İSTANBUL TEKNİK ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

TMS77C82 EMÜLATÖR KARTI

YÜKSEK LİSANS TEZİ

Müh. Ahmet ÖZMEN

Ana Bilim Dalı : ELEKTRONİK VE HABERLEŞME

Programı : ELEKTRONİK VE HABERLEŞME

MAYIS 1991

T. C.
Yükseköğretim Kurulu
Dokümantasyon Merkezi

19259

İSTANBUL TEKNİK ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

TMS77C82 EMÜLATÖR KARTI

YÜKSEK LİSANS TEZİ

Müh. Ahmet ÖZMEN

Tezin Enstitüye Verildiği Tarih : 22 Mayıs 1991

Tezin Savunulduğu Tarih : 6 Mayıs 1991

Tez Danışmanı : Prof. Dr. Ahmet DERVİŞOĞLU

Diğer Jüri Üyeleri : Prof. Dr. Cevdet ACAR

: Prof. Dr. Eşref ADALI

MAYIS 1991

ÖNSÖZ

Bu tezin hazırlanmasında, değerli görüşlerinden ve yapıcı uyarılarından yararlandığım sayın hocam Prof. Dr. Ahmet DERVİŞOĞLU' na, kendi laboratuvarlarında çalışma imkanı ve her türlü kaynak sağlanmasında katkılarından dolayı EMPA A.Ş.'ne, ve Müh. Sevilay Özdemir'e teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET..... VI
SUMMARY..... VII
BÖLÜM 1. GİRİŞ..... 1
BÖLÜM 2. TMS77C82 MİKROBİLGİSAYARININ MİMARİ YAPISI. 6
2.1. Yonga-İçi Bellek ve Saklayıcılar..... 8
2.1.1. Saklayıcı Alanı (RF)..... 8
2.1.2. Giriş/Çıkış Saklayıcı Alani (PF)..... 8
2.1.3. Yığın Göstergesi (SP)..... 9
2.1.4. Durum Saklayıcısı (ST)..... 10
2.1.5. Program Sayacı (PC)..... 11
2.2. Yonga-İçi Genel Amaçlı Giriş/Çıkış İskeleleri..... 12
2.3. Bellek Modları..... 13
2.3.1. Tek-Yonga Modu..... 15
2.3.2. Çevre Birim Genişletme Modu..... 15
2.3.3. Tam Genişletme Modu..... 17
2.3.4. Mikroişlemci Modu..... 19
2.4. Sistem Saati..... 19
2.4.1. Düşük Güç Maskesi..... 20
2.5. Düşük Güç Modları..... 22
2.6. Kesmeler ve Sistem Sıfırlama..... 23
2.6.1. Sistem Sıfırlama (Reset)..... 24
2.6.2. Kesme İşlemi..... 25
2.6.3. Kesme Kontrolü..... 27
2.6.4. Çoklu Kesmeler..... 29
2.7. Programlanabilir Zamanlayıcı/Olay sayıcılar..... 30
2.7.1. Zamanlayıcı Kontrol Saklayıcıları..... 31
2.7.2. Zamanlayıcıların Kaskad Bağlanması..... 33
2.7.3. Zamanlayıcı ve Önbölücü İşlemi..... 33
2.7.4. Zamanlayıcı Kesmeleri..... 34
2.7.5. Zamanlayıcı Çıkış İşlevi..... 35

