1.GİRİŞ

1.1 Mikroişlemciler

Mikroişlemci herhangi bir sistemde merkezi işlem birimidir ve bulunduğu sistemde aritmetik ve mantıksal işlemleri yürütür. Merkezi İşlem Birimi (Central Processing Unit: CPU), kontrol devresini, bir ALU (Aritmetik Mantık Birimi) bazı kaydediciler ve bir adres/program sayıcıyı içerir.

Bir klavyeden gelen verilerin, bir basınç algılayıcısından gelen sayısallaştırılmış çıkışın veya bir başka verinin bilgisayara alınması ve bu bilgilerin çıkış olarak sağlanması mikroişlemci tarafından kontrol edilir. Mikroişlemcinin bu tür işlemleri giriş olarak algılayıp, çıkışa yansıtması için mikroişlemci programlanır. Bir mikroişlemci, belleğinde saklı bulunan programı her bir komutu sıra ile okuyarak yürütür. Her komut önce, onu yürütmek için gerekecek işlemleri belirlemek üzere mikroişlemcinin anlayabileceği makina koduna çevrilir ve sonra gereken işlemler yapılır. Mikroişlemci entegre devresi, yazılan programları meydana getiren makina komutlarını yorumlamak ve yerine getirmek için gerekli olan tüm mantıksal devreleri içerir.

Bir mikroişlemci temel olarak üç kısımdan oluşur. Bunlar merkezi işlem birimi (CPU), giriş çıkış birimi (G/Ç) ve bellektir.

1.1.1 Merkezi İşlem Birimi

Bilgisayarın kalbi veya beyni olarak adlandırılan mikroişlemci aynı zamanda merkezi işlem birimi olarak da anılırlar. Merkezi işlem birimi genel olarak aşağıdaki işlemleri yapar:

- Sistemdeki bütün elemanlar ve birimlere zamanlama ve kontrol sinyali sağlar.
- Bellekten komut veya veri alıp getirir ve işler,
- Veriyi giriş/çıkış birimlerine ya da oradan kendisine aktarır,
- Komutların kodunu çözer,
- Komutla birlikte aritmetik ve mantık işlemlerini gerçekleştirir,

- Program işlenirken giriş/çıkış cihazlarından gelen servis isteklerine I bayrağının durumuna göre cevap verirler.

Intel ve Motorola firmalarının üretmiş olduğu mikroişlemcilerde yukarıda bahsedilen işlemleri gerçekleştirmek üzere üç ana bölüm vardır. Bunlar; Kaydediciler, Aritmetik Mantık Birimi (ALU) ve Zamanlama-Kontrol Birimidir.

1.1.1.1 Kaydediciler

Mikroişlemcinin içinde işlem yaparken geçici olarak verilerin saklandıkları saklayıcılardır. Mikroişlemci içerisinde değişik işlemleri gerçekleştirmek üzere akümülatör, indis kaydedicileri, stack pointer, program counter gibi kaydediciler mevcuttur. Mikroişlemciler içerisinde işlem yapılırken bu kaydedicilere veri atılabilir, toplama, karşılaştırma ve kaydırma gibi bazı işlemle gerçekleştirilebilir.

1.1.1.2 Aritmetik Mantık Birimi (ALU)

ALU, yürütülmekte olan komutta belirtilen iki değer üzerinde aritmetik ve lojik işlemleri yerine getirir. Bu iki değerden biri daima özel bir mikroişlemci kaydedicisi olan ve A kaydedicisi olarak adlandırılan kaydedicinin o anki içeriğidir. Diğer ALU girişi ise; mikroişlemcinin bölümünü oluşturan genel amaçlı kaydedici grubundan bir başka kaydedicidir.

1.1.1.3 Zamanlama-Kontrol Birimi

Mikroişlemcinin kontrol birimi, mikroişlemcinin içinde ve dışında olan bütün veri aktarımlarını ve ALU işlemlerini kontrol eder ve çevre birimlerle eş zamanlama için gerekli sinyallemeyi sağlar. Ayrıca, kontrol birimi ALU'da en son yapılan aritmetik ve mantıksal işlemin sonucunu yansıtan bayrakları giriş olarak alır.

1.1.2 Giriş-Çıkış Birimi (G/Ç)

Mikroişlemcinin dış dünya ile ilişkisinin sağlandığı ünitedir. Mikroişlemciye verilen bilgiler bu ünite yolu ile işlemci içerisine alınırken, CPU ve diğer birimlerde işlenen veriler yine bu ünite sayesinde dış ortama aktarılırlar.

1.1.3 Bellek

Bellek bir mikro bilgisayar sistemi içerisinde bilgileri saklar. Bellek iki bölüme ayrılmıştır. Bir bölümü programı saklar, diğer bölüm ise mikro bilgisayara gerek duyduğu bilgi ya da veriyi tutmak için kullanılır.

Genel amaçlı bir mikro bilgisayar sisteminde program sıklıkla değiştirilir. Çamaşır makinası denetleyicisi gibi bir mikroişlemci tabanlı sistemde program, fabrikada üretilirken yüklenmiştir ve asla değişmez. Bu tür bir uygulama için program ROM adı verilen özel bir bellek türünde saklanır.

ROM (Sadece okunur bellek): Program deyimleri, veri vb. sadece ROM bellekten kopya edilebilir. Yapımcı veya kullanıcı tarafından ROM belleğe kaydedilen hiçbir bilgi değiştirilemez ve bu tür belleklere yeni kayıt yapılamaz. ROM belleklerden başka bir de RAM bellekler de vardır.

RAM (Rastgele erişimi bellek): Kural olarak RAM' lar, üzerinde yazma ve okuma işlemleri yapılabilen bellekler olarak tanımlanmıştır. Bir başka deyişle RAM, değişken veri saklayabilen bir rastgele erişimli bellektir. RAM, bir elektronik devre olduğu için, güce gereksinim duyar. Güç kaynağı kesildiği anda çok anlık olsa bile saklanan bir veri yitirilir. Sistem anahtarı kapatıldığı anda, belleğin içeriği yitirilmiş olur. Bu nedenle bu belleğe kalıcı olmayan saklama ortamı adı verilir.

Başka bellek türleri de vardır. Örneğin, EPROM, silinebilir programlanabilir ROM' dur. EPROM, programcıya bir yonga satın alıp bunu programlama imkanı verir.

1.2 Mikrodenetleyiciler

Bir yazılım olmadan hiçbir işe yaramayan, ancak içerisine yazılan program vasıtasıyla istenilen bir işlemi gerçekleştiren kontrol elemanıdır. Mikrodenetleyici yazılım olması halinde neredeyse sınırsız bir kullanım alanına sahiptir.

Aslında mikrodenetleyici bir bakıma bir bilgisayardır. Her nekadar bir klavyesi, monitörü, kasası ve bunun gibi çevre birimleri olmasa da bir bilgisayarın yaptığı herşeyi

yapabilir. Örneğin her bilgisayarın bir merkezi işlem ünitesi (CPU:Central Processing Unit) vardır ve bu ünite makine kodlarını bizim anlayabileceğimiz karakterlere dönüştürür, programları yorumlar, işler, düzenler, bilgisayarın çeşitli birimleriyle irtibat kurar. Ve bu işlemleri yaparken bazı değişkenleri ve geçici olarak elde ettiği bilgileri sakladığı bir rastgele erişimli hafızaya (RAM:Random Access Memory) ihtiyaç duyar. Ayrıca bilgisayarların dış dünyayla bilgi alış ve verişlerinde kullandıkları bazı giriş ve çıkış üniteleri bulunmaktadır. Örnek olarak fare ve klavye giriş yaptığımız elemanlara, monitör ve yazıcı çıkış aldığımız elemanlara birer örnektir. Bilgilerimizi kaydettiğimiz harddiskler ise hem giriş hem de çıkış elemanı olarak çalışmaktadırlar. Aynen bilgisayarda olduğu gibi mikrodenetleyiciye de fare ve klavye gibi çevre elemanlarının işlemlerini nispeten de olsa yerine getirecek elemanlar ekleyerek küçük bir bilgisayar gibi kullanmamız mümkündür.

Bir mikrodenetleyici genel olarak aşağıdaki birimlerden oluşur:

- CPU (Merkezi işlem ünitesi - central processing unit)
- RAM (Rastgele erişimli bellek-Random Access Memory)
- EPROM/PROM/ROM (Silinir, yazılır sadece okunur bellek-Erasable Programmable Read Only Memory)
- I/O (Girdi/çıktı - input/output) - seri ve paralel
- Timers (Zamanlayıcılar)
- Interrupt controller (Kesmeler)

Sadece kullanılacak işe uygun özellikleri bulunan bir mikrodenetleyici seçildiğinde maliyet nispeten düşmektedir.

Mikrodenetleyicilerde işlemler ve komutlar bit bit kontrol edilebildiğinden giriş ve çıkış birimleri ve kesmeler çok etkin bir şekilde kulllanılabilmektedir.

Şu an kullandığımız masaüstü veya dizüstü bilgisayarlar genel amaçlı bilgisayarlardır ve binlerce programı çalıştırıbilirler. Mikrodenetleyiciler ise özel amaçlı bilgisayarlardır. ve programlandıkları şeyi en iyi şekilde yaparlar. Bunun dışında;

- Mikrodenetleyiciler sadece bir iş için programlanmışlardır ve bu program içlerindeki ROM'da değişmemek üzere saklı bulunur.

- Mikrodenetleyiciler düşük güçte çalışan çiplerdir. Bir bilgisayar 50W civarı güç harcarken mikrodenetleyiciler sadece 50 miliWatt civarında güçharcarlar.
- Mikrodenetleyicelere sadece girdi yapılmaz aynı zamanda çıktı da alınabilir. LED göstergelerle, sıvı kristal göstergelerle, ikaz sesleriyle vb.. Örneğin televizyonunuzda ve uzaktan kumandasında bulunan mikrodenetleyicilerden bahsedelim. Kumandada bulunan mikrodenetleyici tuşlara bastığımızda girdisini almış olur ve bunu televizyondaki alıcı mikrodenetleyiciye gönderir. Ve o mikrodenetleyici de gelen sinyale göre çıkış vererek ya kanal seçer, ya ses ayarı yapar ya da televizyonla ilgili işlemleri yapar.
- Mikrodenetleyiciceler genelde küçük ve düşük fiyatlı çiplerdir. Bir çok parçadan oluşan kompleks birdevreyi kolayca küçük boyutlara ve maliyete indirmenizi sağlar.

Ayrıca mikrodenetleyiciceler belleklerine göre de çeşit gösterirler ;

1.2.1 EEPROM Bellekli Mikrodenetleyiciler

Elektriksel olarak silinebilen ve yazılabilen belleklerdir.Çoğu mikrodenetleyicilerde sınırlı sayıda bulunan EEPROM lar, bir defadan fazla yazılıp silinebildikleri için oldukça kullanışlıdır.

1.2.2 Flash (EPROM) Bellekli Mikrodenetleyiciler

Flash bellekler EEPROM lardan daha hızlı ve daha çok yazma silme işlemine izin vermeleri yönünden üstündürler. Flash belleklerde bilgilerin korunması söz konusu değildir.

1.2.3 OTP Bellekli Mikrodenetleyiciler

Bir kez programlanabilen mikrodenetleyicilerdir. OTP bir kez programlanabilen bir ROM'dur. Programınızı bir EPROM programlayıcı ile bir kez yazdıktan sonra silemez veya değiştiremezsiniz. Bu yöntem programınız artık tamamen hazır olduğunda ve bütün hatalarından arındırıldıktan sonra kullanılır.

Mikrodalga fırınlar gibi birçok üründe kullanılan ve maliyeti de oldukça düşük olan bu çiplerin içine düşük güçlü ve düşük maliyetli CPU lar yerleştirildi. Motorola 6811 ve Intel 8051 ailesi mikrodenetleyicileri bu türe iyi birer örnektir. Bunların yanında Microchip firması tarafından üretilen PIC mikrodenetleyicileri de son zamanlarda oldukça popülerdir.

Günümüz standartlarında bu çipler inanılmayacak kadar küçük ve düşük fiyatla satın alınabilmektedir. Sıradan bir mikrodenetleyici içinde en azından 1,000 byte kapasiteye sahip bir ROM ve 20 byte kapasiteye sahip bir RAM, 8 adet I/0 pini bulundurabilir.

1.3 Mikrodenetleyicinin Mikroişlemciye Olan Üstünlükleri

- Mikroişlemcinin kullanımı ve mikroişlemcili sistemin tasarımı mikrodenetleyicili sisteme göre hem daha masraflı hem de daha karmaşıktır.
- Mikrodenetleyicili bir sistemin çalışması için elemanın kendisi ve bir osilasyon kaynağının olması yeterlidir.
- Mikrodenetleyicinin ihtiyaç duyduğu önbellek ve giriş çıkış birimi bir yonga içerisinde bulunmaktadır. Ancak mikroişlemcili bir sistemde önbellek harici olarak bulunur.

2-PIC MİKRODENETLEYİCİ AİLESİNE GENEL BAKIŞ

PIC serisi mikrodenetleyiciler MICROCHIP firması tarafından geliştirilmiştir. Üretim amacı; çok fonksiyonlu mantık uygulamalarının hızlı ve ucuz bir mikrodenetleyici ile yazılım yoluyla karşılanmasıdır.

PIC'in kelime anlamı PERIPHERAL INTERFACE CONTROLLER (Çevresel arabirim denetleyicisi) dir. İlk olarak 1994 yılında 16 bitlik ve 32 bitlik büyük işlemcilerin giriş ve çıkışlarındaki yükü azaltmak ve denetlemek amacıyla çok hızlı ve ucuz bir çözüme ihtiyaç duyulduğu için geliştirilmiştir.

Çok geniş bir ürün ailesinin ilk üyesi olan PIC16C54 bu ihtiyacın ilk ürünüdür. PIC denetleyicileri RISC benzeri işlemciler olarak anılır. PIC16C54 12 Bit komut hafıza genişliği olan 8 bitlik CMOS bir işlemcidir. 18 bacaklı dip kılıfta 13 I/O bacağına sahiptir ve 20 Mhz osilator hızına kadar kullanılabilir. 33 adet komut içermektedir. 512 byte program EPROM'u ve 25 byte RAM`i bulunmaktadır. Bu hafıza kapasitesi CISC işlemciler için düşük gibi görünebilir ancak PIC'in RISC denetleyici olması birçok işin bu kapasitede uygulanmasına olanak vermektedir.

PIC serisi tüm denetleyiciler herhangi bir ek bellek veya giriş/çıkış elemanı gerektirmeden sadece 2 adet kondansatör, 1 adet direnç ve bir kristal ile çalıştırılabilmektedir. Tek bacaktan 40 mA akım çekilebilme ve entegre toplamı olarak 150 mA akım akıtma kapasitesine sahiptir. Entegrenin 4 Mhz osilator frekansında çektiği akım; çalışırken 2 mA, stand-by durumunda ise 20uA kadardır.

PIC 16C54 'un mensup olduğu denetleyici ailesi 12Bit core 16C5X olarak anılır. Bu gruba temel grup adı verilir. Bu ailenin üyesi diğer denetleyiciler PIC16C57, PIC16C58 ve dünyanın en küçük işlemcisi olarak anılan 8 bacaklı PIC12C508 ve PIC 12C509'dur.

Interrupt (Kesme) kapasitesi ilk denetleyici ailesi olan 12Bit Core 16C5X ailesinde bulunmamaktadır. Daha sonra üretilen ve orta sınıf olarak tanınan 14Bit Core- 16CXX ailesi birçok açıdan daha yetenekli bir grup işlemcidir.

Bu ailenin temel özelliği interrupt kapasitesi ve 14 bitlik komut işleme hafızasıdır. Bu özellikler PIC'i gerçek bir denetleyici olmaya ve karmaşık işlemlerde kullanılmaya yatkın hale getirmiştir. PIC16CXX ailesi en geniş ürün yelpazesine sahip ailedir. 16CXX ailesinin en önemli özellikleri seri olarak devre üstünde dahi programlanmasıdır.

PIC 16CXX ailesinin amatör elektronikçiler arasında en çok tanınan ve dünyada üzerinde ençok proje üretilmiş elemanı ise PIC16C84 veya PIC16F84 tur.

PIC 16F84 un bu kadar popüler olması onun çok iyi bir denetleyici olmasından ziyade program belleğinin EEPROM (Elektrikle silinip yazılabilen bellek) olmasından kaynaklanmaktadır. Seri olarak dört adet kabloyla programlanması da diğer önemli avantajıdır. Bugüne kadar amatörce bir işlemciyle uğraşmış herkesin en büyük sıkıntısı EPROM veya EPROM tabanlı denetleyicileri programladıktan sonra UltraViole ışık kaynağı ile silip tekrar programlamaktır. Bu çok zahmetli ve bir amatör için ekipman gerektiren yöntem olmuştur.

Tasarlanan PIC kontrollü çift kanallı dijital termometrede sistemi kontrol etmek için kullanılan PIC 16F877 ise PIC 16F84'ün tüm özelliklerini taşımaktadır.Ayrıca PIC 16F877 'de PIC16F84 'e ek olarak bazı özellikleri yer almaktadır

2.1 PIC Çeşitleri

Microchip ürettiği mikrodenetleyicileri 4 gruba ayırarak isimlendirmiştir. Her bir grubu ise bir PIC ailesi olarak adlandırmıştır. PIC ailelerine isim verilirken kelime boyu (word lenght) göz önüne alınmıştır. Bu kısımda kelime boyunun ne anlama geldiğini açıklamakta fayda vardır. Mikroişemciler (CPU) veya mikrodenetleyiciler (MCU) kendi içlerindeki dahili veri saklama alanları olan kayededicileri arasındaki veri alışverişini farklı sayıdaki bitlerle yaparlar. Örneğin 8088 mikroişlemcisi çip içerisindeki veri alışverişini 16 bit ile yaparken, pentium işlemcileri 32 bitlik verilerle iletişim kurarlar. Bir CPU veya MCU'nun dahili veri yolu uzunluğuna *kelime boyu* denir.

Microchip PIC'leri 12/14/16 bitlik kelime boylarında üretilmektedir ve buna göre aşağıdaki aile isimleri mevcuttur.

- PIC 16C5XX ailesi 12 bit kelime boyu
- PIC 16CXXX ailesi 14 bit kelime boyu
- PIC 17CXXX ailesi 16 bit kelime boyu
- PIC 12CXXX ailesi 12 bit/14 bit kelime boyu

Bir MCU çip dışındaki harici ünitelerle veri alışverişini kaç bitle yapıyorsa buna *veri yolu* bit sayısı denir. PIC'ler farklı kelime boylarında üretilmelerine rağmen harici veri yolu tüm PIC ailelerinde 8 bittir. Yani bir PIC, G/Ç portu aracılığı ile çevresel ünitelerle veri alışverişi yaparken 8 bitlik veri yolu kullanır.

PIC programcıları program kodlarını yazarken bir komutun kaç bitlik kelime boyundan oluştuğu ile pek fazla ilgilenmezler. Seçilen bir çipi programlarken uyulması gereken kuralları ve o çiple ilgili özelliklerin bilinmesi yeterlidir. Bu özellikler PIC'in bellek miktarı, G/Ç portu sayısı, A/D dönüştürücüye sahip olup olmadığı, kesme (interrupt) fonksiyonlarının bulunup bulunmadığı, bellek tipinin ne olduğu (Flash, EPROM, EEPROM vb) gibi bilgilerdir.

2.2 PIC Mikrodenetleyicilerinin Tercih Sebepleri

a-) Fiyatının ucuz olması;
b-) Mantıksal işlemlerde performansının yüksek olması;
c-) Verilere ve belleğe hızlı bir şekilde erişimin sağlanması;
d-) 8 bitlik bir mikrodenetleyici olması ;
e-) Veri ve bellek için arı yolların (bus'ların) ayrılmış olması;
f-) Yüksek frekanslarda çalışabilme özelliği;
g-) Uyku modunda (Sleep mode) 1μA gibi küçük bir akım çekmesi;
h-) 14 bitlik komut işleme hafızası;
i-) Yalnızca 2 kondansatör ve bir direnç ile çalışabilme özelliği;
j-) RISC mimarisine sahip olması;

2.3 PIC'in Özellikleri

- **Güvenirlik:**PIC komutları bellekte çok az yer kaplarlar. Dolayısıyla bu komutlar 12 veya 14 bitlik bir program bellek sözcüğüne sığarlar. Harward mimarisi kullanılmayan mikrodenetleyicilerde yazılım programının veri kısmına atlama yaparak bu verilerin komut gibi çalışmasını sağlamaktadır . Bu ise büyük hatalara yol açmaktadır. PIC' lerde bu durum engellenmiştir.

- **Hız:** PIC oldukça hızlı bir mikrodenetleyicidir. Her bir komut satırı 1µsn'lik bir zaman diliminde işlenir. Örneğin 5 milyon komutluk bir programın 20Mhz' lik bir kristalle işletilmesi yalnız 1sn sürer. Bu süre kabaca 386 diye tanımladığımız sayısal bilgisayarın hızının yaklaşık 2 katıdır. Ayrıca PIC'lerin RISC mimarisine sahip olmasının hıza etkisi oldukça büyüktür.

- **Komut Takımı:**PIC'te bir işlem gerçekleştirmek için kullanılacak komut sayısı oldukça azdır. Örneğin PIC16F8XX ailesinde 33 komutu kullanarak sınırsız sayıda işlem yapabilmek mümkündür.

- **Statik işlem:**PIC mikrodenetleyici tamamıyla statik bir işlemcidir. Bu da demek oluyor ki işlemciye pals sağlayan osilasyon kaynağı durdurulsa bile işlenen veriler muhafaza edilmektedir.

- **Sürme özelliği:** PIC'ler yüksek bir sürme kapasitesine sahiptir. Çıkış olarak tanımlanan pinlerin yalnız birinin aktif olması halinde 40mA çekilebilmektedir. Entegre elemanın tamamı düşünüldüğünde ise 150 mA'e kadar akım çekilebilmektedir.

- **Güvenlik:** PIC üretim özelliği itibariyle bir protect yani koruma bitine sahiptir. Bu bitin programlanması yolu ile PIC içerisine yazılan programın başkaları tarafından okunması ve kopyalanmasına engel olunmuş olunur.

- **Flash olma özelliği:** Bu özellik PIC'in yeniden programlanabilir olması durumunu ifade etmektedir. Yani PIC üzerine yazılan program geliştirme amacı ile silinebilir ve yeni bir program yüklenebilir.

2.4 Bir PIC'in İşlem Yapabilmesi İçin Gerekli Bileşenler

- **Giriş-Çıkış (I/O):** Mikrodenetleyicinin dış dünya ile ilişkisini sağlayan, girdi ve çıktı şeklinde ayarlanabilen bir bağlantı pinidir.

- **Yazılım:** Mikrodenetleyicinin çalışmasını ve işletilmesini sağlayan bilgidir. Başarılı bir uygulama için yazılım hatasız olmalıdır. Yazılım C, Pascal veya Assembler gibi çeşitli dillerde veya ikilik(binary) olarak yazılabilir

- **Donanım:** Mikrodenetleyiciyi, bellek, arabirim bileşenleri, güç kaynakları, sinyal düzenleyici devreler ve bunları çalıştırmak ve arabirim görevini üstlenmek için bu cihazlara bağlanan tüm bileşenlerdir

- **Simülatör:**PC üzerinde çalışan ve mikrodenetleyicinin içindeki işlemleri simüle eden MPSIM gibi bir yazılım paketidir. Hangi olayların ne zaman meydana geldiği biliniyorsa bir simülatör kullanmak tasarımları test etmek için kolay bir yol olacaktır. Öte yandan simülatör, programları tümüyle veya adım adım izleyerek hatalardan arındırma fırsatı sunar. Şu anda en gelişmiş simülatör programı Microchip firmasının geliştirdiği MPLAB programıdır.

- **ICE** : PIC MASTER olarak da adlandırılır. (In- Circuit Emulator / İç devre takipçisi) PC ve Mikrodenetleyicinin yer alacağı soket arasına bağlanmış yararlı bir gereçtir. Bu gereç yazılım, PC de çalışırken devre kartı üzerinde bir mikrodenetleyici gibi davranır. ICE, bir programa girilmesini, mikro içinde neler olduğunu ve dış dünyayla nasıl iletişim kurulduğunun izlenilmesini mümkün kılar.

- **Programlayıcı** : Yazılımın mikrodenetleyicinin belleğinde programlamasını ve böylece ICE' nin yardımı olmadan çalışmasını sağlayan bir birimdir. Çoğunlukla seri port 'a (örneğin ICSTART, PROMASTER) bağlanan bu birimler çok çeşitli biçim, ebat ve fiyatlara sahiptir.

- **Kaynak Dosyası** : Hem asembler' in hem de tasarımcının anlayabileceği dilde yazılmış bir programdır. Kaynak dosya mikrodenetleyicinin anlayabilmesi için önceden assemble edilmiş olmalıdır.

- **Assembler** : Kaynak dosyayı bir nesne dosyaya dönüştüren yazılım paketidir. Hata araştırma bu paketin yerleşik bir özelliğidir. Bu özellik assemble edilme sürecinde hatalar çıktıkça programı hatalardan arındırırken kullanılır. MPASM, tüm PIC ailesini elinde tutan Microchip' in son assemble edicisidir.

- **Nesne dosyası (object file)** : Assembler tarafından üretilen bu dosya; programcı, similatör veya ICE' nin anlayabilecekleri ve böylelikle dosyanın işlevlerinin çalışmasını sağlayabilecekleri bir dosyadır. Dosya uzantısı assemble edicinin emirlerine bağlı olarak , .OBJ veya .HEX olur.

3-PIC 16F877

3.1 Genel Tanımlama

PIC 16F877 yüksek performanslı, CMOS, full-statik, 8 bit mikrodenetleyicidir. Tüm PIC 16/17 mikrodenetleyicileri gibi PIC 16F877 de RISC mimarisini kullanmaktadır. PIC16F87X mikroları birçok esas özelliklere sahiptir. 14 seviyeli, derin küme ve çoklu iç ve dış kesme kaynaklarına sahiptir. 2 aşamalı komut hattı tüm komutların tek bir saykıl' la (çevrimle) işlenmesini sağlamaktadır. Yalnızca bazı özel komutlar 2 saykıl çekerler. Bu komutlar dallanma komutlarıdır. PIC16F87X ailesi dış elemanları azaltacak spesifik özelliklere sahiptir ve böylece maliyet minimuma inmekte, sistemin güvenirliği artmakta, enerji sarfiyatı azalmaktadır. Bunun yanı sıra tüm PIC'lerde 4 adet osilatör seçeneği mevcuttur. Bunlarda tek pinli RC osilatör, düşük maliyet (4 MHZ) , LP osilatör (Kristal veya seramik rezonatör) , enerji sarfiyatını minimize etmekte (asgari akım) (40 KHZ), XT kristal veya seramik rezonatör osilatörü standart hızlı ve HS kristal veya seramik rezonatörlü osilatör çok yüksek hıza sahiptir (20 MHZ). PIC mikrodenetleyicilerinin en büyük özelliği sleep modu özelliğidir. Bu mod sayesinde işlem yapılmadığı durumlarda PIC uyuma moduna geçerek çok düşük akım çeker. Kullanıcı bir kaç iç ve dış kesmelerle PIC' i uyuma modundan çıkarabilmektedir. Yüksek güvenilirlikli Watchdog Timer kendi bünyesindeki çip üstü RC osilatörü ile yazılımı kilitlemeye karşı korumaktadır. PIC16F877 EEPROM program belleği , aynı aygıt paketinin orjinali ve üretimi için kullanılmasına olanak vermektedir. Yeniden programlanabilirliği mikroyu uygulamanın sonundan kaldırmadan kodu güncelleştirmeye izin vermektedir. Bu aygıtın kolayca erişilemediği, fakat prototipinin kod güncelleştirmesi gerekli olduğu durumlarda, bir çok uygulamanın geliştirilmesinde yararlıdır. Bunun yanı sıra bu kodun güncelleştirilmesi diğer ayrı uygulamalarda da yararlıdır.

