. ALU ve MAC(multiply-accumulate) iş hatlarında aynı anda birden çok komutun yayınlanabildiğini görmekteyiz. Bu daha verimli bir çalışma olarak gözükse de sonuç olarak iş hattının karmaşıklığını arttıracak; buna bağlı olarak güç harcaması ve işlemcinin kapladığı alan artacaktır. ARM11 geliştirilirken yapılan araştırmalardan elde edilen sonuçlara göre komut işlemede elde edilen bu hızlanma, oluşan alan artımı ve fazla güç harcanmasını telafi edememektedir. Bu sebeple tek komut yayınlama tercih edilmiştir.



Şekil 2. ARM11 İşhattı Yapısı

2.10.1 İş hattında paralellik

Her ne kadar iş hattı tek komut yayınlamalı olsa da, paralellik ALU(arithmetic logic unit), MAC(multiply-accumulate) ve LS (load-store) komutları için ayrı birimler görevlendirilmesiyle ortaya çıkmaktadır. LS komutlarının ıskavuru durumuna bağlı yürütülmesinden dolayı aritmetik işlemlerden ayrılması daha verimli ve etkin bir çalışma sağlamaktadır.

Aritmetik işlemlerle LS (load / store) komutlarının ayrılması sayesinde iş hattı belleğe yükleme veya bellekten okuma işlemini beklerken aritmetik işlem komutlarını(ALU-MAC) bekletmek zorunda kalmaz. Bu iş hattı yapısı ayrıca derleyiciye komutların sırasını değiştirerek performansı arttırmasını sağlar.

ARM11 mimarisi sırasız komut sonlandırmaya (out-of-order completion) izin vermektedir. Yani komutlar arası bağımlılık yoksa bir komut diğerinin sonucunu beklemiyorsa o komut doğrudan yürütülüp sonlandırır ve kaynaklarını bir sonraki komut için serbest bırakır.

2.11 Dallanma

ARM11 mimarisi dallanma tahmini yapmaktadır.

2.11.1 ARM'de dallanma tahminleri

İki tiptir. İlki dinamik tahmindir ve ilk önce bu yönteme başvurulur Bunun için bir geçmiş kaydı tutulur ve bu dallanmanın daha önceden yapıp yapılmadığı kontrol edilir. Dört

durumlu bu BTAC(branch target adres cahce) yeniden görülmesi daha olası olan son yapılan dallanmaları saklar. Eğer eşleşen bir dallanma bulunursa onun davranışı tahmin olarak alınır. BTAC 64 dallanmayı saklayabilir.

Durumlar:

- Strongly Taken,
- Weakly Taken
- Strongly not Taken
- Weakly not Taken

BTAC'de eşleşen bir kayıt bulunmazsa statik yõteme başvurulur. Kullanılan statik yöntem şöyledir: Dallanılacak adres eğer büyükse yani ileriye dallanma varsa dallanılmayacağı tahmini yapılır; eğer dallanılacak adres küçükse yani gerideyse bu muhtemelen bir döngüdür ve dallanma öngörüsünde bulunulur.

Geri dönüş adreslerini tutan yığın üç adrese kadar geri dönüş adreslerini saklar. Dallanmalar çoğunlukla koşullu dallanmadır.

Dallanma tahmini eğer isabetli olmazsa dallanılan komut iş hattından silinir. Buna "folding" denmektedir.

Yapılan testlerde ARM11 mimarisinde dallanma tahminlerinin %85 inin doğru öngörüldüğü tespit edilmiştir. Her doğru tahmin komut başına 5 saat çevrimi süre kazandırmaktadır.

2.12 Bellek

2.12.1 Bellek Mimarisi

ARM11 cep bellek, TLB(translition lookaside buffer), DMA (Direct Memory Access), TCM (Tightly-coupled memory) işlemcinin üzerinde olan bellek, yazma kuyruğu (write buffer) ve MMU(Memory Management Unit) içermektedir.

2.12.2 Bellek Organizasyonu

ARM11, Harvard mimarisi içerir. Veri ve komut cebi ayrıdır böylece okuma ve yazma komutları tek saat çevriminde yürütülür.

- Cep bellek sanal indeksli ve fiziksel etiketlidir.
- 4-64k arasında boyutu değişebilen 2 cep bulunur.
- Boyutları ayarlanabilir 4 kümeli asosyatif cep bulunur.
- Cep hatları WB veya WT çalışabilir. Aslen WB'dir.
- Komut ve veri cepleri bir çevrimde 2 sekizli(word) okuyabilir.
- Bellek erişiminde DMA kullanılır.

Cepte yer açma için Pseudo-Random veya Round-Robin algoritmaları kullanılır. Bir sayaç ile kullanılacak yöntem seçilir.

2.12.3 MMU

MMU, cep bellek sistemi ile beraber belleğe yazma ve bellekten okuma işlemlerini kontrol eder. Aynı zamanda sanal adresleri fiziksel adreslere çevirir.

ARM11'deki MMU, adres çevrimi ve işlemcinin portları için erişim izni kontrollerini sağlamaktadır. ARM11'deki MMU adres çevrimlerinde TLB (translition lookaside buffer) kullanır.

2.12.4 TLB

ARM11'in yapısındaki TLB iki seviyelidir: Micro TLB ve Macro TLB. Veri ve komut Micro TLB'leri ayrıdır ve 10 birim veriyi tutar. Macro TLB, hem veri hem de komut Micro TLB'sinde ıskala olan verileri tutar.