2.8. Seri İskele..... 35
2.8.1. Seri İskele Saklayıcıları..... 36
2.8.1.1. Seri Mod Saklayıcısı..... 37
2.8.1.2. Seri Kontrol Saklayıcısı-0 (SCTL0).... 38
2.8.1.3. Seri İskele Durum Saklayıcısı (SSTAT). 39
2.8.1.4. Seri Kontrol Saklayıcısı-1 (SCTL1).... 40
2.8.1.5. Zamanlayıcı-3 Veri Saklayıcısı..... 42
2.8.1.6. Alıcı ve Verici Tamponlar..... 42
2.8.1.7. Haberleşme Modunda RX ve TX İşareti... 43
2.8.1.8. Giriş/Çıkış Modunda RX ve TX İşareti.. 44
2.8.2. Seri Haberleşme Modları..... 44
2.8.2.1. Asenkron Haberleşme Modu..... 44
2.8.2.2. İsosenkron Haberleşme Modu..... 45
2.8.2.3. Seri Giriş/Çıkış..... 46
2.8.3. Çoklu-İşlemcili Haberleşme..... 47
2.8.3.1. Motorola (MC6801) Protokolü..... 48
2.8.3.2. Intel (I8051) Protokolü..... 48
2.8.4. Seri İskele Koşullandırma..... 49
2.8.5. Zamanlayıcı-3..... 49
2.8.6. Seri İskele Kesmeleri..... 51
BÖLÜM 3. TMS77C82 SİMGESEL DİLİ VE KOMUT KÜMESİ..... 52
3.1. Simgesel Dilde Kaynak Program Formu..... 53
3.1.1. Kaynak Programlarda Kullanılan Sabitler. 55
3.1.2. Kaynak Programlarda Kullanılan Semboller 57
3.1.3. Kaynak Programlarda Kullanılan İfadeler. 58
3.1.4. Kaynak Programlarda Kullanılan Simgesel direktifler..... 60
3.2. TMS7000 Ailesine Ait Simgesel Dilin Komut Kümesi..... 61
3.2.1. Tanımlar..... 62
3.2.2. Adresleme Modları..... 62
3.2.2.1. Tek Saklayıcı Adresleme..... 63
3.2.2.2. İki Saklayıcı Adresleme..... 64
3.2.2.3. Giriş/Çıkış Genişletme Alanı Adresleme 64
3.2.2.4. İvedi Adresleme..... 65
3.2.2.5. Program Sayacına Göre Bağıl Adresleme. 65
3.2.2.6. Direk Adresleme..... 66
3.2.2.7. Saklayıcı Alanı Dolaylı Adresleme..... 66

3.2.2.8. Sıralı Adresleme..... 67
3.3. Komut Kümesi..... 67
BÖLÜM 4. TMS77C82 EMÜLATÖR KARTI..... 73
4.1. Emülatör Kartının Özellikleri..... 74
4.1.1. Kartın Bellek Düzeni..... 74
4.1.2. Zamanlayıcı ve Giriş/Çıkış İskeleleri... 75
4.1.3. Kart ile PC İle Seri Haberleşme..... 77
4.1.4. Kartın EPROM Programlayıcı Bölümü..... 78
BÖLÜM 5. İŞLETİM SİSTEMİ..... 79
5.1. Bellek İçeriği Görme İşlevi (M)..... 81
5.2. Bellek İçeriği Değiştirme İşlevi (W)..... 82
5.3. Saklayıcı İçeriği Görme ve Değiştirme (R). 83
5.5. Program Yükleme İşlevi (L)..... 84
5.4. Yürütme İşlevi (G)..... 84
5.6. EPROM Programlama (E) İşlevi..... 84
5.7. Kullanıcı Programları..... 86
SONUÇLAR VE ÖNERİLER..... 88
KAYNAKLAR..... 90
EKLER..... 91
ÖZGEÇMİŞ..... 96

ÖZET

Bu tezde, TMS77C82 mikrobilgisayarının simgesel dilde programlanmasını amaçlayan bir uygulama gerçeklenmiştir. Emülatör Kartı olarak adlandırılan sistemin, PC (Personel Computer: Kişisel Bilgisayar) ile birlikte çalışması düşünülmüştür.

TMS77C82 mikrobilgisayarı, TMS7000 mikrodenetleyici ailesinin geliştirilmiş bir uyarlamasıdır. 8-bit'lik yapıda olan bu mikrobilgisayarın, MİB'i (Merkezi İşlem Birimi), sabit ve uçucu belleklerin bir kısmı, ikili Giriş Çıkış iskeleleri, seri iletişim arabirimi UART (Universal Asynchronous Receiver-Transmitter), zamanlayıcıları ve kesmeleri aynı yonga üzerinde gerçeklenmiştir.

Yazılan işletim sistemi programı sayesinde, kullanıcı daha önce PC'de yazmış ve derlemiş olduğu simgesel dildeki programlarını karta yükleyip icra ettirebilecektir. Ayrıca, saklayıcı ve bellek içeriklerinin görülmesi ve değiştirilmesi,kullanıcı programlarının EPROM'a yazılması mümkündür.

Tezde ayrıca, TMS77C82 mikrobilgisayarı tanıtılmış ve bu mikrobilgisayarla gerçeklenen kartın nasıl kullanılacağı hakkında bilgiler verilmiştir.

