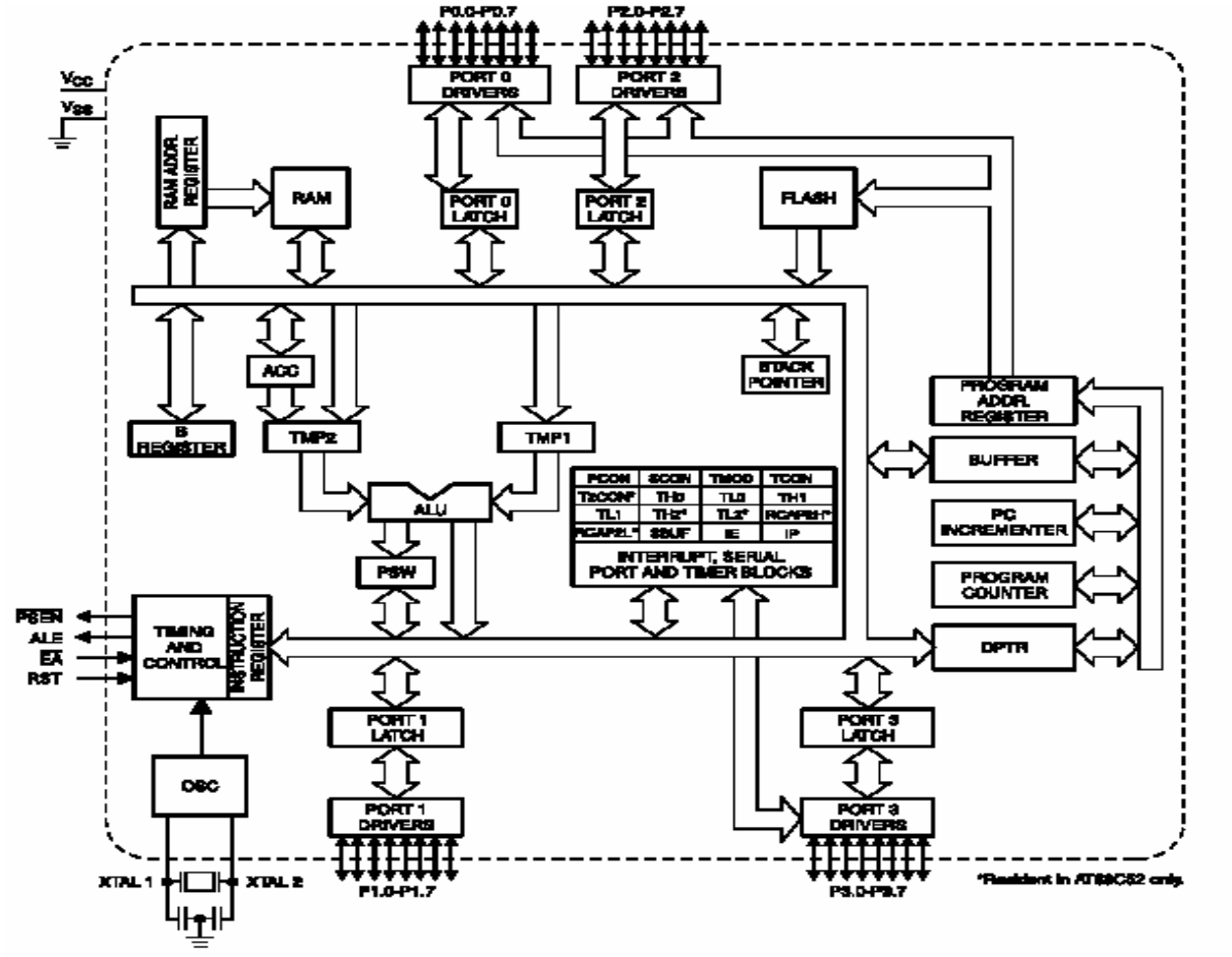
4. 8051 DONANIM ÖZELLİKLERİ



Şekil 4.1 8051 Mikrocontrollerinin mimari Blok Diyagramı

4.1. ÖZEL FONKSİYON REGİSTERLERI (Special Function Registers / SFRs)

8051 mikrodenetleyicisi, pek çok çalışma moduna sahip olan esnek bir entegredir. Programların çalışma modunu değiştirmek için özel fonksiyon registerleri değerleri okunabilir veya değiştirilebilir. Özel fonksiyon registerlerine ulaşmak aynen normal RAM data gözlerine ulaşmak gibidir. İç genel hafıza RAM 00h adresinden 7Fh adresine kadar uzanırken, özel fonksiyon registerleri 80h adresinden FFh adresine kadar uzanır. Bir önceki sayfadaki tablo özel fonksiyon registerlerinin isimlerini ve adreslerini vermektedir.

Tablo 4.1' de görülebileceği gibi, 80h adresinden FFh adresine kadar 128 veri gözü bulunduğu halde, standart 8051 entegresinde bu alanların yalnızca 21 tanesi özel fonksiyon registeri olarak kullanılmıştır. Kullanılmayan bu adreslere yazmak veya okumak beklenmedik program hatalarına sebep olabilir.

	8 Bytes								
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	(T2CON)	(T2MOD)	(RCAP2L)	(RCAP2H)	(TL2)	(TH2)			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

Tablo 4.1 SFR hafıza haritası

8051 entegresinin daha yukarı modellerinde bu gözler farklı özel fonksiyon registerleri olarak tanımlanmış olabilir. O nedenle burada gösterilemeyen register lar kullanılmamalıdır.

Adres değerleri 8 ile bölünebilen (sonu 0 veya 8 ile biten) tüm özel fonksiyon registerlerinde bit işlemleri yapılabilmektedir.

4.1.1. AKÜMÜLATÖR (ACC)

Bu tarz mikrodenetleyicilerde genel amaçlı bir registerda bulunan veri üzerinde aritmetik veya lojik işlem yapılması durumunda bu registerdaki verinin, özel bir register olan ve genellikle A, veya ACC (accumulator) kısaltmasıyla belirlenen bir register a alınması gerekir. Aritmetik ve lojik işlemlerin sadece ACC de yapıldığı bu tür 8 bit mikroişlemciler genellikle ACC tabanlı olarak tanımlanırlar.

8051 ve türevlerinde bir çok lojik ve aritmetik işlem akümülatör üzerinden gerçekleşmektedir ve buna göre bazı komutlara sahiptir.Örneğin:

```
ADD          A,#16H
```

Bu komut, işletildiğinde A nın içeriği ile 16H sayısı ile toplanarak A nın içerisine yazılır. 8051 ve türevleri bu tip birçok komuta sahiptir (MUL, DIV, SUBB, ADD, CJNE, ANL, ORL…v.b).

Akümülatöre direkt olarak ulaşılabildiği gibi bit-bit de ulaşmak mümkündür. Örneğin;

```
MOV     C,ACC.3
```

komutu icra edildiğinde A register ının içerisindeki tüm bilgi (8 bitin tümü) SAYAC olarak adlandırılmış değişkeninin içine yüklenir. Ayrıca ;

```
MOV     SAYAC,A
```

Komutuyla işlem yapıldığında akümülatörün üçüncü bitindeki sayıyı C (carry biti) içerisine yükler (C bir bitlik bir registerdır).

4.1.2. B REGİSTERI (B)

B registerı da A gibi bir register olup bir çok aritmetik işlem için kullanılmaktadır. Kullanım amacı DIV ve MUL komutlarında A registerının yanında iki carpan olarak çalışmaktadır. Ayrıca B dokuzuncu genel amaçlı bir register olarak kullanılır

```
DIV   AB
```

komutu icra edildiğinde A nın içeriğini B içeriğine bölerek, bölüm sonucunu A' nın içerisine, kalanı ise B nin içerisine yazmaktadır.

```
MUL AB
```

Komutunda ise sonuc iki byte (16 bit) olacağından yüksek byte B nin içerisine, düşük byte ise A nın içerisine yazılır.

4.1.3. PROGRAM DURUM SÖZCÜĞÜ (PROGRAM STATUS WORD / PSW)

Mikrodenetleyicinin SFR alanı içerisinde bulunur ve adresi 0D0H' dir. Bir çalışma anında CPU' nun durumunu gösteren çeşitli durum bitlerini (bayrakları-flags) içermektedir. Başka bir değişle PSW, CPU' nun akım durumunu etkileyen durum bitlerini içerir. Bunlar: Carry Biti (elde biti), Auxiliary Carry bit (yardımcı elde biti), iki adet register bank seçme bitleri, bir adet overflow (taşma biti), bir adet parity biti, iki adet kullanıcı kontrollü durum bayrak bitleridir.

Carry Biti Boolean işlemlerinde kullanılırken aritmetik işlemlerde de elde olarak kullanılır. C bayrağı sonuç FFH değerini aşarsa birlenir, aksi halde sıfırlanır. Bu bayrak ayrıca birçok boolean işleminde bir bit "ACC" olarak görev alır.

8051' de dört adet register bank bulunmaktadır. Register banklar arası geçişler yazılım ile PSW içerisindeki RS0 ve RS1 bitleri seçimiyle gerçekleşir (bakınız Tablo 4.1). Değişik komutlar bu RAM bölgesine, R0 ve R7 arasında belirtilen registerlar üzerinden erişmektedir. Dört register banktan yalnız biri bu bitlerin yürütme sırasındaki durumuna göre seçilir.

Parity biti ACC (Akümülatör)' deki birlerin sayısını yansıtır. Eğer ACC tek (Odd) sayıda birler içeriyorsa P=1 olur (Odd Parity). Eğer ACC çift (even) sayıda birler içeriyorsa, P=0 olur (Even Parity). Böylece ACC' deki birler ile P' nin toplamı her zaman çift olur.

Overflow (OV biti) ise, işaretli sayıların toplandığında veya çıkartıldıklarında oluşan bir durumu yansıtır. Taşma, işlem sonucunu hedef register'a sığmadığını gösterir. Örneğin 8-bit register'larda işaretli sayı aritmetiğinde, 07FH (+127), 01H ile toplandığında sonuç 080H (-128) olur. Bu sonuç işaretli toplama için OV bayrağı ile belirtilen bir taşma durumudur ve bu sonucun pozitif bir sayı olarak yorumlanamıyacağını belirtir. İşaretsiz işlemler için OV bayrağının bir önemi yoktur, ihmal edilir.

PSW' deki son iki bit (F0 ve kullanıcı tarafından tanımlanabilen bayrak), bir işlem için atanmadığında genel amaçlı durum bayrakları olarak kullanılabilir.

7	6	5	4	3	2	1	0
C_Y	A_C	F0	RS1	RS0	OV	---	P

BİT 7: C_y, Elde bayrağı boolean işlemlerinde kullanıla bilirken aritmetik işlemlerde de elde olarak kullanılabilir.

BİT 6: A_c, Yardımcı elde bayrağı.Örnek ile açıklarsak 8 bitlik bir toplama işleminde düşük dört bitin toplamından sonra bir elde oluşursa A_c biti 1 lenir.

BİT 5: F_0, Genel amaçlı durum bayrağıdır.

BİT 4, BİT 3: R_{S1}, R_{S0}, Register bankları seçici bitleri

BİT 2: O_V, Taşma bayrağı

BİT 1: ---, Kullanılamayan bit

BİT 0: P, Parity Bit

8051 de 4 adet register bank bulunmaktadır; register banklar arası geçişler PSW içindeki R_{S0} ve R_{S1} bitleri ile yazılım sayesinde seçilir.

RS0	RS1	Register Bloğu	Adresi
0	0	0	00-07H
0	1	1	08-0FH
1	0	2	10-17H
1	1	3	18-1FH

Tablo 4.2

4.1.4. YIĞIN İŞARETÇESİ (Stack Pointer / SP)

Yığın, on-chip RAM'e yerleştirilmiştir. İnterrupt ve subroutine çağırılması esnasında Program Counter'i tutmak için kullanılan, son-giren ilk çıkar mantığı ile çalışan bir depolama mekanizmasıdır. Aynı zamanda POP ve PUSH komutlarını kullanarak veri (data) depo etmek ve tekrar çağırmak için (PSW) kullanılabilir.Yığın işaretçisi, yığının üstündeki adresi içerir.

SP(stack pointer) on-chip RAM hafızasındaki herhangi bir adresi alabilecek 8-bitlik bir registerdır.8051'de hiçbir zaman yığın 127 değerini aşmamalıdır.Eğer aşarsa bütün aldığı verileri kaybedecektir. SP 127 değerinin üstünde ise POP geçersiz veri çağrıcaktır.

SP daima yığına atılmış olan son byte'ın adresini içerir. Power-up da 07H'ye set edilir böylece resetten sonraki yığına atılan ilk byte 08H adresinde olacaktır. Bu adres 8048'in yığını ile uyumludur. 8051 için geliştirilen hemen hemen tüm programlar yığını kullanmadan önce SP'nin içeriğini değiştirerek yığının en altını reset edecektir. Çünkü 08H-1FH adres aralığı pek çok 8051 genel kullanım amaçlı register bankları için ayrılan bir alandır. Aşağıdaki komut yığına atılan bir sonraki byte'in 100 adresine yerleştirilmesini sağlayacaktır.

```
MOV   SP,#99            ;Stack'i  100 adresinden başlatır.
                        ;donanım  SP'yi  PUSH'dan önce bir arttırır.
```

4.1.5. VERİ İŞARETÇİSİ (Data Pointer / DPTR)

DPTR registerinin yüksek değerlikli byte ı DPH (adresi 83H), düşük değerlikli byte ı ise DPL (adresi 82H) olarak adlandırılır. DPL ve DPH registerleri 16 bit bir adres oluşturmak üzere bir arada kullanılan özel fonksiyon registerleridir. Bu registerlar, 16 bitlik adres bilgisini içerir ve bu sayade harici veri hafızasına ulaşılır. Veri işaretleyici (Data Pointer), harici RAM hafıza ile veri alışverişinde ve harici kod hafıza ile ilgili bazı işlemlerde kullanılır.Veri

işaretleyici işaretsiz 16 bit bir sayıyı gösterdiğine göre, veri işaretleyici ile 0000h dan FFFFh (0-65535) adresi aralığında hafıza adreslenebilir.

Bu register iki komutta harici belleğe ulaşımı sağlar.

```
MOVX  A,@DPTR
```

komutu icra edildiğinde DPTR ile gösterilen adresteki bilgi tıpkı @R0 da olduğu gibi A' nın içerisine indirect olarak aktarılır.

```
MOVX  @DPTR,A
```

komutu icra edildiğinde A'nın içeriği DPTR ile gösterilen harici veri belleğinin adresine yazılır.

İç RAM' de FFH-128H aralığındaki sıradan adresleri çağırmak için 16-bitlik adres bilgisini kullanmak gerekmektedir. MOV komutu ile bu işlemi gerçekleştirmemiz mümkün olamadığı için DPTR üzerinden çalışan MOVC komutu kullanılır. DPTR içerisine bu komut icra edilirken adres bilgisi yazılır ve istenilen alandaki bilgi çağrılmış olur.

Akümülator gibi DPTR' de bit-bit adreslenebilir.

```
MOV DPH.5, #1B
```

Aslında DPH ve DPL özel fonksiyon veri (data) gözleri tarafından oluşturulmuş 16 bit bir sayıdır. Pek çok durumda DPTR nin DPH ve DPL baytlarını ayrı ayrı işlemek durumunda kalınabilir. Örneğin DPTR değerini yığına yazmak istediğinizi varsayalım. 16 bit bir değeri tek hamlede yığına yazacak bir 8051 komutu yoktur. Bu nedenle DPH ve DPL değerlerini ayrı ayrı yığına yazmanız gerekir. Ayrıca DPTR işaretleyicisini 1 arttıran 8051 komutu olmasına rağmen DPTR işaretleyicisini 1 azaltan bir 8051 komutu yoktur. Böyle bir durumda DPH ve DPL registerlerini kullanarak bu azaltma işlemi yapılmalıdır. Veri işaretleyicisi en çok harici veri hafıza ile veri alışverişinde kullanılır. Bu amaçla önce veri okunacak veya yazılacak hafıza adresinin değeri veri işaretleyicisine yüklenir. Daha sonra MOVX komutu ile bu değer hafızadan okunur veya hafızaya yazılır. Aşağıdaki komut seti 2300H adresine 13 değerini yazmak için kullanılmıştır.

```
MOV DPTR, #2300H;

MOV A, #13;

MOVX @DPTR, A;
```

Hafızanın 2300H adreindeki bir değeri okumak için ise

```
MOV DPTR, #2300H;

MOVX A, @DPTR;
```

komutları verilir. Harward mimarisine göre yapılan devrelerde, MOVC komutu yine veri işaretleyicisi ile kullanılarak program hafızadan veri okunmasını sağlar. Aşağıdaki komutlar program hafızanın 0300H adresinden bir verinin okunmasında kullanılmaktadır.

```
MOV A, #0;

MOV DPTR, #0300H;

MOVC A, #A+DPTR
```

4.1.6. PCON, POWER CONTROL REGİSTER (Güç Kontrol Registerı)

8051 mikrodenetleyicisinin PCON registerinin içinde sadece SMOD biti bulunmaktadır fakat MCS51 ailesinin CMOS ürünlerinde diğer bitler de vardır.