3.2 PIC 16F877'nin Genel Özellikleri

- Yüksek hızlı RISC işlemciye sahiptir;
- 35 adet komut mevcuttur;
- Tüm komutlar 1 saykıl çeker, (Dallanma komutları 2 saykıl çeker.);
- 20 Mhz'ye kadar işlem hızına sahiptir;
- 8Kx14 word'lük flash program belleği mevcuttur;
- 368x8 bayt'lık data belleği;
- 256x8 byte'lık EEPROM data belleği;
- PIC16C73B/74B/76/77 ile uyumlu pin yapısı;
- Doğrudan ve dolaylı adresleme;
- Power-on Reset (POR), Power-up Timer (PWRT) , üzerinde bulunan RC osilatör ile çalışan Watchdog Timer (WDT);
- Programlanabilen kod koruma;
- Enerji tasarrufu için uyku (SLEEP) modu;
- Düşük güçlü yüksek hızlı CMOSFLASH/EEPROM teknolojisi;
- Tamamen statik dizayn;
- Devre üzerinde seri programlama;
- 5 V'luk kaynak ile çalışma;
- 2 V ile 5.5 V arasında işlem yapabilme özelliği;
- Düşükj güç harcaması;

- < 2 mA typical @ 5V, 4 MHz

-20 mA typical @ 3V, 32 kHz

-< 1 mA typical standby

3.3 PIC 16F877'nin Belirleyici Özellikleri

- Timer0: 8 bit pescaler'e sahip 8bit zamanlayıcı/sayıcı,
- Timer1:Sleep modunda artış gösterebilen ve harici saat darbesiyle artırılabilen Prescaler' li 16 bit zamanlayıcı/sayıcı,
- Timer2:8bit peryot kaydedicili,prescaler ve postscalerli 16bit zamanlayıcı/sayıcı,
- İki adet tutma,karşılaştırma, PWM modülü
- 200ns çözünürlükte 16 bitlik karşılaştırma
- 10 bit çözünürlükte PWM
- 10 bit çok kanallı Analog-Dijital çevirici

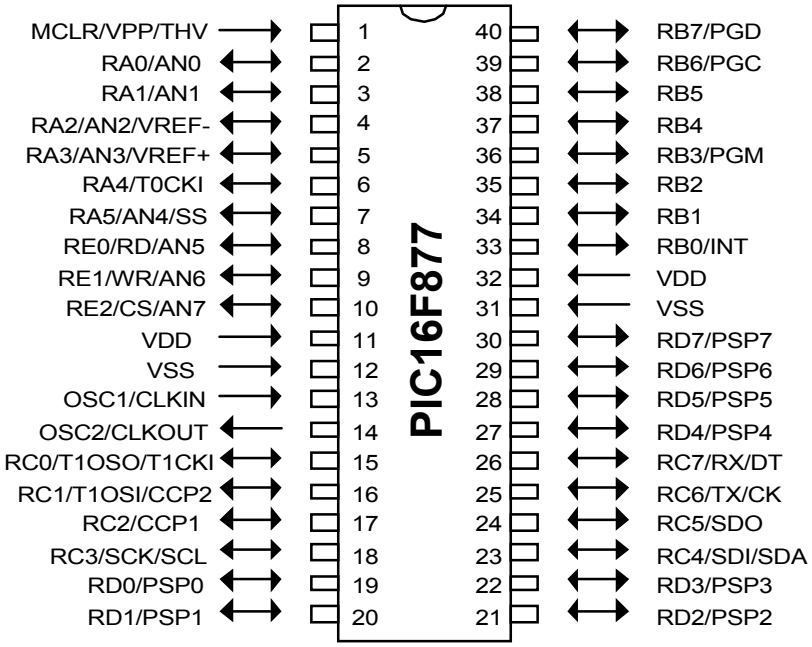
- Seri port ve I^2C modülleri
- 9 bit adres saptamaya sahip USART/SCI
- 8 bit genişliğinde paralel slave port

Aşağıda tablo de PIC16F877'nin PIC 16F84 ile karşılaştırılması yer almaktadır.

PIC 16F877 ile PIC 16F84'ün karşılaştırılması

ÖZELLİKLER	**PIC 16F877**	**PIC 16F84**
Çalışma hızı	DC-20 Mhz	DC-10 Mhz
Program belleği	8Kx14 word Flash ROM	1Kx14 word Flash ROM
Eeprom belleği	256 byte	64 byte
Kullanıcı RAM	368x8 byte	68 x 8 byte
Giriş/Çıkış port sayısı	33	13
Timer	Timer0,Timer1,Timer2	Timer0
A/D Çevirici	8 kanal 10 bit	YOK
Capture/Comp/PWM	16 bit Capture 16 bit compare 10 bit PWM çözünürlük	YOK
Seri çevresel arayüz	SPI ve I^2C modunda SPI portu (senkron seri port)	YOK
Parelel slave port	8 bit, harici RD,WR ve CS kontrollu	YOK
USART/SCI	9 bit adresli	YOK

3.4 PIC 16F877'nin Fiziksel Yapısının İncelenmesi



Şekil 3.PIC 16F877'nin bacak yapısı

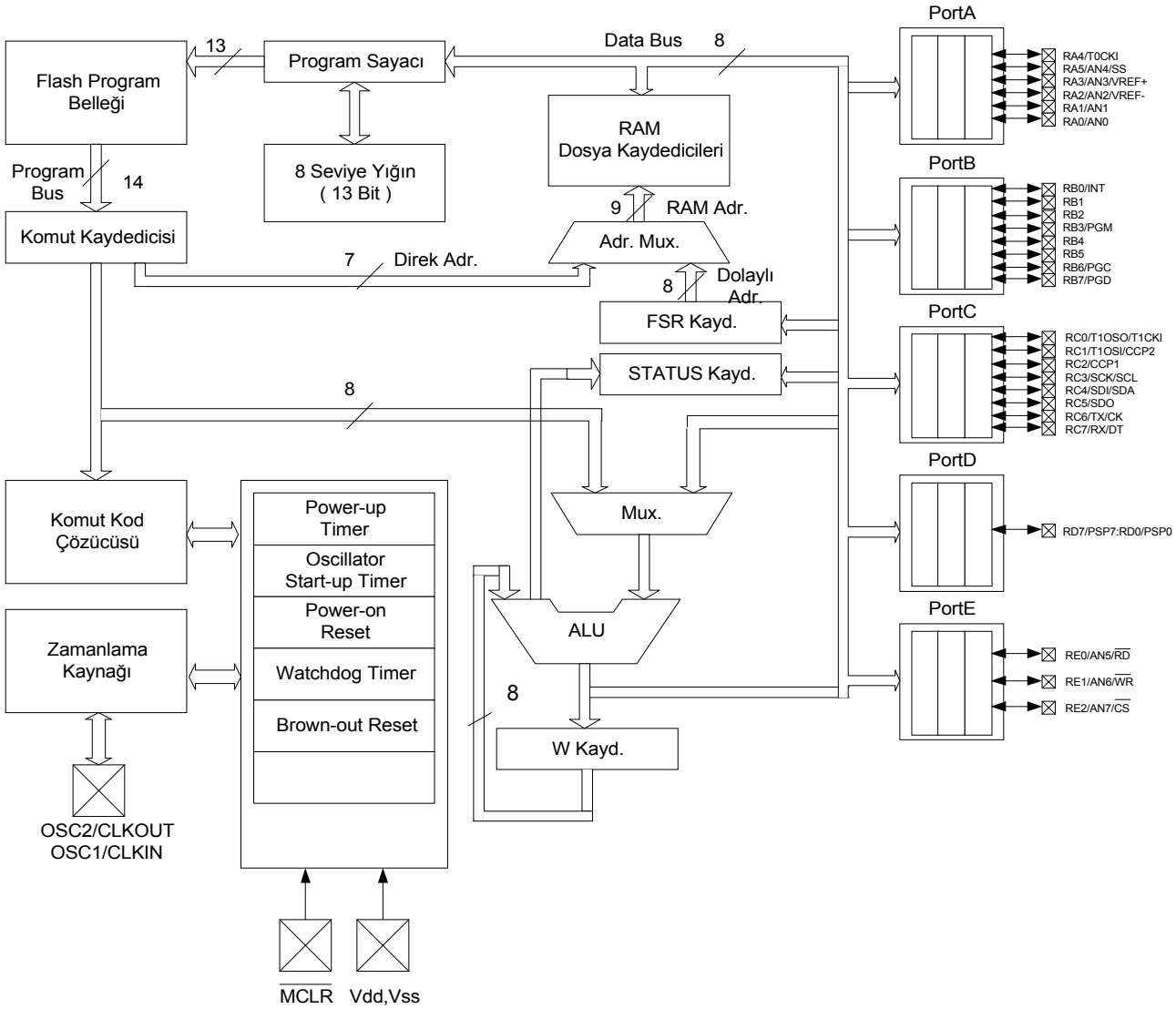
3.5 PIC 16F877 Pin Tanımlamaları

Tablo 3.2:PIC 16F877'de Pin Tanımlamaları

OSC1/CLKIN	13	G	ST/CMOS	Kristal osilatör girişi/Harici osilatör kaynağı girişi
OSC2/CLKOUT	14	Ç	—	Kristal osilatör çıkışı.RC osilatör modunda 1/4 f değerinde frekans çıkışı
MCLR/VPP/THV	1	G/P	ST	Mikrodenetleyici için reset ucu. Normal çalışmada 1 seviyesinde tutulur
RA0/AN0	2	G/Ç	TTL	PORTA:G/Ç olarak yönlendirilebilir port.Pinler G/Ç görevi dışında; RA0: 0. Analog giriş görevi yapar. RA1: 1. Analog giriş görevi yapar. RA2: 2. Analog giriş veya negatif referans gerilimi girişim görevi yapar. RA3: 3. Analog giriş veya pozitif referans gerilimi girişim görevi yapar. RA4: TIMER0 için clock girişi görevi yapar. Açık drain çıkışa sahiptir. RA5: 4. Analog giriş veya SSP için slave seçimi görevi yapar.
RA1/AN1	3	G/Ç	TTL	
RA2/AN2/VREF-	4	G/Ç	TTL	
RA3/AN3/VREF+	5	G/Ç	TTL	
RA4/T0CKI	6	G/Ç	ST	
RA5/SS/AN4	7	G/Ç	TTL	
RB0/INT	33	G/Ç	TTL/ST	PORTB: G/Ç olarak yönlendirilebilir port.Tüm girişleinde yazılımla programlanabilir düşük değerli pull-uplar vardır.Pinler G/Ç görevi dışında; RB0:Harici kesme ucu görevi yapar. RB3:Düşük seviye programlama girişi görevi yapar. RB6:Seri programlama girişi görevi yapar RB7:Seri programlamada data girişi görevi yapar.
RB1	34	G/Ç	TTL	
RB2	35	G/Ç	TTL	
RB3/PGM	36	G/Ç	TTL	
RB4	37	G/Ç	TTL	
RB5	38	G/Ç	TTL	
RB6/PGC	39	G/Ç	TTL/ST	
RB7/PGD	40	G/Ç	TTL/ST	
RC0/T1OSO/T1CKI	15	G/Ç	ST	PORTC: G/Ç olarak yönlendirilebilen port .Pinler G/Ç görevi dışında; RC0:TIMER1 osilatör çıkışı veya TIMER1 clock çıkışı görevi de yapar. RC1:TIMER1 osilatör girişi veya Capture2-G/Compare2-O/PWM2-Ç görevi yapar. RC2:Capture1-G/Compare1-Ç/PWM1-Ç görevi de yapar. RC3: SPI ve I2C modunda senkron seri clock G/Ç görevi yapar. RC4: SPI modunda SPI data giriş, I2C modunda data G/Ç görevi yapar. RC5: SPI modunda SPI data çıkış görevi yapar. RC6:USART asenkron gönderme veya senkron clock görevi yapar. RC7: USART asenkron alma ve senkron data görevi yapar.
RC1/T1OSI/CCP2	16	G/Ç	ST	
RC2/CCP1	17	G/Ç	ST	
RC3/SCK/SCL	18	G/Ç	ST	
RC4/SDI/SDA	23	G/Ç	ST	
RC5/SDO	24	G/Ç	ST	
RC6/TX/CK	25	G/Ç	ST	
RC7/RX/DT	26	G/Ç	ST	
RD0/PSP0	19	G/Ç	ST/TTL	PORTD:G/Ç olarak yönlendirilebilir port veya mikroişlemci hattında arabirim olarak kullanıldığında paralel slave port.
RD1/PSP1	20	G/Ç	ST/TTL	
RD2/PSP2	21	G/Ç	ST/TTL	
RD3/PSP3	22	G/Ç	ST/TTL	
RD4/PSP4	27	G/Ç	ST/TTL	
RD5/PSP5	28	G/Ç	ST/TTL	
RD6/PSP6	29	G/Ç	ST/TTL	
RD7/PSP7	30	G/Ç	ST/TTL	
RE0/RD/AN5	8	G/Ç	ST/TTL	PORTE: G/Ç olarak yönlendirilebilir port . Pinler G/Ç görevi dışında; RE0:Paralel Slave porttan okuma kontrolu veya 5. analog giriş görevi yapar. RE1:Paralel Slave porttan yazma kontrolu veya 6. analog giriş görevi yapar. RE2:Paralel Slave porttan seçim kontrolu veya 7. analog giriş görevi yapar.
RE1/WR/AN6	9	G/Ç	ST/TTL	
RE2/CS/AN7	10	G/Ç	ST/TTL	
VSS	12,31	P	—	Mikrodenetleyici için toprak seviyesini oluşturur.
VDD	11,32	P	—	Mikrodenetleyici için pozitif kaynak gerilimini oluşturur.

Tablo de; P:Power, G/Ç:Giriş/Çıkış, ST: Schmitt Trigger giriş, G:Giriş, Ç:Çıkış, TTL:Transistor-Transistor lojik giriş

3.6 PIC 16F877'nin Basitleştirilmiş İç Yapısı



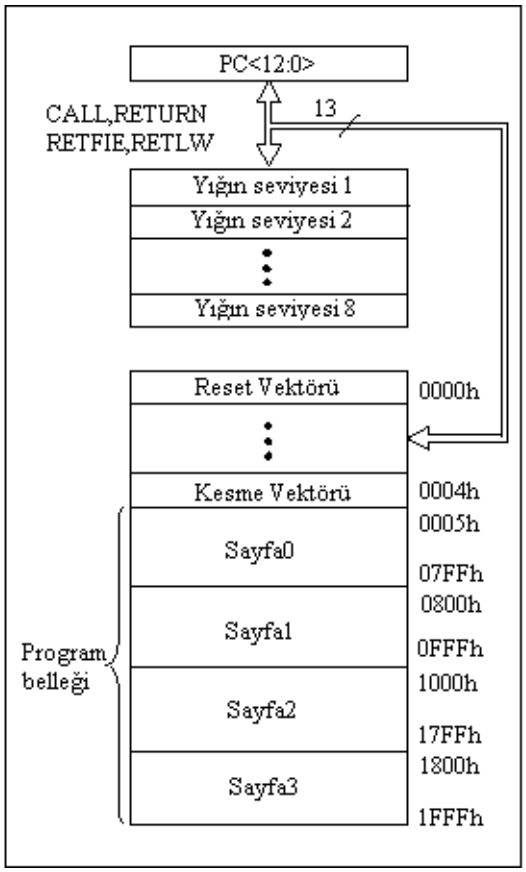
Şekil : PIC 16F877 basitleştirilmiş iç yapısı

3.7 Bellek Organizasyonu

Her PIC mikrodenetleyicisinde 3 bellek bloğu bulunmaktadır. Bunlar program belleği,veri belleği ve bunları ayıran veri hattıdır. Her bir bellek kendi taşıyıcısına sahiptir; böylece her bir bloğa erişim aynı osilatör süreci boyunca meydana gelebilmektedir. Bunun ötesinde, veri belleği genel amaçlı RAM ve özel fonksiyon kayıtları (SFR) olmak üzere ikiye bölünür. SFR`ler her bir bireysel özelleşmiş modülü ele alan bölümde açıklanan özel modülleri kontrol etmek için kullanılmaktadır. Veri belleği EEPROM veri belleğini de içermektedir. Bu bellek, direkt veri belleğine planlanmamış, fakat indirekt olarak planlanmıştır; ve indirekt adres göstergeleri okumak/yazmak için EEPROM belleğinin adresini belirlemektedir.

3.7.1 Program Bellek Organizasyonu

PIC 16F877 13 bit program sayacına ve 8Kx14 adresleme kapasitesine sahiptir. PIC16F877 denetleyicisi 8Kx14 FLASH program belleğine sahiptir. Reset vektörü 0000h ve kesme vektörü 0004h adresindedir. Şekil ’te PIC16F877 program bellek haritası görülmektedir.



Şekil 3.3:PIC16F877 program bellek haritası

3.7.2 Veri Bellek Organizasyonu

Veri belleği genel amaçlı yazmaçlar ve özel işlev yazmaçları (SFR) olmak üzere ikiye ayrılır. RP0 ve RP1 bitleri küme seçimi için ayrılmış bitlerdir. Her bir bank (küme) 7Fh' ye kadar (128 bayt) uzanır. Her bank'ın alt kısımları özel işlev yazmaçları için ayrılır. Üstteki özel işlev yazmaçları ise statik RAM olarak kullanılan yazmaçlardır. Bütün banklarda özel işlev yazmaçları vardır. Özel işlev yazmaçlarındaki yüksek kullanım bir banktan kod indirgemesi ve hızlı erişim için başka bankta gösterilebilir.

RP1	RP0

= 00 Bank0

= 01 Bank1

= 10 Bank2

= 11 Bank3

3.8 Özel Fonksiyon Kaydedicileri

Özel fonksiyon kaydedicileri gerçek bellek birimleri olarak gözükmeseler de, PIC içerisinde veri belleği adreslerinde tanımlanmış sıradan bellek hücreleridir. Bu kaydediciler programlama esnasında bir nevi kayıt tutma görevi üstlenirler. Tablo 3.3'te özel fonksiyon kaydedicilerinin isimleri ve adresleri verilmiştir.

Tablo : PIC 16F877'de Özel Fonksiyon Kaydedicileri ve Adresleri

BANK 0		BANK 1		BANK 2	
00	INDF	80	INDF	100	INDF
01	TMR0	81	OPTION_REG	101	TMR0
02	PCL	82	PCL	102	PCL
03	STATUS	83	STATUS	103	STATUS
04	FSR	84	FSR	104	FSR
05	PORTA	85	TRISA	105	—
06	PORTB	86	TRISB	106	PORTB
07	PORTC	87	TRISC	107	—
08	PORTD	88	TRISD	108	—
09	PORTE	89	TRISE	109	—
0A	PCLATH	8A	PCLATH	10A	PCLATH

0B	INTCON	8B	INTCON	10B	INTCON
0C	PIR1	8C	PIE1	10C	EEDATA
0D	PIR2	8D	PIE2	10D	EEADR
0E	TMR1L	8E	PCON	10E	EEDATH
0F	TMR1H	8F	—	10F	EEADRH
				BANK 3	
10	T1CON	90	—	180	INDF
11	TMR2	91	SSPCON2	181	OPTION_REG
12	T2CON	92	PR2	182	PCL
13	SSPBUF	93	SSPADD	183	STATUS
14	SSPCON	94	SSPSTAT	184	FSR
15	CCPR1L	95	—	185	—
16	CCPR1H	96	—	186	TRISB
17	CCP1CON	97	—	187	—
18	RCSTA	98	TXSTA	188	—
19	TXREG	99	SPBRG	189	—
1A	RCREG	9A	—	18A	PCLATH
1B	CCPR2L	9B	—	18B	INTCON
1C	CCPR2H	9C	—	18C	EECON1
1D	CCP2CON	9D	—	18D	EECON2
1E	ADRESH	9E	ADRESL	18E	—
1F	ADCON0	9F	ADCON1	18F	—

3.8.1 STATUS Kaydedicisi

Aritmetiksel işlemlerin yürütüldüğü, sıfırlama işlemlerinin gerçekleştirildiği ve küme kurma işlemlerinin gerçekleştirildiği kaydedicidir.

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.4: Status kaydedicisinin bit yerleşimi

Status,0 (C): (Carry Bit)Aritmetiksel işlemler sonucunda taşmaların kaydedildiği bittir. Bu bit bir işlem sonucunda 1 ise taşmanın olduğu, 0 ise taşmanın olmadığı anlaşılır.

Status,1(DC):(Digit Carry Bit)Bu bit; işlem sonucunda ilk 4 bitin dışına taşma olup olmadığının kaydını tutar. Eğer bu bit 1 ise 5. bite taşma olduğu, 0 ise 5. bite taşma olmadığı anlaşılır.

Status,2(Z):(Zero Bit)İşlem yapılan hücrede sıfırlamanın olup olmadığının kontrolünü yapar. 1 ise kontrolün yapıldığı hücrenin sayısal değeri 0 olmuş demektir, 0 ise hücre sıfırdan farklı bir sayısal değere sahiptir.

Status,3(PD):(Power Down Bit)Sleep modu ve watchdog timer ile ilgili işlemlerde kullanılır. Watchdog timer sıfırlandıktan sonra (CLRWDT) bu bit 1 yapılmalıdır. Mikrodenetleyici sleep moduna sokulur iken de bu bit 0 yapılarak işleme başlanır.

Status,4(TO):(Time Out Bit)PD biti gibi bu bit de sleep modu ve watchdog timer ile ilgili işlemlerde kullanılır. Watchdog timerin sıfırlanması işlemi sonunda bu bit 0 yapılırken, mikrodenetleyici sleep modundan çıkarılırken bu bit 1 yapılır.

Status,5,6(RP0-RP1):(Register Bank Select Bits) Küme seçimi ile ilgili işlemlerde kullanılırlar. İki bit bir arada düşünüldüğünde 4 konum mevcuttur ve bu dört konumun her birisinde de ayrı bir küme kurulmaktadır.(00=Bank0, 01=Bank1, 10=Bank2, 11=Bank3). Direkt adresleme modunda kullanılan bir bittir.

Status,7 (IRP): Status5,6 gibi bu bit de küme seçimi işlemlerinde kullanılır. Ancak bu bit dolaylı adresleme durumlarında kullanılır.(1 = Bank 2, 3 0 = Bank 0, 1)

3.8.2 OPTION Kaydedicisi

Bu kaydedici üzerinde portB, TMR0 ve dış kesmeleri düzenleyici bitler bulunmaktadır.

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.5: Option Kaydedicinin bit yerleşimi

Option,0,1,2(PS0,PS1,PS2):(Prescaler Rate Select Bits)Timer0(TMR0) ve Watchdog Timer (WDT)'ın süreç oranlarını belirleyen bitlrdir. Üç bit bir arada düşünüldüğü zaman 8 farklı durum ortaya çıkmaktadır. 8 durum için TMR0 ve WDT'ın aldığı durumlar Tablo 3.4 teverilmiştir.

Tablo 3.4:PS2,PS1,PS0'a göre TMR0,WDT oranları

PS2,PS1,PS0	TMRO Oranı	WDT Otranı
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

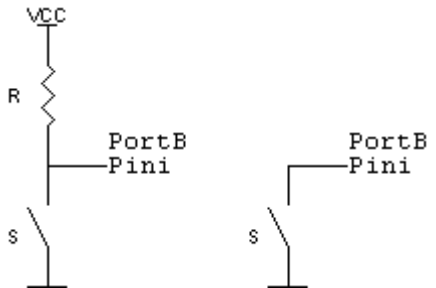
Option,3(PSA):(Prescaler Assignment Bit) Aldığı duruma göre TMR0 veya WDT'a tahsis edilir. Eğer 0 ise TMR0 üzerinde yapılan işlemlerde kullanılır, 1 ise WDT üzerinde yapılan işlemlerde kullanılır.

Option,4(TOSE):(Timer0 Source Edge select bit) Devrede harici clock kaynağı kullanılması durumunda palslerin düşen kenar mı yoksa yükselen kenar mı olduğunun tayininde işlem yapar. Eğer 1 ise RA4/TOCKI pininde kullanılan palslerin düşen kenar, 0 ise RA4/TOCKI pinindeki palslerin yükselen kenar olması gerekir.

Option,5(TOCS):(Timer0 Source select) PIC ile yapılan devrede hangi pals kaynağının kullanılacağına karar vermek için kullanılır. Eğer 1 ise palsler RA4/TOCKI pininden, 0 ise dahili pals kaynağından (CLKOUT) sağlanır.

Option,6(INTEDG):(Interrupt edge select)RBO/INT pininde kesme sinyali olması halinde yükselen kenar veya düşen kenar tetikleme gönderilmesi durumlarını ayarlar. 1 ise RBO/INT pininden yükselen kenar, 0 ise RBO/INT pininden düşen kenar bir tetikleme uygulanmalıdır.

Option,7(RBPU):(PortB Pull-up Control): Bu bit uygulamalarda alan daralmasını sağlayan bir bittir. Bu bitin 1 yapılması ile gerilimin pozitif ucu ile portB pini arasına bir direnç konulur. Bu direnç sayesinde PortB'nin bu pini lojik 1'de tutulmuş ulur. Bu pini lojik 0 yapmak için ise pin ile gerilimin negatif ucu arasını bir push buton ile kısa devre yapmak yeterli olacaktır. Bu bitin 0 yapılması halinde ise pinledeki pull-up dirençleri kaldırılır. Bu biti kullanır iken yalnız portB için pull-up direnci konulduğunu unutmamak gerekir.



Şekil 3.6:RBPU'nun analog düzen karşılığı

Şekil 3.6.a'da RBPU'nun kullanılmadığı durumda kararlılık için B portuna takılması gereken harici direncin bağlantısı görülmektedir. Şekil 3-6-b'de ise RBPU'nun kullanıldığı durum verilmiştir. Bu durumda harici bir dirence ihtiyaç duyulmamıştır.

3.8.3 INTCON Kaydedicisi

Bu kaydedici tüm kesmelerin kontrolü için bazı okunabilir ve yazılabilir bitler sağlayabilmektedir.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.7: Intcon kaydedicisinin bit yerleşimi

Intcon,0 (RBIF): B portuna ait kesme bayrak bitinde değişiklikler yapmaya yarar. Program başında RB7:RB4 bitlerinin sıfırlandığı varsayılır ise RB7:RB4 pinlerinden herhangibirisinin durum değiştirmesi halinde RBIF 1 olur. (0'dan 1'e) 1'den sıfıra düşme durumunda ise RBIF lojik olarak konum değiştirir ve 0 olur.

Intcon,1(INTF):RB0/INT pini ile ilişkilendirilmiştir. Bu bitin 0 olması durumunda RB0/INT aktif değildir, 1 olması durumunda ise aktiftir.

Intcon,2 (TOIF): TMRO da meydana gelen taşmaya göre işlem yapar. Eğerki TOIF 1 ise TMRO'da taşma vardır. Aksi durumda ise taşma yoktur.

Intcon,3 (RBIE): İşlevi RBIF'inkine benzer. PortB kesme yetkilendirme bitidir. 1 olması durumunda PortB kesme yetkilendirmesi aktif, diğer durumda ise aktif değildir.

Intcon,4 (INTE): RB0/INT kesme yetkilendirme bitidir. Bu bitin 1 olması halinde RB0/INT pini kesme yetkilendirmesi verilmiştir. 0 olması halinde RB0/INT pininde kesme yetkilendirmesi yoktur.

Intcon,5 (TOIE): TMR0 kesmesini yetkilendirme bitidir. TOIE=1 ise kesme yetkilendirmesi aktif durumdadır. 0 olması halinde ise TMR0 ‘daki kesme ketkilendirmesi kalkmış durumdadır.

Intcon,6 (PEIE):Çevresel Kesmeler için yetkilendirme bitidir.PEIE=1 olması durumunda çevresel kesmeler etkindir,PEIE= 0 olması durumunda çevresel kesmeler etkin değildir.

Intcon,7 (GIE): Geniş kapsamlı bir yetkilendirme bitidir. Bu bitin 1 yapılması ile program içerisinde tüm yetkilendirmeler aktif hale getirilebilir. 0 olma durumunda ise verilen tüm yetkilendirmeler kaldırılabilir

3.8.4 PIE1 Kaydedicisi

Çevresel kesmelerle ilgili işlemleri gerçekleştirmek için kullanılan kaydedicidir.

R/W-0	R/W-00	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE (1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.8: PIE1 kaydedicisinin bit yerleşimi

Pie1,0(TMR1IE): TMR1’de meydana gelen taşmayı saptamak için gerekli olan kesmeyi kontrol eden bittir. TMR1IE=1 ise TMR1 taşma kesmesi etkin olur,TMR1IE=0 ise TMR1 kesmesi kullanım dışı olur.

Pie1,1(TMR2IE): TMR2 ve PR2 nin denklik kesmesini kontrol eden bittir. Kesmeyi etkinleştirmek için bu bit 1 seviyesinde, kullanım dışı bırakmak için ise 0 seviyesinde tutulur.

Pie1,2(CCP1IE):CCP1 (Capture/Compare/PWM) kesmesini kontrol etmek için kullanılan bittir. Eğer bu bit 1 ise CCP1 kesmesi etkin olur, 0 ise CCP1 kesmesi kullanım dışı olur.

Pie1,3(SSPIE):Eşzamanlı seri port (SSP= Synchronous Serial Port) uygulamaları için kesme yetkilendirme bitidir. Bu bit 1 ise SSP kesmeleri etkin, 0 ise kullanım dışıdır.

Pie1,4(TXIE):USART gönderme kesmesini yetkilendirme bitidir. TXIE 1 ise kesme etkin, 0 ise kesme kullanım dışıdır.