SUMMARY

TMS77C82 EMULATION BOARD

Computers have been in general use since the 1950s. Formerly, digital computers were large, expensive machines used by governments and large businesses. The size and shape of digital computer has changed in the past few years owing to a new device called the microprocessor. The microprocessor is an IC (integrated circuit) that contains much of processing capabilities of a large computer. Computer uses a stored program. Smaller computers, called microcomputers, also use the stored program concept. A microcomputer contains a microprocessor and at least some form of semiconductor memory.

A microcomputer system is a digital computer. It is named as a *micro* because of its small size and low cost. The *microprocessor* generally forms the CPU section of a microcomputer system. The microcomputer contains all five basic sections: The input unit, the control and arithmetic units contained within the microprocessor, the memory unit, and the output unit.[1]

TMS77C82 is a microcomputer,which is produced by Texas Instruments. In this thesis, using this microcomputer an emulation module was developed. Device can emulate TMS 77C82 features on the board, so that it is called TMS77C82 Emulation Board. However, an EPROM programmer was realised at the same board. 8 Kbytes CMOS EPROMs or TMS77C82 microcomputers can be programmed on this board. In addition, TMS77C82 microcomputer is introduced in this thesis.

The TMS77C82 is an EPROM version of the 8-bit TMS7000 microprocessor family. This microcomputer contains 8 K-bytes of on-chip EPROM and is software compatible with the other microcomputers of TMS7000 family. Other features include 256 bytes of on-chip RAM, a flexible serial port (UART: Universal Asynchronous Receiver-Transmitter), three timers, programmable sense interrupts, eight adressing formats, and the same advanced register-to-register architecture that allows direct register arithmetic and logical operations without requiring the use of an accumulator (e.g. ADD R37,R228; add register 37 to 228 and store the result in register 228).[2]

The TMS77C82 is very useful for low-power applications, and for designs where program constraints are likely to change periodically. The CMOS fabrication, coupled with high performance CPU and internal peripherals, allows flexible system designs in industrial, automotive, computer, and telecommunications applications.

The 16-bit timers, with their associated 5-bit prescale 16-bit capture latch, and timer outputs, simplify A/D conversions, pulse width measurements and other time-critical application designs. For real-time applications where accuracy over long periods is essental, the Timer-1 output may be cascaded into the Timer-2 input to effectively form one 32-bit timer.[2]

The unique serial port can operate in any one of three modes: Asynchronous, Isosynchronous, or Serial I/O. Additional features of the serial port include a selectable protocol (data bits, parity, and stop bits), internal or external baud rate generation, and error detection flags. Direct networking for processor-to-processor communications is also supported through two multiprocessor protocols.

There are six prioritized interrupt levels on the TMS 77C82. Level 0 is the non-maskable reset, level-2 is associated with Timer-1, level-4 is associated with the serial port (receive, transmit and Timer-3), and level-5 is generated by Timer-2. Levels 1 and 3 are external interrupts with programmable edge/edge and level triggering, and rising/falling sense detection. All interrupts are routed through a user defined vector to the appropriate service routine; therfore, each service routine can be located anywhere in the TMS77C82 address space. There is a global interrupt enable bit in the status register, as well as individual interrupt enable bits for interrupts 1 through 5.[2]

The TMS77C82 can be programmed like any 8 K-bytes CMOS EPROM (27C64) on a wide variety of PROM programmers with the aid of an adapter socket.

When power cosumption is critical, the TMS77C82 can idle selectable sections of the microcomputer (e.g. Timer-1, Timer-2 or UART) and use power only where needed. Also, the entire processor can be halted while retaining the 256 bytes of internal RAM.

The instruction set is identical to that of all TMS7000 family members, allowing easy transition between members.

An operating system was developed for this emulation board. User can load programs which are developed previously. A PC is used in terminal status as a display and keyboard system. The communication between emulation board and PC is supported with serial way. User can see and change the contents of memory and CPU registers. In addition, programs inside of memory can be copied into EPROM on the same board.

An operating system must provide some features to user that mentioned below.

1. Writing and changing memory must be allowed.

2. Source program must be converted into machine language by using source program interpreter or high level interpreter.[3]

In the system, realized in this thesis, source programs are written and interpreted on the PC. After that, programs are loaded into emulation board by operating systems. The instructions of operating system are shown on Table 2.