7	6	5	4	3	2	1	0
SMOD	-	-	-	GF1	GF0	PD	IDL

BİT 7: SMOD, Asenkron seri haberleşmede Baudrate timer/counter ile elde edilir. SMOD 1 olduğu durumda boudrate iki katı hızına çıkar sistem resetlendiğinde SMOD 0 ile çalışmaya başlar.

BİT 6, BİT 5, BİT 4: Kullanılmıyor.

BİT 3 ve BİT 2: GF0 ve GF1, Genel amaçlı bayrak bitleridir.

BİT 1: PD, CMOS entegrelerde, entegrenin güç tüketimini azaltmakta kullanılır.

BİT 0: IDLE, CMOS entegrelerin güç tüketim kontrolü için kullanılan diğer bir bit.

4.1.7. TCON (Zamanlayıcı Kontrolü Registerı)

Zamanlayıcı kontrolü registeri, standart 8051 entegresinde bulunan iki adet zamanlayıcının ayarlarının yapılmasında kullanılır. Bu register kullanılarak zamanlayıcılar (timerlar) çalıştırılabilir veya durdurulabilir. Bu registerde bulunan bitlerden birisi timerın taşma biti olarak kullanılmaktadır. Böylece her zamanlayıcı (timer) veya sayıcı (counter) taşmasında bu bit aktiflenmektedir. Bu özel fonksiyon registerinde bulunan bitlerden diğer birkaçı, timer ve counterın kesme üretmesi için programlanabilmektedir.

TCON registeri bit adreslenebilir bir registerdir. Bu registerin bitleri timer/counterları kontrol etmekte kullanılır.

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit 7: TF1 (Timer-1 taşma bayrağı). Zamanlayıcı (Timer) taştığı anda bu bayrak 1 (SET) olur. Mikroişlemci ilgili kesme programına sıçradığı anda bu bayrak tekrar temizlenir. Eğer kesme programı yoksa bu bayrak program tarafından temizlenmelidir. Sayıcı FFFFH değerinden 0000H değerine atladığında bir taşma bayrağı oluşur.

Bit 6: TR1 (Timer-1 çalışma kontrol biti). Timer-1' i çalıştırma bitidir. Bu bit '1' yapıldığında Timer-1 çalışmaya başlar.

Bit 5: TF0 (Timer-0 taşma bayrağı). Timer taştığı anda bu bayrak SET olur. Mikroişlemci ilgili kesme programına sıçradığı anda bu bayrak tekrar temizlenir. Eğer kesme programı yoksa bu bayrak program tarafından temizlenmelidir. Sayıcı FFFFH değerinden 0000H değerine atladığında bir taşma bayrağı oluşur.

Bit 4: TR0 (Timer-0 çalışma kontrol biti). Timer0' ı çalıştırma bitidir. Bu bit '1' yapıldığında Timer-0 saymaya başlar.

Bit 3: IE1 (Harici kesme 1 kenar bayrağı). INT1 pininde yüksekten alçağa düşen bir sinyal görüldüğünde, program INT1 kesme adresi 0013h'e sıçrar.

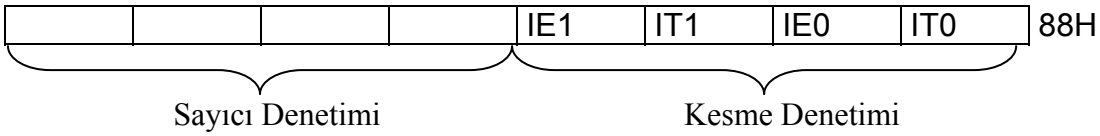
Bit 2: IT1 (Harici kesme 1 INT1 tip belirleme biti). Eğer sinyal yüksekten düşüğe geçtiğinde kesme aktiflenmesi isteniyorsa bu bit SET edilir. Bu bit 0 olduğunda pindeki 0 sinyali kesmeyi aktifler.

Bit 1: IE0 (Harici kesme 0 kenar bayrağı). INT0 pininde yüksekten alçağa düşen bir sinyal görüldüğünde, program INT0 kesme adresi 0003h'e sıçrar.

Bit 0: IT0 (Harici kesme 0 INT0 tip belirleme biti). Eğer sinyal yüksekten düşüğe geçtiğinde kesme aktiflenmesi isteniyorsa bu bit SET edilir. Bu bit 0 olduğunda pindeki 0 sinyali kesmeyi aktifler.

KESME SEVİYESİ KONTROLÜ

Harici kesme ve timer girişleri kenar veya seviye sezici olarak şartlandırılabilirler. Bu işlem TCON register'ının ilgili bitlerinin "1" veya "0" yapılması ile gerçekleştirilir.



"1" yapılan bitler kenar,"0" yapılan bitler seviye sezmeye şartlandırılmıstır. İki seçeneğin de avantajları ve dezavatajları vardır.

Seviye sezmeye şartlanmış bir kesmenini girişinde 0 görüldüğünde interrupt algılanmış olur. Bunun iki dezavatajı vardır:

Interrupt algılandıktan sonra "0" seviyesi kalkmalıdır. Aksi halde interrupt programı tekrar devreye girecektir.

Aynı şekilde bu "0" seviyesinin süresi çok kısa olursa ınterrupt'ın algılanmama tehlikesi doğar.

Bu sistemin avantajı ise aynı anda birden fazla interrupt'a hizmet verebilmesidir. Önceliği yüksek olan interrupt'ın programı bittiğinde ondan sonraki interrupt devreye girer.

Kenar sezmeli olarak şartlanmış interruptlarda yukarıdaki sorunlar olmayacaktır. Sistem interrupt girişinde düşen kenar gördüğü anda devre girer ve bir daha çalışması ise ikinci bir düşen kenarda oluşur.

4.1.8. TMOD (Zamanlayıcı Modu Registerı)

TMOD registerindeki bütün bitlerin ayrı ayrı anlamları olmasına rağmen TMOD registeri bit adreslenebilir bir register değildir. Bu register standart iki zamanlayıcının hangi modda çalışacağını kontrol etmekte kullanılan registerdir. Bu register kullanılarak zamanlayıcılar (timerlar), 16 bit timer, 8-bit tekrar yüklenen timer veya 13 bit timer olarak programlanabilmektedir. Buna ilave olarak timerlar, counter olarak ta programlanabilir. Böylece harici bir sinyalin her değişiminde timerın değeri 1 artacaktır.

TMOD' un alt dört biti Timer/Counter0' ın kontrol bitlerini içermektedir. Aynı registerin üst dört biti ise Timer/Counter1 'in kontrol bitlerini içermektedir.

7	6	5	4	3	2	1	0
Gate	C/T	M1	M0	Gate	C/T	M1	M0

Bit 7: Gate, OR kapısı enable bitidir. Zamanlayıcı1 (Timer-1)'in çalışmaya başlayabilmesi için bu bitin değerinin 0 olması gereklidir. Yani GATE biti ve TCON registerindeki TR1 biti timer-1' in çalışmaya başlamasını kontrol eder. GATE biti 1 ve TR1 biti 1 ise, timerın çalışması INT1 pinindeki sinyale bağlıdır. Bu sinyal 1 olduğu anda timer-1 çalışır. Sonuç olarak, INT1 biti ile GATE'in tersi OR kapısı ile bağlıdır. Bu şekilde, INT1 ile Timer/Counter dışarıdan kontrol edilir.

Bit 6: C/T, Sayıcı (Counter) veya zamanlayıcı (Timer) seçme bitidir. Bu bit '1' (SET) olduğu zaman timer/counter-1 sayıcı (counter) modunda çalışmaya başlar. Bu durumda T0 pinine bağlı sinyal sayılmaya başlar (chip dışında oluşan bir olay sayılabilir). Eğer '0' seçilirse Timer modunda makine periyoduna göre timer olarak çalışır.

Bit 5: M1, Timer/Counter-1 mod seçme biti

Bit 4: M0, Timer/Counter-1 mod seçme biti.

Bit 3: Gate, OR kapısı enable biti. Timer-0'in çalışmaya başlayabilmesi için bu bitin değerinin 0 olması gereklidir. Yani GATE biti ve TCON registerindeki TR0 biti timer-0' ın çalışmaya başlamasını kontrol eder. GATE biti 1 ve TR0 biti 1 ise, timerın çalışması INT0 pinindeki sinyale bağlıdır. Bu sinyal 1 olduğu anda timer-0 çalışır.

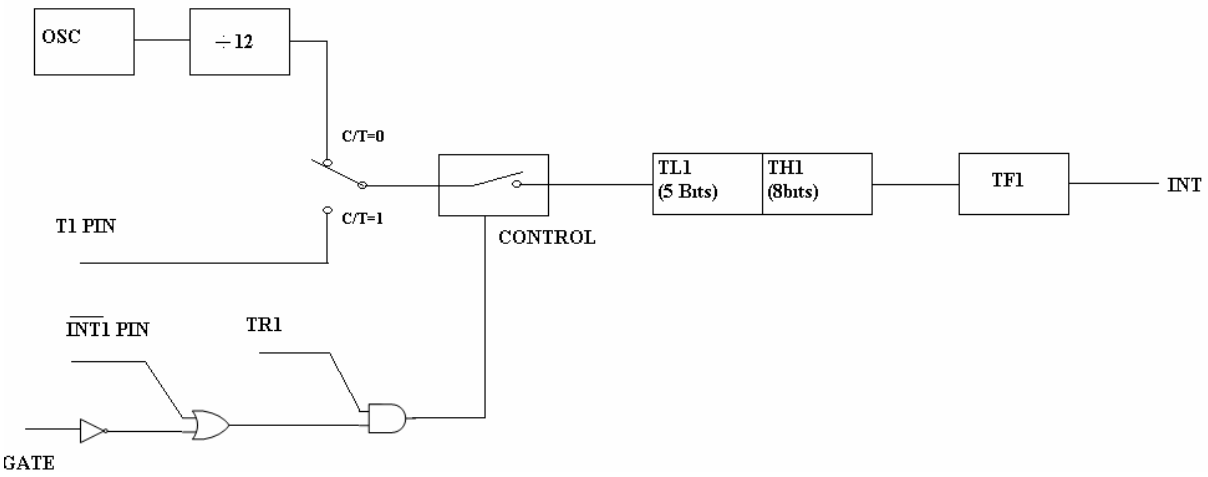
Bit 2: C/T, sayıcı veya timer seçme biti. Bu bit SET olduğu zaman timer/counter-0, sayıcı (counter) modunda çalışmaya başlar. Bu durumda T0 pinine bağlı sinyal sayılmaya başlar.

Bit 1: M1 Timer/Counter-0 mod seçme biti.

Bit 0: M0 Timer/Counter-0 mod seçme biti. Çalışma modları:

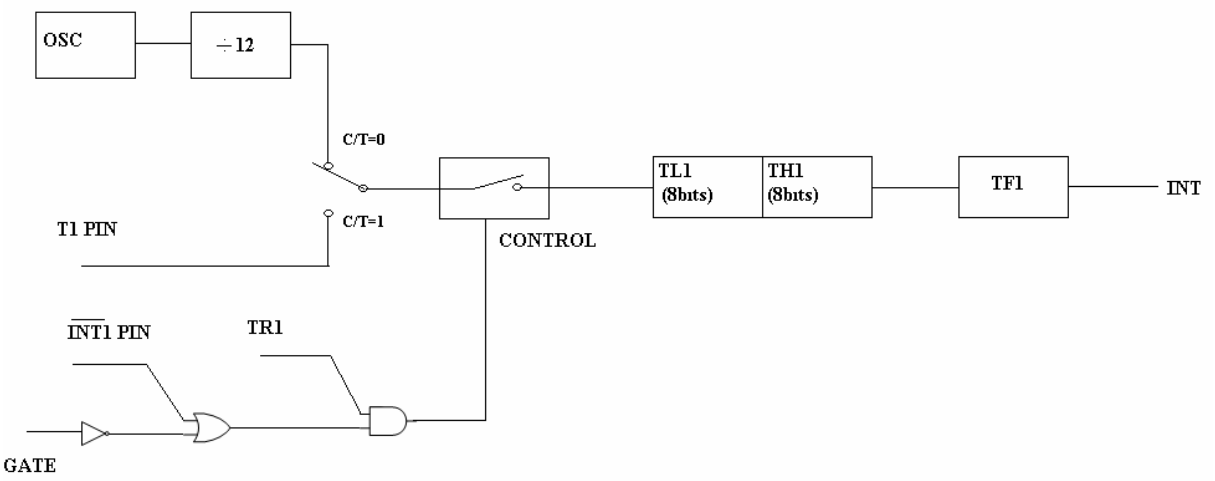
	M1	M0	
MOD0	0	0	TL_x 5 bit ön bölücü, THx 8-bit timer
MOD1	0	1	TL_x ve THx ardışık olarak 16-bit timer
MOD2	1	0	8-bit otomatik yüklemeli çalışma.
MOD3	1	1	Timer 0, TL0 ve TH0 registerları ayrı birer 8-bitlik timer olarak kullanılır.TL0 kontrolü,Timer 0 kontrol kısmından; TH0 kontrolü Timer 1 kontrol kısmından gerçekleştirilir.

MOD0'ın blok diagramı:



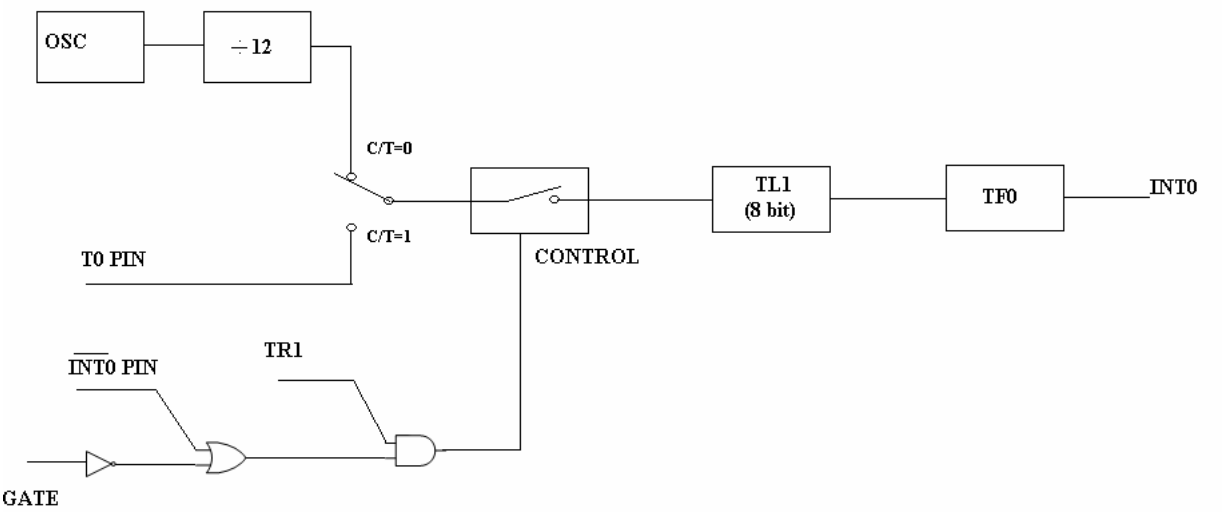
MOD0' da, 13 bit sayaç olarak çalışan counter, taştığında bir kesme üretir. Yani bu moddabir sonraki kesmeyi üretmek için 2^{13} (8192) giriş darbesi gerekir.