Pie1,5(RCIE): USART alma kesmesini yetkilendirme bitidir. RCIE 1 ise kesme etkin, 0 ise kesme kullanım dışıdır.

Pie1,6(ADIE):A/D çevirme kesmesini yetkilendirme bitidir. ADIE=1 ise kesme etkin, ADIE=0 ise kesme kullanım dışı yapılır.

Pie1,7(PSPIE): Parallel Slave Port okuma/yazma kesmesini yetkilendirme bitidir. PSPIE 1 ise kesme etkin, 0 ise kullanım dışıdır.

3.8.5 PIR1 Kaydedicisi

Çevresel kesmelerin işlemlerini bitirip bitirmediklerini kontrol eden kaydedicidir.

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF (1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2If	TMR1If
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.9: PIR1 kaydedicisinin bit yerleşimi

Pir1,0(TMR1IF): TMR1'de meydana gelen taşmayı tutan bittir. Eğer bu bit 1 ise TMR1'de taşma vardır,0 ise taşma yoktur.

Pir1,1(TMR2IF): TMR2 ile PR2 arasındaki denkliği gösteren bittir. Eğer TMR2IF=1 olmuşsa TMR2 ve PR2 arasında bir denklik söz konusudur,TMR2IF= 0 ise böyle bir durum yoktur.

Pir1,2(CCP1IF):CCP1 işlemine dair bilgi veren bittir. Eğer bu bit 1 ise; capture modunda TMR1 kaydedicisi capture edilmiştir ve bu bitin temizlenmesi gerekir, compare modunda TMR1 kaydedicisi eşlik yönünden karşılaştırılmıştır ve bu bitin temizlenmesi gerekir,0 ise CCP işlemleri tamamlanmamıştır.

Pir1,3(SSPIF):Eşzamanlı seri port (SSP) uygulamalarının durumlarını gösteren bittir. Bu bit 1 ise SSP işlemleri tamamlanmıştır, 0 ise tamamlanmamıştır.

Pir1,4(TXIF): USART gönderme hattının durumunu gösteren bittir. TXIE 1 ise USART gönderme hattı boş,0 ise doludur.

Pir1,5(RCIF): USART alma hattının durumunu gösteren bittir. RCIE=1 ise USART alma hattı dolu,RCIF=0 ise boştur.

Pir1,6(ADIF):A/D çevirme işleminin tamamlanıp tamamlanmadığı hakkında bilgi veren bittir. ADIF 1 ise çevirme işlemi tamamlanmış, 0 ise tamamlanmamıştır.

Pir1,7(PSPIF): Parallel Slave Port okuma/yazma işleminin tamamlanıp tamamlanmadığını gösteren bittir. PSPIF 1 ise okuma/yazma işlemi tamamlanmıştır, 0 ise okuma/yazma işlemi henüz tamamlanmamıştır.

3.8.6 PIE2 Kaydedicisi

CCP2 çevresel kesmesi, SSP hat çakışması ve EEPROM yazma kesmesi kontrollerinin yapıldığı kaydedicidir.

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	(1)	—	EEIE	BCLIE	—	—	CCP2IE
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.10: PIE2 kaydedicisinin bit yerleşimi

Pie2,0 (CCP2IE): CCP2 (Capture/Compare/PWM) kesmesini kontrol etmek için kullanılan bittir. Eğer bu bit 1 ise CCP2 kesmesi etkin olur, 0 ise CCP1 kesmesi kullanım dışı olur.

Pie2,1-2: Kullanım dışıdır ve daima 0 seviyesindedir.

Pie2,3(BCLIE): Hat çakışma kesmesidir. BCLIE =1 olması durumunda bu kesme etkin, BCLIE=0 olması durumnda ise kullanım dışıdır.

Pie2,4(EEIE):EEProm yazma kesmesi yetkilendirme bitidir. EEIE=1 ise yazma kesmesi etkin,EEIE= 0 ise kullanım dışıdır.

Pie2,5: Kullanım dışıdır ve daima 0 seviyesindedir.

Pie2,6: Başka işlemler için ayrılmıştır.

Pie2,7: Kullanım dışıdır ve daima 0 seviyesindedir.

3.8.7 PIR2 Kaydedicisi

CCP2 çevresel kesmesi, SSP hat çakışma kesmesi ve EEPROM yazma kesmesinin yaptığı işlemler hakkında bilgi veren bitleri bulunduran bir kaydedicidir.

U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	(1)	—	EEIF	BCLIF	—	—	CCP2IF
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.11: PIR2 kaydedicisinin bit yerleşimi

Pie2,0 (CCP2IF): CCP2 işlemleri ile ilgili kayıt tutan bir bittir. Eğer bu bit 1 ise; CCP2 işlemleri tamamlanmıştır, 0 ise tamamlanmamıştır.

Pie2,1-2: Kullanım dışıdır ve daima 0 seviyesindedir.

Pie2,3(BCLIF): Hat çakışmasını izleyebileceğimiz bir bittir. BCLIF 1 ise SSP'de bir çakışma meydana gelmiştir, 0 ise böyle bir durum yoktur.

Pie2,4(EEIF):EEPROM yazma durumunu gösteren bittir. EEIF= 1 ise yazma İşlemi tamamlanmıştır, 0 ise yazma işlemi tamamlanmamıştır.

Pie2,5: Kullanım dışıdır ve daima 0 seviyesindedir.

Pie2,6: Başka işlemler için ayrılmıştır.

Pie2,7: Kullanım dışıdır ve daima 0 seviyesindedir.

3.9 PCL ve PCLATH Kaydedicileri

Program Sayıcısı (PC=Program Counter) hangi satırda işlem yapılacağının kayıtlarını tutar ve işletim için gidip komut alınması işlemlerini gerçekleştirir. 13 bit genişliğindedir. Düşük bayt PCL kaydedicisi olarak adlandırılır. Bu kaydedici okunabilir ve yazılabilir özelliğe sahiptir. Yüksek bayt ise PCH kaydedicisi olarak adlandırılır ve direkt olarak yazılamaz ve okunamaz. PCH kayıtında bütün güncelleştirmeler, PCLATH kaydedicisini gözden geçirirler.

3.9.1 STACK (Yığın)

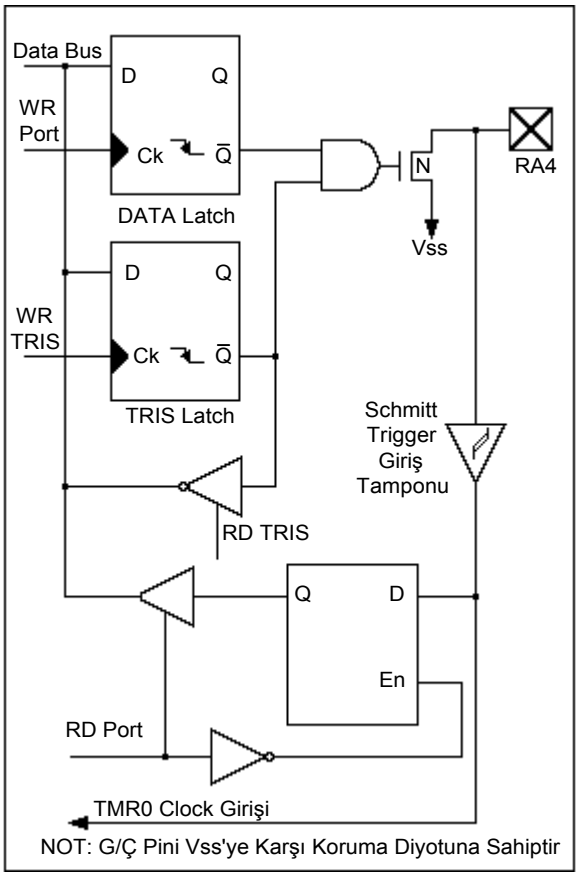
Stack (yığın) program içerisinde kesme ve çağırma neticesinde dallanılan 8 programın satır adreslerini tutar. Program işletilirken ise RETLW,RETURN,RETFIE gibi komutlar görüldüğünde gidilecek satırların adresleri de yine yığınlarda tutulur. 8 adet yığının herbirisi 13 bit genişliğe sahiptir ve bunlara program içerisinde müdahale edilemez.

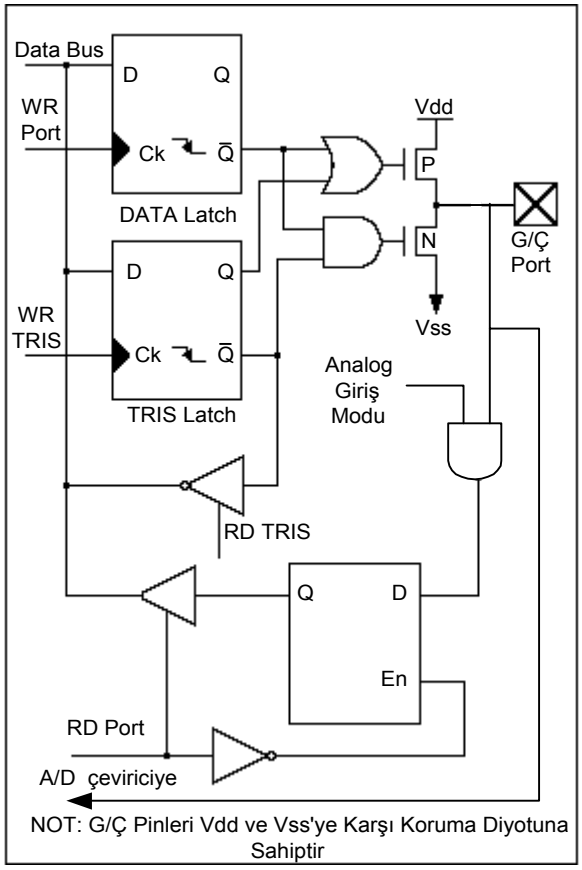
3.10 Giriş/Çıkış Portları (G/Ç Portları)

G/Ç portları giriş/çıkış vazifelerinin dışında bazı çevresel işlemleri de yapacak özelliklere sahiptirler. Çevre birimleri kullanıldığında genel amaçlı giriş/çıkış pini kullanılmaz.

3.10.1 PORTA ve TRISA Kaydedicisi

PORTA 6 bit giriş/çıkış olarak yönlendirilebilir porttur. Bu portu yönlendiren yazmaç ise TRISA yazmacıdır. TRISA kaydındaki herhangi bir bit 1 ise buna uygun çıkış sürücüsü yüksek direnç moduna getirilecektir. TRISA kaydındaki herhangi bir bitin 0 olması durumunda ise çıkış mandalı seçilen pinin üzerine getirilir. Analog giriş kullanıldığında TRISA yazmacı RA pininin yönünü kontrol eder. Şekil 3.12.a ve şekil 3.12.b A portunun iç yapısını göstermektedir.





Şekil 3.12.a :RA0:RA3 ve RA5 İç yapısı

Şekil 3.12.b:R A4/TOCKI İç Yapısı

Örnek 3.1: A portunun G/Ç olarak kurulması:

```
bcf     status, rp0     ;
bsf     status, rp0     ; Bank 1 seçildi.
movlw b'00000101'   ; PortA nın giriş ve çıkış olacak pinleri belirleniyor
movwf trisa         ; Giriş için RA0,RA2
                    ; Çıkış için RA1,RA3,RA4,RA5
```

3.10.2 PORTB ve TRISB Kaydedicisi

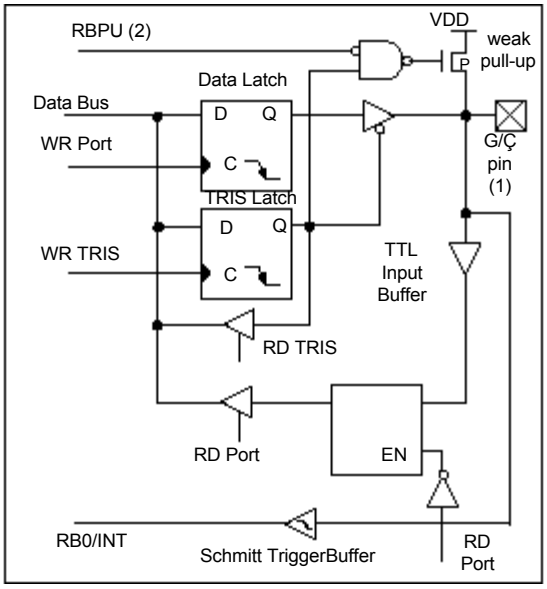
PORTB 8 bit giriş/çıkış olarak yönlendirilebilir porttur. Bu portu yönlendiren yazmaç ise TRISB yazmacıdır. TRISB kaydındaki herhangi bir bit 1 ise buna uygun çıkış sürücüsü yüksek direnç moduna getirilecektir. TRISB kaydındaki herhangi bir bitin 0 olması durumunda ise çıkış mandalı seçilen pinin üzerine getirilir. Analog giriş

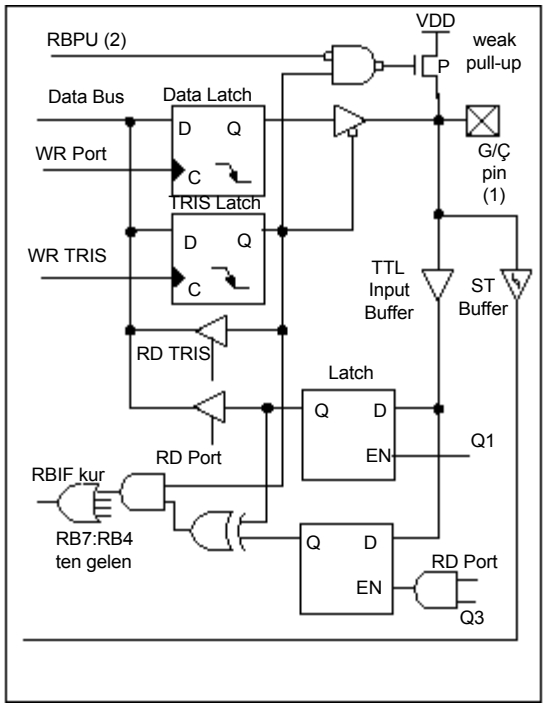
kullanıldığında TRISB yazmacı RB pininin yönünü kontrol eder. (1)

Her bir PORTB pini iç direnç düşürücü engellere sahiptir. RBPU(OPTION – REG<7) bitinin silinmesiyle aktif yapılır. Düşürücü engeller, port pini çıkış olarak konfigüre edildiği zaman otomatik olarak kapanmaktadır. Ayrıca dört PORTB pini, RB7: RB4 değişim özelliklerinde kesmelere sahiptir. Yalnızca giriş olarak konfigüre edilen pinlerkesmenin meydana gelmesine sebep olabilirler. (yani, herhangi bir çıkış olarak şekillendirilen RB7:RB4 pini değişim ilişkisi üzerindeki kesmeden hariç tutulmuştur.) Giriş modundaki pinlerin değeri PORTB` nin önceki okunmasındaki eski değeri ile karşılaştırılır. Pinlerin "uyuşmayan" kısımları RB port değişim kesmesini üretmek için birlikte OR'lanır. şekil 3.13.a ve şekil 3.13.b portunun iç yapısını göstermektedir.

Örnek 3.2 : B Portunun Kurulması:

```
bcf     status, rp0     ;
bsf     status, rp0     ; Bank 1 seçildi.
movlw b'10111111'   ; PortB nın G/Ç olacak pinleri belirleniyor
movwf trisb         ; Giriş için RB<3:0>
                    ; Çıkış için RB<5:4>
```





Şekil 3.13.a: RB3:RB0 Pinleri İç Yapısı

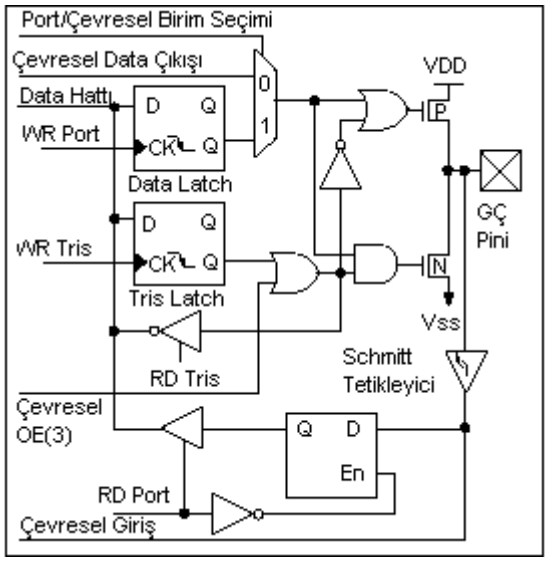
Şekil 3.13.b: R7:RB4 Pinleri İç Yapısı

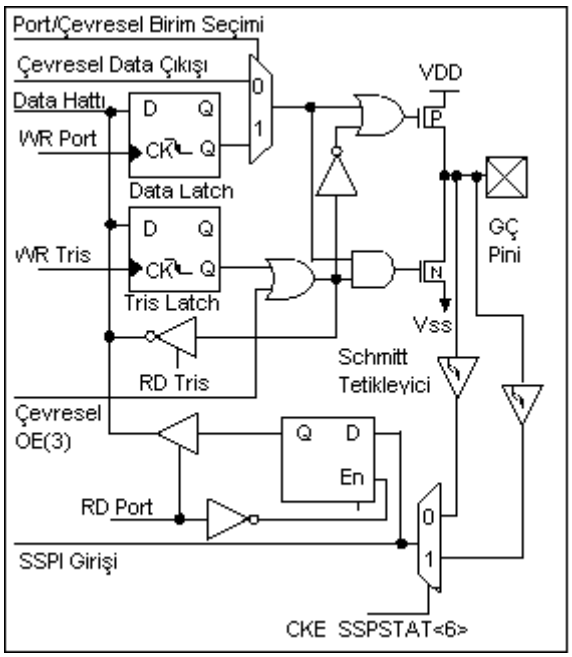
3.10.3 PORTC ve TRISC Kaydedicisi

PORTC 8 bit G/Ç olarak yönlendirilebilir porttur. Bu portu yönlendiren yazmaç ise TRISC yazmacıdır. TRISC Kaydındaki herhangi bir bit 1 ise buna uygun çıkış sürücüsü yüksek direnç moduna getirilecektir. TRISC Kaydındaki herhangi bir bitin 0 olması durumunda ise çıkış mandalı seçilen pinin üzerine getirilir. Analog giriş kullanıldığında TRISC yazmacı RC pininin yönünü kontrol eder. Şekil 3.14.a ve şekil 3.14.a C portunun iç yapısını göstermektedir.

Örnek 3.3: C Portunun Kurulması;

```
bcf     status, rp0     ;
bsf     status, rp0     ; Bank 1 seçildi.
movlw b'10111111'  ; Veri yönlendirmek
                        ; kullanılan değer
                        ; ayarlanıyor.
movwf trisc             ; Giriş için RC<3:0>
                        ; Çıkış için RC<5:4>
```

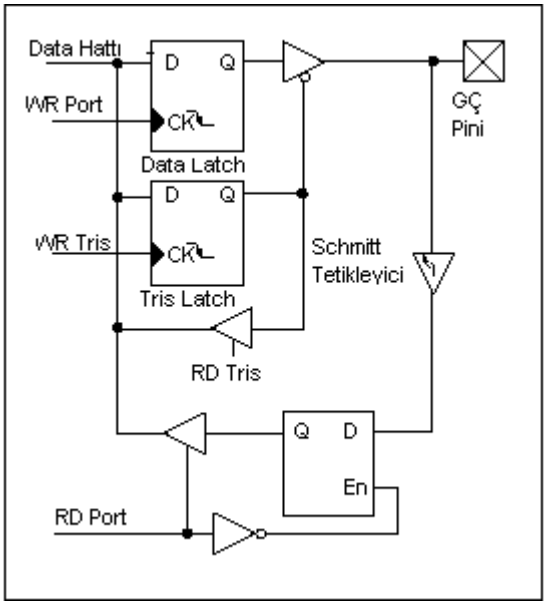




Şekil 3.14.a: RC0:2, RC5:7 Pinlerinin İç Yapısı **Şekil 3.14.b:**RC3:4 Pinlerinin İç Yapısı

3.10.4 PORTD ve TRISD Kaydedicisi

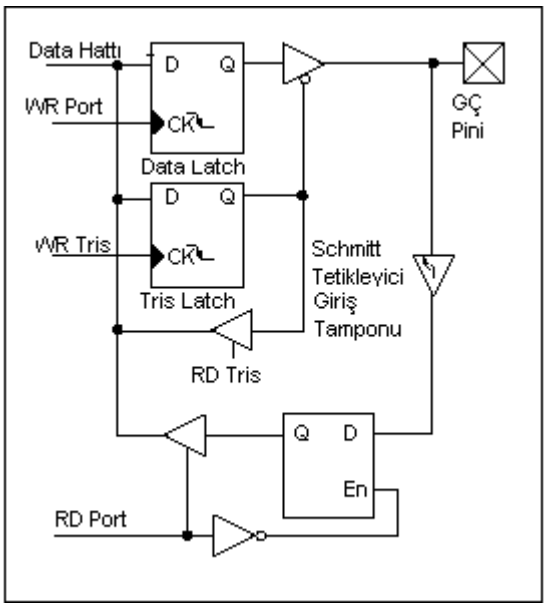
PORTD 8 bit Schmitt trigger tampon girişli porttur. Bu port PSPMODE (TRISD) denetim biti tarafından kurulur. Giriş tamponu TTL tampondur. Şekil 3.15 de D portunun blok diyagramı görülmektedir.



Şekil 3.15: D Portu I/O Blok Diyagramı

3.10.5 PORTE ve TRISE Kaydedicisi

Bu port RE0/RD/AN5, RE1/WR/AN6 ve RE2/CS/AN7 olmak üzere 3 adet pine sahiptir. Giriş veya çıkış portu olarak ayarlanabilir. PORTE kontrolü TRISE tarafından yapılır. PSPMODE(TRISE<4>) denetim biti tarafından ayarlanır. Şekil 3.16 da E portu blok diyagramı görülmektedir.



Şekil 3.16 : E Portu I/O Blok Diyagramı

3.11 Veri EEPROM ve Flash Program Hafızası

Veri EEPROM ve FLASH proram hafızası normal işlem süresinde okunabilme ve yazılabilme özelliğine sahiptir. Veri belleği, kaydedici dosyaya doğrudan planlanmamıştır. Bunun yerine bu bellek, özel fonksiyon kaydı üzerinden dolaylı olarak adreslenmiştir. Burada bu belleği okuyan ve yazan 6 özel kaydedici mevcuttur.

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

EEPROM veri belleği okuma ve yazmaya izin verir. EEDATA yazma/okuma için 8 bitlik veri tutar ve EEADR erişilen EEPROM adreslerini saklar. EEDATH ve EEADRH kaydedicileri veri EEPROM' u kullanmak için erişemezler. Bu aygıtlar Oh ile FFh genişliğindeki adresli EEPROM belleğinin 256 byte' ına sahiptir. EEPROM veri belleği yüksek silme/yazma süreçlerine oranlanmıştır. Yazma süresi yonga (chip) zamanlayıcısı tarafından denetlenmektedir. Yazma süresi chipten chipe olduğu gibi sıcaklık ve gerilimlede değişmektedir. Program hafızası kelime okuma ve yazmasına izin verir. Byte veya word verisi otomatik olarak silinir ve yeni veri yazılır. Program hafızası arabirimi bloklandığı zaman, EEDATH : EEDATA kaydedicileri oku/yaz için 14 bit veriyi 2 byte word halinde tutar. EEADRH ve EEADR kaydedicileri EEPROM'da 13 bitlik 2 byte word adresini tutar. Bu aygıtlar OH ile 3FFFh arasındaki adreste 8K word program EEPROM'una sahiptir. Program hafızasına yazılan değer bir talimat olmayı gerektirmez. Bu yüzden kalibrasyon parametreleri, seri numaraları ASCII kodunda depolanabilir.(6)

3.11.1 EEADR

EEADR kaydı EEPROM verisinin maksimum 256 byte'ını veya FLASH'ın maksimum 8K word'ünü adresleyebilir. Program adres değeri seçilirken EEADRH kaydedicisi adresin MSB'sini yazar, EEADR kaydedicisi ise LSB'sini yazar.(6)

3.11.2 EECON1 ve EECON2 Kaydedicileri

EECON1 belleğe ulaşmak için kontrol kaydedicisidir. EECON2 fiziksel kaydedici değildir. EECON2 yalnızca bellekteki yazı dizisini okumak için kullanılır. Eğer erişim, program veya veri belleğine ulaşma olacaksa buna EEPGD denetim biti karar verir. Veri belleği temizlendiğinde, sonradan yapılan işlemler veri belleğine işlenirler. Bellek ayarlandığında ise yine sonradan yapılan işlemler belleğe işlenirler. RD ve WR kontrol bitleri sırasıyla oku-yaz işlemlerini başlatır.

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	—	WRERR	WREN	WR	RD
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.17: EECON1 Kaydedicinin bit yerleşimi

Eecon1,0(RD): Okuma modu kontrol bitidir.1 olması durumunda EEPROM okuma modundadır, 0 olması durumunda ise EEPROM'dan oku modundan çıkılır.

Eecon1,1(WR):EEPROM'a data yazma işlemini kontrol eden bittir. Bilgi yazımı esnasında 1 yapılır ve yazımın sonlandırılması için 0 yapılır.

Eecon1,2(WREN):EEPROM'a yazma yetkilendirme bitidir. WREN=1 olması durumunda yazım için izin verilmiştir. WREN=0 olması durumunda ise data yazımı için yetki yoktur.

Eecom1,3(WRERR): Yazım işleminde ikaz bitidir. Yazım işleminde bir sebepten dolayı erken sonlandırma var ise bunu bildiren bittir. Eğer ki bu bit 1 ise, yapmak istediğimiz veri yazma işlemi erken sonlandırılmıştır ve bir hata söz konusudur. Bu durumda watchdog timer sıfırlanır veya mikrodenetleyici resetlenerek programın başına dönülür. 0 olma durumunda ise yazım işlemi başarı ile tamamlanmış demektir.

Eecon7,(EEPGD): Program belleği/veri belleği seçim bitidir. Eğer EEPGD=1 ise program belleğine erişim yerkisi verilir,EEPGD= 0 ise veri belleğine erişim yetkisi verilir.

3.11.3 Flash Program Belleğinin Okunması

Program belleğinin yeri EEADR ve EEADRH kaydedicilere adresin 2 byte'ı yazılarak okunabilir. [EEPGD (EECON1<7>) ve RD (EECON1<0>) kontrol bitleri ayarlandığında]

Okuma kontrol biti bir kez ayarlanıp mikrodenetleyici veriyi okumak için komut çevremini kullanacaktır.

Örnek 3.4: FLASH Program okuma:

```
bsf     status, rp1   ;
bcf     status, rp0   ; Bank 2
movlw addrh           ;
movwf eeadrh          ; MSByte Program adresini oku
movlw addrl           ;
movwf eeadr           ; LSByte Program adresini oku
bsf     status, rp0   ; Bank 3
bsf     eecon1, eepgd ; EEPROM Program belleğini
bsf     eecon1, rd    ; oku
```

```
nop                    ; Burada herhangi bir komut olmadığı içim
nop                    ; işlem yapılmaz ve döngü için zaman harcanmaz
bcf   status, rp0      ; Bank 2
```

3.11.4 Flash Program Belleğinin Yazılması

WR Kurulum biti ayarlanır ise FLASH program belleği sadece yazılabilir duruma getirilir. FLASH Belleğe yazma yeri, adresin ilk iki byte'ı EEADR ve EEADRH kaydedicilerine yazılarak ve EEPGD/RD kontrol bitleri 1 yapılarak ilk iki byte EEDATA ve EEDATH kaydedicilerine yazılarak tespit edilir. Program hafızası yazmak için örnek 3.5 takip edilebilir.(6)

Örnek 3.5: EEPROM' veri yazma

```
bsf      status, rp1     ;
bcf      status, rp0     ; Bank 2
movlw addrh              ;
movwf eeadrh             ; MSByte Program adresi oku
movlw addrl              ;
movwf eeadr              ; LSByte Program adresi oku
movlw datah              ;
movwf eedath             ; MS Program bellek değerini yaz
movlw datal              ;
movwf eedata             ; LS Program bellek değerini yaz
bsf   status, rp0        ; Bank 3
bsf   eecon1, eepgd      ; PROGRAM bellek noktası
bsf   eecon1, wren       ; Yazım etkin
bcf      ıntcon, gıe     ; Komutlar etkin değil
movlw 55h                ;
movwf eecon2             ; 55h yaz
movlw aah                ;
movwf eecon2             ; AAh yaz
bsf eecon1, wr           ; WR bitini yazmaya başlamak için ayarla
nop                      ;işlem yok
nop                      ;işlem yok
```

```
bsf intcon, gie          ; Komutlar aktif
bcf eecon1, wren         ; Yazma aktif değil
```

3.11.5 Yazım Doğrulanması

EEPROM verisine yazılan değerlerin yazılması istenen değerlerle doğrulanması gerekmektedir. Bu EEPROM biriminin spesifikasyon limitine yakın uygulamalarda kullanılmalıdır. Toplam kaldırma diski , uygunluk (rahatlık) derecesini belirlemeye yardımcı olacaktır. Genel olarak EEPROM yazım başarısızlığı ”1” olarak yazılan, fakat geriye “0” olarak okunan bitten kaynaklanmaktadır.

