



**KARADENİZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
MİKROİŞLEMCİLER LABORATUARI**



ARM MİMARİSİ VE UYGULAMALARI

1. GİRİŞ

Günümüzde kişisel bilgisayarların yanı sıra akıllı telefonlar, tabletler gibi bir çok mobil cihaz da yaşantımızın bir parçası haline gelmiştir. Mobilitenin sağlaması ve kişilerin kendi veri ve işlem güçlerini yanlarında taşıma istekleri kişisel bilgisayarların daha küçük cep bilgisayarları haline dönüşmesini sağlamıştır. Kişisel bilgisayarlara oranla daha sınırlı kaynaklara sahip olan bu cihazların verimli çalışması son derece önemlidir. Performans kayıpları olmadan daha uzun bir kullanım süresi sunan mobil cihazlar donanımsal olarak kişisel bilgisayarlarla aynı çalışma prensiplerine sahip olmalarına rağmen, gereksinimleri yüzünden farklılaşmışlardır. Bu farklılaşmaya performans, kullanım süresi, maliyet gibi temel parametreler sebep olmuştur. Örneğin, kişisel bir bilgisayar için tasarlanacak olan bir işlemcinin fazla güç tüketmesi önemsenmeyebilirken, mobil cihazlar için tasarlanacak bir işlemcinin güç tüketimi/performans oranı son derece kritik bir öneme sahiptir.

2. ARM (Advanced RISC Machine)

Acorn Bilgisayar firması tarafından ilk olarak 1981 yılında Acorn RISC* Machine adıyla tasarlanmış RISC mimarisidir. Firma adı 1990 yılında Advanced RISC Machine olarak değiştirmiştir. Son olarak 1998 yılında günümüzdeki ismi olan **ARM Ltd.** adını almıştır.

ARMv1 ile başlayıp günümüze kadar geliştirilen bu mimari 32 bitlik yapısı sayesinde 8 bitlik işlemcilere göre çok daha hızlıdır. Ayrıca düşük güç tüketimi ve yüksek performansı sayesinde günümüzde yaklaşık %75'lik oranla gömülü sistemler üzerinde en çok kullanılan işlemcilerdir. ARM bir mimari türü olduğundan geliştirici firma fiziksel işlemci üretmek yerine bu mimariyi işlemci üreten diğer firmalara lisanslayarak gelir elde etmektedir. ARM mimarisi lisansını elinde bulunduran Apple, Samsung, Nvidia gibi başlıca firmalar farklı isimlerde bir çok işlemci üretmişlerdir. Örneğin, Nvidia-Tegra, Samsung-Hummingbird, Exynos ve Apple-A4, A5, A5X vs. ARM firması verilerine göre sadece 2010 yılında 6.1 milyar adet ARM mimarisi tabanlı işlemci üretilmiştir. Bu işlemcilerin çoğu akıllı telefonlarda, bir kısmı dijital televizyonlarda, geri kalanı ise set üstü kutuları (dijital uydu alıcıları, video oynatıcıları vs.) ve mobil bilgisayarlarda kullanılmaktadır.

*RISC (Reduced Instruction Set Computing) : İndirgenmiş(Azaltılmış) komut kümesi ile hesaplama. Bu özelliğe sahip işlemciler karmaşık komut kümelerinden çok daha basit ve az sayıda komut kümesine sahiptirler. Komut kümelerinin az olması işlemcide kullanılan yonga sayısını azaltır ve güç tasarrufu sağlar. Daha hızlı çalışırlar. Tasarlanmaları daha az zaman alır.

3. ARM Mimarisi

ARM mimarisini klasik, gömülü(embedded) sistemlere yönelik ve uygulamaya yönelik olmak üzere 3 temel gruba ayırabiliriz.

Klasik ARM işlemci grubunda ARM7,ARM9 ve ARM11 yer alır. Bu mimari yapıları daha çok motor kontrolü gibi donanıma yakın işlemlerde kullanılır.