Table 2. Operating System Instructions

INSTRUCTION		OPERATION
M: XXXX YYYY		Memory contents between XXXX YYYY addr.
W: XXXX		Writting data on XXXX address.
	/	Changing data on XXXX address.
	F	Searching forward.
	B	Searching backwords.
	.	Computing relative address.
R		Contents of register.
	/	Changing contents of register.
L: XXXX		Loading program XXXX address.
E: XXXX YYYY		Copying program from XXXX address to YYYY address. (For EPROM)
	B	Blank control from XXXX address.
	P	Program to YYYY address (EPROM)
	V	Compare contents of memory and contents of EPROM.
G: XXXX		Execute program on XXXX address.

In second chapter of this thesis, the internal architecture of the TMS77C82 microcomputer has been examined.

TMS77C82 Assembler Language instructions are mnemonic operation codes (or mnemonics) that correspond directly machine instructions. An assembly language program (source program) must be converted to a machine language program (object program) by a process called *assembling* before a computer can execute it. Assembling converts the mnemonics to binary values and associates those values with binary addresses, creating machine language instructions. In third chapter, the instruction set of TMS77C82 and assembler directives has been discussed.

In the fourth chapter of this thesis, features of Emulation board has been introduced.

The Operating System that developed for Emulation Board has been examined and the instructions of operating system has been given in the fifth chapter.

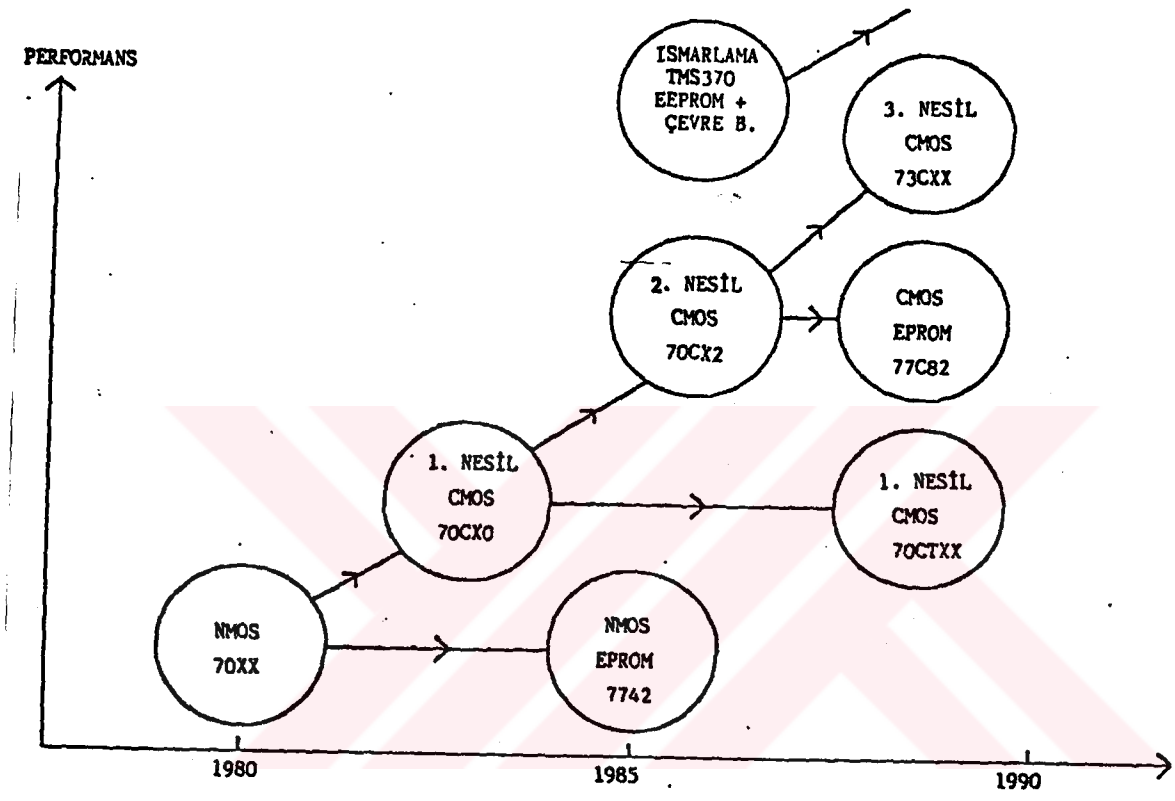
BÖLÜM 1
GİRİŞ

İlk mikroişlemci 1971 yılında Intel firması tarafından piyasaya sürüldü. Veri yolu dört bitlik olan bu mikroişlemci I4004'tür. Aynı firma 1973 yılında 8-bitlik I8080'i piyasaya sürdü. Başlamış olan bu akıma diğer mikroişlemci firmaları da katıldılar ve Motorola 6800'i Zilog Z80'i üretti. 1980'lere gelindiğinde üç mikroişlemci; I8085, M6802 ve Z80 endüstri standardı olarak yerlerini aldılar. Bu arada diğer büyük firmalar da kendilerine ait mikroişlemcileri yaptılar.[3]