MOD1'ın blok diagramı:



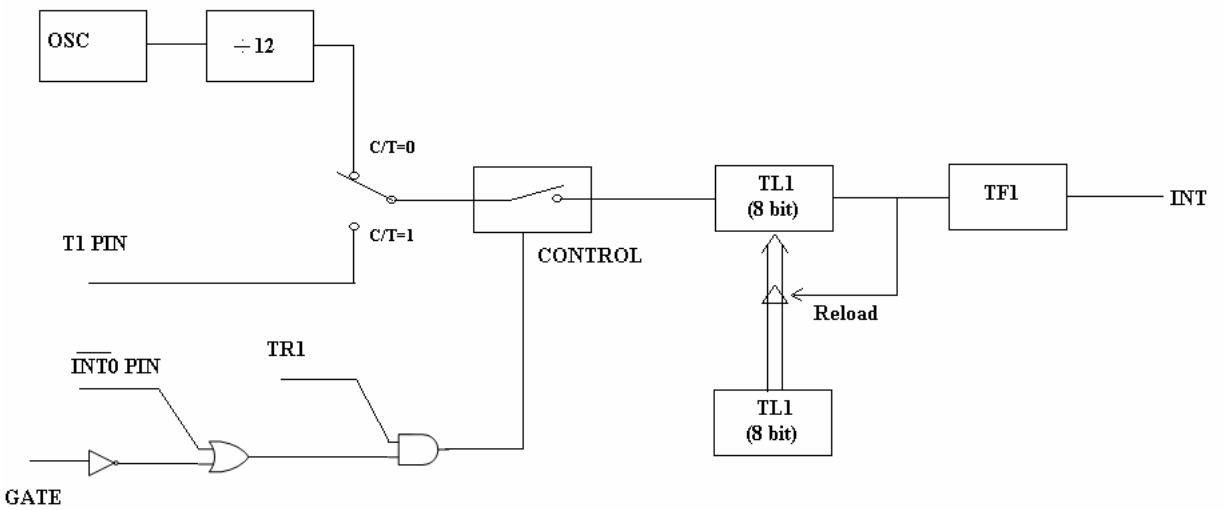
MOD1, MOD0' a benzer. Counter, 16 bit sayaç olarak çalışır.MOD1' de bir sonraki kesmeyi üretmek için 2^{16} veya 65536 giriş darbesi gerekir.

MOD2'nin blok diagramı:



MOD2, 8 bit tekrar yüklemeli tarzda çalışır. TL_i 8 bit Timer/Counter olarak çalışır. Counter taştığı zaman, TH_i 'de saklı olan değer, TL_i 'ye kopyalanır ve sayma devam eder. Counter her taştığında ve tekrar yükleme olduğunda bir kesme üretilir. (i=0 veya i=1)

MOD3'un blok diagramı:



MOD3'de, zamanlayıcı-1 (Timer-1) pasiftir ve sadece sayma değerini tutar. Timer- 0, iki ayrı 8 bit timer olarak çalışır. TL0, timer-0 kontrol bitleri ile kontrol edilir ve taşmada timer-0 kesmesi üretir. TH0, 12 ile ölçeklenmiş sistem saati tarafından sürülen zamanlayıcı olarak çalışır ve taşmada timer-1 kesmesine neden olur.

4.1.9. TL0/TH0 (TIMER-0 DÜŞÜK VE YÜKSEK)

Bu iki özel fonksiyon registeri timer-0' ı temsil eden özel fonksiyon registerleridir. Bu registerlerin nasıl davranacakları TMOD registerine yazılan kod ile konfigüre edilir. Bilinmesi gereken önemli noktalardan birisi de timerların yalnızca yukarı doğru saymasıdır. TL0 ve TH0 counterın sayma değerlerinin bulunduğu registerlerdir.

4.1.10. TL1/TH1 (TIMER-1 DÜŞÜK VE YÜKSEK)

Bu iki özel fonksiyon registeri timer-1' i temsil ederler. Timer/Counter-1'in sayma değerleri bu registerlerde bulunur. Herhangi bir anda sayma değeri okunmak istendiğinde, TL1 ve TH1 counterlarının değerleri değişkenlere okunur. Örneğin kullanacağımız iki değişken TIMERH ve TIMERL olsun.

```
MOV TIMERH, TH1;

MOV TIMERL, TL1;
```

4.1.11. P0 (PORT0, BİT ADRESLENEBİLİR)

Bu port giriş/çıkış portlarından birisidir. Bu özel fonksiyon veri gözündeki bitlerin herbiri 8051 entegresinin bacaklarından birisine karşılık gelmektedir. Örneğin port0 registerinin 0 nolu biti, entegrenin P0.0 bacağına, 7 nolu bit P0.7 nolu bacağa karşılık gelmektedir. Entegredeki P0 bacaklarından herhangi birisini 1 veya 0 yapmak için, P0

registerinde, entegre bacağına karşılık gelen bit değeri 0 veya 1 yapılır. Dolayısıyla port 0 pinleri bit adreslenebilir pinlerdir. Port 0'a yazmak veya okumak için aşağıdaki komutları kullanabilirsiniz.

```
MOV  P0, #0FH;
```

Yukarıdaki komut port 0'ın pinlerinin düşük nibbelını 1 yüksek nibbelını 0 yapacaktır. Yukarıdaki komut verildikten sonra port 0 pinlerinin değerleri aşağıdaki gibi olacaktır.

P0.0, P0.1, P0.2, P0.3 = 1 ve P0.4, P0.5, P0.6, P0.7 = 0

Bu porttan bir okuma yapmak için ise aşağıdaki komutlar verilir.

```
MOV  P0, #0FFH;

MOV  A, P0;
```

Görüldüğü gibi porttan okuma yapmadan önce, bütün port giriş moduna getirilir. Port giriş moduna getirildikten sonra ise okuma işlemi gerçekleştirilir. Yukarıdaki komutlar port 0'a 1 bayt bilgiyi yazmak veya okumakta kullanılır. Daha önce söylediğimiz gibi port 0 istenirse pin pin de kontrol edilebilir. Bit bit kontrol için kullanılabilecek 8051 assembly komutları, SETB bit , CLR bit , CPL bit, MOV bit, C, MOV C, bit komutlarıdır. Aşağıda bu komutların nasıl kullanılabileceğine dair örnekler verilmiştir.

```
SETB  P0.1        ; P0.1 pinini 1 yap.

CLR  P0.5         ; P0.5 pinini 0 yap

SETB C            ; Elde bayrağı 1

MOV  P0.4, C      ; P0.4 pinine Elde bayrağının değerini yaz.

SETB P0.7         ; P0.7 pinini giriş için hazırla.

MOV C, P0.7       ; P0.7 pininin değerini Elde bayrağına oku.
```

Daha önceden de bahsedildiği gibi harici hafıza kullanılan devrelerde port 0 düşük adres baytını ve veri alışverişini gerçekleştirmek üzere multiplex görev yapar. Adres ve veri alışverişi CPU kontrolü altındadır.

4.1.12. P1 (PORT1, BİT ADRESLENEBİLİR)

Bu port giriş/çıkış portlarından biridir. Bu özel fonksiyon veri gözündeki bitlerin herbiri 8051 entegresinin bacaklarından birisine karşılık gelmektedir. Örneğin port1 registerinin 0 nolu biti, entegrenin P1.0 bacağına, 7 nolu bit P1.7 nolu bacağa karşılık gelmektedir. Entegredeki P1 bacaklarından herhangi birisini 1 veya 0 yapmak için, P1 registerında, entegre bacağına karşılık gelen bit değeri 0 veya 1 yapılır. Dolayısıyla port 1

pinleri bit adreslenebilir pinlerdir. Port 1'e yazmak veya okumak için aşağıdaki komutları kullanabilirsiniz.

```
MOV  P1, #0FH;
```

Yukarıdaki komut port 1'ın pinlerinin düşük nibelini 1 yüksek nibelini 0 yapacaktır. Yukarıdaki komut verildikten sonra port 1 pinlerinin değerleri aşağıdaki gibi olacaktır.

P1.0, P1.1, P1.2, P1.3 = 1 ve P1.4, P1.5, P1.6, P1.7 = 0

Bu porttan bir okuma yapmak için ise aşağıdaki komutlar verilir.

```
MOV P1, #0FFH;

MOV A, P1;
```

Görüldüğü gibi porttan okuma yapmadan önce, bütün port giriş moduna getirilir. Port giriş moduna getirildikten sonra ise okuma işlemi gerçekleştirilir. Yukarıdaki komutlar port 1'e 1 bayt bilgiyi yazmak veya okumakta kullanılır. Port 1 istenirse pin pin de kontrol edilebilir. Bit bit kontrol için kullanılabilecek 8051 assembly komutları, SETB bit , CLR bit , CPL bit, MOV bit, C, MOV C, bit komutlarıdır. Aşağıda bu komutların nasıl kullanılabilecğine dair örnekler verilmiştir.

```
SETB  P1.1        ;P1.1 pinini 1 yap.

CLR  P1.5         ;P1.5 pinini 0 yap

SETB C            ;Elde bayrağı 1

MOV  P1.4, C      ;P1.4 pinine Elde bayrağının değerini yaz.

SETB P0.7         ;P0.7 pinini giriş için hazırla.

MOV C, P0.7       ;P0.7 pininin değerini Elde bayrağına oku.
```

4.1.13. P2 (PORT2, BİT ADRESLENEBİLİR)

Bu portta diğer portlar gibi bit adreslenebilir bir porttur. Harici hafıza kullanan devrelerde bu portun görevi, harici adres hattının yüksek baytını sağlamaktır. Mikroişlemci harici program hafızadan kod okuma yaptığı sırada kodun adresinin yüksek baytı bu port tarafından sağlanır. Harici veri veya kod hafızadan MOVX veya MOVC komutlarıyla veri alışverişi sırasında da bu port yine okunacak adresin yüksek baytını sağlama işlevini görür. Port 2 pinlerinin harici hafızaya nasıl bağlandığını görmek için Harward ve Von Neumann mimarisine göre 8051 devre şemalarına bakınız.

Eğer devrede harici hafıza kullanılmayacaksa, bu port da diğer portlar gibi basit giriş çıkış işlemleri için kullanılabilir. Bu portta Port1 ve Port3 gibi dahili pull-up dirençlerine sahiptir. Bu yüzden dışarıya veri yazarken dış pull-up dirençlerine gerek yoktur. Basit giriş

çıkış birimi olarak kullanıldığında, bu özel fonksiyon veri gözündeki bitlerin herbiri 8051 entegresinin bacaklarından birisine karşılık gelmektedir. Örneğin port2 registerinin 0 nolu biti, entegrenin P2.0 bacağına, 7 nolu bit P2.7 nolu bacağa karşılık gelmektedir. Entegredeki P2 bacaklarından herhangi birisini 1 veya 0 yapmak için, P2 registerinde, entegre bacağına karşılık gelen bit değeri 0 veya 1 yapılır. Port 2'ye yazmak veya okumak için aşağıdaki komutları kullanabilirsiniz.

```
MOV  P2, #0FH;
```

Bu komutla Port2'ye bir seferde 1 bayt veri yazılmıştır. Port2'den bir seferde 1 bayt veri okumak için ise aşağıdaki komutları verebilirsiniz.

```
MOV P2, #0FFH;

MOV A, P2;
```

Port 2'yi bit bit kontrol etmek için ise 8051 assembly bit komutları, SETB bit , CLR bit , CPL bit, MOV bit, C, MOV C, bit kullanılabilir.

```
SETB  P2.1        ;P2.1 pinini 1 yap.

CLR  P2.5         ;P2.5 pinini 0 yap

SETB C            ;Elde bayrağı 1

MOV  P2.4, C      ;P2.4 pinine Elde bayrağının değerini yaz.

SETB P2.7         ;P2.7 pinini giriş için hazırla.

MOV C, P2.7       ;P2.7 pininin değerini Elde bayrağına oku.
```

4.1.14. P3 (PORT3, BİT ADRESLENEBİLİR)

Bu port da diğer portlar gibi bit adreslenebilir bir porttur. Port 3 pinlerinin hepsi çift fonksiyona sahip pinlerdir. Örneğin seri iletişim için kullanılan TxD, RxD pinleri, harici hafıza okuma yazma pinleri WR ve RD, timer/counter pinleri T0 ve T1, Harici kesme pinleri INT0 ve INT1 hep bu portta bulunur. Bütün bu fonksiyonlar, kendi özel fonksiyon registerleri aracılığı ile programlanır. Bu fonksiyonların kullanılmadığı pinler basit giriş/çıkış pinleri olarak kullanılabilir. Port3 de Port1 ve Port2 gibi iç pull-up dirençlerine sahiptir. Bu sebeble bu port basit giriş çıkış için kullanıldığında, harici pull-up dirençlere gerek yoktur. Diğer portlarda olduğu gibi basit giriş/çıkış işlemleri P3 özel fonksiyon registeri aracılığı ile yapılır. Aşağıdaki komutlar P3 pinlerinin P3 özel fonksiyon registeri aracılığı ile nasıl kontrol edilebileceğini göstermektedir.

```
MOV  P3, #0FH;
```

Yukarıdaki komut verildikten sonra port 3 pinlerinin değerleri aşağıdaki gibi olacaktır.

P3.0, P3.1, P3.2, P3.3 = 1 ve P3.4, P3.5, P3.6, P3.7 = 0

Bu komutla Port3'e bir seferde 1 bayt veri yazılmıştır. Port3'den bir seferde 1 bayt veri okumak için ise aşağıdaki komutları verebilirsiniz.

```
MOV P3 #0FFH;

MOV A, P3;
```

Port 3'ü bit bit kontrol etmek için ise 8051 assembly bit komutları, SETB bit , CLR bit , CPL bit, MOV bit, C, MOV C, bit kullanılabilir.

```
SETB  P3.1        ;P3.1 pinini 1 yap.

CLR  P3.5         ;P3.5 pinini 0 yap

SETB C            ;Elde bayrağı 1

MOV  P3.4, C      ;P3.4 pinine Elde bayrağının değerini yaz.

SETB P3.7         ;P3.7 pinini giriş için hazırla.

MOV C, P3.7       ;P3.7 pininin değerini Elde bayrağına oku.
```

4.1.15. SCON (SERİ KONTROL REGİSTERI)

SCON özel fonksiyon registeri 8051 seri giriş/çıkış biriminin davranışını kontrol etmekte kullanılır. Örneğin bu register kullanılarak seri haberleşme hızı (baud rate) belirlenir. Bu registerde, aynı zamanda seri bir karakter başarıyla yollandığında veya başarıyla okunduğunda aktiflenen bayraklar bulunur. Seri port kullanılmak istendiğinde yalnızca SCON registerinin konfigüre edilmesi yeterli olmayabilir. Bu durumda TCON ve TMOD registerlerinde de gerekli değişimler yapılmalıdır. Çünkü seri haberleşme hızını ayarlamak için timerlardan birisini kullanmak gerekebilir. Aşağıda SCON registerinin yapısı ve bitleri görülmektedir.

Bit	İsim	Bit Adresi	Açıklama
7	SM0	9Fh	Seri port mod bit 0
6	SM1	9Eh	Seri port mod bit 1.
5	SM2	9Dh	Birden çok işlemci ile haberleşme modu
4	REN	9Ch	Alıcı aktif biti. Karakter okuması için bu bit aktif yapılmalıdır.
3	TB8	9Bh	Yolla biti, bit 8. Mod 2 ve 3 de karakter yollamak için kullanılan bit.
2	RB8	9Ah	Mod 2 ve 3 de karakter okumak için kullanılan bit.
1	TI	99h	Yolla bayrağı. Bir karakter yollandığı zaman bu bit aktif olur.
0	RI	98h	Alıcı bayrağı. Bir karakter okunduğu zaman bu bit aktif

SM0 VE SM1 BİTLERI VE ANLAMLARI.

SM0	SM1	Seri Mod	Açıklama	Haberleşme hızı
0	0	0	8-bit kaydırma Registeri	Osilatör frekansı / 12
0	1	1	8-bit UART	Timer-1 ile ayarlanır.(*)
1	0	2	9-bit UART	Osilatör frekansı / 32
1	1	3	9-bit UART	Timer-1 ile ayarlanır. (*)

(*) Timer-1 kullanılarak haberleşme hızı seçildiği durumda, eğer PCON.7 biti aktif yapılırsa, haberleşme hızı iki katına yükselmiş olur.

SCON registeri bit adreslenebili bir registerdir. Bu registerin bitleri isimleriyle kullanılabilir. MCS-51 assembly programlama dilinde, bu registerin bitleri tanımlanmış durumdadır.

SCON registerinin üst 4 biti (bit 4 - bit 7) konfigürasyon bitleridir.