İkinci grup olarak gömülü işlemci grubuna bakarsak bunlar da Cortex M ve R serileridir. Bunlar matematiksel formüllere dayalı olarak kesin sonuca yönelik çalışması gereken, düşük güç tüketimi isteyen uygulamalarda kullanılır.

Son olarak da uygulama işlemcileri Cortex-A5, Cortex-A8, Cortex-A9, Cortex-A15 olarak sınıflandırılabilir. Bunlar da akıllı telefonlar, tablet bilgisayarlar gibi yüksek performans gerektiren, son kullanıcı uygulamalarına yönelik platformlarda kullanılır.

ARM tabanlı mikroişlemcilerin yanı sıra mikrodenetleyiciler de üretilmektedir. Mikroişlemciler genellikle sadece işlem ünitesi ve temel birimlerden oluşurlar, harici olarak bellek ve diğer giriş/çıkış birimlerine ihtiyaç duyarlar. Mikroişlemcileri daha çok uygulama yazılımlarının çalıştığı sistemlerde görüyoruz. Mikrodenetleyiciler ise kendi içerisinde giriş-çıkış birimleri, çeşitli donanım birimleri (SPI, I2C, PWM), RAM ve kalıcı hafıza (genellikle NOR veya NAND flash) bulunan komple sistemlerdir. Mikrodenetleyiciler daha çok elektronik tasarım alanında kullanılmaktadırlar. Çevremizdeki cihazlardan sadece cep telefonlarımız, tablet bilgisayarlarımız değil, çamaşır makinalarımızdan otomobillerimize, fotoğraf makinalarımızdan yazıcılarımıza kadar birçok ürünün içinde ARM tabanlı işlemciler/denetleyiciler bulunmaktadır.

4. ARM Mikroişlemci/denetleyici Programlama

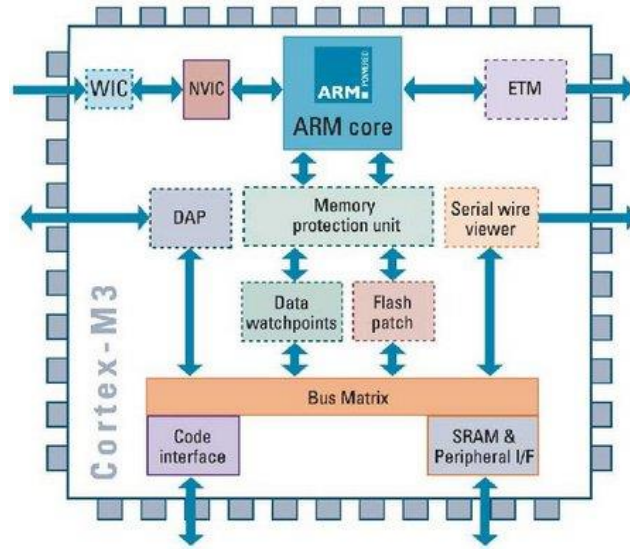
Elektronik tasarımlara yönelik alanlarda *mikrodenetleyici*, uygulama yazılımlarına yönelik alanlarda ise *mikroişlemci* kullanmak daha mantıklı bir seçim olacaktır. Uygulama yazılımları doğrudan donanım seviyesinde tasarlanmaktan ziyade daha çok bir platform (Android,iOs, Windows Mobile vs.) aracılığı ile gerçekleştirilirler. Bu nedenle ARM mimarisine sahip bir *mikrodenetleyici uygulaması* gerçeklemek, bu tarz işlemcilerin çalışma mantıklarını daha iyi kavramamızı sağlar. ARM mimarisine sahip mikrodenetleyicilere örnek olarak NXP firmasının LPC1000,2000 serileri, ST firmasının STM32 serisi, AMTEL firmasının AT90SAM serisi gösterilebilir. Deneyde kullanılan EasyMX-Pro_v7 board'ı varsayılan (default) olarak STM32 model bir işlemci ile gelmektedir.