Texas Instruments firması 1980 yılında 8-bitlik veri uzunluğuna sahip TMS7000 mikroişlemcisini piyasaya sürdü. Aynı firma daha sonraki yıllarda aynı komut kümesini kullanan, geliştirilmiş ilave özelliklere sahip mikroişlemciler üretti. Böylece TMS7000 mikroişlemci ailesi oluştu. Teknolojinin gelişmesiyle aynı kırmık içine MİB'e ilave olarak bellekler (RAM, ROM), giriş-çıkış iskeleleri de yerleştirildi. Böylece mikrodenetleyiciler ve mikrobilgisayarlar üretilmeye başlandı (Şekil 1.1.). Bu tezde Texas İns. firmasına ait TMS77C82 mikrobilgisayarı incelenmiş ve bir uygulama gerçeklenmiştir.

MİB'in giriş-çıkış arabirimlerinin ve belleklerin aynı kırmık üzerinde ve ucuza üretilmeleri sonunda mikrobilgisayarlı kontrol devreleri gerçeklenmesi olağan olmuştur. Artık otomatik çamaşır makinesi, yazar kasa hatta oyuncaklar için mikrobilgisayarlar kullanılmaktadır. Örneğin, bir otomatik çamaşır makinasının mikrobilgisayar ile denetlenmesi düşünülsün. Bu işi görecek mikrobilgisayar donanım olarak şu üç bölümden oluşabilir:

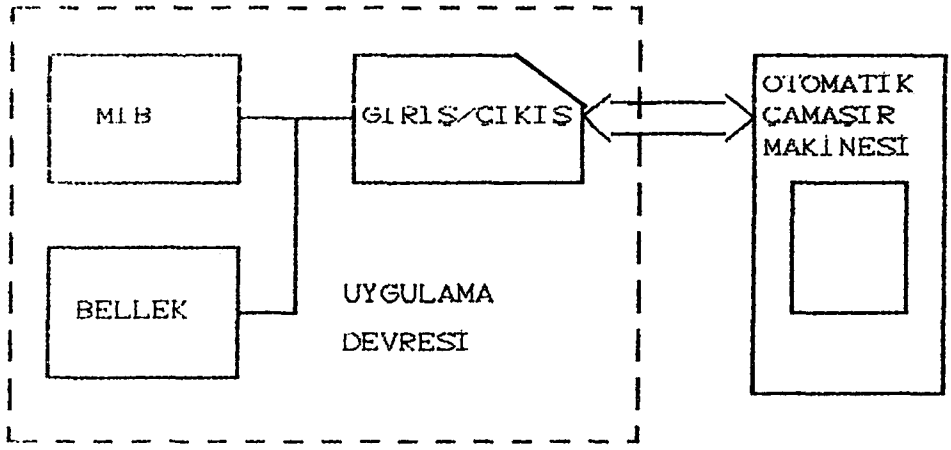
1. MIB (8-bit)
2. Bellek (Yaklaşık 2K-sekizli)
3. Giriş-Çıkış (8-bit paralel)



Şekil 1.1. TMS7000 mikroişlemci ailesi

Otomatik çamaşır makinesi ile mikrobilgisayar birlikte Şekil 1.2. 'de gösterilmiştir.

Tek yonga ile kurulabilen bu uygulama devresinin toplam maliyeti çok düşüktür ve bu nedenle kullanımı gittikçe yaygınlaşmaktadır. Ancak bu mikrobilgisayarı geliştirmek için uzun süren çalışma ve özel bir geliştirme bilgisayarı gereklidir.[3]



Şekil 1.2. Mikrobilgisayarlı bir uygulama devresi

Geliştirme bilgisayarı, uygulama devrelerinin donanım ve yazılım özelliklerini içeren, ancak işletim sistemi ve esnek donanım yapısına sahip olan bir bilgisayar sistemidir. Uygulama devresi gerçeklenmeden önce geliştirme bilgisayarı çalışma bilgisayarına bağlanır. Gerekli olan tüm yazılım çalışmaları tamamlanır. Sonuçta elde edilen program bir ROM'a (ya da EPROM v.b.) aktarılır.