SM0 ve SM1 bitleri seri haberleşme modunu seçmekte kullanılır. 4 farklı seri haberleşme modu vardır. Seri haberleşme modunun seçilmesiyle haberleşme hızının nasıl hesaplanacağı da belirlenmiş olur. Mode 0 ve Mod 2 de haberleşme hızı kristal frekansı ile orantılı bir sayıdır ve sabittir. Mod 1 ve 3 de ise haberleşme hızı timer-1' in birim zamanda taşma sayısına bağlı olarak değişkendir.

SM2 biti, birden fazla işlemci haberleşmesi bayrağıdır. Genelde 8051 seri hattan bir karakter okuduğunda, RI (alındı kesmesi) bayrağı aktiflenecektir. Böylece program bir karakterin alındığını ve işlenmek üzere hazır olduğunu bilecektir. Fakat SM2 biti aktiflenirse, RI bayrağı, ancak okunan 9'uncu bit "1" ise aktiflenecektir. Bu bit gelişmiş seri haberleşme olayı için zaman zaman son derece faydalıdır. Fakat biz hemen tüm programlarımızda bu biti "0" yapacağız Dolayısıyla her karakter okunuşunda, RI bayrağı aktiflenecektir.

Bir sonraki bit REN (receiver enable) alıcı aktifle bitidir. Bu bitin anlamı çok açıktır. Eğer seri port üzerinden okuma yapmak istiyorsanız bu biti aktiflemeniz gerekir.

SCON registerinin son 4 biti, seri haberleşme sırasında kullanılan işlem bitleridir. Bu bitler seri portu konfigüre etmek için değil seri yazma/okuma işlemlerini gerçekleştirmek için kullanılır.

TB8 biti mod 2 ve mod 3 de kullanılır. Mod 2 ve mod 3 de toplam 9 bit veri yollanıp okunmaktadır. İlk 8 bit karakter değeridir, 9'uncu bit TB8 den okunarak yollanır.

RB8 biti de mod 2 ve mod 3 de kullanılır. Mod 2 ve mod 3 de okuma yapıldığında toplam 9 bit veri okunur. Okunan ilk 8 bit SBUF registerine kaydedilir. 9'uncu bit RB8 bitine

okunur.

TI (transmit interrupt) yollandı kesmesi bitidir. Program seri porttan bir bilgi yolladığında, değerin seri porttan tamamen yollanması için belli bir zaman geçecektir. Eğer bu karakterin yollanması bitmeden, yeni bir karakter yollanmak üzere SBUF registerine yazılırsa, veriler birbirine karışacaktır. Bunu önlemek için yollama tamanlandı (TI) biti kullanılır. TI biti "1" ise bir önceki karakter yollanmış demektir ve yeni bir karakter yollanmasında sakınca yoktur. Dolayısıyla programcı, yeni bir karakter yollamadan önce TI bitini okuyarak kontrol etmelidir.

RI (receive interrupt) alma kesmesi bitidir. Bu bit de TI bitine benzer bir görev görmektedir. Fakat bu defa, dışardan bir karakter okunduğunda, okumanın bittiğini bildirmek üzere bu "1" olmaktadır.

4.1.16. SBUF (SERİ KONTROL)

Seri port bufferı, seri port üzerinden verilerin yazılmasında ve okunmasında kullanılır. SBUF registerine yazılan herhangi bir değer entegrenin yolla (TXD) bacağı üzerinden seri bilgi halinde yollanılır. Benzer şekilde entegreye oku (RXD) bacağı üzerinden gelen bir bilgi de SBUF registerinde okunmaya hazır bir şekilde tutulur. Yani bir karakter yollanacağı zaman SBUF registerine yazılır. Bir karakter okunacağı zaman ise SBUF registeri okunur. Aşağıda seri porta bir karakter yazılması için verilmesi gereken komutlar görülmektedir.

```
CLR TI            ;Önce TI bitini temizle.

MOV SBUF,#'A'     ;Yollanacak karakteri seri porta yaz.

JNB TI,$          ;Karakter yollamasının bitişini bekle.
```

Seri porttan bir karakter okunması için ise aşağıdaki komutlar verilmelidir.

```
JNB RI,$           ;8051 RI bayrağının aktiflenmesini bekle

MOV A,SBUF         ;Gelen karakteri akümülatöre oku.
```

4.1.17. IE (KESME İZİN REGİSTERI)

IE özel fonksiyon registeri, kesmelerin aktiflenmesinde ve pasiflenmesinde kullanılan registerdir. Bit 0-1-2-3-4-5-6 , herhangi bir özel kesmeyi aktiflemekte veya pasiflemekte kullanılan bitlerdir. Bit 7 tüm kesmeleri aktiflemek ve pasiflemekte kullanılır. Bu bit pasiflendiğinde, herhangi bir özel kesmeyi aktifleyecek bit 1 olsa bile, kesme pasif kalır.

7	6	5	4	3	2	1	0
EA	---	ET2	ES	ET1	EX1	ET0	EX0

BİT 7 : EA, genel kesme izin biti.Herhangi bir kesmeyi aktif etmek için öncelikle bu bitin aktif edilip daha sonra istenilen kesmenin özel izin biti aktif edilmelidir.

BİT 5 : ET2, sadece 8052'de bulunan Timer-2 aktif etme bitidir.

BİT 4 : ES, seri port kapısının özel izin bitidir.Seri portun kesmesinin (interrupt ının) açılması için "1" yapılmalıdır.

BİT 3: ET1, Timer-1'in kesmesinin özel izin bitidir. Timer-1 kesmesini aktif etmek için bu bit "1" yapılmalıdır.

BİT 2: EX1, Harici kesme-1'in özel izin bitidir. Harici kesme -1'i aktif etmek için "1" yapılır.

BİT 1 : ET0, Timer-1'in kesmesinin özel izin bitidir. Timer-0'in kesmesini aktif etmek için bu bit "1" yapılmalıdır.

BİT 0 : EX0, Harici kesme-0'in özel izin bitidir. Harici kesme -0'i aktif etmek için "1" yapılır.

İNTERRUPTLARDA SEVİYE BAZINDA ÖNCELİK :

Aynı düzeydeki öncelik durumunun amacı aynı düzeydeki öncelik seviyesinde eşzamanlı kesmelerin ayrımıdır.

Mikrodenetleyici sistemine aynı anda birden fazla kesme geldiğinde hangisinin öncelikli olduğu öncelik kaydedicisine bakılarak anlaşılır.İlgili kesmenin biti bir ise öncelik düzeyi yüksek,0 ise düşüktür.

En yüksekten düşük seviyeye öncelik seviyesi;

IE0
TF0
IE1
TF1
RI or TI
TF2 or EXF2

Herhangi bir kesmeyi aktiflemek için, once hangi kesme aktifleneckse o bit SET edilir. Daha sonra EA biti SET edilir. Örneğin Harici kesme 0'ın çalışmaya başlaması için,

```
SETB EX0;

SETB EA;
```

komutları verilir. Bazı yerlerde bit kontrolü yerine bayt seviyesinde komutlar da görebilirsiniz. Fakat program okunulabilirliği açısından yine de bit komutları tercih

edilmelidir. Yukarıdaki 2 satır kodu, örneğin;

```
MOV IE, #01H ;

ORL IE, #080H;
```

veya,

```
ORL IE, #01H;

ORL IE, #080H;
```

Veya başka formlarda görebilirsiniz. Bunların hepsinin aynı işi yaptığına dikkat ediniz.

Bit	İsim	Bit Adresi	Açıklama
7	EA	AFh	Genel kesme aktifle/engelle
6	-	AEh	Tanımsız
5	-	ADh	Tanımsız
4	ES	ACh	Seri kesme aktifle
3	ET1	ABh	Timer-1 kesmesi aktifle
2	EX1	AAh	Harici kesme 1 aktifle
1	ET0	A9h	Timer-0 kesme aktifle
0	EX0	A8h	Harici kesme 0 aktifle

4.1.18. IP (Kesme Öncelik Registerı, Bit Adreslenebilr)

IP özel fonksiyon registeri kesme önceliklerini tanımlamakta kullanılır. Birden çok kesme yaratan bir sistemde, aynı anda oluşan iki kesmeden hangisinin daha önce işleme sokulacağını belirlemek gereklidir. 8051 entegresinde bir kesme ya düşük önceliğe (0) yada yüksek önceliğe (1) sahip olarak tanımlanabilir. Örneğin birden fazla kesme kullanan bir sistemde, seri port kesmesini yüksek, diğer tüm kesmeleri düşük öncelikli olarak tanımladığımızı varsayalım. Bu durumda, her bir seri port kesmesi, kesme altprogramının çalışmasını sağlayacaktır. Seri port kesmesi sırasında, başka bir kesme aktif olsa bile önce seri port kesmesi programı tamamlanacaktır. Aşağıda IP registerinin bitleri görülmektedir. Herhangi bir kesmeyi öncelikli yapmak için o kesmenin bitini SET etmek yeterlidir. Örneğin seri port kesmesinin diğer kesmelerden daha öncelikli olmasını istiyorsak,

```
SETB PS;
```

komutunu vermek yeterlidir.

7	6	5	4	3	2	1	0
---	---	PT2	PS	PT1	PX1	PT0	PX0

BİT 5: PT2, Timer-2 kesme önceliğini belirler.

BİT 4: PS, seri portun kesme önceliğini belirler.

BİT 3: PT1, Timer-1'in kesme önceliğini belirler.

BİT 2: PX1, Harici Kesme-1'in önceliğini belirler.

BİT 1: PT0, Timer-0 kesme önceliğini belirler.

BİT 0: PX0, Harici Kesme-0'ın önceliğini belirler.

Bit	İsim	Bit Adresi	Açıklama
7	-	-	Tanımsız
6	-	-	Tanımsız
5	-	-	Tanımsız
4	PS	BCh	Seri port önceliği
3	PT1	BBh	Timer-1 önceliği
2	PX1	Bah	Harici kesme-1 önceliği
1	PT0	B9h	Timer-0 önceliği
0	PX0	B8h	Harici kesme- 0 önceliği

Şekil 4.2 Kesme önceliği registeri bitleri ve anlamları.

4.2. PORT YAPILARI VE KULLANIMLARI

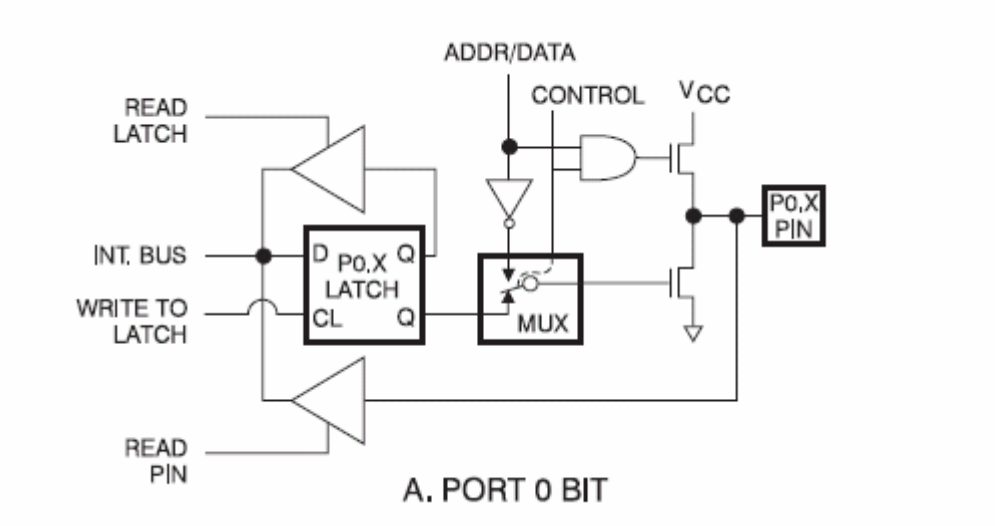
GENEL AMAÇLI GİRİŞ/ÇIKIŞ PORTLARI, ÖZEL FONKSİYONLU PORTLAR, ADRES VE VERİ YOLUNU SÜREN PORTLAR

Giriş - Çıkış Portları

8051 giriş - çıkış port mimarisinde çift yönlüdür ve oldukça esnektir. 8051'de 32 giriş-çıkış pini, 4 tane sekizli paralel port olarak konfigüre edilmiştir. Her pin yazılım kontrolünde giriş veya çıkış olarak ayarlanabilir. Bu giriş-çıkış pinlerine program çalışması esnasında direkt olarak hafıza komutları ile erişilebilmektedir.

4.2.1. PORT 0 YAPISI

Port-0 normal port işlevinden başka,çevre birimler kullanıldığında adres bilgisinin alt 8-bitini(LSB) gönderme ve veri yolu için kullanılır. Bu sebepten dolayı port-0'ın yapısı diğerlerinden biraz daha karışıktır.



A. PORT 0 BIT

Şekilde görüldüğü gibi port-0'ın normal bir port veya adres/veri yolu olarak kullanımına KONTROL girişi ile karar verilir. KONTROL girişi lojik sıfır olduğunda normal bir giriş/çıkış birimi olarak çalışır.fakat diğer portlardan bir farkı Pull-Up devresinin olmamasıdır. Bu nedenle Port-0 çıkışı TTL uyumlu değildir, bu portun TTL uyumlu olması için çıkışılarına dışarıda Pull-Up dirençleri bağlanması gerekmektedir.

KONTROL girişi lojik "1" olduğunda ise çıkış adres/veri yoluna bağlanmış olur. Bu durumda önce adres bilgisinin alt 8-biti(LSB) porttan çıkacak, bu bilgini dışarıda tutucu tarafından ALE sinyali vasıtasıyla tutulmasından sonra, port veri yolu olarak kullanılacaktır. Bu modda çalışırken KONTROL=1 ve Adres/Veri=1 olursa AND (VE) kapısının çıkışı lojik "1" olacak ve Q_1 FETi iletime girecektir.Bu anda eviricinin çıkışı lojik sıfır olduğunda Q_2 kesime gidecek ve çıkış lojik 1 olacaktır. Bu çalışma şeklide FET'lerin Simetrik (Push-Pull) Bağlanması ile gerçekleşir.

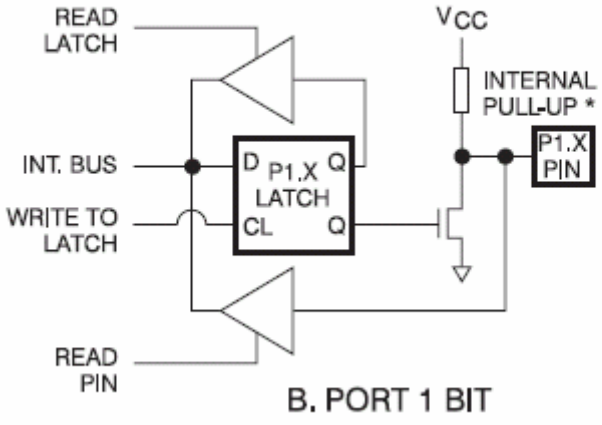
4.2.2. PORT 1 YAPISI

Bu arada iki temel unsur,tutucu (D-flip-flop) ve pull-up devreleridir. Tutucu devre porta yazılan veya okunan bilginin, diğer bilgi gelinceye kadar tutulmasını sağlar.Yukarı tutucu devre ise porta lojik "1" yazıldığında çıkışta yüksek direnç göstermesi için konulmuştur. Bu devrenin dirençlerle değil de, FET'ler ile yapılmasının sebebi ise; dirençlere göre FET'lerin, chip üzerinde daha az yer kaplamasıdır.Sabit bölümü oluşturan DFET 80 µA, 0'dan 1'e geçişi hızlandıran EFET ise 2 saat periyodu süresince DFET'in yüz katı akım akıtmaktadır.

PORT 1'e bilgi yazılması :

Portun herhangi bir bitine bir yazıldığında ,bu bilgi tutucu tarafından tutulur. Aynı zamanda Q' ucu sıfır olur. Bu ucun sıfır olması Q_2 FET'ini açık devre yapar. Pull-Up devresi vasıtasıyla V_{CC}'ye çekilerek lojik 1 yapılır.Bu durumda pulseup devresi üzerinden 80µA akım akar.

Aynı bite lojik sıfır yazıldığında Q ucu lojik bir olur ve Q_2 FET'i iletime geçerek çıkışı toprağa çekip lojik sıfır yapar. Bu durumda çıkışta Q_2 FET' i üzerinden içeriye doğru 1.6 mA akım akar. Bu akımlar arasındaki simetrik olmayan durum sistemin TTL olmasını sağlamaktadır.