4.1 Cortex – M Serisi Mikrodenetleyiciler

ARM mimarisinin mikrodenetleyici ailesinden biri olan Cortex M serisi, piyasada mevcut bulunan 8 ve 16 bitlik mikrodenetleyicilere rakip olarak ortaya çıkmış 32 bitlik işlemci mimarisine sahip işlemci serisidir. Çok düşük enerji tüketimlerinin yanı sıra maliyetleri de düşüktür. Bu aileye mensup mikrodenetleyiciler endüstriyel kontrol sistemleride, beyaz eşyalarda, medikal cihazlarda vb. kullanılmaktadırlar.

Günümüzde Cortex-M3 serisi mikrodnetleyiciler daha yaygın olarak kullanılmaktadır.Bunun nedeni hem daha önce üretilmeleri hem de daha ucuz olmalarıdır. Ancak son olarak tasarlanan M4 serisine eklenen DSP(Digital Signal Process) özelliği gelecekte çok daha etkili işlerin yapılabileceğinin bir göstergesidir.

Deney setindeki yer alan *STM32F107VCT6* model numaralı chip Cortex ailesinin M3 sınıfına ait bir işlemcidir.



Şekil.1 Cortex –M3 İç Yapısı

5. Açıklamalı Örnek Kodlar

1) MikroC Led Yakıp Söndürme

```
void main() {  
  
    GPIO_Digital_Output(&GPIOD_BASE, _GPIO_PINMASK_ALL);  
    GPIOD_ODR = 0;  
  
    while(1) {  
        GPIOD_ODR = ~GPIOD_ODR;  
        Delay_ms(1000);  
    }  
}
```

Main'den sonrası için satır satır açıklayacak olursak;

- D portu dijital çıkış olarak ayarlanıyor.
- D portunun ilk değeri 0 olarak setleniyor.
- While yapısı kullanılarak sonsuz döngü oluşturuluyor.
- D portunun çıkışı bir önceki çıkış değerinin tersi olacak şekilde setleniyor.
- Milisaniye cinsinden 1000 ms bekleme yapılıyor.

2) Led Kaydırma

```

void main() {
    int counter = 0;
    GPIO_Digital_Output(&GPIOD_BASE, _GPIO_PINMASK_ALL);
    GPIOD_ODR = 0;

    while(1) {
        for (counter = 0; counter < 4; counter++){
            GPIOD_ODR = 0x8888 >> counter;
            Delay_ms(100);
        }
    }
}

```

Main'den sonrası için satır satır açıklayacak olursak;

- int türünde counter isimli bir değişken tanımlanıyor.
- D portu dijital çıkış olarak ayarlanıyor.
- D portunun ilk değeri 0 olarak setleniyor.
- while yapısı kullanılarak sonsuz döngü oluşturuluyor.
- counter değişkeni baz alınarak bir for döngüsü tanımlanıyor.
- 16 bitlik D portuna 0x8888 ataması yapılarak counter sayısı kadar kaydırma yapılıyor.
- Milisaniye cinsinden 100 ms bekleme yapılıyor.

3) Button ile Led Yakma

```

void main() {
    int oldstate = 0;
    GPIO_Digital_Input(&GPIOA_IDR, _GPIO_PINMASK_0);
    GPIO_Digital_Output(&GPIOD_ODR, _GPIO_PINMASK_ALL);

    do {
        if (Button(&GPIOA_IDR, 0, 1, 1))
            oldstate = 1;
        if (oldstate && Button(&GPIOA_IDR, 0, 1, 0)) {
            GPIOD_ODR = ~GPIOD_ODR;
            oldstate = 0;
        }
    } while(1);
}

```

Main'den sonrası için satır satır açıklayacak olursak;

- int türünde oldstate isimli bir değişken tanımlanıyor.
- A portunda bulunan PA0 butonu giriş olarak ayarlanıyor.
- D portu dijital çıkış olarak ayarlanıyor.
- do-while yapısı kullanılarak sonsuz döngü oluşturuluyor.
- if yapısı kullanılarak butona basılıp basılmadığı kontrol ediliyor ve oldstate 1 yapılıyor.