Örnek 3.6: Yazım doğrulanması:

```
bcf     status, rp0     ; Bank 0
                        ; Herhangi kod buraya gidebilir
movf    eedata, w       ; Bank 0 da olmalıdır.
bsf     status, rp0     ; Bank1
bsf     eecon1, rd      ; Evet yazılan değeri oku
bcf     status, rp0     ; Bank0
                        ; (Wreg) yazılan ve(EEDATA) okunan değerler aynımı?
subwf   eedata, w
```

3.11.6 Taklit Yazılımlara Karşı Koruma

Bazı durumlarda, aygıt EEPROM veri belleğine yazmak istemeyebilir. Taklit yazılımlara karşı korunmak için, değişik mekanizmalar monte edilmiş, kurulmuştur. Yüksek güçte WREN temizlenir. Bunun yanı sıra , yüksek güç timer’ i (72 msn süreli) EEPROM yazımını önler. Yazılımı başlatan ardışık ve WREN biti ikisi birlikte ‘Brown-Out’,güç arızası veya yazılım aksamasında tesadüfi yazılımları önlemeye devam eder.

3.11.7 Kod Koruma Süresince EEPROM Veri İşlemi

Microdenetleyici, kod korumalı durumdayken düzene sokulan verileri okuyabilir ve EEPROM verisine yazabilir. ROM aygıtlarında iki koruma biti mevcuttur. Birisi ROM program belleği diğeri ise EEPROM veri belleği içindir.

3.12 Zamanlama0 (TIMER0) Modülü ve TMR0 Kaydı

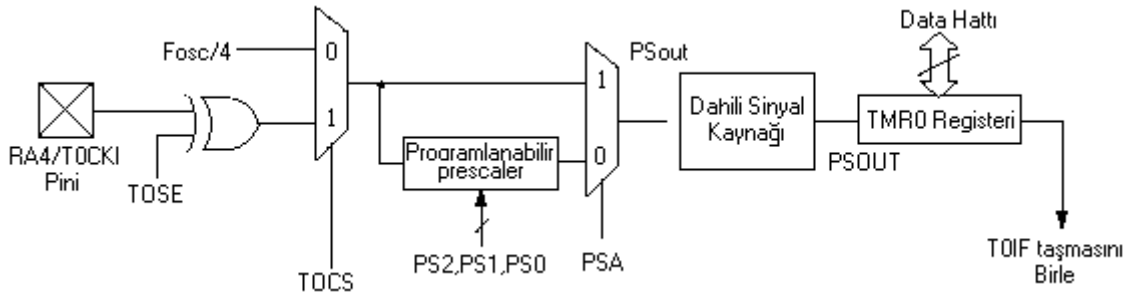
Timer0 modül, timer/sayaç aşağıdaki özelliklere sahiptir.

- 8 bitlik timer/sayaç ,
- Okunabilir ve yazılabilir ,
- 8 bitlik programlanabilir prescaler.,
- İçten veya dıştan saat ayarı ,
- FFh` tan 00h` ye taşma üzeri kesme ,
- Dış saatin sınır seçimi ,

Timer modu TOCS bitinin (OPTION<5>) temizlenmesiyle seçilir. Timer modunda Timer0 modülü her bir komut sürecini uzatır. Eğer TMR0 kaydı yazılıysa, uzama takip eden 2 süreci engeller. Kullanıcı ayarlanan değeri TMR0 kaydına yazarak, bunun etrafından çalışabilir. Sayaç modu TOCS bitinin (OPTION<5>) ayarlanmasıyla seçilir. Bu modda, TMR0, RA4/TOCK1 pininin sınırlarının herbir artışında ya da düşüşünde artacaktır. Genişleyen sınır, TO kaynak sınır seçim biti tarafından, TOSE (OPTION<4>) tarafından belirlenmektedir. TOSE bitinin temizlenmesi artan sınırları seçecektir.

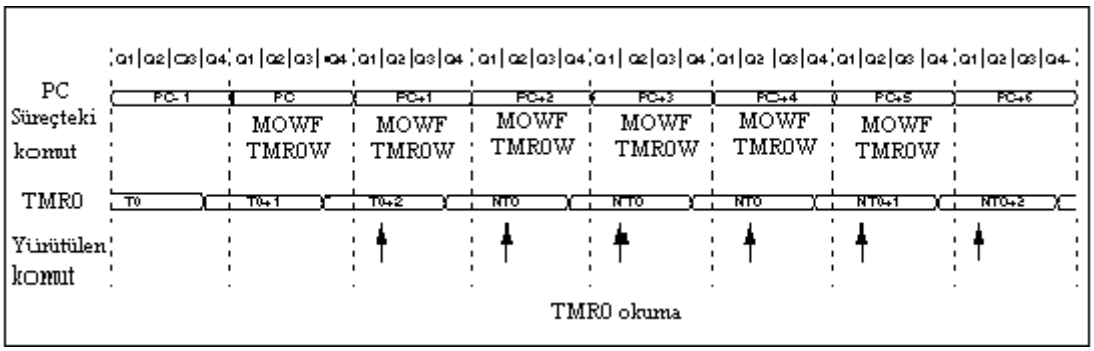
Prescaler, Timer0 modülü ile Watchdog Timer arasında paylaşılmaktadır. Prescaler ataması, yazılımda PSA biti kontrolü tarafından denetlenmektedir. (OPTION<3>) PSA bitinin temizlenmesi, prescaler' ı Timer0 modülüne atayacaktır. Prescaler okunabilir veya yazılabilir değildir. Prescaler, Timer0 zamanlama modülüne atandığında prescaler değeri (1:2,1:4,1:8,1:16,1:32,1:64,1:128,1:256 olmak üzere) yazılım tarafından seçilebilirdir. (1)

3.12.1 TMR0 Kesmesi



Şekil 3.18:TMR0 blok diyagramı

TMR0 kesmesi, TMR0 kaydı FFH`dan 00h`ye akışında üretilmektedir. Bu fazla akım TOIF bitini (INTCON<2>) kurar (ayarlar). Kesme, aktif TOIE bitinin (INTCON<5>) temizlenmesi ile gizlenebilir. (INTCON<5>) TOIF biti, Timer0 modülü tarafından, bu kesmenin yeniden aktifleştirilmesinden önce yazılımdan silinmelidir. TMR0 kesmesi (Şekil 3.18) denetleyiciyi SLEEP` ten çıkaramaz, çünkü, SLEEP boyunca timer kapalıdır.



Şekil 3.19: TMR0 Zamanlaması: Prescaler' siz İçten Clock

3.12.2 TMR0'ın Dıştan Saat İle Kullanımı

Dıştan saat girişleri TMR0 için kullanıldığında, bazı ön şartların gerçekleştirilmesi gerekir. Dıştan saat gereksinimi , içten faz saati senkronizasyonundan kaynaklanmaktadır. Bunun yanısıra, TMR0 kaydının senkronize edilmesinden sonra, fiili artmada gecikme mevcuttur.

3.12.2.1 Dıştan saat senkronizasyonu

Hiç bir prescaler kullanılmadığı taktirde , dıştan saat girişi prescaler çıkışındaki gibidir. RA4/TOCKI pininin içten faz saati ile senkronize edilmesi iç faz saatlerinin Q2 ve Q4 süreçlerindeki prescaler çıkışını örneklemek yoluyla yerine getirilir.

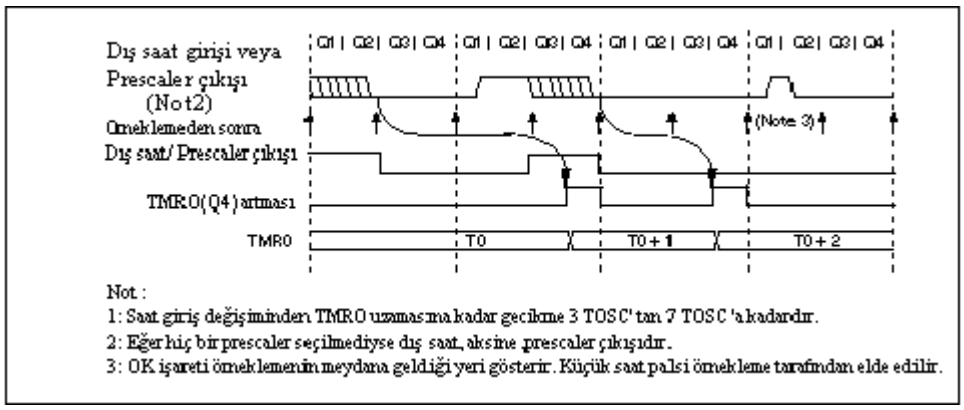
Bunun için , TOCKI' nin düşük değerinin en azından 2TOSC (artı ufak RC gecikmesi) olması gerekir. Prescaler kullanıldığında, dış saat girişi asenkron sayıcı tipi prescaler' a bölünür ve böylece prescaler çıkışı simetrik olur. Dış saatin örnekleme gereksinmelerini karşılamak için sayaç (counter) dikkate alınmalıdır. Böylece prescaler değerine bölünen en azından 4 TOSC peryot uzunluğuna sahip olmalıdır. (1)

3.12.2.2 TMR0 gecikme uzatılması

Prescaler çıkışı , iç saat ile senkronize edildiği için, dış saat sınırlarının meydana gelmesindeki zamandan TMR0 modülünün fiili olarak uzatılması zamanına kadar küçük bir gecikme vardır. Şekil 3.20 dış saat sınırından Timer uzamasına kadar gecikmeyi göstermektedir.

3.12.3 Prescaler (Bölücü)

8 Bitlik sayaç Timer0 modülünde veya Watchdog timer'ında bulunur. Prescaler dışarıdan verilen sinyali 256 ya kadar bölmeye yarar. Timer0 modülü ile watchdog timer'ı arasında karşılıklı istisna tutulan yalnızca bir tek prescaler mevcuttur. Böylece Timer0 modülüne prescaler ataması, watchdog timer'ın prescaleri olmadığı anlamına gelmektedir. PSA ve PS2 : PSO bitleri (option <3:0>) prescaler atamasını ve prescaler oranını belirlemektedir.



Şekil 3.20: Dış saatli TIMER0 zamanlayıcısı

Timer0 modülüne yazılan bütün komutlar, timer0 modülüne atandığında prescaler'i ölçecektir. WDT ye atandığında , CLRWDT komutu watchdog timer boyunca prescaler' i temizleyecektir. Prescaler yazılabilir veya okunabilir değildir.

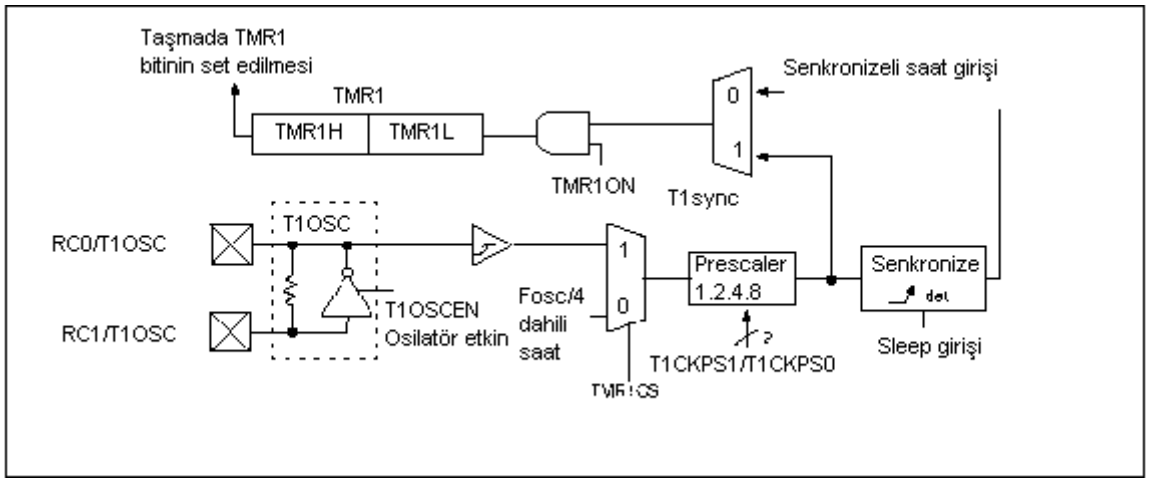
3.13 Zamanlayıcı 1 (TIMER1) Modülü

Timer1 modül: Timer/sayaç aşağıdaki özelliklere sahiptir.

- 16 bit timer/sayaç,
- Okunabilir ve yazılabilir,
- İçten ve dıştan saat seçimi,

- FFFFh'den 000h'a taşma üzeri kesme,
- CCP modülünden resetleme,

Şekil 3.21'de Timer1'in aktif/pasif, ayarlama/sıfırlama işlemleri gösterilmektedir.



Şekil 3.21: TIMER1 Blok diyagramı

3.13.1 Zamanlayıcı1 (TIMER1) İşlemleri

Timer1 aşağıdaki modlardan birini işletebilir.

- Zamanlayıcı olarak
- Senkronize sayacı olarak
- Asenkronize sayacı olarak

TMR1CS (T1CON<1>) biti bu modlardan hangisinin çalıştırılacağına karar verir. Zamanlayıcı modunda Timer1 her komut çevrimi artışları, sayaç modunda her harici saat girişi yükselmesini belirler. Timer1 osilatör aktifken RC1/T10SI ve RC0/T10S0/T1CKI pinleri giriş olurlar. TRISC<1:0> değeri ihmal edilir.

3.13.2 T1CON: TIMER1 Kontrol Kaydedicisi

TIMER1'deki işlemlerin kontrol edildiği ve işlemler yürütülürken bazı bilgilerin alındığı kaydedicidir.

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.22: T1CON Kaydedicinin bit yerleşimi

T1con,0(TMR1ON): Timer1' e yetki veren bittir. TMR1ON biti 1 yapıldığında Timer1 kullanılır durumdadır,0 ise kullanım dışıdır.

T1con,1(TMR1CS): Timer 1 için clock kaynağının seçildiği bittir. TMR1CS 1 ise clock kaynağı olarak RC0/T1OSO/T1CKI pininden gelen sinyalin yükselen kenarları seçilir. TMR1CS'nin 0 olması halinde ise dahili clock kaynağı kullanılır.

T1con,2(T1SYNC): Harici clock kaynağının senkron kontrolünü yapan bittir. Eğer harici kaynak ile Timer1 eşzamanlı çalışmıyorsa T1SYNC 1 seviyesinde olur. Eğer eşzamanlı bir çalışma var ise T1SYNC 0 seviyesindedir.

T1con,3(T1OSCEN): Timer1 osilatör kaynağı yetkilendirme bitidir. T1OSCEN 1 ise osilatör kaynağı etkindir, 0 ise kullanım dışıdır.

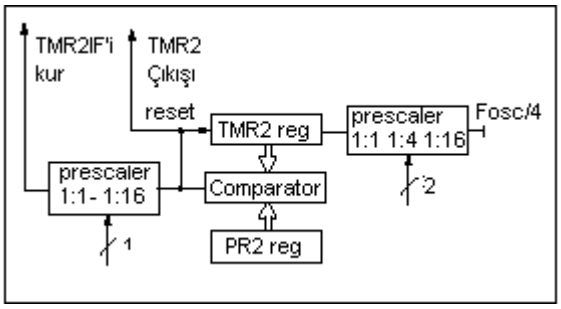
T1con,4-5(T1CKPS0,T1CKPS1): Timer1 giriş sinyalinin bölme (prescale) oranının seçildiği bilerdir. Bu bitlerin aldığı değerlere göre bölme prescaler değeri değişmekte ve buna bağlı olarak Timer1 artma frekansı değişmektedir.

11 = 1:8 Prescale değeri
10 = 1:4 Prescale değeri
01 = 1:2 Prescale değeri
00 = 1:1 Prescale değeri

3.14 Zamanlayıcı2 (TIMER2) Modülü

Timer2 modül aşağıdaki özelliklere sahiptir.

- 8 bit zamanlayıcı (TMR2 kaydedici),
- 8 bit peryot kaydedici (PR2),
- Okunabilir ve yazılabilir,
- Yazılım ile programlanabilir prescaler,
- TMR2, PR2 eşlemesinde kesme,
- Saat kaymasını üretmek için TMR2 çıkışının seçimli kullanımı SSP modülü,



Şekil 3.23: TIMER2 Blok diyagramı

Timer2 şekil 3.23'te gösterilen denetim yazmacına sahiptir. Timer2 güç tüketimini en aza indirgemek için TMR0 denetim bitini temizleyerek kapanabilir.

3.14.1 Zamanlayıcı2 (TIMER2) İşlemleri

Timer2, CCP modülünün PWM modu için PWM zaman tabanı olarak kullanılabilir. Giriş saati (Fosc/4) 'in prescale seçeneğine sahip olur. Aşağıdaki durumlardan herhangi biri gerçekleştiğinde prescaler ve postscaler sayaçları temizlenirler.

- TMR2 kaydedicisine yazıldığında ,
- T2CON kaydedicisine yazıldığında,
- Herhangi bir aygıt hazır olduğunda ,

***T2CON yazıldığında, TMR2 temizlenemez.

3.14.2 T2CON: TIMER2 Kontrol Kaydedicisi

Timer2'deki işlemlerin kontrol edildiği ve işlemler yürütülürken bazı bilgilerin alındığı kaydedicidir.

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.24: T2CON Kaydedicinin bit yerleşimi

T2con,0-1(T2CKPS0,T2CKPS1): Timer2'de prescaler'in ayarlandığı bitlerdir. Bu bitlerin aldığı değerlere göre prescaler değeri değişmektedir.

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

T2con,2(TMR2ON): Timer2' e yetki veren bittir. TMR2ON biti 1 yapıldığında Timer2 kullanılır durumdadır,0 ise kullanım dışıdır.

T2con,3:6(TOUTPS3:TOUTPS0): Timer2'de postscaler ayarlarının yapıldığı bitlerdir. Bu bitlerin aldığı değerlerle timer2 değişik postscaler değerlerine ulaşmaktadır.

0000 = 1:1 Postscaler

0001 = 1:2 Postscaler

•

•

1111 = 1:16 Postscaler

3.15 CAPTURE/COMPARE/PWM(CCP) Modülleri

CCP modülleri yakalama, karşılaştırma, pals genişliği modülasyonu gibi işlemleri gerçekleştirmek üzere bazı özelliklerle donatılmışlardır. Her bir CCP modülü 16 Bitlik yakalama (capture) kaydedicisi, 16 bitlik karşılaştırma (compare) kaydedicisi veya PWM kaydedicisine sahiptir. Tablo 3.5 de CCP modülünün her elemanının ilişkide bulunduğu zamanlama kaynakları verilmiştir.

Tablo 3.5:CCP Timer İlişkisi

CCP Modu	Timer Kaynağı
Capture	Timer1
Compare	Timer1
PWM	Timer2

PIC 16F877'nin CCP modülleri olan CCP1 ve CCP2'nin çalışmaları özel tetikleyicileri dışında hemen hemen aynıdır. Bu nedenle bundan sonraki anlatımlarımız CCP1 modülü üzerinde yoğunlaştırılacaktır.

3.15.1 CCP1 Modülü

CCP modülü, düşük kısmı 8 bit olan CCPR1L ,yüksek kısmı 8 bit olan CCPR1H olan ve CCPR1 olarak adlandırılan bir kaydediciden oluşmaktadır. CCP1 modülünün işlemleri ise CCP1CON kaydedicisi tarafından kontrol edilmektedir. Bu modüldeki kaydedicilerin tümü yazılabilir ve okunabilir özelliktedirler.

3.15.2 CCP1CON Kaydedicisi

Yukarıda da belirtildiği gibi bu kaydedici CCP1 modülünün yaptığı işlemleri denetler ve yetkilendirmeleri yapar.

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.25: CCP1CON Kaydedicinin bit yerleşimi

Ccp1con,0:3(CCPxM0:CCPxM3): CCP için mod seçim bitleridir. Bu bitlerin aldığı değerlere göre CCP1 modülü capture/compare/PWM modlarına sokulmaktadır.

0000 = Capture/Compare/PWM etkin değil (CPM kullanım dışı)
0100 = Capture modu, her düşen kenarda
0101 = Capture modu, her yükselen kenarda
0110 = Capture modu,her 4. yükselen kenarda
0111 = Capture modu, every 16. yükselen kenarda
1000 = Compare modu, denklikte durumunda çıkışı 1'le(H) (CCPxIF=1)
1001 = Compare modu, denklik durumunda çıkışı 0'la (L) (CCPxIF=1)
1010 = Compare modu, denklikdurumnda yazılım kesmesi üret
1011 = Compare modu, özel tetikleme (CCPxIF=1, CCP1 TMR1'i resetler, CCP2 TMR1'i resetler ve A/D çevirme başlar.(Eğer A/D çevirme modülü yetkilendirilmişse).)
11xx = PWM modu

Ccp1con,4:5(CCPxX:CCPxY): CCP modülün PWM modundaki Lsb ayarlarını gerçekleştiren bitlerdir.

3.15.3 CAPTURE (Yakalama) Modu

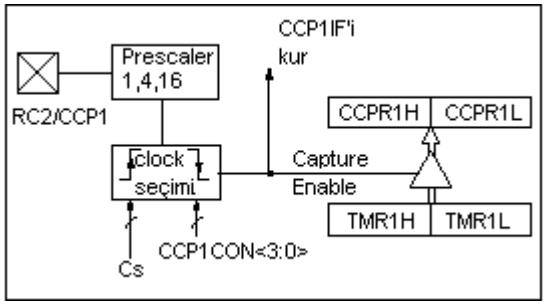
Capture modunda RC2/CCP1'deki bir takım olaylarla; CCPR1H:CCPR1L, TMR1'in 16 bitlik değerini yakalar ve kaydeder. RC2/CCP1'deki olaylar şu şekilde tanımlanmaktadır:

- Her düşen kenar
- Her yükselen kenar
- Her 4. yükselen kenar
- Her 16. yükselen kenar

RC2/CCP1'deki bu durumlardan birisi CCPxM0:CCPxM3 bitleri sayesinde seçilir. Bir yakalama işlemi gerçekleştiğinde CCP1IF (PIR1,2) kurulur ve bir kesme talebinde bulunulur.Eğer yeni bir yakalama gerçekleşirse CCP1R registerine yeni değer kaydedilir ve eski değer tamamiyle silinir.

3.15.3.1 Capture Modunda Pin Durumu

Capture modunda RC2/CCP1 pini Trisc<2> vasıtasıyla giriş olarak ayarlanmalıdır. Bu sayede RC2/CCP1 pini capture işlemi için gerekli clockları alabilecektir. Şekil 3.26'da capture modunda veri yakalama işleminin nasıl gerçekleştiğini gösteren diyagram bulunmaktadır.



Şekil 3.26: Capture Modunda Veri Yakalama İşleminin Gerçekleşmesi

3.15.3.2 CCP Prescaler

CCP'de prescaler ayarlarını gerçekleştirmek üzere, CCP1M3:CCP1M0 bitleri ayrılmıştır. CCP modülü kullanım dışı ise veya CCP modülü capture modunda kullanılmıyorsa prescaler sayıcısı temizlenmiş vaziyette durur. Bu herhangi bir resetleme

durumunda prescalerin temizleneceği anlamına gelir. Capture prescaler'i diğer bir kesmenin üretilmesi durumunda değişim gösterir. Örnek 3.7'de CCP prescalerin kurulmasına dair bir örnek verilmiştir. Bu örnek programda prescaler sayıcı temizlenmiştir ve hata kesmelerinin üretilmesi engellenmiştir.

Örnek 3.7

```
clrf    ccp1con        ;CCP modülünü kapalı duruma getir.
movlw new_capt_ps   ;CCP'yi kur, prescaleri  ile yükle
movwf  ccp1con        ; CCP1CON kaydedicisini yükle
```

3.15.4 COMPARE (Karşılaştırma) Modu

Karşılaştırma modunda 16 bitlik CCPR1 ile TMR1'in karşılıklı bitleri baytları karşılaştırılır. Eğer bir eşitlik söz konusu ise RC2/CCP1'de ;

- Yüksek seviyeye (H) çekilme
- Düşük seviyeye (L) çekilme
- Değişiklik olmama.

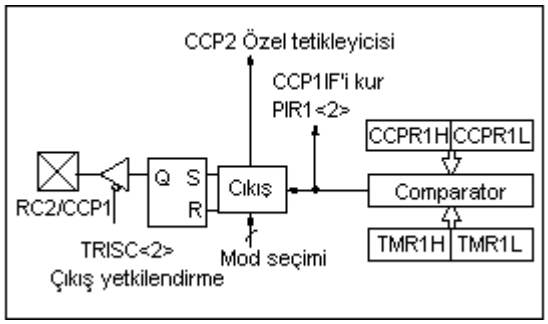
durumlarından birisi gerçekleşir. Gerçekleşmesi istenilen durum program yardımıyla belirlenir.

RC2/CCP1 pinindeki değişme durumu CCP1M0:CCP1M0 (CCP1CON<0:3>) bitleri yardımı ile yapılmaktadır.

1000 = denklik durumunda çıkışı 1'le(H)
1001 = denklik durumunda çıkışı 0'la (L)
1010 = denklik durumnda yazılım kesmesi üret (pinde değişiklik olmaz)

Denklik durumunda daima CCP1IF biti yüksek (H) durumuna çekilmektedir.

Compare modunda görev alan kaydedici, pin ve diğer çevre elemanlarının ilişkileri şekil 3.27'de verilmiştir.



Şekil 3.27: Compare Modu İşlemlerinin Blok Diyagramı

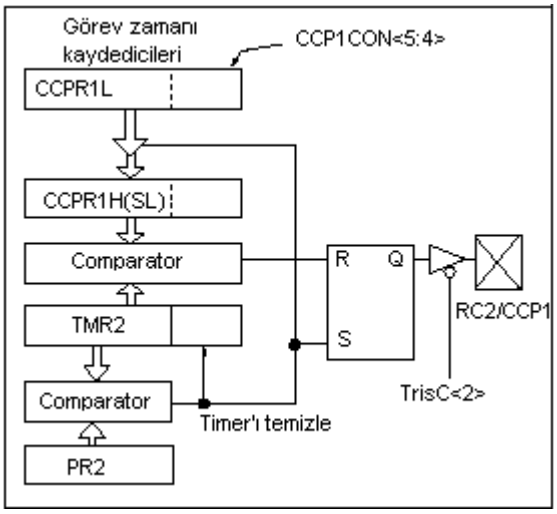
3.15.4.1 Özel Hal Tetikleyicisi

Bu modda dahili donanım tetikleme kaynağı başka bir yerde herhangi bir işlemi başlatmak için tetikleme üretir. CCP1'in özel hal tetikleyici çıkışı TMR1 kaydedici çiftini resetler. Bu durum Timer1'de yeni bir 16 bitlik programlanma devresini başlatır.

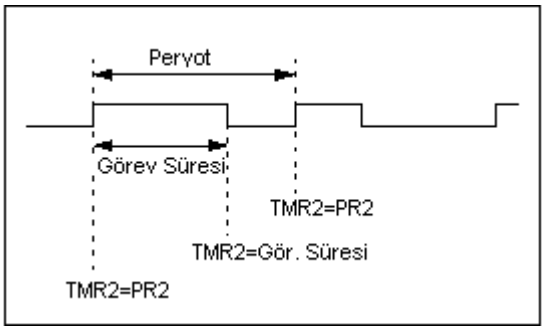
CCP2'nin özel hal tetikleyicisi ise TMR1'i resetler ve eğer A/D modülü kullanılır vaziyette ise yeni bir A/D çevirme işlemi başlar.

3.15.5 PWM (Pals Genişlik Modülasyonu) Modu

Bu modda CCP1 pininden 10 bit çözünürlükte PWM çıkış alınır. PWM modunda işlem yapabilmek için TRISC<2> biti çıkış konumunda tutulmalıdır. Şekil 3.28'de PWM modunda gerçekleştirilen işlemlerin basit blok diyagramı verilmektedir. Şekil 3.29 da ise PIC 16F877'den elde edilecek bir PWM çıkışın şekli verilmiştir.



Şekil 3.28: PWM Moduna Ait Basitleştirlmiş Blok Diyagram



Şekil 3.29: PWM Çıkış Şekli

3.15.5.1 PWM Süresi

Bir PWM işlemi ile verinin aktarılması PR2 kaydedicisinin değerine bağlıdır. (PR2=TMR2 prescaler değeri). PWM süresi formül 3.1 den faydalanılarak hesaplanabilir.