Porttan Bilgi Okunması:

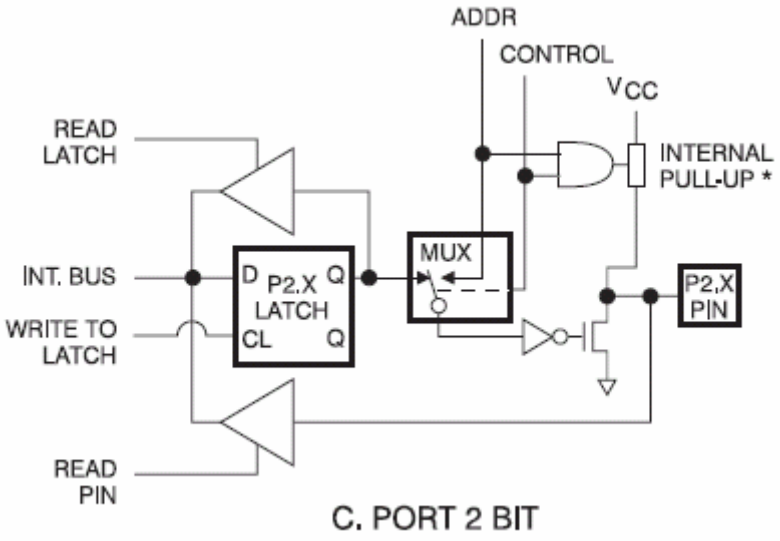
Portlar vasıtasıyla dışarıdan bilgi okunurken dikkat edilmesi gereken nokta; okuma yapmadan önce, bu portun bir yapılmasıdır.Bunun nedeni; port, lojik "0" da bırakıldığında çıkış doğrudan toprağa bağlanmış olacağından dolayı gelen bilgi bozulacak ve doğru bir okuma yapılamayacaktır.

Sistem resetlendiği anda tüm portların çıkışı lojik "1" olur.

Porta girilen bilgi bir bufferdan geçerek iç veri hattına çıkar. Aynı zamanda tutucuya da yazılarak ikinci bilgi gelinceye kadar, hatta kalması sağlanır.

4.2.3. PORT 2

PORT 2'nin normal bir giriş/çkış kapısı olmasından başka diğer bir görevi de; harici belleğe ulaşmada gerekli olan adres bilgisinin üst 8-bitini (MSB) taşımasıdır.Bu işlem bir anahtarlama düzeneği ile sağlanmaktadır. Bu sistem FET'lerden oluşturulmuştur.



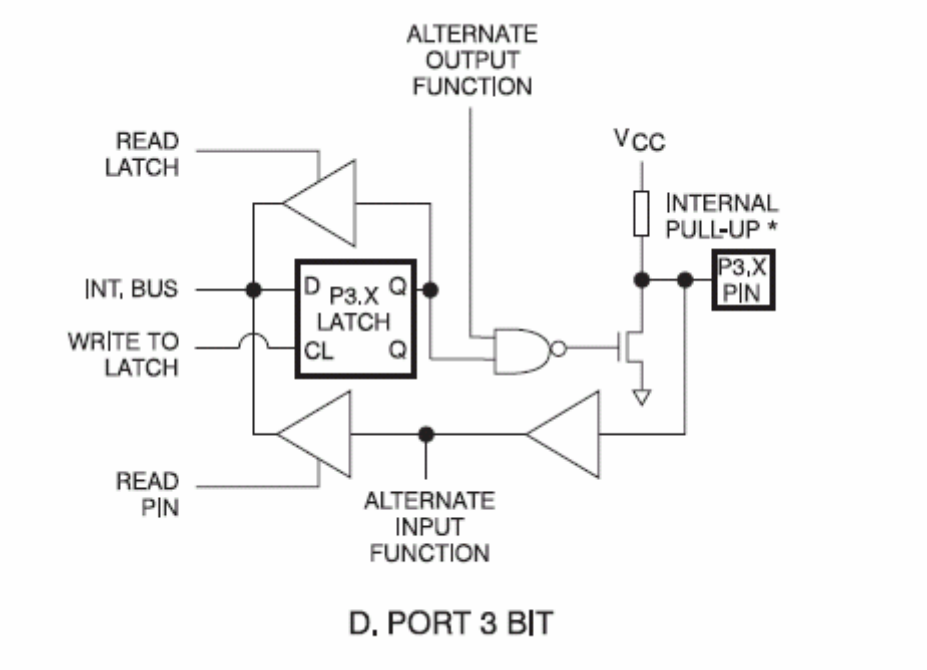
”

Harici belleğe ulaşılmak istenildiğinde, yani MOVX <Hedef >,<Kaynak> komutu kullanıldığında otomatik olarak denetim ucu yoluyla port2 iç adres yoluna bağlanmış olur. Bu kapının port1 den diğer bir farkı ise; tutucunu Q çıkışında yararlanılmış ve anahtarlamadan sonra bir evirici kullanılmış olmasıdır.

4.2.4. PORT 3 YAPISI

Port3'ün, port2 gibi ikinci bir görevi daha vardır.Bu görevi ise,sistemin dış dünya ile haberleşmesi için gerekli olan kontrol uçlarının bu port üzerinde olmasıdır.Bu uçlar ve görevleri ise şunlardır:

Port-3.0	RxD	Seri bilgi alma ucu
Port-3.1	TxD	Seri bilgi gönderme ucu
Port-3.2	INT0	Harici Kesme-0
Port-3.3	INT1	Harici Kesme-1
Port-3.4	T0	Timer/Counter-0 girişi
Port-3.5	T1	Timer/Counter-1 girişi
Port-3.6	WR'	Harici RAM Belleğine yazma denetim çıkışı
Port-3.7	RD'	Harici RAM belleğinden okuma denetim çıkışı



D. PORT 3 BIT

Port3'ün Port2 den tek farkı, anahtarlama ve evirici yerine bir NAND (Ve Değil) kapısı kullanılmış olmasıdır. Bu porttan bilgi okunurken port1 ve 2 de olduğu gibi, çıkış lojik bir olmalıdır. Aksi halde seçenek işlevi bilgisi NAND kapısı çıkışında kendini gösteremeyecektir.

4.3. ZAMANLAYICI / SAYICILAR (TIMER / COUNTERS)

Klasik 8051 çekirdeği 2 tane dahili 16 bit Timer/Counter' a sahiptir. Herbir timer/counter kendisine ait 2 adet SFR ' ye sahiptir. Timer/Counter1 için TH1 ve TL1, Timer/Counter0 için TH0 ve TL0 registerleridir.

Bu 8 bit registerlar birleştirildiklerinde, Timer/Counterları 16 bit yapar. Çalışma esnesında, herbir giriş darbesinde bir veya iki SFR' de saklı olan sayma (count) değeri arttırılır.
8051' de iki adet olan bu Timer/Counterlar hem zamanlayıcı hem de sayıcı olarak işletilmek için düzenlenmişlerdir.

Bir zamanlayıcı (timer) olarak register her makine saykılında arttırılır. Yani register makine saykıllarını sayar. Eğer bir makine saykılı 12 osilatör periyodundan oluşuyorsa, sayma oranı osilatör frekansının 1/12' sidir. Zamanlayıcı sistem saatini giriş darbelerinin kaynağı olarak kullanır. Yani timer arttırılması periyodiktir. Buna karşın counter harici darbeleri sayma değerini arttırmak için kullanır. Harici sayma darbeleri, Port3' ün iki biti üzerinden (

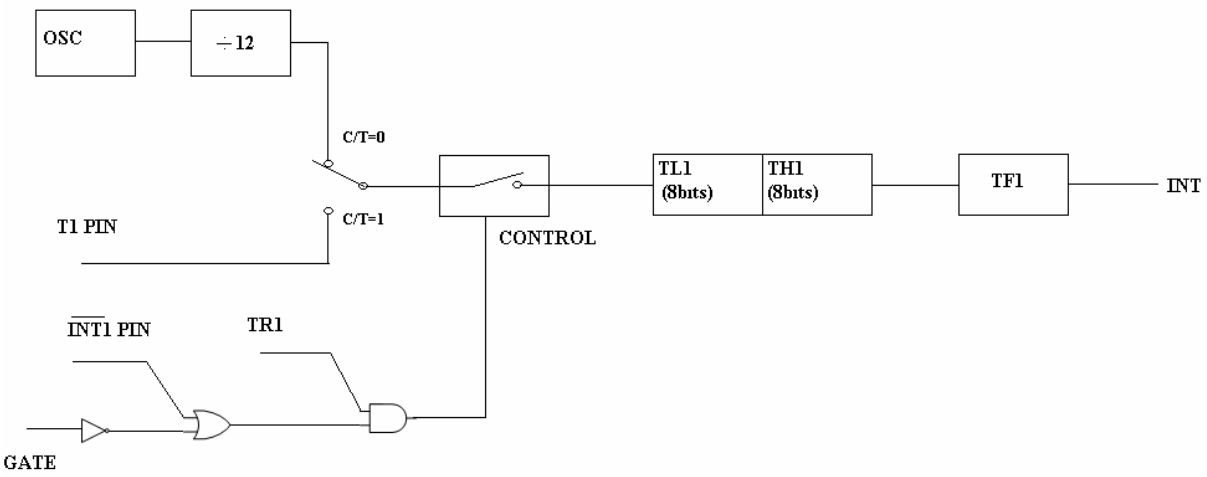
P3.4(T0) veya P3.5(T1)) alınabilir. Giriş osilatör frekansının 1/12' si veya Port3 uçlarından Timer0 için T0; Timer1 için T1 kullanılabilir.

Bir Timer/Counterların bir andaki sayma değeri, TH_i ve TL_i ' den okunabilir (i=0 veya 1). Timer/Counterlar kesme üretebilir. Bunun için ilgili kesmelerden IE (Interrupt Enable) bayraklarının aktif olması gerekir. Kesmeli çalışma modunda, sayma değeri taştığı zaman (0'dan tekrar başladığında) bir kesme üretilir. Ilgili kesmelerin durumu SFR TCON registerinden gözlenebilir.

Bir 8051 Timer/Counter' ın çalışmasını açıklamak için, işlemlerini 3 ayrı bölüme ayırıcak olursak:

1-) Giriş Kaynaklarını Kontrol
2-) Çalışma Kontrol
3-) Yenileme Modu

Giriş, osilatör frekansının 1/12' si veya Port3 uçlarından T0 (P3.4), Timer0 için ve T1 (P3.5), Timer1 için olabilir. Giriş kaynağı TMOD registerinin C/T kontrol biti ile seçilir. Gerçekte her Zamanlayıcı (Timer) için bir tane olmak üzere, topla 2 tane kontrol biti vardır.



Şekil 4.3: Timer-0'ın basit blok diagramı aşağıda verilmiştir;

Zamanlaycı/Sayıcı (Timer/Counter) işlemi, kaynaktan sayma veri registerlarına (TH0,TL0,TH1,TL1) gelen darbeleri bloklayarak veya geçirerek kontrol edilir. Örneğin Timer 0 veri registerlerine saat kaynağının gidebilmesi için aşağıdaki durumun lojik 1 olması gerekmektedir.

(GATE' OR INT0') AND TR0

GATE', TMOD registerinin Timer-0 kontrol bitidir. TR0, TCON registerlerinde bulunan diğer Timer-0 kontrol ucudur. INT0', 8051' in P3.2 ucudur. Yukarıdaki boolean eşitliğinin ifade ettiği durum, Timer-0 veri registerlerinin; TR0 kontrol biti 1 ve GATE'

temizlendiğinde (0) veya P3.2 lojik seviye 1 olduğunda, giriş kaynağına göre yenileneceğini belirtir.

TR0, Timer- 0' ı açıp/kapayan (on/off), diğer durumlardan bağımsız, ana yazılım anahtarı olarak yorumlanabilir. TR0, 1 yapıldıktan sonra, Timer-0, yazılım tarafından GATE' kontrol biti temizlenerek veya donanım tarafından P3.2 lojik 1 seviyesine getirilerek çalıştırılabilir.

4.4. STANDART SERİ HABERLEŞME ARABİRİMİ (UART)

8051 entegrenin güçlü özelliklerinden birisi de yapısında bulundurduğu seri haberleşme birimidir (UART). 8051 entegrenin seri bir haberleşme birimine sahip olması demek, 8051 ile standart RS-232 seri haberleşmesinin yapılabilmesidir. Böyle bir birim entegre içerisinde bir parça olarak verilmeseydi, seri bir hattan bilgi yollayıp okumak oldukça uğraştırıcı bir iş olarak karşımıza çıkacaktı. Seri bir hattan bilgi yollamak için bit değerini entegrenin bir bacağına yazmak, daha sonra dikkatli bir zamanlama kullanarak sırasıyla bitleri hatta yerleştirmek gerekecekti. Bir pin üzerinden seri bilgi okunması daha da külfetli bir iş olarak karşımıza çıkacaktı. Yollanan veya okunan her bir karakterin, başlangıç biti (start bit), bitiş biti veya bitleri (stop bit), parity kontrolünün 8051 programcısı tarafından yapılması gerekecekti. Fakat günümüz mikrodenetleyicilerinin çoğunda seri haberleşme biriminin bulunması bu sorunu gidermiştir.

Entegrenin seri portunu kullanmak için yapmamız gereken yalnızca bir kaç işlem vardır. Bunun için entegrenin seri port iletişim modunu ve seri port haberleşme hızını seçmemiz gereklidir. Bu işlemler yapıldıktan sonra yapmamız gereken tek şey yazacağımız karakteri SBUF özel fonksiyon registerine yazmak ve okuyacağımız karakteri SBUF registerinden okumaktır. 8051 bir karakteri yolladığında bize karakteri yolladığını, veya seri porttan bir karakter okunduğunda okunan bir karakterin olduğunu bize bildirecektir. Böylece programlayıcı karakterin bit bazında gönderilme işleminin sıkıntılarından kurtulmuş olmaktadır.

Seri port asenkron iletişim, bir start biti, bir veya daha fazla stop biti kullanır.

Şekil 4.4

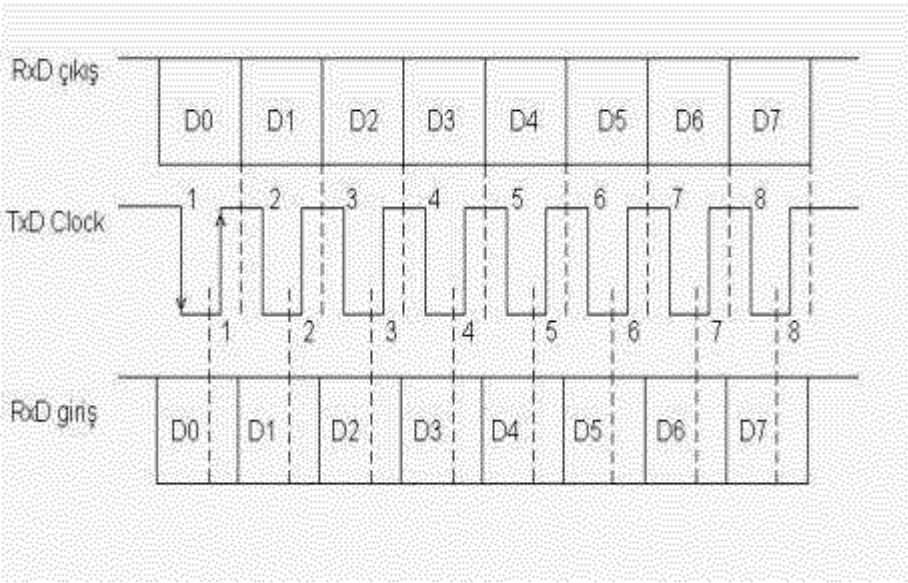
4.4.1. SERİ HABERLEŞME VE AĞ YAPILARI

Bir dizi bilgisayarı veya bir dizi işlemciyi birbirine bağlamak için en uygun yöntem olarak bir kaç temel yöntem vardır. Bunlardan en önemlileri, yıldız (star) ve loop (döngü) bağlama yöntemleridir.

Yıldız bağlama yönteminde ana kontrolcüden birer hat tüm diğer ikincil kontrolcülere götürülür. Bu yöntem de tüm kontrol kararı ana kontrolcü tarafından verilir ve her bir ikincil kontrolcü yalnızca ana kontrolcü ile iletişimde bulunur.