Micro TLB'deki veri değişimleri için round-robin ve rasgele belirleme yöntemleri kullanılır. Varsayılan olarak round-robin yöntemi kullanılır.

2.12.5 Yazma Tamponu

Tüm belleğe yazma işlemleri önce yazma tamponuna uğrar. Tampon FIFO olarak çalışır. Yazma tamponundaki bir birim veri; adres bilgisi, veri ve boyut bilgisinden oluşur.

Şayet ardışık adreslerdeki veriler belleğe yazılacaksa bunlar birleştirilir ve tek bir adres bilgisi ile belleğe yazdırılır, böylece yazma tamponunda adres bilgisi için gereksiz yer harcanmamış olur.

2.12.6 TCM

“Tightly-coupled memory” ifadesinin kısaltmasıdır. İki tiptir:

- *Komut TCM - Instruction TCM (ITCM)*
- *Veri TCM - Data TCM (DTCM)*

TCM düşük gecikmeli bellek sağlamak için kullanılmıştır. Bu tip bellekler kesme veya gerçek zamanlı çalışan kritik programcıkları tutmak için kullanılır.

TCM etkin olduğu zaman sonsuz bir bellek alanı sağlar. Sistemin fiziksel bellek haritasının bir parçası olarak kullanılır ve aynı fiziksel adresi taşıyan aynı seviye bellekle örtüşmez. Bu sebeple TCM, belleğin WT davranışlı ön bellek olarak işaretlenmiş alanları için ceplerden farklı davranış gösterir. TCM'ye dışarıdan yazma (external write) yapılmaz.

2.13 Veri Yolu

ARM11 veri ve komut cepleri ile işlemci arasında ve yardımcı işlemci birimler arasında 64 bitlik veri yolu bulundurmaktadır. Böylece tek saat çevriminde komut cebinden 2 komut alınır ve veri cebinden 2 veri okunur yani 2 LS komutu tek çevrimde yürütülür, 2 ARM kütüğü transfer edilir.

ARM11 kayan noktalı sayı işlemlerini destekler. İsteğe dayalı olarak kayan noktalı sayı ile işlem yapan veya yapmayan çekirdekler üretilmektedir.

3. SONUÇ

3.1 ARM11'in Eski Versiyonlarla Karşılaştırılması

Feature	ARM9E™	ARM10E™	Intel® XScale™	ARM11™
Architecture	ARMv5TE(J)	ARMv5TE(J)	ARMv5TE	ARMv6
Pipeline Length	5	6	7	8
Java Decode	(ARM926EJ)	(ARM1026EJ)	No	Yes
V6 SIMD Instructions	No	No	No	Yes
MIA Instructions	No	No	Yes	Available as coprocessor
Branch Prediction	No	Static	Dynamic	Dynamic
Independent Load-Store Unit	No	Yes	Yes	Yes
Instruction Issue	Scalar, in-order	Scalar, in-order	Scalar, in-order	Scalar, in-order
Concurrency	None	ALU/MAC, LSU	ALU, MAC, LSU	ALU/MAC, LSU
Out-of-order completion	No	Yes	Yes	Yes
Target Implementation	Synthesizable	Synthesizable	Custom chip	Synthesizable and Hard macro
Performance Range	Up to 250MHz	Up to 325MHz	200MHz – >1GHz	350MHz - >1GHz

Şekil 3. ARM11'nin diğer mimarilerle karşılaştırılması

ARM11' in mimarisi ile eski mimarilerin karşılaştırılması yukarıdaki tabloda ele alınmıştır. Daha önce belirttiğimiz gibi Intel® Xscale™ ARM mimarisi üzerine tasarlanıp üretilmiştir. Bu sebeple karşılaştırmada ele alınmıştır.

Referanslar

- David Cormie, Nisan 2002, The ARM11™ Microarchitecture, ARM11MicroarchitectureWhitePaper.pdf.
<http://www.arm.com>
- http://222.66.109.20/jpkc/arch/document/student/16-033230_Zu_Min.doc
- Chris Williams, Future ARM based processors for RISC OS? ,
<http://www.drobe.co.uk/riscos/artifact763.html>
- ARM Technical Tidbits, <http://www.go-ecs.com/arm/armtek1.htm>
- ARM architecture, http://en.wikipedia.org/wiki/ARM_architecture
- What ARM Does, <http://www.arm.com/aboutarm/>
- Hardware Directory: ARM11 Implements ARMv6 ISA,
<http://electronicdesign.com/Articles/Index.cfm?ArticleID=2474>
- ARM Bir Sonraki Jenerasyon İşlemcilerini Duyurdu,
<http://www.donanimhaber.com/arama2.asp?Sayfa=4&plus=xscale&num=10"e=&any=&minus=&time=&mode=0>
- ARM11 Family, <http://www.arm.com/products/CPUs/families/ARM11Family.html>
- ARM Architecture Overview,
http://www.hitex.com/products.html?con_arm_architecture_overview.html~content
- ARM1176JZ(F)-S, <http://www.arm.com/products/CPUs/ARM1176.html>
- Matthew Byatt, Delivering high-performance embedded signal processing with ARM OptimoDE™
<http://www.all-electronics.de/ai/resources/ae88782d994.doc>