PWM süresi = [(PR2) + 1] . 4 . TOSC

Formül 3.1

PWM frekansı ise 1/PWM süresinden bulunabilir. TMR2'nin PR2'ye eşit olması durumunda ise, sonraki peryotta şu üç durum meydana gelir.

- TMR2 temizlenir.
- CCP1 pini yüksek seviyeye (H) çekilir.
- PWM görev saykılı CCPR1L den CCPR2H içerisine atılır.

3.15.5.2 PWM Görev Saykılı

PWM görev saykılı CCPR1L ve kaydedicisi ve CCP1CON<4:5> bitlerine bağlıdır. PWM'de ki 10 bitlik çözünürlük; CCPR1L kaydedicisinin 8 bitlik kısmının MSB, ve CCP1CON'un 4 ve 5. Bitlerinin LSB olarak kabul edilmesiyle elde edilir. Bu 10 bit çözünürlük CCPR1L:CCP1CON<5:4> şeklinde gösterilmektedir. PWM görev saykılının hesaplanmasında ise formül 3.2 kullanılabilir.

(CCPR1L:CCP1CON<5:4>) •Tosc • (TMR2 prescaler değeri)

Formül 3.2

3.15.5.3 PIC 16F877'nin PWM İşlemi İçin Kurulması

- PR2 kaydedicisi yazılarak PWM süreci başlatılır,
- CCPR1L kaydedicisine ve CCP1CON<5:4> bitlerine PWM görev saykıl'ı yazılır,
- TRISC<2> biti temizlenerek CCP1 pini çıkış olarak ayarlanır,
- TMR2 prescaler değeri ayarlanır ve T2CONOn biti vasıtasıyla Timer2 yetkilendirilir,
- CCP1 modülü PWM işlemi için düzenlenir.

3.16 MSSP (Asıl Eşzamanlı Seri Port) Modülü

MSSP The Master Synchronous Serial Port) modülü başka çevre birimlerle veya mikrodenetleyicilerle seri olarak haberleşmeyi sağlayan bir arabirimdir. Burada bahsedilen çevre birimleri; seri EEPROM, shift register, display sürücüleri, A/D çevirici gibi birimler olabilir. MSSP modülünün iki tip çalışma modu bulunmaktadır.

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

MSSP'nin çalışma modlarına geçmeden önce bu modülün kontrolünü sağlayan kaydedicilerden bahsetmekte fayda vardır. MSSP modülünde işlemleri kontrol eden başlıca üç kaydedici; SSPSTAT,SSPCON,SSPCON2 kaydedicileridir.

3.16.1 SSPCON Kaydedicisi

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.30: SSPCON Kaydedicisinde Bit Yerleşimi

Sspcon,0:3(SSPM0:SSPM3): SSP (Synchronous Serial Port) için mod seçim bitleridir. Bu bitlerin aldığı değerlere göre MSSP modülü daha önce de bahsettiğimiz modlardan birisine sokulur ve özel bir işlem için tahsis işlemi yapılır.

0000 = SPI master modu, clock = FOSC/4
0001 = SPI master modu, clock = FOSC/16
0010 = SPI master modu, clock = FOSC/64
0011 = SPI master modu, clock = TMR2 çıkışı/2

0100 = SPI slave modu, clock = SCK pin. SS pin kontrolü yetkilendirmesi.
0101 = SPI slave modu, clock = SCK pin. SS pini kullanım dışı.
0110 = I^2C slave modu, 7-bit addres
0111 = I^2C slave modu, 10-bit addres
1000 = I^2C master modu, clock = FOSC / (4 * (SSPADD+1))
1011 = I^2C sabit kontrollü master mod
1110 = I^2C sabit kontrollü master mod, başlat ve durdur kesmeleri ile birlikte 7 bit adres yetkilendirmesi
1111 = I^2C sabit kontrollü master mod, başlat ve durdur kesmeleri ile birlikte 7 bit adres yetkilendirmesi
1001, 1010, 1100, 1101 = Başka işlemler için ayrılmıştır.

Sspcon,4(CKP): Clock'un polaritesini seçmek için kullanılan bittir.

SPI modunda; CKP 1 ise, yüksek seviye etkizis kabul edilir, CKP 0 ise düşük seviye etkisiz kabul edilir.

I^2C slave modunda; CKP 1 ise clock yetkilendirmesi yapılır, CKP 0 ise clock düşük seviyede tutulur.

Sspcon,5(SSPEN): MSSP modülü için yetkilendirme bitidir.

SPI modında;SSPEN =1 ise seri port kullanımdadır ve SCK ,SDO, SDI ve SS uçları kullanılabilir. SSPEN =0 ise seri port işlemleri kullanım dışı bırakılır.

I^2C modunda; SSPEN=1 ise seri port işlemleri etkin yapılır ve SDA ve SCL pinleri kullanılabilir duruma getirilir. SSPEN= 0 ise seri port işlemleri kullanım dışıdır.

Sspcon,6(SSPOV): MSSP modülünde alma işleminde taşmaları izleyen bittir.

SPI modunda; SSPOV 1 ise, SSPBUF önceki veriyi tutarken yeni bir bayt alınır. SSPSR verisi üzerindeki taşma kaybolur. Slave modunda kullanıcı SSPBUF kaydedicisini okumak zorundadır, sadece data iletimi sırasında bu bit önemsizdir. Master modda taşma biti SSBBUF kaydedicisine yeni bir veri yazılana kadar kurulmaz. Yeni bir işlem gerçekleştirebilmek için SSPOV biti programdan temizlenmelidir. SSPOV biti 0 ise herhangi bir taşma yoktur.

I^2C modunda; SSPOV 1 ise, SSPBUF önceki veriyi tutarken bir baytlık veri alınır. SSPOV biti iletim modunda önemsizdir. Alım modunda yeni bir işlem için SSPOV bitinin programdan temizlenmesi gerekir.

Sspcon,7(WCOL): Yazım çakışması bitidir. Bir nevi hata kontrolü yapar.

SPI modunda;WCOL 1 ise I^2C hattı meşgul iken SSPBUF kaydedicisine veri yazılmaya kalkışılmıştır. Bu durum bir çakışmadır. WCOL 0 ise herhangi bir çakışma yoktur.

I^2C modunda; Hala önceki data iletilirken SSPBUF kaydedicisine veri yazılmıştır. Bu bir çakışmadır ve WCOL bitinin programdan silinmesi gerekir. WCOL 0 ise herhangi bir çakışma yoktur.

3.16.2 SSPSTAT Kaydedicisi

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.31:SSPSTAT Kaydedicisi Bit Yerleşimi

Sspstat,0(BF): Tamponun doluluk durumu hakkında bilgi tutan bittir.

Alım durumunda (SPI ve I^2C modunda);BF= 1 ise alım işlemi tamamlanmıştır ve SSPBUF kaydedicisi doludur. BF=0 ise alım işlemi tamamlanmamıştır ve SSPBUF boştur.

İletim durumunda (I^2C modunda); BF =1 ise veri iletimi sürmektedir ve SSPBUF kaydedicisi doludur. BF=0 ise veri iletim işlemi tamamlanmıştır ve SSPBUF kaydedicisi boştur.

Sspstat,1(UA): 10 bitlik I^2C modunda adres güncelleme bitidir. UA=1 ise SSPADD kaydedicisinde gösterilen adres güncellenir. UA=0 ise güncelleme gerçekleşmez.

Sspstat,2(R/W): SPI veya I^2C modunda işlem yapılırken okuma mı yoksa yazma mı yapıldığının bilgisini tutan bittir.

SPI modunda; R/W =1 ise okuma, R/W=0 ise yazma yapılıyordur.

I^2C modunda;R/W=1 ise veri iletimi yapılmakta, R/W=0 ise veri iletim işi yapılmamaktadır.

Sspstat,3(S): I^2C modunda başlangıç bitidir. S biti 1 yapıldığında start biti son bulur,S biti 0 yapıldığında ise start biti son bulmaz.

Sspstat,4(P): I^2C modunda sonlandırma bitidir. S biti 1 yapıldığında durdurma biti son bulur,S biti 0 yapıldığında ise durdurma biti son bulmaz.

Sspstat,5(D/A): I^2C modunda veri ve adres arasındaki durumları gösteren bittir. D/A=1 ise data alınmıştır yada iletilmiştir. D/A=0 ise adres yollanmıştır veya bir adres alınmıştır.

Sspstat,6(CKE):SPI clock kaynağının türünü belirlemekte kullanılan bittir.

SPI modunda; CKP=0,CKE=1 ise SCK'nın yükselen kenarında veri iletilir

SPI modunda; CKP=0,CKE=0 ise SCK'nın düşen kenarında veri iletilir

SPI modunda; CKP=1,CKE=1 ise SCK'nın düşen kenarında veri iletilir

SPI modunda; CKP=1,CKE=0 ise SCK'nın yükselen kenarında veri iletilir

I^2C modunda; CKE=1 ise giriş seviyesi olarak SMBUS izlenir, CKE=0 ise I^2C izlenir.

Sspstat,7(SMP): Örnekleme bitidir.

SPI master modda;SMP=1 ise giriş sinyali data çıkışının sonunda örneklenir, SMP=0 ise giriş sinyali çıkış sinyalinin orta noktasında örneklenir.

SPI slave modda; SMP biti 0 (L) seviyesinde tutulmalıdır.

I^2C modunda; SMP=1 ise standart hız için (100KHz ve 1 MHZ) slew rate kontrolü kullanım dışıdır. SMP=0 ise yüksek hız modunda kullanım için (400KHz) slew rate yetkilendirilmiştir.

3.16.3 SSPCON2 Kaydedicisi

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.32:SSPCON2 Kaydedicisi Bit Yerleşimi

Sspcon2,0(SEN): I^2C modunda start durum ayarının yapuldığı bittir. SEN=1 ise SDA ve SCL pinlerinde start vaziyeti alınır. SEN=0 ise start durumu yetkisizlendirilir.

Sspcon2,1(RSEN): I^2C modunda start durum yineleme bitidir. RSEN=1 ise SDA ve SCL pinlerinde alınan start vaziyeti tekrarlanır. RSEN=0 ise start durum yineleme kullanım dışı olur.

Sspcon2,2(PEN): I^2C modunda stop (durdurma) durum yetkilendirme bitidir. PEN=1 ise SDA ce SCL pinlerinde stop vaziyeti başlatılır. PEN=0 ise stop vaziyeti kullanım dışıdır.

Sspcon2,3(RCEN): I^2C modunda alım yetkilendirme bitidir. RCEN=1 ise I^2C modunda veri alım vaziyetine geçilir. RCEN=0 ise alım modu kullanım dışıdır.

Sspcon2,4(ACKEN): I^2C modunda eşzamanlı ard arda veri alımı yetkilendirme bitidir. ACKEN=1 ise SDA ve SCL pinlerinden eşzamanlı ard arda veri iletimi başlatılır ve ACKDT veri biti iletilir. Bu bit donanım tarafında otomatik olarak temizlenir.

Sspcon2,5(ACKDT): I^2C modunda master alım durumunda veri kabul bitidir. ACKDT=1 ise iletilen veri kabul edilmemiştir (alınmamıştır.) .ACKDT=0 ise iletilen veri kabul edilmiştir (alınmıştır).

Sspcon2,6(ACKSTAT): I^2C modunda iletim durumunda statü kabul bitidir. ACKSTAT=1 ise alınan veri kabul edilmemiştir. ACKSTAT=0 ise alınan veri kabul edilmiştir.

Sspcon2,7(GCEN): I^2C modunda genel çağırma yetkilendirme bitidir. GCEN=1 ise SSPSR'den gelen çağırma adresi (0000h) alındığında kesme etkinleştirilir. GCEN=0 ise genel çağırma adresi yetkisizleştirilir.

3.16.4 SPI (Seri Çevresel Arabirim) Modu

SPI modu 8 bit uzunluğundaki verinin aynı anda eşzamanlı olarak alınmasına ve iletilmesine olanak tanır. SPI modunda haberleşme temel olarak 3 pin ile sağlanmaktadır.

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

Bu üç pine ek olarak slave modda işlem yapmak üzere bir dördüncü pin olarak SLAVE SELECT (SS) mevcuttur.

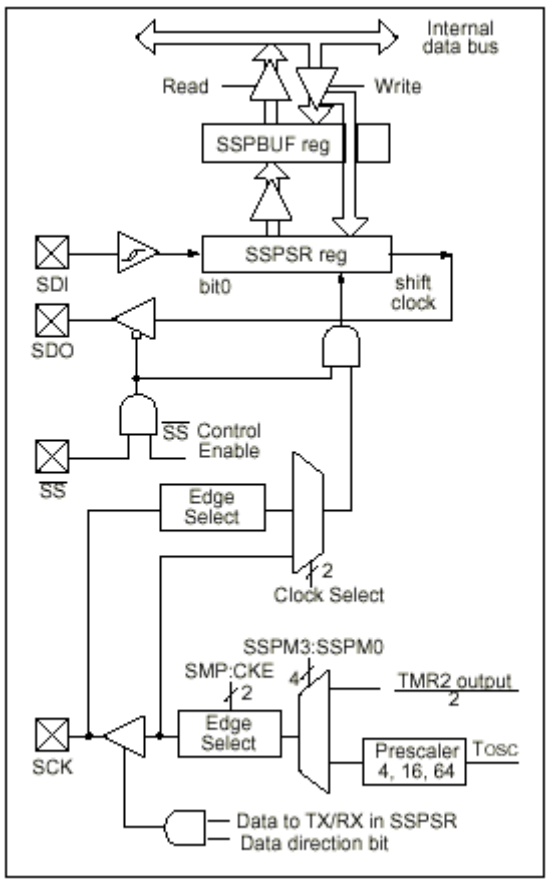
3.16.4.1 SPI İşlemleri

SPI modunda işlem yapabilmek için birkaç ayarın yapılması gerekmektedir. Bu ayarlar SSPCON<5:0> ve SSPSTAT<7:6> bitleri üzerindeki değişikliklerle yapılabilir. Bu ayarlar sayesinde SPI modunda aşağıdaki değişiklikler gerçekleştirilebilir.

- Master mod seçimi (SCK clock çıkışı)
- Slave mod seçimi (SCK clock girişi)
- Giriş verisi örnekleme evresi (sonda veya ortada)
- Clock türü (SCK nın düşenkenarı veya yükselen kenarında veri aktarma)

Şekil 3.33'te MSSP modülünde SPI modunda işlemlerin nasıl gerçekleştirildiğini gösteren blok diyagram bulunmaktadır.

MSSP modülünde yapılan işlemler bir iletme/alma kaydırmalı kaydedici (SSPSR) ve bir tampon kaydedicisine bağlıdır. SSPSR verileri önce MSB biti olmak üzere içerisinde kaydırır. Data alma işlemi tamamlanana kadar SSPBUF'ta tutulan verileri SSPSR'ye yazar. İlk önce SSBUF'tan yollanan verinin 8 bitlik kısmı alınır. Sonra BF biti (SSPSTAT<0>) kontrol edilir ve kesme bayrağı SSPIF (PIR<3>) kurulur. SSPBUF registerine veri yazma durumunda iletim veya data alma işlemi olmasına bakılmadan WCOL (SSPCON<7>) biti kurulur. Yeni bir işlemin yapılabilmesi için program içerisinde bu bitin temizlenmesi gereklidir. Örnek 3.8, SSPBUF kaydedicisinin (SSPSR) yüklenmesini göstermektedir.



Şekil 3.33: SPI Modunda İşlemleri Gösteren Blok Diyagram

ÖRNEK 3.8

```
        bsf     status, rp0         ;Bank1 seçildi
devam   btfss   sspstat, bf         ;Veri iletimi/alımı tamam mı?
        goto    devam               ;tamalanmadı
        bcf     status, rp0         ;Bank0 seçildi
        movf    sspbuf, w
        movwf   rxdata
        movf    txdata, w
        movwf   sspbuf              ;yeni veri sspbuf'ta
```

3.16.4.2 SPI G/Ç Yetkilendirmesi

Seri port yetkilendirmesi için öncelikle MSSP yetkilendirme biti SSPEN (SSPCON<5>) kurulmalıdır. Herhangi bir resetleme veya yeni biçimlendirme durumunda SSPEN temizlenmeli SSPCON kaydedicisi yeniden düzenlenmeli ve SSPEN tekrar

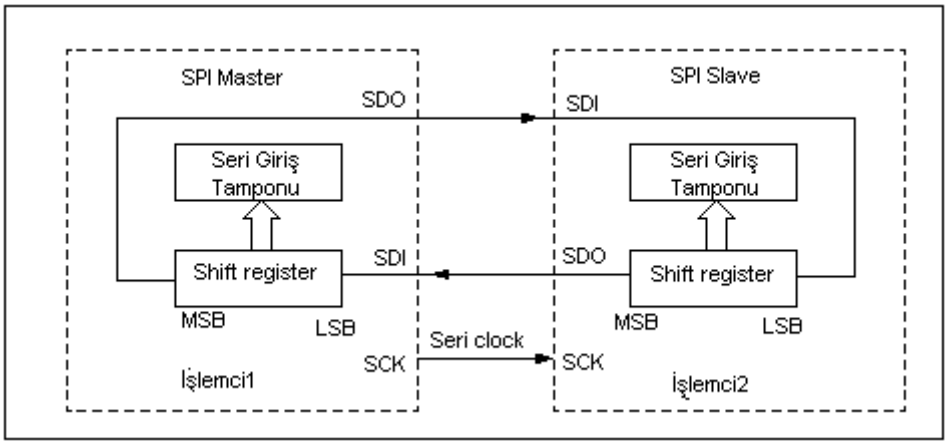
kurulmalıdır. SDI, SDO, SCK ve SS' pinlei seri port pinleridir. Bu pinlerin birer seri port elemanı olarak çalışabilmeleri için programdan bazı TRIS kaydedicilerinin ilgili bitlerinin ayarlanması gerekmektedir.

- SDI,SPI modulü tarafından otomatik olarak kontrol edilir.
- SDO için TRISC<5> temizlenmelidir.
- Master modunda SCK için TRISC<3> temizlenmelidir.
- Slave modda SCK için TRISC<3< kurulmalıdır.
- SS' için TRISA<5> kurulmalıdır.

3.16.4.3 Tipik Bağlantılar

Şekil 3.34 iki denetleyicinin seri olarak nasıl haberleşeceğini göstermektedir. Burada işlemcilerden birisi master modda diğeri slave modda çalışmaktadır. 1. denetleyici master modda çalışmaktadır ve görevi 2. denetleyiciye ihtiyaç duyduğu veriyi ve clock darbelerini göndermektir. Slave modda çalışan 2. İşlemci ise 1. denetleyiciden gönderilen veriyi clock darbeleri ile alır ve işler. Bu bağlantının en önemli özelliği tek clock kaynağı ile her iki denetleyicinin eşzamanlı seri veri haberleşmesi yapmalarıdır. Bu bağlantıda;

- Master veri gönderir, slave bu veriyi kopyalar ve gönderir
- Master veri gönderir, slave veri gönderir
- Master kopyalanan veriyi gönderir, slave veri gönderir.

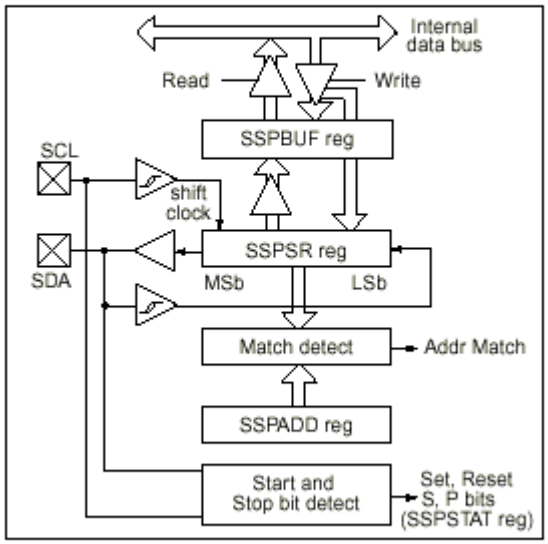


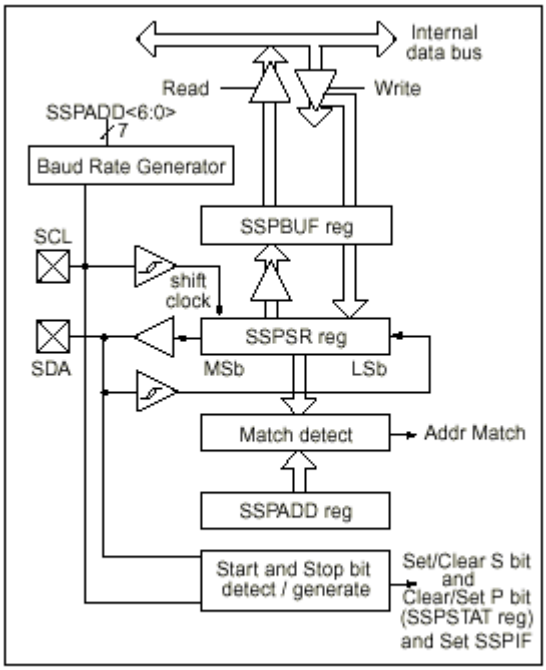
Şekil 3.34: SPI Modda Master/Slave Bağlantısı

3.16.5 MSSP'de I²C Modu

MSSP modülünün I²C modunda master ve slave olmak iki durumda çalışma vardır. MSSP modülü I²C modülü için 7 bit ve 10 bit adresleme olanağı sunmaktadır.

SCL ve SDA pinleri giriş olarak kullanıldığında bu pinlere hataları önlemek için filtre eklentisi yapılır. Bu filtre 100KHz ve 400 KHz arasında işlem yapar. Bu pinler çıkış olarak kullanıldığında slew rate aygıt frekansı tarafından otomatik olarak kontrol edilir. Şekil 3.35.a ve şekil 3.35.b I²C modunun master/slave durumunda çalışmalarının blok diyagramını göstermektedir.





Şekil 3.35.a: Slave Modu Blok diyagramı **Şekil 3.35.b:** Master Modu Blok Diyagramı

I²C modunda 2 pin veri iletimi için kullanılır. Bunlardan SCL clock, SDA veri ucudur. I²C modu yetkilendirildiğinde SDA ve SCL pinleri otomatik olarak kurulur. SSP modülü fonksiyonları SSPEN (SSPCON<5>) bitinin kurulması ile yetkilendirilir.

MSSP modülü içerisinde I²C modunda işlem yapabilmek için 6 adet kaydedici vardır.

- SSP Kontrol Kaydedicisi (SSPCON)
- SSP Kontrol Kaydedicisi2(SSPCON2)

- SSP Statü Kaydedicisi (SSPSTAT)
- Seri Alış/İletim Tamponu (SSPBUF)
- SSP Kaydırmalı Kaydedici (SSPSR)
- SSP Adres Kaydedicisi (SSPADD)

SSPCON kaydedicisi I^2C işlemlerinin kontrol edilmesi ne olanak sağlar. SSPCON'da bulunan 4 bit (SSPCON<3:0>) I^2C modunda ayarlamaları yapar. Bu bitleri ayarlayarak şu seçimler yapılabilir:

- I^2C slave modu (7 bit adres)
- I^2C slave modu (10 bit adres)
- I^2C master modu, clock=osc/4

Herhangi bir I^2C modu seçildikten sonra, SCL ve SDA pinleri TRIS bitleri yardımıyla giriş olarak programlanmalıdır. SSPEN bitinin kurulmasıyla I^2C modu seçilmiş olur ve SCL-SDA pinleri etkin hale gelir.

3.17 USART (Adreslenebilir Evrensel Senkronize Asenkronize Alıcı Verici) Modülü

USART modül verilerin senkron veya asenkron iletimi için kullanılan ve belli bir protokol dahilinde işlem yapan bir birimdir. Bu modül dahilinde

- Verinin seriden parelele dönüştürülmesi ve parelelden seriye dönüştürülmesi,
- Eşlik bitlerini eklemek ve bu bitleri kontrol etmek suretiyle hata bulmak,
- Başlatma ve durdurma bitlerini eklemek ve bulmak, gibi işlemler yapılabilir.

İşlevsel açıdan, USART alıcı ve verici olarak iki kısma ayrılır. Her iki yönde de veri aktarmadan önce USART denetim kaydedicisine, verinin niteliğini gösterecek bir denetim sözcüğü programlanmalıdır. Örneğin; veri bitlerinin sayısı, eşlik kullanılıp kullanılmadığı, eğer eşlik kullanılmış ise, bunun tek eşlik mi yoksa çift eşlik mi olduğu ve durdurma bitlerinin sayısı gibi. Temel olarak, başlatma biti isteğe bağlı olmayan tek bittir ve her zaman yalnızca bir başlatma biti vardır. Bir başlatma durumu için bu bitin mantıksal 0 seviyesinde olması gerekmektedir.

PIC 16F877'de USART modülünün kontrolü için TXSTA ve RCSTA kaydedicileri mevcuttur.

3.17.1 TXSTA Kaydedicisi

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.36:TXSTA Kaydedicisinin Bit Yerleşimi

Txsta,0(TX9D): İletilen verinin 9. Bitini oluşturur. Bu bit bir parite biti olarak düşünülebilir.

Txsta,1(TRMT): Transmit shift register (TSR=İletim kaydırmalı kaydedicisi) durum bitidir. TRMT=1 ise TSR boştur. TRMT=0 ise TSR dolu vaziyettedir.

Txsta,2(BRGH):Yüksek baud oranı seçim bitidir. BRGH=1 ise yüksek hız, BRGH=0 ise yavaş hız seçilmiş olur.

Txsta,4(SYNC): USART modülünde mod seçim bitidir. SYNC=1 ise senkron (eşzamanlı) çalışma modu, SYNC=0 ise asenkron (eşzamanlı olmayan) çalışma modu seçilir.

Txsta,5(TXEN): USART modülünde iletim gerçekleştirmek için yetkilendirme bitidir. TXEN=1 ise iletim yetkisi verilir. TXEN=2 ise iletim yetkizi verilmez.

Txsta,6(TX9): 9 bitlik iletim yetkilendirme bitidir. TX9=1 ise 9 bitlik iletim seçilir. TX9=0 ise 8 bitlik iletim durumu seçilir.

Txsta,7(CSRC): Clock kaynağı seçim bitidir. USART eşzamanlı çalışma modunda iken; CSRC=1 ise master mod seçilmiş olur ve clock kaynağı dahilidir,CSRC=0 ise slave mod seçilmiştir ve clock kaynağı olarak harici kaynak kullanılır.

3.17.2 RCSTA Kaydedicisi

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.37:RCSTA Kaydedicisinin Bit Yerleşimi

Rcsta,0(RX9D): Alınan verinin 9. Bitini oluşturur. Bu bit bir parite biti olarak düşünülebilir.

Rcsta,1(OERR): Bitiş süresi hata bitidir. OERR=1 ise bilgi alımında belli bir süre içerisinde veri alınamamıştır. OERR=0 ise herhangi bir süre aşımı söz konusu değildir.

Rcsta,2(FERR):Çerçeve hata bitidir. FERR=1 ise çerçeve hatası vardır, FERR=0 ise çerçeve hatası yoktur.

Rcsta,3(ADDEN): Adres ortaya çıkarma durumunu etkinleştirme bitidir. ADDEN=1 ise adres ortaya çıkarma durumu etkinleştirilir, ilgili kesme aktif hale getirilir ve RSR<8> biti=1 olduğunda alış tamponu yüklenir. ADDEN=0 ise adres ortaya çıkarma durumu etkin değildir ve 9. bit parite biti olarak kullanılabilir.

Rcsta,4(CREN): Sürekli alış yetkilendirme bitidir. Asenkron modda CREN=1 ğise sürekli alış etkindir, CREN=0 ise sürekli veri alımı kullanım dışıdır. Senkron modda CREN=1 ise CREN biti temizlenene kadar sürekli veri alımı etkin olur, CREN=0 ise sürekli veri alımı kullanım dışıdır.

Rcsta,5(SREN): Tek veri alımı yetkilendirme bitidir. Master senkron modda SREN=1 ise tek veri alımı etkindir. Bu durumda işlem tamamlandığında bu bitin tekrar sıfırlanması gerekir. SREN=0 ise tek veri alımı kullanım dışıdır.

Rcsta,6(RX9): Veri alma durumu yetkilendirme bitidir. RX9=1 ise 9 bitlik veri alımı formatı seçilir. RX9=0 ise 8 bitlik veri alımı formatı seçilir.

Rcsta,7(SPEN): Seri port yetkilendirme bitidir. SPEN=1 ise seri port aktif yapılır. Bu durumda RC7/RX/DT ve RC6/TX/CK pinleri seri port pinleri olarak tahsis edilir. SPEN=0 ise seri port kullanım dışıdır.