Döngü (loop) bağlama yönteminde ise bir kontrol hattı ile ana kontrolcü tüm diğer kontrolcülere bağlanır. Hangi yöntemin seçileceği bu işi düzenleyen kişinin tercihine ve pek çok dış etkene bağlı olabilir. Temel olarak yıldız bağlama yöntemi güvenilirlik ve hata bulma açısından döngü yöntemine göre daha üstündür. Döngü yöntemi ise maliyet ve kablo sayısı yönünden üstündür. Yıldız yöntemi eğer ikincil kontrolcülerin sayısı azsa ve ana kontrolcü ile ikincil kontrolcüler arasındaki mesafe azsa tercih edilen yöntem olabilir. Fakat ikincil kontrolcülerin sayısı ve bu kontrolcülerin ana kontrolcüye olan uzaklığı arttıkça tercih döngü yöntemine doğru gider.

4.4.2. SERİ VERİ MODU 0



8051 entegrenin seri haberleşme biriminin Seri veri modu 0'a getirilmesi için SM0 ve SM1 bitleri 0 yapılır. Seri veri modu 0'da haberleşme hızı sabittir ve osilatör frekansının 12 de birine eşittir. 12Mhz kristal kullanılan bir 8051 devresinde bu haberleşme hızı 1 Megabyte/s dir. Bu haberleşme modunda RxD ucu hem karekterlerin okunmasında, hem de karakterlerin yollanmasında kullanılır. Entegre seri haberleşme modu 0'a getirildiğinde, TxD pininde kristal frekansının 1/12 sinde kare dalga sinyal bulunur. Bu sinyal yalnızca haberleşme sırasında aktif duruma gelir. Hatta haberleşme olmadığı zaman, TxD hattı 1 seviyesinde bulunur. Entegreden veri yollanırken, TxD hattının ilk yükselmesinden, 1 clock63

sonra yollanacak karakterin ilk biti RxD hattına kaydırılır. Daha sonra TxD hattının her yükselişinden 1 clock sonra diğer bitler RxD hattına konularak karakter hattan yollanmış olur. Entegreye veri okunurken de, karakterler RxD hattından okunur. TxD hattında ise yine senkronizasyon sinyali bulunur. Gelen bitler TxD hattının düşen kenarından 2 veya 3 clock sonra seri buffer'a (SBUF) alınır. Mod 0 bilgisayar ve 8051 arasında haberleşmeden daha ziyade 8051 ve mod 0 benzeri haberleşme kullanan diğer entegrelerle veya diğer 8051 işlemcilerle haberleşme için kullanılmak üzere düşünülmüş bir haberleşme modudur.

4.4.3. SERİ VERİ MODU 1

Seri port konfigüre edilirken yukarda da belirtildiği gibi program seri port haberleşme hızını belirleyecek düzenlemeleri yapmalıdır. Yalnızca mod 1 ve mod 3 de seri port haberleşme hızının konfigürasyonuna gerek vardır. Mod 2 ve mod 4 seri port haberleşme hızını direkt olarak kristal frekansından ürettiğinden bu modlarda haberleşme hızı ile ilgili işlem yapmaya gerek yoktur.

Mod 1 ve Mod 3 de seri port haberleşme hızı timer-1 taşmalarıyla belirlenir. Timer-1 çok sık taşıyorsa, haberleşme hızı yüksek, daha seyrek taşıyorsa haberleşme hızı daha düşük olacaktır. Timer-1' i haberleşme hızını belirlemek üzere çok farklı biçimde programlama olanağı vardır. Fakat en çok kullanılan metod, timer-1' i 8 bit otomatik yükle modunda (timer modu 2 -reload) kullanmak ve TH1 registerine gerekli tekrar yükleme değerini kaydetmektir. TH1 registerine yüklenecek değeri bulmak için (PCON.7 = 0) iken aşağıdaki formülü kullanmamız gerekmektedir.

TH1 = 256 - ((Kristal frekansı / 384) / Haberleşme Hızı)

Eğer (PCON.7 = 1) ise, haberleşme hızı iki katına çıkıyordu. Dolayısıyla PCON.7 = 1 olduğu durumda kullanmamız gereken formül,

TH1 = 256 - ((Kristal frekansı / 192) / Haberleşme Hızı)

olacaktır. Örneğin 11.059 Mhz lik bir kristalimiz olduğunu düşünelim. Bu kristal frekansında, timer-1' i kullanarak saniyede 19,200 bit veri haberleşme hızında haberleşme yapmak isteyelim. TH1 registerine yazmamız gereken değeri bulmak için, yukarıdaki formüllerden birincisini kullanalım.

TH1 = 256 - ((Kristal frekansı / 384) / Haberleşme hızı)
TH1 = 256 - ((11059000 / 384) / 19200)
TH1 = 256 - ((28,799) / 19200)
TH1 = 256 - 1.5 = 254.5

Yukarıdaki hesaplamalardan da görülebileceği 19,200 haberleşme hızında çalışabilmek için TH1'e yazmamız gereken değer 254.5 elde edilmektedir. Bir registere küsuratlı bir sayı koyma imkanı yoktur. Dolayısıyla bu sayıyı TH1 registerine yazamayız. Bunun yerine 254 yazarsak elde edeceğimiz haberleşme hızı 14,400, 255 yazarsak ise elde

edeceğimiz haberleşme hızı 28,800 olacaktır. Öyleyse 19,200 haberleşme hızını gerçekleştirmek için Pcon.7 bitini "1" yapıp, ikinci formülü kullanalım.

TH1 = 256 - ((Kristal frekansı / 192) / haberleşme hızı)
TH1 = 256 - ((11059000 / 192) / 19200)
TH1 = 256 - ((57699) / 19200)
TH1 = 256 - 3 = 253

Burada elde ettiğimiz değer, küsüratlı bir değer değildir. Dolayısıyla bu değeri TH1 registerine yazarak 19,200 baud için gerekli işlemleri tamamlamış oluruz. Dolayısıyla, 11,059 Mhz bir kristalle 19, 200 haberleşme hızında haberleşebilmek için yapmamız gerekenler sırasıyla şunlardır.

1. Seri portu mod 1 veya mod 3 e getirmek gerekir.
2. Timer-1'i mod 2'ye getirmemiz gerekir.

3. TH1'e 253 değerinin yazılması gerekir.
4. PCON.7 (SMOD) bitini "1" yapmamız gerekir.

Şimdi yukarıdaki işlemleri, ASSEMBLY dilinde gerçekleştirelim.

```
MOV   SCON, #048H       ; seri port mod 1
MOV   TCON, #020H       ; timer-1 mod 2
MOV   TH1,  #0FDH       ; TH1 = 253
MOV   TL1,  #0FDH       ; TL1 = 253
ORL   PCON, #080H       ; PCON SMODE biti = 1
```

Eğer kullandığımız frekans 12 Mhz ise, 19200 haberleşme hızı için ne yapabileceğimizi görelim. Önce PCON.7 = 0 alıp birinci formülü deneyelim.

TH1 = 256 - ((Kristal frekansı / 384) / Haberleşme hızı)
TH1 = 256 - ((12,000,000 / 384) / 19200)
TH1 = 256 - ((31,250) / 19200)
TH1 = 256 - 1..63 = 254.37

Gördüğünüz gibi, küsüratlı bir sayı elde ettik. Bu sayının TH1'e konulmasına olanak yoktur. Şimdi PCON.7 = 1 alıp ikinci formülü deneyelim.

TH1 = 256 - ((Kristal frekansı / 192) / haberleşme hızı)
TH1 = 256 - ((12,000,000 / 192) / 19200)
TH1 = 256 - ((62500) / 19200)
TH1 = 256 – 3.26 = 252.74

Yine küsüratlı bir sayı elde ettik. Buna göre 12 Mhz kristal kullanıldığı zaman 19200 haberleşme hızı olanaklı değildir.

Şimdi 9600 haberleşme hızını deneyelim. İlk önce PCON.7 = 0 alalım.

TH1 = 256 - ((Kristal frekansı / 384) / Haberleşme hızı)

TH1 = 256 (12,000,000 / 384) / 9600)
TH1 = 256 - ((31,250) / 9600)
TH1 = 256 – 3.25 = 252.75

Bu sayı da TH1'e konmak için uygun değil. PCON.7 = 1 alalım.

TH1 = 256 - ((Kristal frekansı / 192) / haberleşme hızı)
TH1 = 256 - ((12,000,000 / 192) / 9600)
TH1 = 256 - ((62500) / 9600)
TH1 = 256 – 6.5 = 249.5

Bu sayıda TH1'e yerleştirmek için uygun değil.

Şimdi 4800 haberleşme hızını deneyelim. PCON.7 = 0

TH1 = 256 - ((Kristal frekansı / 384) / Haberleşme hızı)
TH1 = 256 - ((12,000,000 / 384) / 4800)
TH1 = 256 - ((31,250) / 4800)
TH1 = 256 – 6.5 = 249.5

Bu sayıda uygun değil. PCON.7 = 1 alalım.

TH1 = 256 - ((Kristal frekansı / 192) / haberleşme hızı)
TH1 = 256 - ((12,000,000 / 192) / 4800)
TH1 = 256 - ((62500) / 4800)
TH1 = 256 – 13.02 = 242.98

Bu sayıyı 243 (F3H) olarak alıp, TH1'e yerleştirebiliriz. Dolayısıyla yukarıdaki hesaplardan da görüldüğü gibi, 12 MHz bir kristal kullanıldığında, timer-1 kullanılarak oluşturulabilecek maksimum haberleşme hızı, 4800 dür. 12 MHz kristalde 4800 haberleşme hızını programlamak için yazmamız gereken satırlar aşağıdadır.

```
MOV SCON, #048H         ; seri port mod 1
MOV TCON, #020H         ; Timer-1 mod 2
MOV TH1,  #0F3H         ; TH1 = 243
MOV TL1,  #0F3H         ; TL1 = 243
ORL PCON, #080H         ; PCON SMODE biti = 1
```

Biz 8051-8031 entegrelerini kullandığımız devrelerde, standart olarak 11,0592 Mhz kristali kullanacağız. İnceleyeceğimiz 8052 entegresi üçüncü bir timera sahiptir. Bu timer da haberleşme hızının kontrolünde kullanılabilmektedir. Bu timer 12 MHz kristalde 9600 haberleşme hızında seri iletişime olanak vermektedir. 80535 entegresi ise, kendi içinde haberleşme hızı kontrol birimi bulundurmaktadır. Dolayısıyla bu entegrelerle

yaptığımız devrelerde 12 Mhz kristal kullanacağız. Çünkü 12 Mhz kristal kullanıldığında, zaman ölçme hesapları daha kolay olmaktadır.

4.4.4. SERİ PORTA YAZMA

Seri port yukarıda açıklandığı şekilde, konfigürasyonu yapıldığında, seri port yazma ve okuma için hazır hale gelmiştir.

Seri porta bir karakter yollamak için yapmanız gereken sadece, bu karakteri SBUF (99h) özel fonksiyon registerine yazmaktır. Örneğin seri porta "X" karakteri yollamak istiyorsanız, vermeniz gereken komut,

```
MOV SBUF,#'X'  komutudur.
```

Yukarıdaki komut işlemci tarafından okunduğunda, işlemci "X" karakterini seri port üzerinden yollayacaktır. Elbetteki seri porttan yollama işlemi anlık bir işlem değildir ve bir zaman dilimi içerisinde gerçekleşecektir. Standart 8051 bir seri çıkış bufferına sahip olmadığından, yeni bir karakter yollamadan önce, önceki karakterin yollandığından emin olmak zorundayız. 8051 entegresi bir karakterin yollanmasının bittiğini TI bitini "1" yaparak bildirir. Dolayısıyla bu bitin değeri "1" ise karakterin yollanması bitmiştir ve yeni bir karakter yollanabilir. Bu kontrol 8051 kodu olarak aşağıdaki şekilde yazılır.

```
SENDCHR :
CLR TI                  ;Önce TI bitini temizle.
MOV SBUF,#'A'           ;Yollancak karakteri seri porta yaz.
JNB TI,$                ;Karakter yollamasının bitişini bekle.
RET
```

Yukarıdaki üç komut ile bir karakter seri port üzerinden yollanır ve yollama işleminin bitişi beklenir.

4.4.5. SERİ PORTTAN OKUMA

Seri porttan okuma yapmak da seri porta yazma kadar kolay bir işlemdir. Seri porttan okuma yapmak için RI bayrağına bakılır. Bu bayrak "1" ise seri porttan bir karakter gelmiş demektir. Seri porttan gelen bu karakter SBUF registerinde bulunur.

Örneğin, program kodumuz bir karakter okumak üzere beklesin ve karakter geldiğinde bu karakteri akümülatöre okusun. Bunun için yazmamız gereken kod yalnızca iki satır olacaktır.

```
GETCHR:
JNB RI,$          ;8051 RI bayrağının aktiflenmesini bekle
MOV A,SBUF        ;Gelen karakteri akümülatöre oku.
RET;
```

Haluk Turksoy Sk, No:11 ALTUNIZADE, 34662 ISTANBUL-TURKEY
TEL: +90 216 6518000 FAX: +90 216 6518009
http://www.elektro.com.tr info@elektro.com.tr

Yukardaki kodun birinci satırı RI biti "1" oluncaya kadar beklemektedir. Eğer seri hattan bir karakter gelirse RI biti "1" olacaktır . RI biti "1" olduğunda program ikinci satıra geçecek ve MOV komutunu işleme sokacaktır.

4.4.6. SERİ VERİ MODU 2 (ÇOKLU İŞLEMCİ MODU)

Seri haberleşme modu 2, seri haberleşme modu 1'e benzer. Mod 2'de haberleşmede 11 bit kullanılır. Bu 11 bitin biri start biti, birisi stop biti ve 9 tane veri bitidir. Gönderilen verinin 9'uncu biti SCON registerinin TB8 bitine yazılır. Veri okuma sırasında ise bu 9'uncu bit RB8 bitine gelir. Seri veri modu 2'de iletişim hızı,

$f_{baud2} = (2^{SMOD} / 64) \times f_{clock}$

formulü ile hesaplanır. Dikkat edilirse, bu iletişim modunda da haberleşme hızı mod 1 haberleşme hızından çok daha yüksektir. Pek çok işlemci kullanılan bir ağ üzerinde, toplanan verilerin çok hızlı bir şekilde aktarılabilmesine gerek vardır. Bu iletişim modunun en önemli özelliği kullanılan 9uncu veri bitinde yatmaktadır. Bu veri biti sayesinde yollanan bir mesajın ağdaki belli işlemciler tarafından algılanması, diğerleri tarafından ise ihmal edilmesi sağlanabilmektedir. Seri iletişim modlarında, 8051 entegresine bir karakter geldiğinde karakterin son bitinin alınmasıyla RI bitinin 1 yapıldığını biliyoruz. Seri veri modunda RI bitinin 1 olabilmesi extra bazı şartlara da bağlanmıştır. RI bitinin set olabilmesi için, ya son veri biti 1 olmalı yada SCON registerindeki SM2 biti 0 olmalıdır. Dolayısıyla ağdaki bir işlemci gönderilen mesajın ağdaki tüm işlemciler tarafından alınmasını istiyorsa gönderdiği bilginin son bitini 1 yapmalıdır. Eğer ağdaki yalnız belli işlemcilerin bu mesajı alması isteniyorsa, mesaj yollayan işlemci gönderdiği bilginin son bitini 0 yapar. Bu sayede ağdaki işlemcilerden yalnızca SM2 biti 0 olanlar, gelen mesajı alırlar.

Bu sayede bir yerel ağ oluşturulup, zaman zaman sadece belli entegrelerin sorgulama sonuçlarına cevap verebilmeleri sağlanabilir. Bu yöntem işlemci ve bilgisayar ağlarında sık kullanılan, konuşan ve dinleyenlerin belirlenmesi esasına dayanan bir yöntemdir.

4.4.7. SERİ VERİ MODU 3

Bu işlemci modu seri veri modu 2 ile aynı çalışma mantığına sahiptir. Yalnızca bu modda haberleşme hızı sabit değildir ve aynı seri mod 1' de olduğu gibi timer-1 ile belirlenir.