- Diğer if yapısı ile beraber hem butona basıldığı kontrol edilirken aynı zamanda kullanıcının elini butonda kaldırmasının da kontrolü yapılıyor.
- D portunun çıkışı bir önceki çıkış değerinin tersi olacak şekilde setleniyor.
- oldstate değişkeni tekrar 0 yapılıyor.

4) TFT Ekranı Üzerinde Button ve Label vb. Yapıların Kullanılması

```
void Goster_Click() {  
    Label1.Caption = "Merhaba Dünya!";  
    DrawScreen(&Screen1);  
}
```

Burada TFT ekranı üzerinde Button ve Label yapılarının yerleştirme ve tanımlama işlemleri Visual TFT programı üzerinde yapılarak mikroC PRO for ARM platformuna aktarılmaktadır. Bu kısım ile ilgili açıklama deney föyünde verilmeyecek olup deney sorumlusu tarafından laboratuvar ortamında ayrıca anlatılacaktır.

6. Deney Hazırlığı

Deneye gelmeden önce deney föyünün bir ön bilgi teşkil etmesi açısından okunması, anlaşılmayan kısımların deneyde sorulması için notlar ya da sorular oluşturularak hazırlanması deneyin hızlı ve eksiksiz bir şekilde ilerlemesi açısından önemlidir. Öğrencilere sorulacak olan soruların kapsamı sadece föy içerisindeki bilgiler ile sınırlı kalmayacağından, öğrencilerin mikroişlemci ve mimari yapılarını genel olarak araştırması faydasına olacaktır. Deney föyünden elde edilemeyen bilgiler çevrimiçi kaynaklar kullanılarak araştırılabilir.

Deneydeki uygulama kısımlarının daha sağlıklı yürütülebilmesi için kullanılacak olan deney seti (EasyMX-Pro board) için deneye özel oluşturulan kataloğun (Ek-1'de sunulan) okunarak incelenmesi gerekmektedir.

Açıklaması ile verilen kod parçalarının yapılacak uygulamaların benzerlerini teşkil ettiğinden elde edilecek başarıyı direk etkileyeceği bilinmeli ve bu kodlara çalışılmalıdır. Ayrıca uygulamaların yapılacağı platformlar olan "mikroC PRO for ARM" ve "Visual TFT" deney öncesinde incelenmesi verimliliği arttıracaktır.

7. Deney Tasarımı ve Uygulanması

Bu deney ARM mimarisine sahip mikrodeneleyici setlerine geçiş deneyi niteliğinde olduğundan temel kavramların öğretilmesi amaçlanmıştır. Uygulama aşamasında deneydeki mevcut donanım incelenecek ve genel bilgiler verilecektir. Ardından donanımı programlayabilmemize olanak sağlayan ana ve yardımcı yazılımlar ana hatları ile incelenecek ve bir kaç örnek uygulama gerçekleştirmeniz istenecektir.


8. Deney Raporu

Deneyde anlatılan kısımların pekiştirilmesi ve kalıcı bilgi haline getirilmesi açısından tekrar edilmelidirler. Bu nedenle deney raporları bireysel olarak Ek-2'deki şablon kullanılarak yazılacak (bilgisayar ya da el ile) ve bir sonraki deneye kadar teslim edilecektir.

Deneye hazırlık kısmında da vurlandığı üzere deney raporunuza verdiğiniz önem deneye verdiğiniz önemin bir göstergesidir.

9. Ekler

Ek-1 : EasyMX-Pro_v7 board katalog

 *Deneye hazırlanma durumunuz deneye verdiğiniz önemin bir göstergesidir. Deneye hazırlıklı gelmeniz deneyden alacağınız verimi artıracaktır. Ayrıca deneye olan ilginizin yoğunluğu deney sorumlusunun size aktarabileceği bilgilerin yoğunluğu ile doğru orantılıdır.*