3.17.3 USART Baud Oran Jeneratörü (BRG)

USART modülde clock'un frekansını belirleyen aygıttır. USART baud jeneratörü USART' ın Asenkronize ve Senkronize modlarını desteklemektedir. BRG 8 bit'lik bir jeneratördür. Bu cihaz 8 bit'lik serbest zamanlı bir clock ile kontrol edilir. Asenk mod'da bit BRGH (TXSTA<2>) ile kontrol edilir. Senkronize mod'da ise bit BRGH istenmez.

Verilen ve istenilen baud rate ve Fosc SPBRG (register) değerleri aşağıdaki formül ile ayarlanabilir ve hesaplanabilir.(6)

İstenen Baud oranı = Fosc / (64 (X + 1))

9600 = 16000000 /(64 (X + 1))

X = [25.042]= 25

Hesaplanan Baud oranı=16000000 / (64 (25 + 1))

= 9615

$$\text{Hata} = \frac{\text{Hesaplanan Baud Oranı-İstenen Baud Oranı}}{\text{İstenen Baud Oranı}}$$

= (9615 - 9600) / 9600

= 0.16%

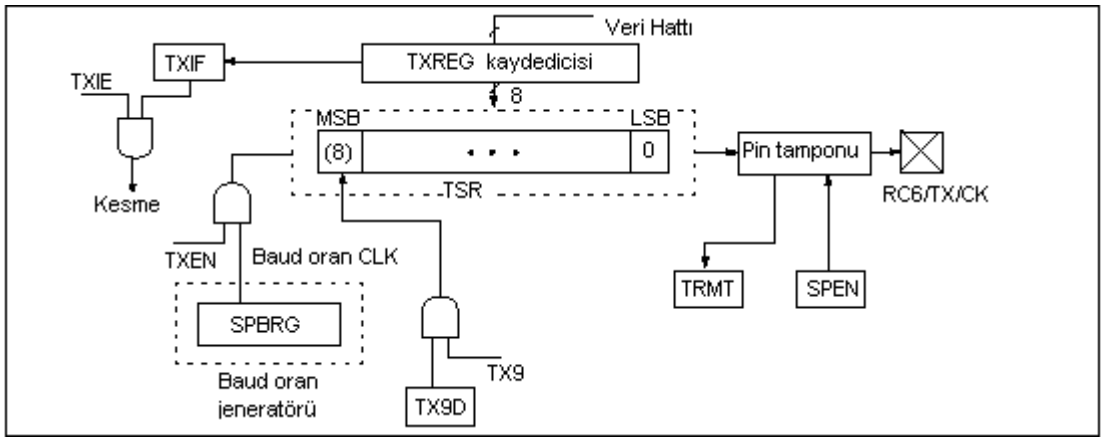
3.17.4 USART Asenkronize Durumu

Asenkron iletimde genellikle 8 bitlik iletim formatı kullanılmaktadır. PIC 16F877'de datalar denetleyici üzerinde bulunan osilatörden standart baud 'un frekans oranına göre iletilir. USART gönderici ve alıcıdan alınan ve iletilen ilk bit LSB'dir. USART iletici ve kaydedici fonksiyonel olarak bağımsız fakat aynı data formatı şeklinde ve baud oranında kullanılır. Baud jeneratörü saat gibi çalışır veya x16 yada x64 lük bitlerin oranı şeklinde üretim yapar ve BRGH biti ile kontrol edilir.. (TXSTAXZ) Bu oranlar hiçbir zaman donanım tarafından desteklenmez fakat sistem içerisinde 9 bitlik veriler halinde sistemde kullanılır. Asenkronize mod'da işemi durdurmak için denetleyici uyku moduna sokulur. Asenkronize mod SYNC bitinin durumuna göre seçilir (TXSTA <4>) USART

3.17.5 USART Asenkronize İletici

USART ileticinin kalbi TSR'dir. TSR ileticiye okuma/yazma işlemleri için geregen veriler tampon tarafından iletilir.(TXREG) TXREG kaydedicisi TSR ileticiye veri ile beraber yüklenir. İletilecek olan veriler TSR'nin yüklenmesinden sonra TXREG kaydının tamamlanması ile başlar. TXREG'de ise veri iletimi TXIF (PIR<4>) bitinin müsadesi ile olur. Yani bilgi transferi için daha önceki iletim işleminin tamamlanmış ve tamponun boşalmış olması gerekmektedir. Yeni veri iletimine başlanmadan evvel TXIF'in mutlaka

temizlenmesi gerekmektedir. Bu bitin temizlenmesi ile birlikte yeni veriler TXREG' yüklenir ve başlatma biti ile beraber veriler iletilmiş olur.



Şekil 3.38: USART İletim Blok Diyagramı

Asenkronize ileticinin kümelenmesi aşağıda adım adım açıklanmıştır.

1. SPBRG kaydedicisi baud rate'in durumuna göre harekete geçer.
2. Kullanılabilir asenkronize seri bağlantılarının temizlenebilmesi için SYNC bitinin temizlenmesi ve SPEN bitinin yetkilendirilmesi gerekir.
3. İletim için hazır vaziyete gelinmiş ise TXIE bitinin yetkilendirilmesi gerekir.
4. Eğer 9 bit'lik çevirici kullanılıyorsa, kümede TX9 biti seçilmelidir.
5. Bu çeviricinin kullanılabilmesi için TXEN bitinin yetkilenmesi ve de TXIF bitinin kümede yer alması gerekmektedir.
6. Eğer 9 bitlik çeviri seçilmiş ise TXD bitine 9 bitlik veri yüklenmelidir.
7. TXREG kaydedicisine bilgi yüklenir ve iletim başlar.

3.18 ADC (Analog Dijital Konvertör) Modülü

PIC 16 F877'de 8 kanallı 10 bit'e kadar çevirme işlemi yapabilen bir analog-dijital çevirici (ADC) modülü bulunmaktadır.

PIC 16F877 üzerindeki ADC modülün çalışması şu şekildedir. Analog giriş örnekle ve tut kondansatörünü şarj eder. Örnekle ve tut kondansatörünün çıkışı dönüştürücünün girişine uygulanır. Dönüştürücü, ardışık yaklaştırma yoluyla bu analog düzeyin sayısal sonucunu üretir. Bu A/D dönüşümde, analog giriş sinyali 10 bitlik sayı karşılaştırma ile sonuçlanır. ADC eşsiz bir özelliğe sahiptir. İşlem yapmazken uyuma moduna geçer.

Uyuma modunda ADC'nin saatinde bir iç RC osilatörü üretilmelidir. ADC Modül dört (4) kaydediciye sahiptir.

Bunlar;

1- A/D Yüksek sonuç kaydedicisi (ADRESH)
2- A/D Düşük sonuç kaydedicisi (ADRESL)
3- A/D Kontrol kaydedici 0 (ADCON0)
4- A/D Kontrol kaydedici 1 (ADCON1)

A/D çeviricinin kontrolünü ADCON0 ve ADCON1 kaydedicileri sağlamaktadır.

3.18.1 ADCON0 Kaydedicisi

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.39:ADCON0 Kaydedicisinin Bit Yerleşimi

Adcon,0(ADON):A/D çeviriciyi yetkilendirme bitidir. ADON=1 ise A/D çevirici açıktır ve işlem yapılabilir. durumdadır. ADON=0 ise A/D çevirici kapalıdır.

Adcon,2(GO/DONE): Eğer ADCON biti de 1 ise A/D çevirici statü biti görevini üstlenir. GO/DONE=1 ise A/D çevirici işlem yapıyor demektir. GO/DONE=0 ise A/D çevirici üzerinde herhangi bir işlem yapılmıyordur.

Adcon,3:5(CHS0:CHS2):A/D çevirici için kanal seçim bitlerini oluşturur. Bu bitlere verilecek değerlerle A/D çevirme için hangi kanalın seçileceği belirlenir. Daha önceden de belirtildiği gibi PIC 16F877'de A/D çevirici için 8 kanal mevcuttur.

000 = kanal 0, (RA0/AN0)
001 = kanal 1, (RA1/AN1)
010 = kanal 2, (RA2/AN2)
011 = kanal 3, (RA3/AN3)
100 = kanal 4, (RA5/AN4)
101 = kanal 5, (RE0/AN5)
110 = kanal 6, (RE1/AN6)
111 = kanal 7, (RE2/AN7)

Adcon,6:7(ADCS0:ADCS1): A/D çevirici için clock frekansı seçim bitleridir. Bu bitlere verilecek değerler ile A/D çevirme işlemi esnasında kullanılacak frekans değeri bize sunulan değerler içerisinden seçilir.

00 = FOSC/2

01 = FOSC/8

10 = FOSC/32

11 = FRC (Harici bir RC osilasyon kaynağından gelen clock darbeleri kullanılır.)

3.18.2 ADCON1 Kaydedicisi

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.40:ADCON1 Yazmacının Bit Yerleşimi

Adcon1,0:3(PCFG0:PCFG3): A/D çevirici portunun biçimini düzenlemeyi sağlayan bitlerdir. Yani A/D çevirme işleminde kullanılacak pinlerin nasıl davranacağını belirlememize olanak sağlarlar.

Tablo 3.6: PCFG3:PCFG0 Bitlerinin Aldığı Değere Göre Yaptığı İşlemler

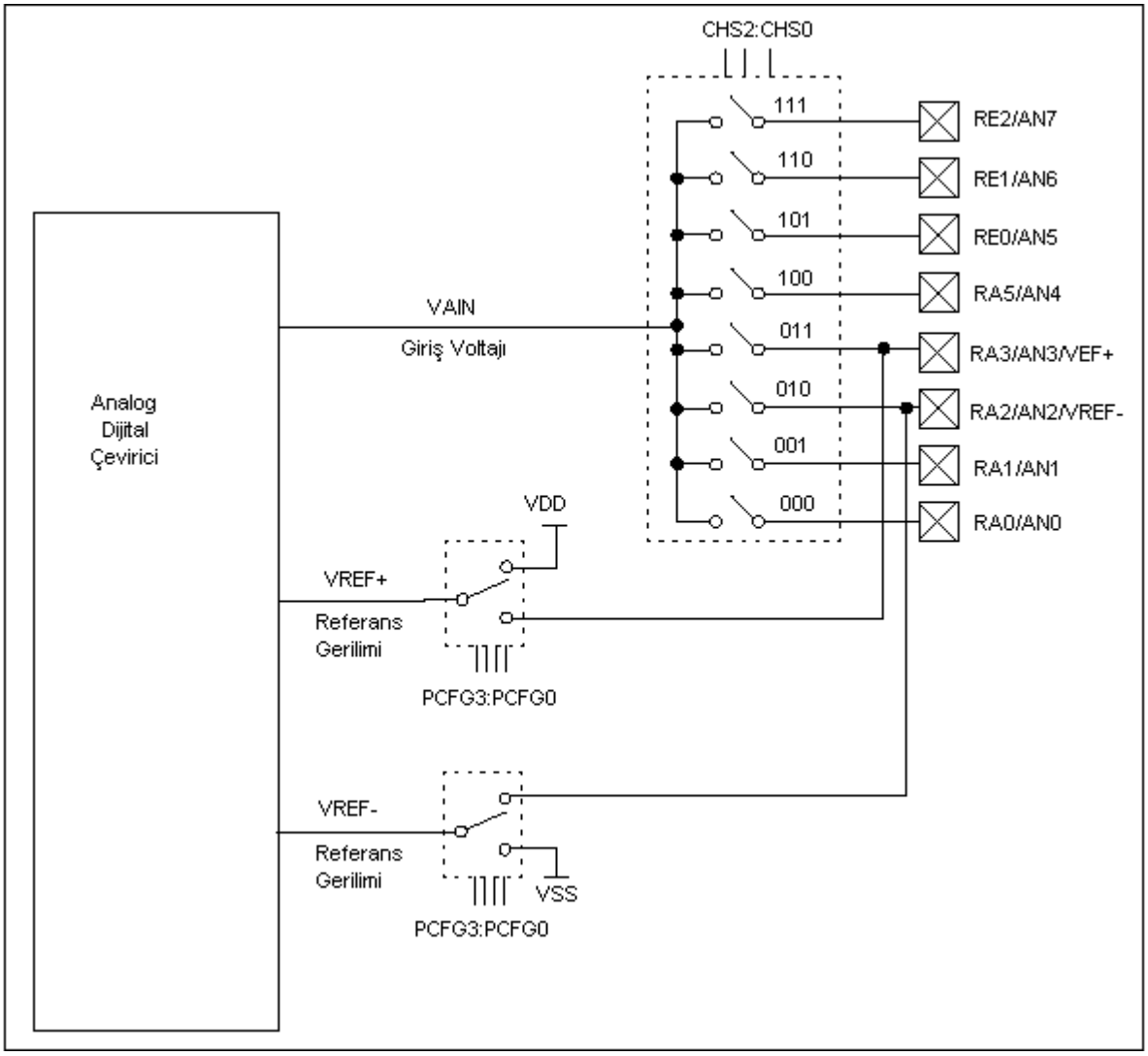
PCFG3: PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	KANAL/ REF
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0111	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

Bu tabloda A=Analog giriş D=Dijital giriş/çıkış anlamına gelir.

Adcon1,7(ADFM): A/D çevirme işlemi esnasında meydana gelen verinin biçimini belirlemeye yarayan bittir. ADFM=1 ise ADRESH kaydedicisinin MSB kısmındaki altı

biti 0 kabul edilir ve A/D çevirme sonucunda elde edilen veri ADRESH'ın 2 bitlik LSB kısmına ve ADRESL'ye yazılır. ADFM=0 ise ADRESL'nin Lsb kısmındaki 6 biti 0 kabul edilir ve A/D çevirme sonucu elde edilen veri ADRESL'nin son iki bitine ve ADRESH'a yazılır.

ADRESH: ADRESL kaydedicileri A/D dönüşümün 10 bit sonucunu kapsar. A/D dönüşümü bittiği zaman, sonuç A/D sonuç kaydedicisinin içine yüklenir. A/D modülü şekil 3.41'de görülmektedir.



Şekil 3.41: A/D Blok Diyagramı

A/D Modülü biçimlendirildikten sonra, dönüştürme işlemi başlamadan önce kanal seçilmiş olmalıdır. Analog giriş kanallarında ilgili TRIS bitleri giriş için seçilmiş olmalıdır.

Aşağıdaki adımlar, A/D dönüşüm yapmak için takip edilmelidir.

1- A/D Modülü Konfigürasyonu
   - Analog pinler, referans voltajları ve digital I/O konfigürasyonu (ADCON1)
   - A/D giriş kanalı seçimi (ADCON0)
   - A/D dönüşüm saat sekimi (ADCON0)
   - A/D Modülünü açma

2- A/D Kesme Konfigürasyonu
   - ADIF bitinin temizlenmesi
   - ADIE bitinin ayarlanması
   - GIE bitinin ayarlanması

3- Gerekli zamanı bekleme işlemi

4- Dönüşümün başlaması
   - GO/DONE bitinin ayarlanması (ADCON0)

5- A/D dönüşümünün beklenmesi

6- A/D dönüşüm sonucunu okuma ve kaydetme

7- Diğer dönüşüm için 1. ve 2. kez adımları tekrarlama

3.18.3 A/D Girdileri İçin Gereksinimler

Belirlenmiş doğruluğu karşılaştırmak için A/D çeviricinin CHOLD kondansatörü giriş gerilimine şarj edilmelidir. CHOLD kondansatörü kaynak empedansı (RS) ve anahtar iç direnci (RSS) üzerinden şarj olur. Anahtar iç direnci, kaynak voltajının değerine göre değişir. Analog kaynaklar için tavsiye edilen maksimum empedans 10 KΩ dur. Dönüşüm yapılmaya başlamadan önce analog giriş kanalı seçilmiş olmalıdır.(6)

Minimum giriş zamanı hesabı;

TCAQ=TAMP+TC+TCOFF

TACQ=Minimum girdi zamanı

TAMP=Yükselteç yerleşme zamanı

TC= CHOLD şarj zamanı

TCOFF= Sıcaklık katsayısı

Örnek 3.9:

TACQ = TAMP + TC + TCOFF

TACQ = 2 µs + Tc + [(Temp – 25 ^{o}C)(0.05 µs/ ^{o}C)]

TC = -CHOLD (RIC + RSS + RS) ln(1/2047)

-120 pF (1 kΩ+ 7 kΩ+ 10 kΩ) ln(0.0004885)

-120 pF (18 kΩ) ln(0.0004885)

-2.16 µs (-7.6241)

16.47 µs

TACQ = 2 µs + 16.47 µs + [(50 ^{o}C – 25 ^{o}C)(0.05 s/ ^{o}C)]

18.447 µs + 1.25 µs

19.72 µs

3.18.4 A/D Dönüşüm Saatinin Seçimi

TAD bit başına A/D dönüşüm zamanı olarak tanımlanır. 10 bit A/D dönüşüm için maksimum 12 TAD gerekir. TAD seçimi için mümkün olan 4 seçenek vardır.

- 2TOSC
- 8TOSC
- 32TOSC
- Dahili RC Osilatörü

A/D Dönüşümün doğru olarak yapılması için , TAD minimum 1,6 µS seçilmiş olmalıdır.

3.18.5 Analog Port Pinlerini Yapılandırma

ADCON1 ve TRIS kaydedicileri kontrol ve işletim port pinleridir. Analog girişlerin TRU bitlerinin karşılıklı olarak ayarlanması gerekir. TRIS biti temizlenmiş ise dijital çıkış seviyesine (VOH veya VOL) dönüştürme yapılmış demektir.

3.18.6 A/D Dönüşümü

Örnek 3.10'de bir A/D dönüşümünün nasıl yapıldığı gösterilmektedir. Analog pinler, analog girdiler olarak yapılandırılır. Analog referans gerilimleri VDD ve VSS dir. A/D kesmesi seçildi ve A/D dönüşüm saat frekansı (FRS) ile sola yanaşık olarak sonuçlandırıldı. Dönüştürme RA/0/AN0 pinleri ile gerçekleşti.(6)

Örnek 3.10: A/D Dönüşümü

```
bsf status, rp0          ; Bank 1
bcf status, rp1          ;
clrf adcon1              ; A/D girişleri kuruldu
bsf pie1, adıe           ; A/D komutları etkin
bcf status, rp0          ; Bank 0
movlw 11000001           ; RC saat, A/D açık, kanal 0 seçildi.
movwf adcon0             ;
bcf pir1, adif           ; A/D komut bayrak bitleri temizlendi
bsf intcon, peie         ; çevresel komutlar etkin
bsf intcon, gie          ; tüm komutlar etkin
bsf adcon0, go           ; A/D dönüşümü başla
```

3.18.7 A/D Dönüşümü Sırasında Uyuma

A/D Modül işletim sırasında uyuma modunda olacaktır. Bunun için A/D saat kaynağı ayarlanmalıdır. (ADCS1:ADCS0=11). RC saat kaynağı seçildiği zaman dönüştürme başlamadan önce A/D modül bir saat çevrimi süresi kadar bekler. Bu uyku talimatına izin verir. Dönüşümdeki tüm sayısal anahtarlama gürültüsü elemine edilmiş olur. Dönüştürme işlemi bittiği zaman GO/DONE biti temizlenir ve sonuç adres kaydedicisi içine yüklenir. A/D kesmesi etkinleştirilirse aygıt uykudan uyanır. A/D kesmesi pasifleştirilirse A/D modülü kapalı duruma dönse de bir süre açık kalacaktır.(6)

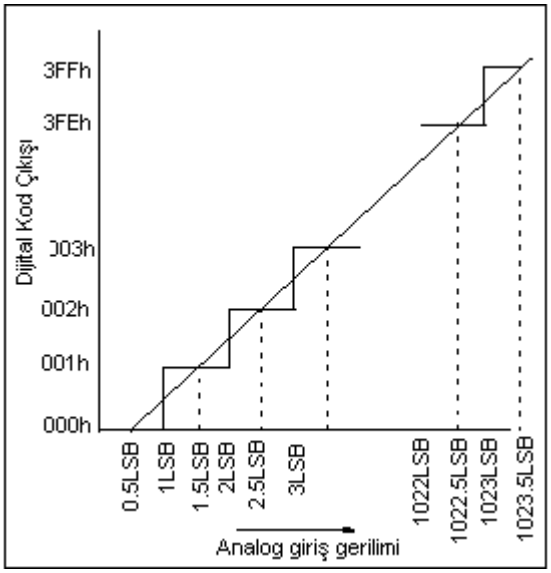
3.18.8 A/D Doğruluk/Hata

Aygıt frekansı RC saatin alçak kullanımı olduğu sistemlerde tercih edilir. Yüksek frekans azaltılırken, TAD aygıt osilatörü türetilmelidir. A/D Dönüştürücü için belirtilen salt doğruluk, miktar ölçme hatası, integral hata, türevsel hata, tam skala hata, sapma

hatalarının katkıları toplamını içerir. Herhangi bir kod için ideal geçişe karşı, güncel geçişten maksimum sapma olarak tanımlanır. Verilen bir analog giriş aralığı için sayısal çıkış kodu aynı olur. Bu sayısal kod analog girişten ölçülen miktar kadardır. Analogtan dijitale ölçme işleminde hata tipik olarak ½ LSB kadardır.

3.18.9 Transfer Fonksiyonu

A/D Dönüştürücünün transfer fonksiyonu aşağıda gösterildiği gibidir. Analog giriş voltajı/1024 ile bulunur.



Şekil 3.42:A/D Transfer Fonksiyonu

3.19 Güç Sarfiyat Bilgileri

Tablo 3.7: EEPROM verisi ile uyumlaştırılan Kayıtlar/Bitler

Adres	Ad	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Power-on resetindeki değer	Bütün dişer resetlerdeki değerler
08h	EEDATA	EEPROM VERI KAYDI								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM ADRES KAYDI								xxxx xxxx	uuuu uuuu
88h	EECON1	----	------	-----	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM KONTROL KAYDI								---------	--------

Not: EADRR <7:6> biti temizlenmelidir. Bu bitlerden herhangi birisi kurulduğunda micronun maximum IDD si her iki bitin de temizlenmiş olması halindekinden daha yüksektir. Spesifikasyon 400mA' dir. Silinen EADRR<7:6> ile maximum 150mA civarındadır. İşaretler: x =bilinmeyen, u = değişmeyen, ----- = '0' olarak tamamlanmamış okuma Q = Şartlara bağımlı değer. Bölgelendirilen hücreler EEPROM tarafından kullanılmamaktadır.

3.20 CPU' nun Spesifik Özellikleri

Mikrodenetleyici'yi diğer işlemcilerden ayıran şey , gerçek zaman uygulamalarının gereksinmeleri ile ilgili özel devreleridir. PIC16F877' te sistem güvenliğini maksimize eden, dış elemanları ayırarak maliyeti minimize eden , güç tasarrufu, çalışma modu ve kod koruma gibi özellikleri taşımaktadır. Bu özellikler;

- OSC seçimi
- Reset
    - Güç kaynağı reseti (POR)
    - Yüksek güç timerı (PWRT)
    - Osilatör başlangıç Timer ı (OST)
- Kesmeler
- Watchdog Timer
- Sleep
- Kod koruma
- ID yerleşimleri
- Devre içi seri programlama

PIC16F877' te yalnızca konfigrasyon bitleri tarafından kapatılabilen Watchdog Timer mevcuttur. Güvenliği arttırmak için bu kendi RC osilatörünü de çalıştırmaktadır. Yüksek güçte gereken esas gecikmeleri sağlayan 2 Timer mevcuttur. Bunlardan birisi Osilatör Başlangıç Timer 'ıdır. Bu timer , kristal osilatör durgunlaşıncaya kadar çipi resette tutar. Diğer timer ise yalnızca nominal yüksek güçte 72 ms sabit gecikme üreten Yüksek Güç Timer' ıdır. Bu güç kaynağı stabilize olurken aygıtı resette tutar. Bu iki çip üzeri Timer ile , uygulamaların çoğu hiçbir reset devrelerini gerektirmemektedir. SLEEP modu çok düşük enerjili alçak güç modunu sunmaktadır. Kullanıcı SLEEP ten çıkmak için dış

reset, Watchdog Timer zaman aralığı veya kesmeleri kullanabilir. Bazı osilatör seçenekleri, kısımları uygulamaya yerleştirmek için elde edilmektedir. RC osilatör seçeneği sistem maliyetini, LP kristal seçeneği ise güç sarfiyatını düşürmektedir. Çeşitli seçenekleri seçmek için konfigürasyon bitler seti kullanılmaktadır. (1)

3.20.1 Biçimlendirme (Konfigürasyon) Bitleri

Biçimlendirme bitleri çeşitli aygıt işlevlerini seçmek için programlanabilir('0' olarak okur) yada programlamadan bırakılabilir. ('1' olarak okur) Bu bitler 2007h program bellek yerleşiminde saklanır. 2007h adresi kullanıcı program bellek biriminin ötesindedir ve özel test/biçim bellek birimine (2000h- 3FFFh) aittir. Bu birime yalnızca programlama sürecinde erişilebilirdir.

CP1	CP0	BKBUG	-	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0
B13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Şekil 3.43: Biçimlendirme bitlerinin bellekteki bit dizilimi

Bit 0,1 (FOSC0,FOSC1): Osilatör seçme bitleridir. Yapılacak uygulamada hangi tür osilatör kullanılacağını tayin eder. (RC,XT,HS,LP) FOSC0 ve FOSC1 bitlerinin aldığı durumlara göre kullanılacak osilatör türleri tablo 3.8'de verilmiştir.

Tablo 3.8: Osilatör seçim tablosu

FOSC1	FOSC0	OSC
1	1	RC
1	0	HS
0	1	XT
0	0	LP

Bit 2 (WDTE): Watchdog timer'in kontrol edildiği bittir. Bu bitin 1 olması WDT'ı devreye sokarken, 0 olması ile WDT devreden çıkarılır.

Bit 3(): Power up timer kontrol bitidir. Bu bitin 1 olması ile PWRT devreden çıkarılır, 0 olması ile ise PWRT devreye sokulur.

Bit 6 (BODEN): Brown out reset özelliğine ait yetkilendirme bitidir. BODEN=1 ise BOR kullanım dışı, BODEN=0 ise BOR etkindir.

Bit 7 (LVP): Düşük gerilim programlaması için yetkilendirme bitidir. LVP=1 ise RB3/PGM pinin PGM özelliği devreye girer. LVP=0 olduğunda ise RB3 özelliği kullanılabilir.

Bit 8(CPD): Bellek EEPROM verisi kod koruması için yetkilendirme bitidir. CPD=1 ise kod koruması devre dışıdır. CPD=0 ise kod koruma etkindir ve EEPROM içerisine kayıtlı veriye bir teşebbüste bulunulamaz.

Bit 9(WRT): Flash program belleğine yazma yetkilendirme bitidir. WRT=1 ise korumasız program belleğine EECON kontrolünde veri yazılabilir. WRT=0 ise korumasız program belleğine EECON kontrolünde veri yazılamaz.

Bit 11(DEBUG): Devre üzeri seri programlama yetkilendirme bitidir. DEBUG=1 ise RB6,RB7 normal G/Ç işlemlerinde kullanılır. DEBUG=0 ise RB6, RB7 aygıt içerisine program yerleştirme işleminde kullanılır.

Bit 12:13 (CP0:CP1): Yapılan uygulamanın kopyalamaya veya herhangi bir teşebbüse karşı kod koruma özelliğinin konması işlemini kontrol eden bitlerdir. Bu bitlerin aldığı farklı değerler ile farklı kod koruma seçenekleri söz konusu olmaktadır.

11 = Kod koruma kapalı
10 = 1F00h tan 1FFFh'a kadar kod koruması
01 = 1000h tan 1FFFh'a kadar kod koruması
00 = 0000h tan 1FFFh'a kadar kod koruması

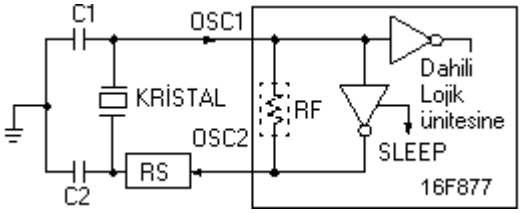
3.20.2 Osilatör Tipleri

PIC16F877 mikrodenetleyicilerinde 4 çeşit osilatör tipi bulunmaktadır. Kullanıcı bu 4 moddan birini seçerek iki biçimlendirme bitini (FOSC1 ve FOSC2) programlayabilir.