4.4.8. RS-232 İLETİŞİMİNİN TEMEL KAVRAMLARI

RS-232 bilgisayar dışındaki cihazların bilgisayar ile haberleşmelerinde en çok kullanılan iletişim standartlarından birisidir. RS-232 temel olarak bir seri iletişim birimidir. Seri iletişim biriminde karakterler bir hat üzerinden bit bit yollanır. Seri iletişimin paralel iletişime göre en önemli üstünlüğü bağlantı kolaylığıdır. Bilgisayardan cihaza karakter yollamak için bir hat , cihazdan gelen karakterleri okumak üzere bir hat ve bir toprak hattı

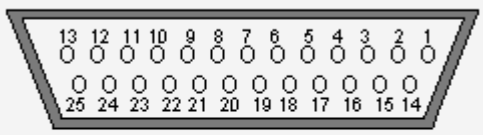
olmak üzere toplam üç hat kullanılarak iletişim gerçekleştirilebilir. Bu standardın önemli dezavantajı, haberleşme hızı arttıkça bilgi kaybına yo açmamak için kablo uzunluğunun da kısalması gerekliliğidir. Standard RS-232 19200 haberleşme hızında en fazla 20m kablo uzunluğuna izin vermektedir. Yeni seri iletişim standartlarından RS-422, RS-449 çok daha yüksek haberleşme hızlarında çok daha uzun kablolamaya olanak vermektedir. Örneğin RS-422 1600 m uzunluğunda bir kablo üzerinden bir magabit/saniye haberleşme hızlarını desteklemektedir.

Her bilgisayar en az bir seri RS-232 iletişim birimine sahiptir. Yeni bilgisayarlarda RS-232 konnektörleri hemen hemen tamamen 9 pin erkek konnektörlerdir. Öncelikle bilgisayarınızın arkasında böyle bir konnektör olup olmadığına bakın. Bilgisayar herhangi bir RS-232 portuna o portun adresi aracılığı ile ulaşır. Bilgisayarınızda bulunan RS-232 portunun adresini bilgisayar denetim masasından sistem seçeneği içinden araçlar menüsüne geçerek ulaşabilirsiniz. Araçlar menüsünde bilgisayarda var olan tüm iletişim birimlerinin (seri portlar, USB, printer portu, ekran kartı vs.) adresleri bulunur. Seri iletişim biriminizin adresini buradan bulabilirsiniz. Bazı bilgisayarlarda yalnızca bir RS-232 çıkışı bulunabilir. Eğer bu çıkışı seri fare için kullanılıyorsa mikroişlemci kartınızı bilgisayar ile haberleştirecek hiç bir RS-232 biriminiz yok demektir. Yapmanız gereken seri fareden vazgeçip, bilgisayarınızın fare girişine bağlanabilecek bir fare satın almak olacaktır. Böylece RS-232 portunuz mikroişlemci kartınıza bağlamak üzere size kalacaktır.

Bu aşamada biraz da RS-232 konnektörünün yapısından ve seri iletişimden bahsetmek faydalı olacaktır. RS-232 seri iletişim standardı elektriksel özelliklerinde "1" biti -3 volttan –25 volta kadar bir elektrik sinyali olarak tanımlanmıştır. "0" sinyali ise +3 volttan +25 volta kadar elektrik sinyali olarak tanımlanmıştır. –3 volt ile +3 volt arasındaki herhangi bir sinyal ise belirsiz bir sinyaldir.

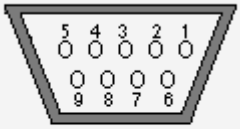
Daha önce de belirttiğimiz gibi yeni bilgisayarlarda RS-232 iletişimi için 9 pin erkek konnektörler kullanılmıştır Fakat bazı bilgisayarlarda ve seri iletişim kurulacak bazı cihazlarda 25 pin erkek konnektörler de bulunabilmektedir. Aşağıda 25 pin ve 9 pin DB kablolarda RS-232 sinyallerinin nasıl bağlandığını görebilirsiniz.

DB25S konnektör:



Pin	Sinyal
2	TxData
3	RxData
4	RTS
5	CTS
6	DSR
7	GND
20	DTR

DB9S konnektör:



Pin	Sinyal
2	RxData
3	TxData
4	DTR
5	GND
6	DSR
7	RTS
20	CTS

Mikrokontrolcülerle bilgisayarların iletişminde temel olarak TxD, RxD ve GND hatları kullanılır. Pek çok mikrokontrolcüde, bilgisayarla haberleşmek üzere TxD ve RxD hatları bulunur. Fakat mikrokontrolcüler de bu hatların sinyalleri 0 volt ve 5 volttur. Bilgisayarla haberleşmede gereken RS-232 sinyalleri ise +12 volt ve –12 volttur. Bu yüzden mikrokontrolcü ve bilgisayar arasındaki iletişimde gerilim dönüşümünü sağlamak üzere bir gerilim dönüştürücü kullanılır. Bu tür gerilim dönüştürücülerin en çok kullanılanı MAX232 entegresidir.

Genelde mikrokontrolcü tarafında kullanılacak RS232 konnektörü dişi yapılır. Dolayısı ile mikrokontrolcü ile bilgisayarı bağlayacak ara kablonun bir ucuna 9 pin erkek konnektör, bir ucuna da 9 pin dişi konnektör bağlanır Daha önce de söylediğimiz gibi gereken kablo sayısı yalnızca 3 tür. Konnektörlerin 5 nolu uçları (GND) direkt olarak birbirine bağlanır. 2 (TxD) ve 3 (RxD) uçları ise birbirine çarpraz bağlanır. Böylece mikrokontrolcünün GONDER (TxD) ucu bilgisayarın AL (RxD) ucuna ve bilgisayarın YOLLA ucu mikrokontrolcünün AL ucuna bağlanmış olur.

4.5. KESMELER (INTERRUPTS)

Kesme isminden de anlaşılacağı üzere , bir kesme normal program çalışmasını kesen bir mekanizmayı tanımlamaktadır. Mikrodenetleyiciler, programları sıra ile işleyen cihazlardır. Bu sıra yalnızca sıçra (JMP) ve çağır (CALL) gibi komutlarla değiştirilebilmektedir. Kesmelerde ise normal program akışı, bir kesme ile kırılmakta ve kesme programı tamamlandıktan sonra normal program akışı, tam kesmeden önceki noktadan normal akışına devam etmektedir. Bu tür bir altprograma kesme hizmet programı (Interrupt Handler) adı verilir. Bu altprogram yalnızca bir kesme olduğunda çalışır. Kesme olmaması durumunda programa hiç bir katkısı yoktur. Kesme yaratan mekanizma , "taşma (overflowing)", " ser porttan bir karakter okuma", seri porta bir karakter yazma", veya INT0 ve INT1 uçlarına bağlı sinyallerin aktif olmasıyla gerçekleşebilir.

8051 uygun bir şekilde programlandığı zaman bu kesmelerden birisi gerçekleştiğinde, program akışının kesme hizmet programına yöneltilmesi mümkündür. Bu kesme tamamlandığında program kaldığı yerden normal akışına devam edecektir. Ana program böyle bir sıçramanın farkında bile olmayacaktır. Normal program akışının bu şekilde kesilebilmesi, belli koşulları kontrol etmek açısından son derece önemli olabilir. Eğer kesmeler olmasaydı, ana program sürekli olarak timernın taşıp taşmadığını, seri porttan bir karakter gelip gelmediğini veya INT0 ve INT1 uçlarının değerlerinin "1" olup olmadığını sürekli olarak kontrol etmek zorunda kalacaktı. Bu durumda yazılan kod hem çok uzun ve okunması zor olacaktı, hemde işlemci zamanının önemli bir kısmı, çok nadir olarak gerçekleşen bir olayın, gerçekleşip gerçekleşmediğini kontrol etmekle harcanacaktı. Örneğin elimizde 30 kilobaytlık, pek çok altprogramdan oluşan bir kodumuz olsun. Programımızın yapması gereken işlerden birisi de, örneğin timer 0 her taştığında, P3.0 bacağının değerini "1" ise "0", "0" ise "1" yapmak olsun. Böyle bir işlemin kodu son derece kolaydır.

```
JNB    TF0, SONRAKI_KOMUT
CPL    P3.0
CLR   TF0
SONRAKI_KOMUT: ...
```

Timer-0 her taştığında TF0 biti "1" olacağından, yukardaki kod her taşmada P3.0 bitinin değerini değiştirecektir. Böylece program kodu istenilen görevi yapacak şekilde yazılmış olmaktadır. Bu kod istenen görevi yapmaktadır fakat önemli ölçüde sistem zamanı kullanmaktadır.

Yukarıdaki kısa kodun ne kadar işlemci zamanı kullandığını hesaplamaya çalışalım. JNB komutu 2 makine çevriminde tamamlanan bir komuttur. Timer 0'ın taşması durumunda, CPL ve CLR komutları işlenecektir. Bu komutların her biri bir makine çevriminde tamamlanan komutlardır. Dolayısıyla iki makine çevrimi de bu komutların işlenmesine gidecektir. Matematiği biraz kolaylaştırmak için, programın geri kalan kısmının 98 makine çeviriminden oluştuğunu varsayalım. Böyle bir durumda, timer 0'ın taşıp taşmadığının kontrolü için 2 makine çevrimi, programın geri kalan kısmı için 98 makine çevrimi, toplam 100 makine çevrimi zaman geçecektir. Timer 0'ın 16 bit timer modunda (mod 0) çalıştığını düşünelim. Bu durumda timer 0, her 65,536 makine çevriminde bir defa taşacaktır. Bu zaman içerisinde 655 kere JNB komutu işlenmiş olacaktır. 655 JNB komutu 1310 makine çevrimi alacaktır. Timer taştığında iki makine çevrimi de CPL ve CLR komutlarının işlenmesinde kullanılacaktır. Buna göre timer 0 taştığında, P3.0 bitinin değerini değiştirebilmek için toplam 1312 makine çevirimi işlemci zamanı kullanmış olduk. Bu zaman, program işlenmesi zamanında geçen zamanın yaklaşık %2 sidir. Öte yandan her döngüde timer 0'ın taşıp taşmadığının kontrolü, taşma olduysa P3.0 bitinin değiştirilmesi, ana program içerisinde gerçekleştirildiğinden, program kodu da okunulabilirlik açısından kötüdür.

Kesmeler bize bu tür sorunları çözme olanağı vermektedir. Kesme kullanımında, timer-0'ın taşıp taşmadığını bizim kontrol etmemiz gerekmeyecektir. Bu kontrolü mikroişlemci kendisi yapacaktır. Taşma oluştuğunda, kontrolcü kesme hizmet programına sıçrayıp, kesme programını işleyecek ve normal program akışına geri dönecektir. Bunun için

yazılması gereken kesme hizmet programı yalnızca 2 satırdır.

```
CPL  P3.0

RETI
```

Öncelikle kod içerisinde "CLR TF0" komutunun kullanılmadığına dikkat ediniz. Çünkü, 8051 timer-0 kesmesi için programlandığında, taşma olduğunda TF0 biti otomatik olarak mikroişlemci tarafından temizlenmektedir.

Aynı zamanda bir altprogramdan dönüşte kullanılan normal RET komutu yerine RETI (return from interrupt) komutu kullanılır. RETI komutu RET komutu ile aynı işi yapar, fakat işlemciye kesme hizmet programının sonlandığını belirtir. Her kesme hizmet programı mutlaka RETI komutu ile sonlandırılmalıdır.

Buna göre kesme kullandığımızda, P3.0'ın değerini değiştirmek için yalnızca 3 makine çevrimi zaman kullanmamız gerekiyor. Bir makine çevrimi CPL komutu için, 2 makine çevrimi de RETI komutu için kullanılacaktır. Bu iki komut 65,536 makine çevrim zamanı içerisinde yalnızca bir kez işleme sokulmuş olacaktır. Kesme kullanılmadığında aynı görevi yapmak için 1312 makine çevrim zamanı kullandığımız düşünülürse, kesme kullanıldığında programımız 437 kere daha verimli bir şekilde işlemci zamanını kullanmaktadır. Öte yandan program daha kolay anlaşılabilir ve okunulabilir bir duruma gelmiştir. Çünkü program içinde sürekli olarak timer-0'ın taşıp taşmadığını kontrol etmek zorunda değiliz. Yapmamız gereken yalnızca kesme programını yazmak ve unutmaktır. Gerektiği durumda işlemci kesme programını devreye sokacaktır.

Aynı fikiri seri port haberleşmesinde veri alırken zaman zaman son derece faydalı olmaktadır. Seri porttan bir karakter okumak için kullanılabilecek yollardan birisi, sürekli olarak RI bitini kontrol etmek ve RI biti "1" olduysa, gelen karakteri SBUF registerinden okumaktır. Böyle bir yaklaşımda, ana program kodu büyüdükçe gelen karakterlerden birini kaçırma riski de vardır.

Çünkü RI bitini kontrol ettikten hemen sonra seri karakter gelmiş olabilir, RI hattını ikinci kere kontrol edinceye kadar başka bir karakter daha gelebilir ve böylece birinci karakter kaybolmuş olur. Seri port kesmesi yardımıyla bu sorunu da çözmüş oluruz.

Seri porttan bir karakter okunduğunda hemen kesme programımızı devreye sokuyoruz ve gelen karakteri okuyoruz. Program içerisinde de sürekli olarak karakter geldi gelmedi kontrolü yapmaktan kurtuluruz.

Haluk Turksoy Sk, No:11 ALTUNIZADE, 34662 ISTANBUL-TURKEY
TEL: +90 216 6518000 FAX: +90 216 6518009
http://www.elektro.com.tr info@elektro.com.tr

4.5.1. KESME KAYNAKLARI VE ADRESLERİ

8051'in temel kesme kaynakları aşağıda verilmiştir. 8051 bu kesmelerden biri veya birden fazlası kesme oluşturacak şekilde programlanabilir

Timer 0 taşması
Timer 1 taşması
Seri bir karakter konması veya yazılması
Harici kesme-0
Harici kesme-1

Diğer bir deyişle 8051, timer-0 taştığında veya seri porttan bir karakter okunduğunda veya yazıldığında uygun kesme hizmet programı çağrılacak şekilde programlanabilir.

Farklı kesmeler oluştuğunda, işlenmesi gereken kesme hizmet programı farklı olacaktır. Bunun için mikroişlemci içerisinde kesme hizmet programı başlangıç adresleri oluşturulmuştur. Bu adresler 8051'in standart adresleridir ve oluşan kesmeye göre program akışı bu adrese dallanacaktır.

Kesme	Bayrak	Kesme Adresi
Harici Kesme-0	IE0	0003h
Timer-0	TF0	000Bh
Harici Kesme-1	IE1	0013h
Timer-1	TF1	001Bh
Seri Kesme	RI/TI	0023h

Tablo 4.3

Yukarıdaki tablodan (Tablo 4.3) da anlaşılabileceği üzere, eğer timer-0 kesmesi oluşabilecek şekilde 8051 ayarlansaydı, timer-0'ın her taşmasında program akışı 000Bh adresine gidecektir. Bu adreste timer-0 taştığında yapılacak işlem kodları bulunmalıdır.

4.5.2. KESME İZİNLERİ

8051 entegresi ilk çalışmaya başladığında, tüm kesmeler engellenmiş durumdadır (disabled). Kesmeler engelli durumda olduğunda, örneğin timer-0 kesme bayrağı taşsa bile, bir kesme oluşturmayacaktır. Kesmelerin aktiflenmesi için programda 8051 kesmeleri aktif hale getirilmeli ve hangi kaynakların kesme yaratacağı belirtilmelidir. Kesmeleri aktif hale getirmek veya engellemek için özel fonksiyon registerlerinden kesme aktifleme registeri (IE, adres A8h) kullanılır IE registerinin bit adreslenebilir bir registerdir. Aşağıdaki tabloda bu registerin her bir bitinin anlamı gösterilmiştir.

Bit	İsim	Bit Adresi	Açıklama
7	EA	AFh	Genel kesme aktifle/engelle
6	-	AEh	Tanımsız
5	-	ADh	Tanımsız
4	ES	ACh	Seri kesme aktifle
3	ET1	ABh	Timer 1 kesmesi aktifle
2	EX1	AAh	Harici kesme 1 aktifle
1	ET0	A9h	Timer 0 kesme aktifle
0	EX0	A8h	Harici kesme 0 aktifle

Tablo 4.4

Yukarıdaki tablodan (Tablo 4.4) da görülebileceği gibi, her bir kesme IE registerinde kendi aktifleme/engelleme bitine sahiptir. Herhangi bir kesmeyi aktiflemek için IE registerinde o kesmeye ait bit aktiflenmelidir. Örneğin timer 1 kesmesini aktiflemek için ya,

```
MOV IE,#08h veya SETB ET1
```

komutunu vermek gereklidir. Yukarıdaki komutlardan her ikiside IE registerinin 3 nolu bitini, yani timer-1 kesmesini aktiflemektedir. Bu kesme aktiflendiğinde, ne zaman TF1 biti "1" olsa 8051 001Bh adresine gidip orada bulunan kesme programını işleyecektir.