LP Kristal veya seramik rezonatör-asgari akım - 40Khz
XT Kristal veya seramik rezonatör- genel amaçlı - 4Mhz
HS Kristal veya seramik rezonatör- yüksek hız - 20 Mhz
RC Direnç/ Kapasitör zaman sabitli - düşük maliyet - 4Mhz

3.20.2.1 Kristal Osilatör / Seramik Rezonatör

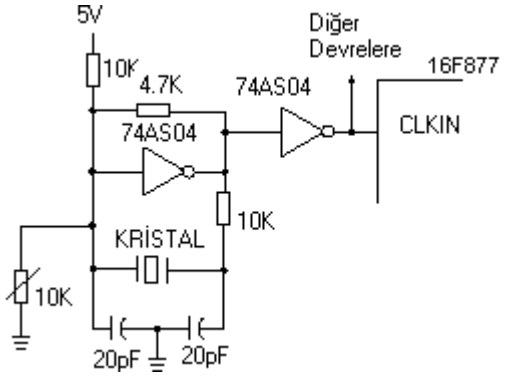
XT, LP ve HS modları, kristal veya seramik rezonatörlerin, OSC1/CLKIN ve SC2/CLKOUT pinlerine bağlanmalarıyla kurulur.PIC16F84' te osilatör dizaynı paralel kesim kristali kullanmayı gerektirir. Kesim kristallerinin seri bağlanması ile, kristallerin üzerindeki frekans değerlerinden farklı bir frekans değeri oluşabilir. XT, LP ve HS modlarında OSC1/CLK1 dışardan sürülebilir



Şekil 3.44: PIC 16F877'de Kristal Osilatör Kullanımı

3.20.2.2 Harici Kristal Osilatör Devresi

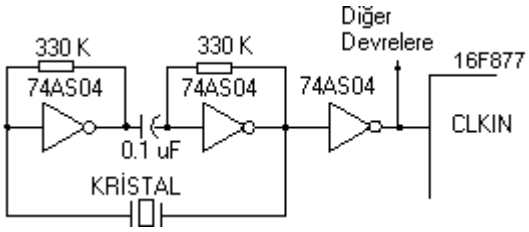
Ambalaj öncesi osilatör TTL girişli basit osilatör devresi kurulabilir. Ambalaj öncesi osilatör geniş işlem alanı ve denge sunmaktadır. İyi tasarlanmış kristal osilatör TTL girişleri ile iyi performans sağlayacaktır. İki tip kristal osilatör devresi mevcuttur. Birisi seri rezonanslı ve diğeri de paralel rezonanslı osilatördür.



Şekil 3.45: Paralel Rezonanslı Osilatör Devresi

Şekil 3.45 Paralel resonanslı osilatör devresini göstermektedir. Devre, kristalin temel frekansını kullanmak için tasarlanmıştır. 74AS04 Inverter, paralel osilatörün gerektirdiği 180 dereceli faz kaymasını yürütmektedir. 4.7 KΩ direnci kararlılık için negatif geri besleme sağlamaktadır. 10 KΩ potasiyometre 74AS04'ü lineer bölgede çalıştırır. Bu devre osilatör tasarımı için de kullanılabilir.

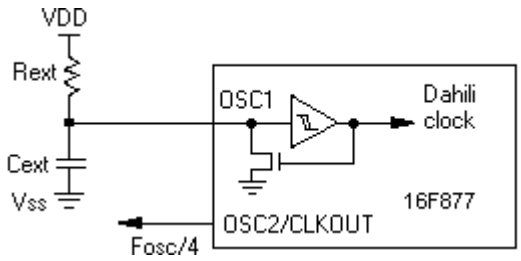
Şekil 3.46 ise seri rezonanslı osilatör devresini göstermektedir. Bu devre kristalin esas frekansını kullanmak için tasarlanmıştır. Inverter 180^o faz kaymasını yürütmektedir. 330 KΩ lık direnç, Inverter'lere kendi lineer bölgesinde etkilenmesi için negatif geri beslemeyi sağlamaktadır.



Şekil 3.46: Seri Rezonanslı Osilatör Devresi

3.20.2.3 RC Osilatörü

Zamanlamaya duyarsız uygulamalar için RC osilatörü fazla maliyeti azaltmaktadır. RC osilatör frekansı, voltaj ihtiyacına, direnç (Rext) değerine, kondansatör (Cext) değerine, çalışma ısı derecesinin değerine bağlıdır. Bunun ötesinde ambalaj tipindeki şekil kapasitansındaki farklılıkları da, özellikle düşük Cext değerlerinde, osilatörün frekansını etkileyebilmektedir. Kullanıcı dış R ve C elemanlarının toleransı nedeniyle meydana gelen değişiklikleri de dikkate almaktadır. Şekil 3.47 de RC kombinasyonunun PIC16F877'ye nasıl bağlandığı görülmektedir.



Şekil 3.47:RC Osilatör Bağlantısı

Rext < 2.2KΩ için, osilatör işlemi kararsız hale gelebilir hatta tamamıyla durabilir. Çok yüksek Rext değerleri için (yani 1 MΩ gibi) osilatör, gürültüye, rutubete ve sızmalara karşı duyarlı hale gelir. Bunun için Rext değeri 3Ω ile 100Ω arasında tutulmalıdır.

Osilatörün hiçbir dış kondansatörü olmaksızın çalışmasına rağmen (Cext=0pf) , gürültü ve kararsızlık nedenleri ile 20pf'ın üzerindeki değerlerin kullanılması yerinde olur. Çok düşük kapasitans veya hiç kapasitans olmadan, osilatör frekansı , devredeki kaçak

kapasitans gibi dış kapasitanslardaki değişmeler nedeniyle önemli ölçüde değişebilmektedir.

Frekastaki değişme ;yüksek R değerleri (çünkü gerilim sızıntı değişmeleri R den daha büyük değerde RC frekansını etkileyecektir.) ve düşük C değerlerinin etkisi ile artar. (Çünkü giriş kondansatörü değişmeleri RC frekansında daha büyük etkiye sahiptir).

OSC2/CLKOUT pininde 4 ile bölünen ösilatör frekansı mevcuttur ve bu frekans, sürücüleri test etmek için yada diğer lojik üniteleri test etmek için kullanılabilir.

4-PIC ASSEMBLY VE PIC 16F877 PROGRAMLAMA TEMELLERİ

4.1.Assembler ve PIC Assembly

4.1.1 Assembler

Bir text editöründe assembly kurallarına göre yazılmış olan komutları PIC'in anlayabileceği hexadecimal kodlara çeviren (derleyen) bir programdır. Microchip firmasının hazırladığı MPASM bu işi yapan assembler programıdır. Assembler'e çoğu zaman compiler (derleyici) de denir.

4.1.2 PIC Assembler

Assembly dili, bir PIC'e yaptırılması istenen işlerin belli kurallara göre yazılmış komutlar dizisidir. Assembly dili komutları İngilizce dilindeki bazı kısaltmalardan meydana gelir. Bu kısaltmalar genellikle bir komutun çalıştırılmasını ifade eden cümlenin baş harflerinden oluşur. Böylece elde edilen komut, bellekte tutulması kolay (mnemonic) bir hale getirilmiştir. Örneğin:BTFSC (**B**it **T**est **S**kip if **C**lear): İlgili biti test et, eğer sıfırsa bir sonraki komutu atla, anlamında kullanılan İngilizce cümlenin kısaltmasıdır.

4.2 PIC Assembly Dili Yazım Kuralları

PIC assembly programlarının bilgisayar ortamında yazılabilmesi için notepad, edit gibi yazım aracılarına ihtiyaç vardır. Microchip firmasının PIC uygulamaları için özel olarak hazırlamış oduğu MPLAB programını kullanmamız halinde bu gibi yazım aracılarına ihtiyaç duyulmaz. Çünkü MPLAB içerisinde assemply PIC assembly dilinde program yazmak için text tabanlı bir yazım editörü ve ayrıca MPASM mevcuttur.

MPASM assembler programının yazılan komutları doğrun olarak algılayıp, PIC'in anlayabileceği hexadecimal kodlara dönüştürebilöesi için şu bilgilerin program içerisinde özel formatta yazılması gerekir.

- Komutların hangi PIC'e ait olduğu (PIC 16F877,PIC 16F84, PIC 16C71...vb)
- Programın bellekteki hangi adresten başlayacağı.
- Komutların ve etiketlerin neler olduğu

- Programın bitiş yeri

Örnek 4.1

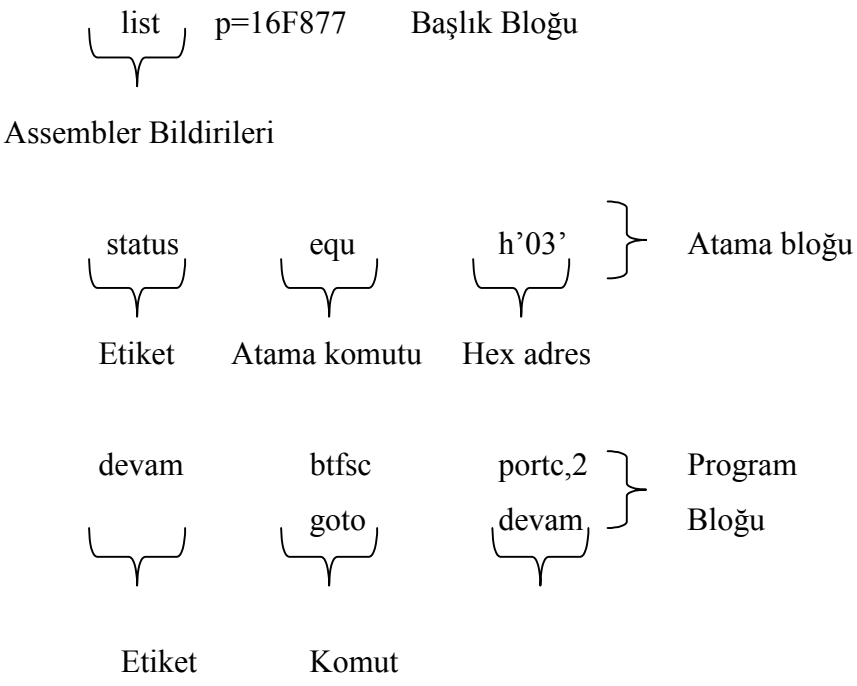
PIC 16F877'de PORTC'nin 4 bit LSB bitlerini giriş olarak, 4 bit MSB bitlerini çıkış olarak tanıtan ve PORTC'nin 2. Bitinin 1 olması durumunda PORTC nin 6. Bitini 1 yapan programın yazımı:

```
;-------------Örnek Program 1-----------
        list    p=16f877        ;kullanılacak PIC in seçimi
;---------------------------------------------
;Adres tanımlama bloğu
status  equ     h'03'
portc   equ     h'07'
trisc   equ     '87'
;---------------------------------------------
        org     h'00'           ;programı 00 adresinden başlat
;---------------------------------------------
;PORTC nin G/Ç tanımlamasının yapılması
        clrf    portc
        bsf     status,5        ;bank1 seçildi
        movlw   h'0F'
        movwf   trisc
        bcf     status,5        ;bank0 seçildi
;---------------------------------------------
;Program bloğu
basla   bcf     portc,6
        btfss   portc,2
        goto    basla
        bsf     portc,2
devam   btfsc   portc,2
        goto    devam
        goto    basla
        end                     ;sonlandırma
```

4.2.1 Program Yazımında Noktalı Virgül (;) Kullanımı

Baş tarafında (;) bulunan satırlar, assembler tarafından hexadecimal kodlara dönüştürülmez. Bu satırlar programın geliştirilmesi esnasında hatırlatıcı açıklamaların yazılmasında kullanılır. Ayrıca program bölümlerni birbirinden ayırmak için (------- veya =======) çizgileri kullanmak, programı görsel olarak daha okunur halegetirdiği gibi bu çizgiler arasına uyarılar ve açıklamalar da yazılabilir. Bu sayede programda bir hata veya sorun meydana geldiğinde takip işlemi daha rahat yapılır.

4.2.2 Bir Program Satırının Kısımları



4.2.2.1 Etiket

PIC belleğindeki bir adresin atandığı, hatırlamayı kolaylaştıran kısaltmalardan meydana gelen sembolik işaretlere etiket denir. Örneğin portc etiketi, PIC 16F877'nin kaydedici dosyası belleğindeki C portunun bulunduğu adresi temsil eden etikettir. Etiketler program içerisinde 1. kolana yazılır.

Portc equ h'07' yazıldıktan sonra C portunun hangi adreste olduğunu akılda tutmaya gerek yoktur. Programın herhangi bir yerinde portc etiketi kullanıldığında,C portunun adresi olan h'07' yazılmış gibi işlem görülür.

Birinci kolona yazılan ve adres atanmayan etiketler de kullanılabilir. Örneğin basla ve devam bu tip etiketlerdendir. Bu etiketler program akışını istenilen yere dallandırmak için kullanılırlar. Bu tip etiketlerin adresi özel kaydedici adresi gibi fiziksel bir adres değildir. Bu şekilde tanımlanan bir etikete assembler otomatik olarak adres verir. Bu adresi programcının bilmesi gerekmez.

Etiket tanımlarken uyulması gereken kurallar şunlardır.

- Etiketler birinci kolona yazılmalıdır.
- Etiketler bir harfle veya alt çizgi (_) ile başlamalıdır.
- Etiketler içerisinde Türkçe karakterler kullanılamaz.
- Etiketler bir asembly komutundan oluşamaz.
- Etiketlerin içerisinde alt çizgi, rakam, soru işareti bulunabilir.
- Etiketler en fazla 31 karakter uzunluğunda olabilir.
- Etiketlerde büyük/küçük harf duyarlılığı vardır. (“Devam” diye tanımlanmış bir etiketi program içerisinde “devam” yazarak kullanmak mümkün değildir.)

4.2.2.2 Atama Deyimi (EQU)

EQU deyimi PIC 16F877’nin belleğindeki bir hexadecimal adresi belirlenen bir etikete atamak için kullanılır. Aşağıda bu atama deyimine birkaç örnek gösterilmiştir.

```
portb   equ     h’06’
say1    equ     h’10’
portc   equ     h’07’
```

4.2.2.3 Sabitler

PIC assembly dilinde hexadecimal,binary ve decimal sayılar birer sabittir. Sabitler movlw, sumlw, andlw gibi bazı komutların içerisinde ve atama işlemlerinde kullanılırlar. Sabitler program içerisinde kullanılırken hexadecimal olduuna dair “h” veya 0x.., binary olduğuna dair “b”, decimal olduğuna dair “d” karakterleri ile birlikte kullanılmalıdırlar.

```
movlw       b’01000111’   ;binary sabit
sublw       d’126’        ;decimal sabit
sublw       h’0F’         ;hexadecimal sabit
```

```
movlw       0x0F        ;hexadecimal sabit
```

4.2.2.4 ORG Deyimi

ORG İngilizcedeki "origin" kelimesinden gelmektedir. ORG deyimi program içerisinde iki amaç için kullanılır.

- Program komutlarının hangi adresten itibaren başladığını gösterir.

```
org     h'00'
```

- PIC 16F877'nin interrupt (kesme) alt programlarının başlangıç adresini belirlemede kullanılır.

```
org     h'04'
```

4.2.3 PIC Assembly Komutlarının Yazılışı

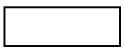
PIC 16f877'nin toplam 35 tane komutu vardır. Bu komutların yazılış biçimini dört grupta toplayabiliriz.

- Byte yönlendirmeli komutlar
- Bit yönlendirmeli komutlar
- Sabit işleyen komutlar
- Kontrol komutları

Komutların yazılış biçimlerini açıklarken bazı tanımlama harfleri kullanacağız. Öncelikle bu harflerin anlamlarını vermekte fayda vardır.

f=File register (Kaydedici)
d=destination (gönderilen yer)
    d=0→w kaydedicisi (akümülatör)
    d=1→f kaydedicisi
k=sabit veya adres etiketi
b=bit tanımlayıcı
b=binary sayıları belirleyen harf
d=decimal sayıları belirten harf
h=hexadecimal sayıları belirten harf

4.2.3.1 Byte Yönlendirmeli Komutlar



Komut f,d→f: Hexadecimal adres veya kaydedici adı

d:Komutun çalıştırılmasından sonra verinin yazılacağı yer.

d=0→w kaydedicisi

d=1→f kaydedicisi

Örnek

```
movf   h'03',0    ;h'03' adresindeki kaydedicinin içeriğini w kaydedicisi içerisine kopyalar
movf   status,0   ;STATUS kaydedicisinin içeriğini w kaydedicisine kopyalar.
movf   status,1   ;STATUS kaydedicisinin içeriğini yine STATUS kaydedicisine kopyalar.
```

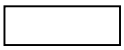
Not: Bit yönlendirmeli komutlarda destination (gönderilecek yer) belirleyen d'nin yazıldığı yere 0 veya 1 yazmak hatırlatıcı olmayabilir. MPASM bunu dikkate alarak 0 yerine w, 1 yerine f yazmaya izin verir. MPASM'nin MS-DOS versiyonunda ise w ve f harflerinin otomatik olarak kullanılmasına izin verilmez. Bu durumda her programın tanımlama bölümüne aşağıdaki eşitlikler yazılmalıdır.

```
w      equ    0
f      equ    1
```

Bu eşitliklerden sonra komutlarda destination belirlemek için w ve f harfleri kullanılabilir. Örneğin:

```
decfsz         say1,f
movf           say2,w
```

4.2.3.2 Bit Yönlendirmeli Komutlar



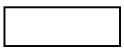
Komut f,b→f: Hexadecimal adres veya kaydedici adı

b:0-7 arasında hexadecimal sayı veya etiket (EQU komutu ile adresi tanımlanmış olmalıdır)

Örnek

```
bcf    h'03',5      ;h'03' adresindeki verinin 5. Bitini sıfırla
bsf    porta,2      ;PORTA'nın 2. bitini birle
```

4.2.3.3 Sabit İşleyen Komutlar

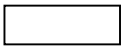


Komut k→k: sabit (b'00110011', h'0F', d'255' gibi)

Örnek

```
movlw   h'2f'     ;w kaydedicisine 2F hexadecimal sayısını yükler
addlw   d'221'    ;w kaydedicisindeki sayıya 221 decimal sayısını ekler.
```

4.2.3.4 Kontrol Komutları



Komut k→k: Adres etiketi

Örnek

```
goto    dongu     ;programı dongu ile belirtilen yere dallandır.
call    timer     ,program akışı timer etiketi ile belirtilen alt programa dallanır
```

4.2.4 Program Bölümleri

Bu kısımda programın metin belgesi olarak nasıl yazıldığı konusuna değinilecektir. Yazılan programı temel olarak 5 kısımda toplayabiliriz.

- PIC serisinin tanıtılması ve kaynak dosyanın belirtilmesi
- Kaydedici tanımlamalarının yapılması
- Bit tanımlamalarının yapılması
- Giriş ve çıkış pinlerinin belirlenmesi
- Pic'in yapacağı işlemlerin yazılması (Brogram bloğu)

4.2.4.1 Pic Serisinin Tanıtılması Ve Kaynak Dosyanın Belirtilmesi

Piyasada değişik isimlerle değişik işlemleri gerçekleştiren PIC'ler satılmaktadır. Her bir PIC'in değişik yapı ve tanımlamaları vardır. Bu tanımlamaların kayıtları ise PIC'in özellikleri göz önünde tutularak yazılmış kaynak dosyalarda tutulur. Ancak her bir seriye ait özellikleri tek tek bilme imkanımız çok azdır. Bu nedenle PIC üreten firmalar üretilen

elemana ait özellik ve kaynak dosyaları hazır halde bize sunmaktadır. Bizler bu kaynak dosyalardan kullanacağımız PIC'e ait olanını alabilir ve kendimize göre değişiklikler yaparak kullanabiliriz. Ancak yapılacak değişiklikler standart özellikleri etkilememelidir. Bir kaynak dosyada temel olarak standart register tanımlamaları (W, F, STATUS, EEDATA, EEADR,PORTA,PORTB….gibi), standart bit tanımlamaları (WREN ZERO, CARRY….gibi) ve var ise kendimize ait macrolar bulunmaktadır.

Her program metninin baş tarafında yer alması gereken seri tanıtımı ve kaynak dosya belirtme işlemi aşağıdaki şekillerde gerçekleştirilebilir.

```
list p=16f877
#include <16f877.ınc>
```

Yukarıdaki tanımlamanın ilk satırında kullanılacak pic'in serisi belirtilmiştir.2. satırda ise program metninin bulunduğu klasör içerisinde aranması istenen kaynak dosya (INCLUDE dosyası) belirtilmiştir.

4.2.4.2 Kaydedici Tanımlamalarının Yapılması:

Yukarıda bize sunulan kaynak dosyaların içerisinde standart kaydedicilerin tanımlandığından söz etmiştik. Ancak çoğu çalışmada tanımlanan bu kaydediciler yetersiz kalmaktadır ve yeni RAM bölgeinden kaydediciler tanımlanması gerekmektedir. Bu durumda kullanılmak istenen kaydedici bellek bölgesinin adresi ile eşleştirilip bir isim verilerek program içerisinde belirtilebilir ve ilerleyen işlemlerde rahatlıkla kullanılabilir. Ancak bu işlem gerçekleştirilirken kaynak dosya içerisinde tanımlanan standart adresleri bilmek ve tanımlayacağımız kaydedicilerin adresleini bu adreslerin dışında seçmemiz gerekmektedir. PIC 16F877 kaynak dosyası içerisinde yer alan standart kaydediciler aşağıdaki gibidir.

```
indf        equ   h'0000'
tmr0        equ   h'0001'
pcl         equ   h'0002'
status      equ   h'0003'
fsr         equ   h'0004'
porta       equ   h'0005'
```

```
portb           equ   h'0006'
portc           equ   h'0007'
portd           equ   h'0008'
porte           equ   h'0009'
pclath          equ   h'000a'
intcon          equ   h'000b'
pir1            equ   h'000c'
pir2            equ   h'000d'
tmr1l           equ   h'000e'
tmr1h           equ   h'000f'
t1con           equ   h'0010'
tmr2            equ   h'0011'
t2con           equ   h'0012'
sspbuf          equ   h'0013'
sspcon          equ   h'0014'
ccpr1l          equ   h'0015'
ccpr1h          equ   h'0016'
ccp1con         equ   h'0017'
rcsta           equ   h'0018'
txreg           equ   h'0019'
rcreg           equ   h'001a'
ccpr2l          equ   h'001b'
ccpr2h          equ   h'001c'
ccp2con         equ   h'001d'
adresh          equ   h'001e'
adcon0          equ   h'001f'
option_reg      equ   h'0081'
trisa           equ   h'0085'
trisb           equ   h'0086'
trisc           equ   h'0087'
trisd           equ   h'0088'
trise           equ   h'0089'
pie1            equ   h'008c'
```

```
pie2        equ   h'008d'
pcon        equ   h'008e'
sspcon2     equ   h'0091'
pr2         equ   h'0092'
sspadd      equ   h'0093'
sspstat     equ   h'0094'
txsta       equ   h'0098'
spbrg       equ   h'0099'
adresl      equ   h'009e'
adcon1      equ   h'009f'
eedata      equ   h'010c'
eeadr       equ   h'010d'
eedath      equ   h'010e'
eeadrh      equ   h'010f'
eecon1      equ   h'018c'
eecon2      equ   h'018d'
```

Tanımlanabilecek kişisel kaydedicilere ise şu örnekleri verebiliriz.

```
say1        equ    h’10’
say2        equ    h’11’
reg         equ    h’20’
reg1        equ    h ‘1a’
```

4.2.4.3 Bit Tanımlamalarının Yapılması

Program içerisinde programdaki işlemleri daha rahat anlamak ve yazım esnasında hatayı azaltmak için çıkış pinlerine ve entegre içerisinde kontrolü sağlayacak bitlere kolay anlaşılabilecek isimler verilebilir. İsimlendirilecek bu bitlerin ise programda belirlenmesi gerekir. Yani verilecek ismin hangi kaydedicinin hangi bitine ait olduğu programa yazılmalıdır. Bu kısımda yine kaynak dosyada kullanılan bazı standart bitler karşımıza çıkar. Bitleri isimlendirir iken bu standart bitleri göz önünde bulundurmak gerekir. Standart olarak tanımlanan bitlerin bazıları aşağıda sunulmuştur.

```
;----- status bitleri ----------------------------------------------------
#define     irp         status,7
#define     rp1         status,6
#define     rp0         status,5
#define     not_to      status,4
#define     not_pd      status,3
#define     z           status,2
#define     dc          status,1
c           status,0

;----- intcon bitleri ----------------------------------------------------
#define     gie         intcon,7
#define     peie        intcon,6
#define     t0ie        intcon,5
#define     inte        intcon,4
#define     rbie        intcon,3
#define     t0if        intcon,2
#define     intf        intcon,1
#define     rbif        intcon,0

;----- adcon0 bitleri ----------------------------------------------------
#define     adcs1       adcon0,7
#define     adcs0       adcon0,6
#define     chs2        adcon0,5
#define     chs1        adcon0,4
#define     chs0        adcon0,3
#define     go          adcon0,2
#define     not_done    adcon0,2
#define     go_done     adcon0,2
#define     chs3        adcon0,1
#define     adon        adcon0,0
```

Bu standart bit tanımlamaları yanında standart kaydedicilerin bazılarının ve kişisel kaydedicilerin bitlerine özel isimler verme olanağımız da vardır. Buna birkaç örnek aşağıda sunulmıştur.

```
#define     tus         porta,2
#define     clock       portb,1
#define     kontrol     say1,7
#define     en          reg1,0
```

4.2.4.4 Giriş ve Çıkış Tanımlamaları

PIC 16F877 giriş ve çıkışın yapıldığı pinler PORTA, PORTB, PORTC, PORTD, PORTE pinleridir. Bu portlara ait tüm pinleri hem giriş hem de çıkış olarak kullanabiliriz. Ancak program içerisinde pinlerin hangisini giriş için hangisini çıkış için kullandığımızı belirtmek gerekir. Çıkış için tanımlanan bir pine gerilim girişi yapmak bu pinin bozulmasına neden olabilir. Bu yüzden işlemlerde kullanılmayan pinleri programda giriş olarak tanımlamak faydalı olmaktadır. Bir program içerisinde yapılabilecek giriş ve çıkış tanımlamaları aşağıda örnekler halinde sunulmuştur.

```
bank1
movlw    b'00011100'
movwf    trisa
movlw    b'01100110'
movwf    trisb
bank0
```

4.2.4.5 PIC'in Yapacağı İşlemlerin Yazılması

PIC'in yapacağı işlemleri yazmak demek komutları bir algoritma veya akış şemasına göre programda yerlerine yazmak demektir. (Bu kısımda yapılacak işlemler sonraki kısımda örnek ve uygulamalar ile genişçe anlatılacaktır.)

4.3 PIC 16F877 Komut Takımı

PIC 16F877'nin programlanmasında kullanılan 35 komutluk komut takımı tablo 4.1' de verilmiştir.

Tablo 4.1: PIC 16F877 Komut Takımı

Komut		Süresi(T)	Etkilenen Bit
ADDWF	f,d	1	C,DC,Z
ANDWF	f,d	1	Z
CLRF	f	1	Z
CLRW	-	1	Z
COMF	f,d	1	Z
DECF	f,d	1	Z
DECFSZ	f,d	1(2)	
INCF	f,d	1	Z
INCFSZ	f,d	1(2)	
IORWF	f,d	1	Z
MOVF	f,d	1	Z
MOVWF	f	1	
NOP	-	1	
RLF	f,d	1	C
RRF	f,d	1	C
SUBWF	f,d	1	C,DC,Z
SWAPF	f,d	1	
XORWF	f,d	1	Z
BCF	f,b	1	
BSF	f,b	1	
BTFSC	f,b	1(2)	
BTFSS	f,b	1(2)	
ADDLW	k	1	C,DC,Z
ANDLW	k	1	Z
CALL	k	2	
CLRWDT	-	1	TO,PD
GOTO	k	2	
IORLW	k	1	Z
MOVLW	k	1	
RETFIE	-	2	
RETLW	k	2	
RETURN	-	2	
SLEEP	-	1	TO,PD
SUBLW	k	1	C,DC,Z
XORLW	k	1	Z

4.4 Komut Açıklamaları

ADDWF f,d: Akümülatördeki herhangi bir veri ile f kaydedicisindeki veriyi mantıksal olarak toplar ve sonucu d'nin değeri f ise kaydediciye w ise akümülatöre kaydeder.

İşlem: (w)+(f) → d

Örnek:addwf porta,f

İşlemden önce:W=00110011 PORTA:00010101

İşlemden sonra:W=00110011 PORTA:01001000

ANDWF f,d: Akümülatördeki veri ile f kaydedicisindeki veriyi ve (AND) işlemine sokar ve sonucu d'nin değeri f ise kaydediciye w ise akümülatöre kaydeder.

İşlem: (w).AND.(f) → d

Örnek:andwf portb,w

İşlemden önce: W=00110011 PORTB=00000011

İşlemden sonra: W=00000011 PORTB=00000011

CLRF f:f kaydedicisinin içeriğini siler (sıfırlar).

İşlem: f=0

Örnek: clrf portb

İşlemden önce: PORTB=00000011

İşlemden sonra: PORTB=00000000

CLRW: Akümülatördeki verileri siler.

İşlem: W=0

Örnek: clrw

İşlemden önce: W=11110001

İşlemden sonra: W=00000000

COMF f,d:f kaydedicisinde yer alan verinin tümleyenini (complementini) alır ve d'nin aldığı değere göre sonuç akümülatöre veya f kaydedilir.

İşlem: $\overline{f}$ =d

Örnek: comf portb,w

İşlemden önce: W=01010101 PORTB=00101010

İşlemden sonra: W=11010101 PORTB=00101010

DECF f,d: f kaydedicisindeki veriden 1 azaltır ve d'nin aldığı değere göre sonucu akümülatöre veya f kaydedicisine kaydeder.

İşlem: f-1=d

Örnek: decf pcl,w

İşlemden önce: PCL=10001111 W=11010101

İşlemden sonra: PCL=10001111 W=10001110

DECFSZ f,d: f ile belirtilen kaydedicideki değeri 1 azaltır ve sonucu d değerinin aldığı değere göre f veya akümülatöre kaydeder. Daha sonra sayıcı değerinin 0 olup olmadığına bakılır. Eğer değer 0 ise bir alttaki satır atlanır. Eğer 0 değil ise bir alttaki satır okunur.

```
Örnek: devam    decfsz  say1,f
                goto    devam
                end
```

Yukarıdaki program parçasında say1'in değeri 1 azaltılır. Eğer say1'in değeri 0 olmuş ise program sonlanır. Eğer 0 değilse program devam etiketinin olduğu satıra gelir.

INCF f,d=f kaydedicisindeki veriye 1 ekler ve d'nin aldığı değere göre sonucu akümülatöre veya f kaydeder.

İşlem: f+1=d

Örnek: incf pcl,f

İşlemden önce:pcl=01110000 W=10001110

İşlemden sonra:pcl=01110001 W=10001110

INCFSZ f,d: f ile belirtilen kaydedicideki değeri 1 arttırır ve sonucu d değerinin aldığı değere göre f veya akümülatöre kaydeder. Daha sonra sayıcı değerinin 0 olup olmadığına bakılır. Eğer değer 0 ise bir alttaki satır atlanır. Eğer 0 değil ise bir alttaki satır okunur.

Örnek: devam incfsz say1,f
goto devam
end

Yukarıdaki program parçasında say1'in değeri 1 arttırılır. Eğer say1'in değeri 0 olmuş ise program sonlanır. Eğer 0 değilse program devam etiketinin olduğu satıra gelir.

IORWF f,d: Akümülatördeki bilgi ile f kaydedicisindeki veriyi veya (OR) işlemine sokar ve d'nin aldığı değere göre sonucu akümülatöre veya f kaydedicisine kaydeder.

İşlem: (w).OR.(f) → d

Örnek: iorwf porta,w
İşlemden önce: W=10001110 PORTA=01010101
İşlemden sonra: W=11011111 PORTA=01010101

MOVF f,d: f kaydedicisinde yer alan veriyi akümülatöre taşır. Burada d değeri her nekadar f değerini de alabiliyor gibi gözükse de d=w olmalıdır.

İşlem: (f) →(w)

Örnek: movf porta,w
İşlemden önce: W=00000000 PORTA=01010101
İşlemden sonra: W=01010101 PORTA=01010101

MOVWF: Akümülatörde bulunan veriyi f kaydedicisine taşır.

İşlem: (w) → (f)

Örnek: movwf portb
İşlemden önce:W=01010101 PORTB=00000111
İşlemden sonra: W=01010101 PORTB=01010101

NOP: Bu komutun bulunduğu satıra gelindiğinde hiçbir işlem yapılmaz.1µs'lik bir beklemeden sonra program kaldığı yerden devam eder. Bu komut genellikle zaman gecikmesi istenen yerlerde bir döngü ile birlikte kullanılır

RLF f,d: f kaydedicisindeki veriyi sola kaydırma işlemini yapar. Çarpma işlemini yapan döngülerde oldukça çok kullanılmaktadırlar. Aslında bir dijital veriyi 1 kere sola kaydırmak demek sayıyı 2 ile çarpmak demektir. Sayıyı n defa sola kaydırmak demek ise sayıyı 2^n ile çarpmak demektir. Kaydırma işleminden sonra kaydedicinin değeri d'nin değerine göre kaydediciye ya da akümülatöre atılır.

Örnek: rlf status,f
İşlemden önce: STATUS=01110010
İşlemden sonra: STATUS=11100100

RRF f,d: f kaydedicisindeki veriyi sağa kaydırma işlemini yapar. Bölme işlemini yapan döngülerde oldukça çok kullanılmaktadırlar. Aslında bir dijital bilgiyi 1 kere sağa kaydırmak demek sayıyı 2'ye bölmek demektir. Sayıyı n defa sağa kaydırmak demek ise sayıyı 2^n'e bölmek demektir. . Kaydırma işleminden sonra kaydedicinin değeri d'nin değerine göre kaydediciye ya da akümülatöre atılır.

Örnek: rrf portb,f
İşlemden önce: PORTB=01010101
İşlemden sonra: PORTB=00101010

SUBWF f,d: f kaydedicisinde bulunan veriden akümülatörde bulunan veriyi mantıksal çıkarma işlemine tabi tutar. Sonucu d değerine göre f kaydedicisine veya akümülatöre kaydeder.

İşlem: (f) –(w) $\rightarrow$ d

Örnek: subwf reg1,f
İşlemden önce:REG1 = 00000011
W = 00000010 C = ? Z = ?
İşlemden sonra:REG1 = 00000001
W = 00000010 C = 1;sonuç pozitif Z = 0

Örnek: : subwf reg1,f
İşlemden önce:REG1 = 00000010
W = 00000010 C = ? Z = ?
İşlemden sonra:REG1 = 00000000

W = 00000010 C = 1; sonuç 0 Z = 1

Örnek: : subwf reg1,f

İşlemden sonra:REG1 = 00000001

W = 00000010 C = ? Z = ?

İşlemden sonra:REG1 = 11111111

W = 00000010 C = 0; sonuç negatif Z= 0

SWAPF f,d:f kaydedicisinde bulunan veride 4'er bitlik bloklar halinde yer değiştirme yapar. Yani f kaydedicisinin ilk 4 biti düşük son 4 biti yüksek bitler olarak isimlendirilirse; bu komut ile yüksek bit ile düşük bitler yer değiştirir. Elde edilen sonuç ise d değerine bağlı olarak f kaydedicisine ya da akümülatöre kaydedilir.

Örnek: swapf say1,w

İşlemden önce:SAY1=01010011 W=11010111

İşlemden sonra:SAY1=01010011 W=00110101

XORWF f,d: Akümülatördeki veri ile f kaydedicisindeki veriyi özel veya (EXCLUSİVE-OR) işlemine sokar ve d'nin aldığı değere göre sonucu akümülatöre veya f kaydedicisine kaydeder.

İşlem: (w).EXOR.(f) $\rightarrow$d

Örnek: xorwf status,f

İşlemden önce: W=10111111 STATUS=11100100

İşlemden sonra: W=10111111 STATUS=01011011

BCF f,b:f kaydedicisinin b. bitini temizler (sıfırlar)

İşlem: f,b=0

$0 \leq b \leq 7$

Örnek: bcf porta,0

İşlemden önce: PORTA=01110101

İşlemden sonra:PORTB=01110100

BSF f,b: f kaydedicisinin b. bitini 1 yapan komuttur.

İşlem: f, b=1

0≤b≤7

Örnek: bsf portc,4

İşlemden önce: PORTC=0110000

İşlemden sonra:PORTC=0111000

BTFSC f,b:f kaydedicisinin b. bitine göre işlem gerçekleştiren komuttur. İstenen bitin sonucu 1 ise bir alt satırdan işlemlere devam edilir. Eğer sonuç 0 ise bir alttaki satır atlanır.

İşlem: f,b=0 alttaki satırı atla

Örnek:addlw b'11111110'

btfsc carry

bcf status,3

andwf porta,w

İşlemden önce:W=00000011 CARRY=0 STATUS=01110100 PORTA=01001000

İşlemden sonra:W=00000000 CARRY=0 STATUS=01110000 PORTA=01001000

Yukarıdaki program parçasında öncelikle akümülatöre 11111110 sabit sayısı eklendi. Bu aşamada akümülatöre 00000001 sayısı yerleşti ve CARRY bitine 1 sayısı yerleşti. Bu durumda işlem BCF STATUS,3 satırından devam etti. Eğer ki toplama sonucu CARRY biti 0 olsa idi işlem ANDWF PORTA,W satırından devam edecek idi.

BTFSS f,b:f kaydedicisinin b. bitine göre işlem gerçekleştiren komuttur. İstenen bitin sonucu 0 ise bir alt satırdan işlemlere devam edilir. Eğer sonuç 1 ise bir alttaki satır atlanır.

İşlem: f,b=1 alttaki satırı atla

Örnek:btfss buton1

addlw 00001111

bsf porta,2

İşlemden önce:W=00000000 BUTON1=0 PORTA=01001000

İşlemden sonra:W=00000000 BUTON1=0 PORTA=01001010

Yukarıdaki program parçasında BUTON1 biti işlem kolaylaştırmak için daha önceden tanımlanmış olması gereken bir bilgidir. Eğer ki BUTON1 1 ise PORTA' nın 2. Biti 1 yapılacaktır, 0 ise önce akümülatöre 0nce 00001111 sabiti yüklenecek daha sonra PORTA'nın 2. Biti 1 yapılacaktır. (Yukarıda yaptığımız işlemler BUTON1'in 1 konumu içindir.)

ADDLW k: Herhangi bir sabit ile akümülatördeki veriyi toplayıp akümülatöre kaydeden komuttur.

İşlem:(w)+k $\rightarrow$ w $0 \leq k \leq 255$

Örnek: addlw b'00110110'

İşlemden önce: W=00111101

İşlemden sonra: W=01110011

Eğer ki toplama işlemi sonucunda bir taşma oluşmuş ise bu CARRY bitine kaydedilir.

ANDLW k: Herhangi bir sabit ile akümülatördekiveriyiand (ve) işlemine sokan komuttur.

İşlem: (w).AND.k $\rightarrow$(w) $0 \leq k \leq 255$

Örnek:andlw b'00111011'

İşlemden önce: W=01110011

İşlemden sonra:W=00110011

CALL k: k ile belirtilen etiketli satıra gider ve return komutunu görene kadar işlemleri gerçekleştirir. Return komutunu gördüğü anda ise call k satırının bir altındaki satıra dönüş yapılır. k yerine herhangi bir bilgiyi içeren sayı yazabileceğimiz gibi etiket ismi olabilecek bir kelime de yazabiliriz.

CLRWDT: Watchdog Timer'ın sıfırlandığı komuttur. Watchdog Timer genellikle frekans bölme işlemlerinde kullanılır ve her yeni bölme işleminin başında sıfırlama yapılması gerekmektedir.

GOTO k:CALL' ın yaptığı işleme benzer bir işlem yapar. Yani k ile belirtilen satıra gider ve işlemlere devam eder. Ancak return ve retlw gibi sonlandırma komutları yoktur.

```
Örnek: başla    btfsc   portb,3
                goto  topla
                goto  bitir
        topla   addlw       b'00001111'
        bitir   end
```

İşlemden önce:PORTB=00000111 W=00000001

İşlemden sonra: PORTB=00000111 W=00010000

IORLW k: Akümülatördeki veri ile k sabit sayısını veya (OR) işlemine sokar ve sonucu akümülatöre kaydeder.

İşlem: (w).OR.(k) → w

Örnek: iorlw b'11001010'

İşlemden önce: W=10100000

İşlemden sonra: W=11101010

MOVLW k: k ile belirtilen bir sabit sayıyı akümülatöre atar.

İşlem: k → w

Örnek: movlw b'10011010'

İşlemden sonra:W=10011010

RETFIE: Çağırılan bir kesmeden çıkmayı sağlar.

İşlem: TOS → PC 1 → GIE

Örnek: retfie

Kesmeden çıktıktan sonra

PC = TOS GIE = 1

RETLW k: Bu komut return'a benzer bir işlem yapar. Yani call ile çağırılan işlemin sonunda akümülatöre k gibi bir sabiti yükler ve program call'ın bulunduğu satırın bir alt satırına dallanır.

RETURN: Bu komut call ile çağırılan işlemin sonlandırılmasında kullanılır ve program bu satırı okuduğunda işlem call'ın bulunduğu satırın bir alt satırına dallanır.Bu işlem yapılır iken akümülatördeki bilgide hiçbir değişiklik olmaz.

SLEEP: Pic' in uyku modunda çekilen akım 1μA'den daha da azdır. Herhangi bir işlem yapılmadığı istendik durumlarda pic bu modda çalıştırılarak az akım çekilmesi sağlanabilir.

SUBLW k:k gibi bir sabiti akümülatördeki veriden çıkarır, sonucu akümülatöre kaydeder.

İşlem: (w)-(k) → w

Örnek: sublw b'00000101'
İşlemden önce: W=10100101
İşlemden sonra: W=10100000

XORLW k: Akümülatördeki veri ile k sabit sayısını özel veya (EX-OR)işlemine sokar ve sonucu akümülatöre kaydeder.

İşlem: (w).EXOR.(k) → w

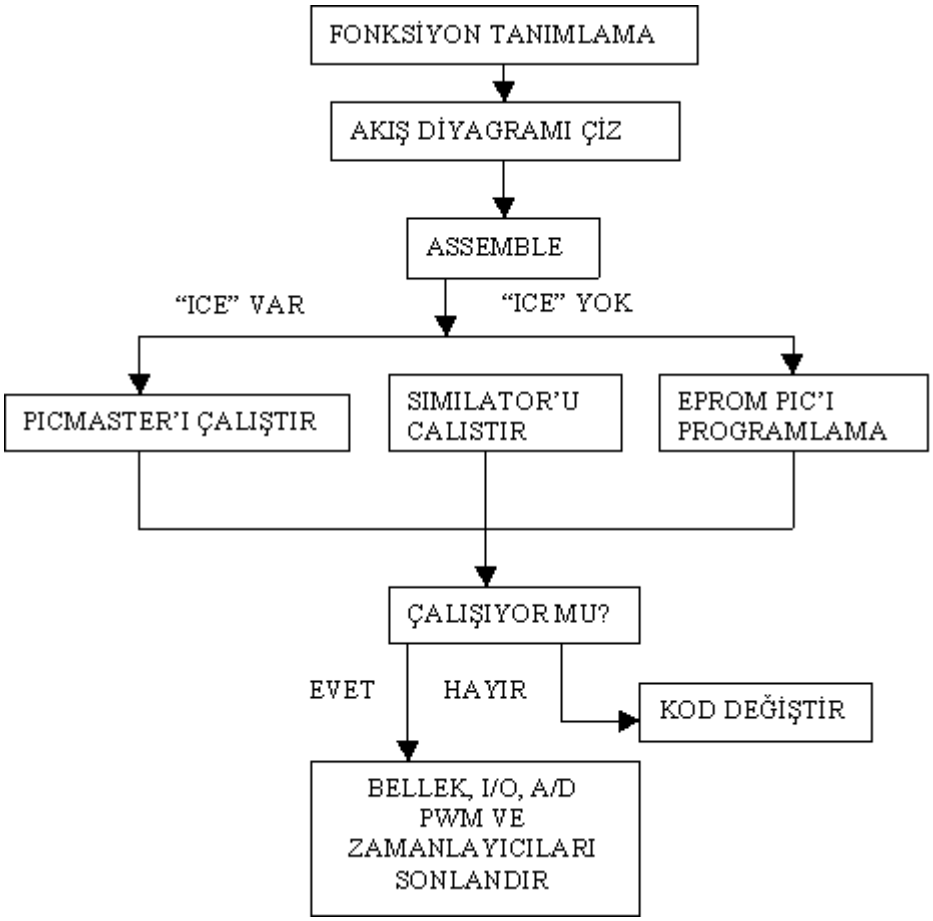
Örnek:xorlw b'00011010'
İşlemden önce:W=10111111
İşlemden sonra:W=10100101

4.5 Program Geliştirme

Yazılımı geliştirmede izlenebilecek üç yol vardır: ICE (in circuit emulator) kullanmak, programcı ve EPROM temelli bir miktar PIC kullanmak veya simülatör .

ICE kullansanız bile başarılı bir ürün geliştirme için bazı EPROM sürümlü PIC'lere ihtiyacınız olacaktır. PIC'in en az 5 EPROM sürümüyle başlayın. Sonradan bir PIC'in silinmesinin 10 – 20 dakika aldığını göreceksiniz.

Not: Aygıtlar büyüdükçe , silinme süreleri de artar.



Şekil 4.1:PIC Geliştirme Akış Diyagramı

Bundan sonraki kısımlarda PIC 16F877'nin özelliklerini kullanmaya yönelik program parçalarının üzerinde durulacaktır. Bu program parçacıkları kullanılarak daha büyük ve daha kapsamlı işlemleri gerçekleştiren programların yazılması mümkündür.

5. PIC 16F877 PROGRAM UYGULAMASI

Bu bölümde, en çok kullanılan sistemlerden biri olan analog bir verinin PIC 16F877 ile dijitale dönüştürülüp, LED displayde nasıl görüntülendiğini inceleyeceğiz.

5.1 Programın Yazılımı

```
;Analog verinin Dijitale çevrilmesi
list p=16f877
#include <p16f877.inc>
__config_cp_off&_pwrte_on&_boden_on&_xt_osc&_lvp_off&_cpd_off&_debug_off

zero        equ     =z
carry       equ     =c
azalt0      equ     h'32'
azalt1      equ     h'33'
azalt2      equ     h'34'
azalt3      equ     h'35'
ram         equ     h'36'   ;programın ilerleyen satırlarında kullanılacak geçici
                            değişkenler ve değişkenlerin yerleştirileceği hafıza
                            alanları tanımlar
soldwit     equ     h'37'
ortadigit   equ     h'38'
sağdigit    equ     h'39'
saat        equ     h'40'
sys_wsay    equ     h'41'
sys_ssay    equ     h'42'
sys_psay    equ     h'43'

org0
clrf        azalt0
clrf        azalt1
clrf        azalt2
```

```
clrf        azalt3
clrf        ram             ;geçici hafızada tanımlanan, program içerisinde
                            kullanılacak tüm değişkenlerin içeriğinin program
                            başlangıcında silinmesi
clrf        soldigit
clrf        ortadigit
clrf        sağdigit
clrf        saat
bcf         status,rp0
bcf         status,rp1      ;rp0 da 0 yapılarak bank 0 seçiliyor
movlw       h'20'
movwf       intcon
movlw       h'00'
movwf       porta           ;porta çıkış olarak ayarlandı
movlw       h'00'
movwf       portb
movlw       h'00'           ;portb çıkış olarak ayarlandı
movwf       portc
movlw       h'00'           ;portc çıkış olarak ayarlandı
movwf       portd
movlw       h'00'           ;portd çıkış olarak ayarlandı
movwf       porte
movlw       h'81'           ;porte'de 7-4-0 bitleri giriş diğer bitleri çıkış yapıldı
movwf       adcon0
bsf         status,rp0
bcf         status,rp1      ;bank1 seçiliyor
movlw       h'87'
movwf       option_reg      ;prescaler 1:128 olarak seçiliyor
movlw       h'0e'
movwf       adcon1
movlw       h'01'           ;porta nın 0 biti giriş diğerleri çıkış yapılıyor
movwf       trisa
movlw       h'00'
```

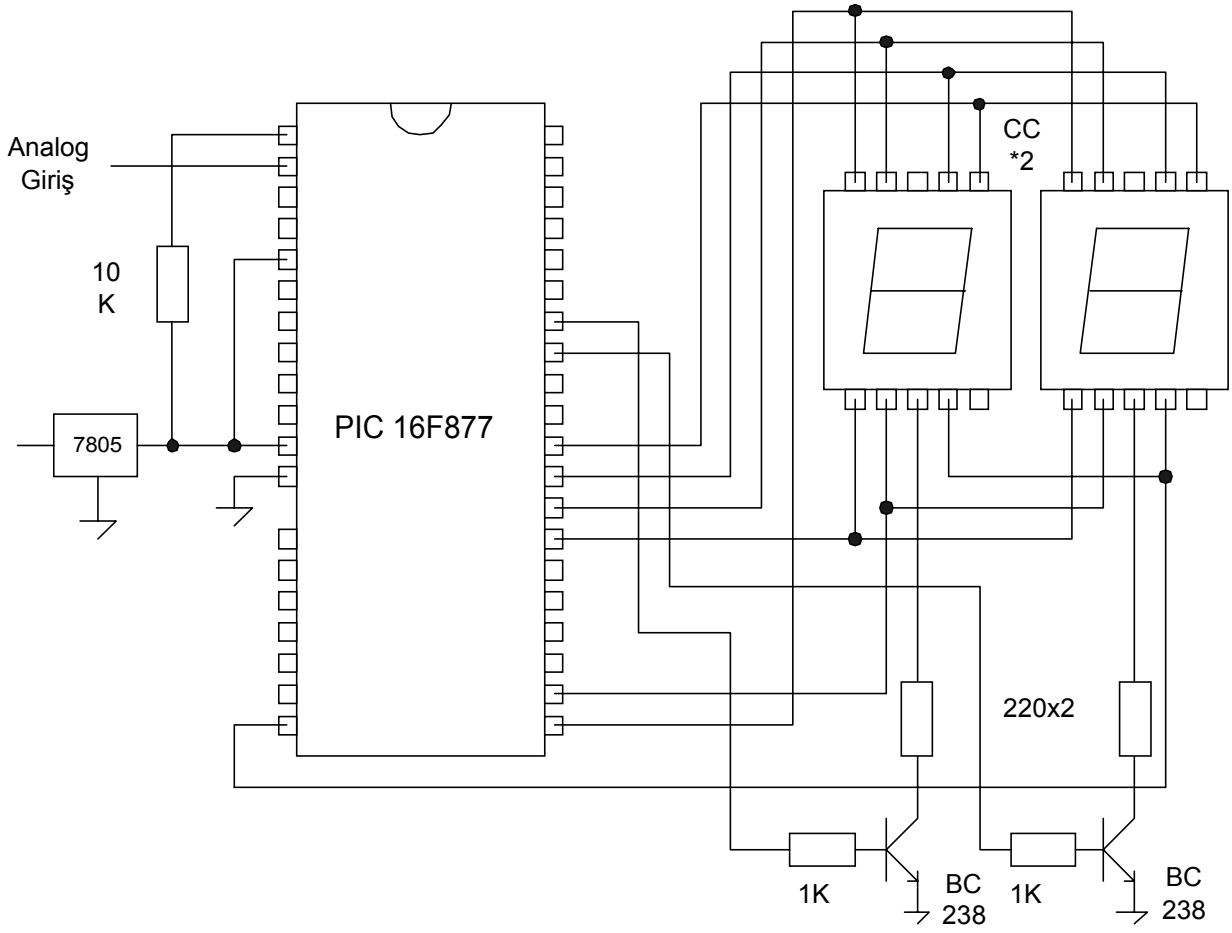
```
movwf       trisb                ;portb nin tüm pinleri çıkış yapıldı
movlw       h'00'
movwf       trisc                ; portc nin tüm pinleri çıkış yapıldı
movlw       h'00'
movwf       trisd                ;portd nin tüm pinleri çıkış yapıldı
movlw       h'00'
movwf       trise                ; portb nin tüm pinleri çıkış yapıldı
bcf         status,rp0
bcf         status,rp1           ;bank1 seçiliyor
main
yenile
        btfss       intcon,t0if
        goto        yenile
        bcf         intcon,t0if
        bsf         adcon,go     ;A/D işlemi aktif hale getirildi
wait
        btfss       pir1,adif    ;taşma kontrolü yapılır
        goto        wait
        movf        adresh,w
        movwf       sagdigit
        movlw       h'0a'
        subwf       sagdigit,w
        btfsc       status,0
        goto        kucuk10
        goto        atlason
kucuk10
        incf        soldigit,f
        movwf       sagdigit
        movlw       h'0a'
        subwf       sagdigit,w
        btfsc       status,0
        goto        kucuk10
        btfss       soldigit,4
```

```
        goto        atlason
        clrf        soldigit
atlason
        movlw       h'0a'
        movwf       saat
kesme
        movwf       sagdigit,w
call    segment                 ;segment altrutinine dallanılıyor ve karakter oluşumu
                                için başlangıç yapılıyor
        movwf       portd
        bsf         portb,0
        call        time
        bcf         portb,0
        movwf       soldijit,w
        call        segment
        movwf       portd
        bsf         portb,1
        call        time
        bcf         portb,1
        decfsz      saat,f
        goto        kesme
        clrf        soldigit
        clrf        sagdigit
        goto        yenile
segment
        addwf       pcl,f       ;bu alt rutin sayesinde digit kaydedicilerindeki
                                verilerin displerdeki karşılıkları belirlenir
        retlw       h'04'   ;0
        retlw       h'7d'   ;1
        retlw       h'23'   ;2
        retlw       h'31'   ;3
        retlw       h'51'   ;4
        retlw       h'91'   ;5
```

```
        retlw       h'c1'   ;6
        retlw       h'3d'   ;7
        retlw       h'00'   ;8
        retlw       h'19'   ;9
time
        movlw       .5              ;5 sabit değeri akümülatöre atıldı
        movwf       azalt1
dongu1
        movlw       .10             ;10 sabit değeri akümülatöre atıldı
        movwf       azalt2
dongu2
        movlw       .30             ;30 sabit değeri akümülatöre atıldı
        movwf       azalt3
dongu3
        decfsz      azalt3,f
        goto        dongu3
        decfsz      azalt2,f
        goto        dongu2
        decfsz      azalt1,f
        goto        dongu1
        return
end
```

Bu programda temel olarak dışarıdan algılanan analog bir veri; PIC 16F877'nin ADC modülünün aktif edilmesi ile dijital olarak aygıt içerisine alınır. Bu dijital verinin sayısal olarak değeri iki haneli displayde görüntülenebilecek şekilde sağ digit ve sol digit olarak iki parçaya ayrılır. Daha sonra her digite ait veriler insan gözünün algılayamayacağı bir hızla displaylere yollanır. Böylelikle dış ortamdaki analog verinin sayısal değeri displaylerle gözlenmiş olur.

5.2 Uygulama Devresine Ait Şema



Şekil 5.1: Uygulama Devresine Ait Şema

SONUÇ VE DEĞERLENDİRME

Yapılan bu çalışma ile geniş bir kullanım alanına sahip mikrodenetleyicilerin bir üyesi olan PIC aygıtları ve PIC 16F877 geniş olarak ele alınmıştır.

Yazılan bu tez ve bunun gibileri sayesinde PIC mikrodenetleyicileri hakkında Türkiye'de az bulunan Türkçe kaynakları artmaktadır.

PIC16F84 ve PIC 16F877 gibi aygıtlar kolay seri programlanma özelliklerine sahiptir. Bu özellikleri sayesinde PIC 16F84 ve PIC 16F877 dijital uygulamalarda sıkça tercih edilmektedirler. Bu aygıtların kolay temin edilmesi ise diğer bir avantajdır.

PIC aygıtlarını programlamak için birkaç alternatif vardır. Bunlardan ilki microchip veya diğer firmaların ürettiği geniş bir kütüphaneye sahip olan programlayıcılar, ikincisi ise internetten kolayca ulaşabileceğimiz ancak kullanım alanı biraz daha dar olan programlayıcılardır.

PIC 16F877'de I^2C, SPI, ADC, PWM, CPM gibi veri iletim protokollerinin yeralması; aynı anda birçok işlemin gerçekleştirilmesi ve bir tasarım içerisindeki elektronik eleman sayısının azalması açısından önemlidir.

KAYNAKLAR

1. Daldal , NİHAT Gazi Üniversitesi Teknik Eğitim Fakültesi Lisans Tezi
2. http://www.antrak.org.tr/gazete
3. http://www.rentron.com
4. Gümüşkaya, HALUK 1999, Mikroişlemciler ve 8051 Ailesi
5. http://www.pjwdesign.demon.co.uk
6. MICROCHIP 2000 Technical Library CD-ROM
7. MICROCHIP 1998 Technical Hand Book

ÖZGEÇMİŞ

29.05.1979 tarihinde Ankara'da doğdu. İlk ve orta öğrenimini Ankara'nın Mamak ve Altındağ ilçelerinde tamamladı. Lise öğrenimini ise Yıldırım Beyazıt Anadolu Meslek Lisesi Telekomünikasyon Bölümü'nde tamamladı. 1997 yılında Gazi Üniversitesi Elektronik Öğretmenliği Programına girmeye hak kazandı. Halen Gazi Üniversitesi Elektronik Öğretemenliği Programında öğrenimine devam etmektedir.

Eğitimi esnasında bilgisayar network, telefon sistemleri, asansör sistemleri gibi alanlarda faaliyet gösteren firmalarda çalışmıştır. Şu anda FULLTECH Elektronik ve Bilgisayar Tasarım Hizmetleri adıyla hizmet veren firmayı arkadaşları ile birlikte işletmektedir.