Bununla beraber, timer-1 kesmesinin tam anlamıyla aktiflenmesi için aynı zamanda genel kesme aktifleme bitinin de (bit 7) "1" yapılması gereklidir. Bu bit bütün kesmeleri aktifler veya engeller. Başka bir deyişle bit 7 "0" olduğunda, diğer kesme aktifleme bitleri "1" olsa bile kesme oluşmayacaktır. Bu bit "1" yapıldığında, seçilen kesmeler aktif hale gelecektir. Bu durum, zaman kritik kodlarda önemli yararlar sağlar. Belli bir zaman içerisinde mutlaka işlenip bitirilmesi gereken bir kod varsa ve bu kodun kesmeler tarafından kesilmesi istenmiyorsa, bu kodun başlangıcında tüm kesmelerin engellenmesi yerinde olacaktır.

Örnek olarak timer-1 kesmesini aktif hale getirmek için gereken kod yalnızca iki satırdır. Önce timer 1 kesme aktifleme biti "1" yapılır. Daha sonra genel kesme aktifleme biti "1" yapılır.

```
 SET  ET1
SETB EA
```

Bu iki satırla, timer-1 kesmesi aktif hale gelmiş demektir. Bundan sonra ne zaman timer-1 taşması oluşsa program 001Bh adresindeki kesme programını çalıştıracaktır.

4.5.3. KESME ÖNCELİKLERİ

8051 otomatik olarak her bir komuttan sonra bir kesme oluşup oluşmadığını kontrol eder. Kesme koşullarını kontrol ederken, hangi kesmenin oluştuğunu belli bir sırada kontrol eder. Bu kontrol sırası aşağıdaki gibidir:

Harici kesme-0 /Timer-0 kesmesi / Harici kesme-1 /Timer-1 kesmesi /Seri port kesmesi

Buna göre örneğin, seri port kesmesi ve dış kesme 0 aynı anda olarak oluşmuşsa, önce dış kesme 0 programı işlenecektir. Dış kesme 0 programı tamamlandıktan sonra da seri kesme programı tamamlanacaktır 8051 kesme önceliği için yalnız iki seviye tanımlamaktadır; düşük ve yüksek kesme önceliği. Kesme önceliklerini kullanarak aynı zamanlı oluşan kesmelerden hangisinin daha önce işleme sokulacağını belirlemek mümkündür.

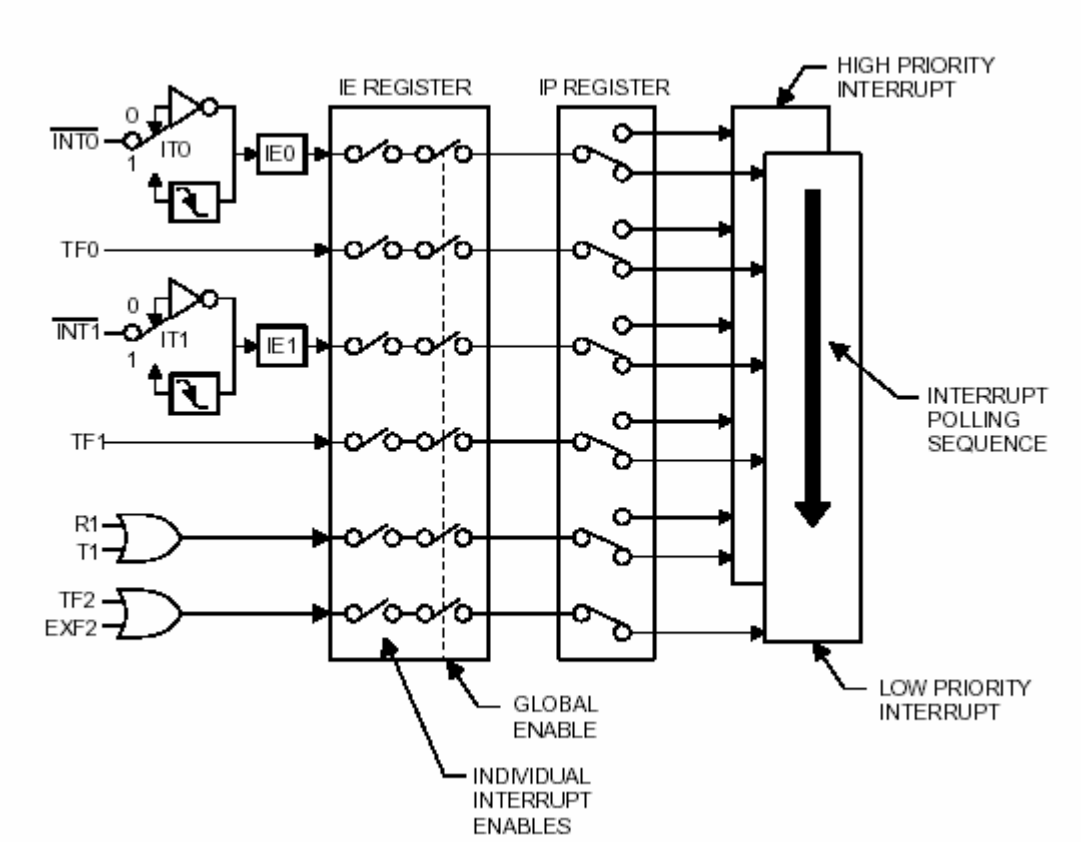
Bit	İsim	Bit Adresi	Açıklama
7	-	-	Tanımsız
6	-	-	Tanımsız
5	-	-	Tanımsız
4	PS	BCh	Seri port önceliği
3	PT1	BBh	Timer 1 önceliği
2	PX1	Bah	Harici kesme 1 önceliği
1	PT0	B9h	Timer 0 önceliği
0	PX0	B8h	Harici kesme 0 önceliği

Tablo 4.5

Örneğin, timer-1 her taştığında, timer-1 kesmesi oluşacaktır. Şekilde ve seri port üzerinden bir karakter geldiğinde seri port kesmesi aktiflenecek şekilde 2 kesme kullandığımızı varsayalım. Programınızda, seri porttan bir karakter geldiğinde bunu okumak, zamanlama kesmesinden daha önemli olabilir. Bu durumda eğer seri kesme geldiğinde, timer 1 kesmesi çalışır durumdaysa, seri port kesmesinin timer 1 kesmesini durdurmasını ve öncelikle seri port kesmesi işlemini tamamlamasını isteyeceksiniz. Seri port kesmesi tamamlandıktan sonra kontrol tekrar timer 1 kesmesine geçecektir. Bu amaçla yapmanız gereken sadece, seri port kesmesine yüksek öncelik, timer 1 kesmesine düşük öncelik vermek olacaktır. Kesme önceliği kontrolü, kesme önceliği (Interrupt Priority, IP) özel fonksiyon registeri (adres B8h) kullanılarak yapılır. Kesme önceliği kontrol registeri bit adreslenebilir bir registerdir ve aşağıdaki bitlerden oluşmuştur.

Kesme öncelikleri ile ilgili aşağıdaki kurallar geçerlidir.Yüksek öncelikli bir kesme, başka yüksek öncelikli bir kesme tarafından kesilemez. Yüksek öncelikli bir kesme düşük öncelikli bir kesme akışını keser. Düşük öncelikli bir kesmenin çalışabilmesi için yüksek öncelikli bütün kesmelerin çalışıp tamamlanmış olması gerekir.

Eğer iki kesme aynı zamanlı oluşmuşsa önce yüksek öncelikli kesme çalışacaktır. Eğer yüksek öncelikli iki kesme aynı anlı oluşmuşsa, 8051 kontrol sırasında daha önce sırada olan kesme önce işleme sokulacaktır.



4.5.4. BİR KESME OLUŞTUĞUNDA 8051 ENTEGRESİNİN YAPTIĞI İŞLEMLER

Bir kesme oluştuğunda 8051 otomatik olarak aşağıdaki işlemleri gerçekleştirir. Program sayıcısı (program counter) yığına yazılır, önce program counterın düşük baytı daha sonra program counterın yüksek baytı yığına yazılır. Aynı öncelikli ve düşük öncelikli diğer kesmeler bloke edilir.

Timer kesmesi ve harici kesmelerde, kesmeye karşılık gelen kesme bayrağı temizlenir. Program çalışması kesme vektör adresine geçirilir.Kesme programı çalıştırılır.

Eğer işlenen kesme timer kesmesi veya harici kesmelerden biriyse, bu kesmeye karşılık gelen kesme bayrağı otomatik olarak temizlenmektedir. Bu nedenle program içerisinde bu bayrakları temizlemeye gerek yoktur.

4.5.5. BİR KESME SONUNDA 8051 ENTEGRESİNİN YAPTIĞI İŞLEMLER

Bir kesme programı, RETI komutu ile sonlandırılır. RETI komutuna rastlandığında 8051 otomatik olarak aşağıdaki iki işlem yapar. Yığına yazılan program counter değeri yığından geri alınır. Kesme statüsü kesmeden önceki durumuna getirilir.

Seri port kesmeleri diğer kesmelerden bazı farklılıklar gösterir. Bunun sebebi seri port kesmesinin iki farklı kesme bayrağına (RI ve TI) sahip olmasıdır. Eğer bu bayraklardan birisi aktif olursa seri port kesmesi aktiflenmiş olacaktır. Daha önceki bölümlerden de hatırlanabileceği gibi seri porttan bir karakter okunduğunda RI kesmesi, seri porta bir karakter yazıldığında ise TI kesmesi aktiflenecektir.

Buna göre eğer bir seri port kesmesi oluşmuşsa, ya bir karakter okunmuştur, yada bir karakter yollanmıştır. Dolayısıyla yazılacak kesme programı bu iki durumdan hangisinin gerçekleştiğini anlamak ve ona göre gerekli işlemleri yapmak durumunda kalacaktır. Aynı zamanda 8051 RI ve TI bayraklarını otomatik olarak temizlemediği için , yazılan kod bu bayrakları kendisi temizlemek durumundadır. Aşağıda basit bir seri kesme programı gösterilmektedir.

```
SERI_KESME :
      JNB   RI,CHECK_TI       ; RI "0"  ise TI_KONTROL
      MOVA,SBUF               ;RI=1,karekterioku

      CLR    RI               ;RI bitini temizle
CHECK_TI:
      JNB    TI, KESME_SONU   ;TI "0"
ise  KESME_SONU
      CLR   TI                ;TI bitini
      temizle
      MOV   SBUF,#'A'         ;Seri porta "A" karakteri yolla
KESME_SONU:
      RETI
```

Yukarıdaki kodda da görülebileceği gibi, program iki seri port kesme bayrağını kontrol etmektedir. Eğer her iki bayrakta aktif durumdaysa, kodun iki bölümü de çalıştırılacaktır. Dikkat edilmesi gereken diğer bir nokta da her iki bölümün sonunda da ilgili kesme bayrağı temizlenmektedir. Eğer kesme bayrağını temizlemeyi unutursanız, kesme tekrar tekrar sonunda kesme bayrağı temizleninceye kadar çalışacaktır. Bu nedenle seri kesme programının sonunda, kesme yaratan bayrağı temizlemek son derece önemlidir.

4.5.6. KESMELERDE REGİSTERLERIN KORUNMASI

Kesmelerle ilgili unutulmaması gereken çok önemli bir kural vardır. Kesme programı sonucunda, mikroişlemci tamamen kesme başlamadan önceki duruma gelmelidir. Çünkü kesme mantığı, ana programın hiç haberi olmayan bir program parçasının kendiliğinden ve gelişigüzel bir zamanda devreye girmesi demektir.

```
CLR   C              ;Elde bayrağını temizle
MOV   A,#25h         ;Akümülatöre 25h değeri yükle
ADDC   A,#10h        ;10h değerini elde ile birlikte akümülatöre ekle
```

Yukarıdaki kod tamamlandığı zaman akümülatörde 35h değeri bulunacaktır. Fakat MOV komutundan hemen sonra bir kesme oluştuğunu varsayalım. Bu kesme programı, elde bayrağını "1" yapıyor ve akümülatöre 40h değeri yüklüyor olsun. Kesme programı bittiğinde, program akışı tekrar ana programa dönecek ve ADDC A, #10h komutu ile program çalışmaya devam edecektir. Kesme programı içerisinde elde bayrağı "1" yapıldığı ve akümülatöre 40h değeri yüklendiği için toplama işlemi sonucunda akümülatörde 51h değeri kalacaktır. Bunun ana program için anlamı açıktır. Ana programda 25h ve 10h değerleri toplandığında elimizde kalan sonuç 51h olacaktır. Böyle bir sonuç son derece anlamsız olduğundan kesme programları ile fazla çalışmamış bir programcı işlemcinin bozulduğunu düşünecektir.

Aslında ortaya çıkan olay, kesme programının, kullandığı registerleri koruma altına almamış olmasıdır. Bir kesme programı işlemciyi, kesme başlamadan önceki durumda bırakmalıdır. Bu kurala göre, eğer kullandığınız kesme programı, akümülatörü kullanıyorsa, akümülatörün değeri kesme başında ve sonunda aynı olmalıdır. Bu olay genelde PUSH ve POP komutları kullanılarak yapılır. Aşağıdaki kod bu kullanıma bir örnektir.

```
PUSH  ACC
PUSH  PSW


MOV   A,#0FFh
ADD   A,#02h


POP   PSW
POP   ACC
```

Yukaridaki kesme programı temel olarak MOV ve ADD komutlarından oluşmuştur. Bu komutlardan her ikisi de akümülatörün içeriğini değiştirmektedir. Öte yandan ADD komutu elde bayrağının durumunu değiştirmektedir. Bir kesme programı, registerların kesme başındaki durumlarını korumak zorunda olduğuna göre, kesme programı başında PUSH komutlarıyla korunacak registerlar yığına yazılır. Bu registerlar yığına yazıldıktan sonra kesme programı içerisinde, bu registerları istendiği gibi değiştirme özgürlüğü yakalanmış olur. Kesme programı tamamlandığı zaman, registerların orjinal değerleri yığından geri alınır ve kesme programı RETI komutu ile sonlandırılır. Dolayısıyla

bir kesme oluştuğunda, ana program kesme oluşup oluşmadığını hiç bir şekilde bilmeyecektir.

Genelde kesme programlarında aşağıdaki registerlar korumaya alınmalıdır.

PSW

DPTR (DPH / DPL)

ACC

B

R0-R7 registerları

Bu registerlardan PSW çeşitli 8051 komutları ile değişebilen bitlere sahip bir registerdır. Dolayısıyla, yaptığınız işten hiç şüphesiz emin olmadığınız durumlarda, PSW registeri korunması gereken bir registerdir.

Öte yandan pek çok assembler,

```
PUSH R0
```

gibi bir komuta izin vermez. Çünkü böyle bir komut seçilen register banka bağlı olarak 00h, 08h, 10h, veya 18h iç hafıza elemanına karşılık gelecektir. Dolayısıyla R0, PUSH ve POP komutlarının kullanabileceği geçerli bir hafıza adresini göstermemektedir. Dolayısıyla, kesme programınızın içerisinde bir "R" registeri kullanıyorsanız, doğrusu direkt olarak register adresini PUSH etmektir. Örneğin, PUSH R0 yerine, kullanılan register bankına bağlı olarak,

PUSH 0H, PUSH 8H, PUSH 10H veya PUSH 18H komutlarından birini kullanmak gerekir.

4.6. ÇALIŞMA MODLARI (DÜŞÜK GÜÇ TÜKETİMİ MODLARI)

Üç çesit çalışma modu vardır. Bunlar aşağıda incelenmiştir.

4.6.1. NORMAL MOD

Bu modda, mikrodenetleyici normal işlemleri yürütmeye devam eder.

4.6.2. IDLE MOD

Aylak çalışma modudur. Bu modda CPU durur. Fakat chip üzerindeki tüm işlevler (Sayıcılar (counterlar),seri kapı,kesme düzeni) devam ettirilir. Registerlar ve RAM içleri korunur. Bu modan çıkış için sistem resetlenmelidir. İkinci yol ise; yetkili kılınan bir kesmenin (sayıcılardan (counterlardan), seri kapıdan veya dışarıdan) aktif olmasıdır.

4.6.3. POWER-DOWN MOD

Kesik güç modudur. Bu modda sadece iç RAM in içeriği korunur ve herhangi bir işlem için 5 V' dan düşük gerilim seviyeleri kullanılır bu modan çıkmak için sistemin resetlenmesi gerekmektedir.