Yukarıda bir örnekle anlatılan geliştirme bilgisayarının bir benzeri bu tezde gerçeklenmiştir. Texas Instruments firmasının ürettiği TMS77C82 mikrobilgisayarı kullanılarak gerçeklenen ve bu mikrobilgisayarın özelliklerini emüle eden devre, TMS77C82 Emülatör Kartı olarak adlandırılmıştır.

TMS77C82 mikrobilgisayarı, 8-bit'lik veri uzunluğuna sahip TMS7000 mikrodenetleyici ailesinin, geliştirilmiş ve 8K-byte EPROM ilave edilmiş bir uyarlamasıdır. Diğer bazı özellikleri şunlardır: 256 sekizli yonga-içi RAM, kullanışlı seri iskele (UART: Universal Asynchronous Receiver-Transmitter), üç tane zamanlayıcı, programlanabilir kesmeler, sekiz adresleme formatı ve akümülatör kullanmadan saklayıcılar arası aritmatik ve lojik işlemleri mümkün kılan, saklayıcı-saklayıcı mimarisi (Register to Register Architecture).[4]

5-bit'i kullanıcı tarafından belirlenebilen, 16-bit'-lik tutucu (Latch) ve zamanlayıcı, A/D dönüşümlerini, darbe genişlik ölçümlerini ve zamanlayıcı ile ilgili diğer tasarımları kolaylaştırmaktadır. Uzun periyotlar süresince alınan sonuçların temel alındığı, gerçek zaman uygulamaları için, birinci zamanlayıcının çıkışları ikincisinin girişlerine bağlanarak 32-bit'lik bir tane zamanlayıcı oluşturulabilir.

Seri iskele, bilinen üç moddan herhangi birinde çalıştırılabilir: Asenkron, isosenkron ve seri G/Ç. Seri iskelenin diğer özellikleri: Protokol seçilebilmesi (Veri biti, parite ve dur biti), hata saptama bayrakları ve veri akış hızının otomatik veya kullanıcı tarafından belirlenebilmesidir. Başka işlemcilerle haberleşme, çoklu işlemci protokolü ile desteklenmiştir.[4]

TMS77C82 mikrobilgisayarı altı kesme (Interrupt) seviyesine sahiptir: 0 maskelenemez (Reset), 1 ve 3 kesmeleri programlanabilir (kenar/kenar, seviye tetikleme ve yükselen/düşen), 2 kesmesi birinci zamanlayıcı ile ilgilidir, 4 seri iskele ile ilgilidir (veri gönder, veri al ve üçüncü zamanlayıcı), 5. kesme ikinci zamanlayıcı tarafından kullanılır. Tüm kesmeler kullanıcı tarafından belirlenen vektörlerle uygun servis programlarına yönlendirilebilirler. Ayrıca, tüm servis programları mikrobilgisayarın adres alanı içinde herhangi bir yere yerleştirilelebilir. Birden beşe kadar, her kesme için izin biti olduğu gibi, durum saklayıcısında bir de genel kesme izin biti vardır.

Güç tüketiminin kritik olduğu durumlarda TMS77C82'nin kullanılmayan bazı bölümleri, program kontrolü ile devre dışı bırakılabilir, (seçilebilir örn: zamanlayıcılar veya uart) ve sadece gerektiği yerlerde güç tüketilir. Hatta tüm işlemci, belleğindeki bilgiler kaybolmaksızın durdurulabilir.[4]

Ikinci bölümde TMS77C82 mikrobilgisayarının iç yapısı açiklanmış, yonga-içi bellek ve saklayıcılar, genel amaçlı G/Ç iskeleleri, düşük güç modları, kesmeler, zamanlayıcılar ve seri iskele ayrıntılı bir şekilde anlatılmıştır.

Üçüncü bölümde bu mikrobilgisayara ait simgesel dil ve bu simgesel dilin kullandığı adresleme modları ele alınmış, her komutun nasıl kullanıldığı ve icra edildiğinde bayrakları nasıl etkilediği belirtilmiştir.

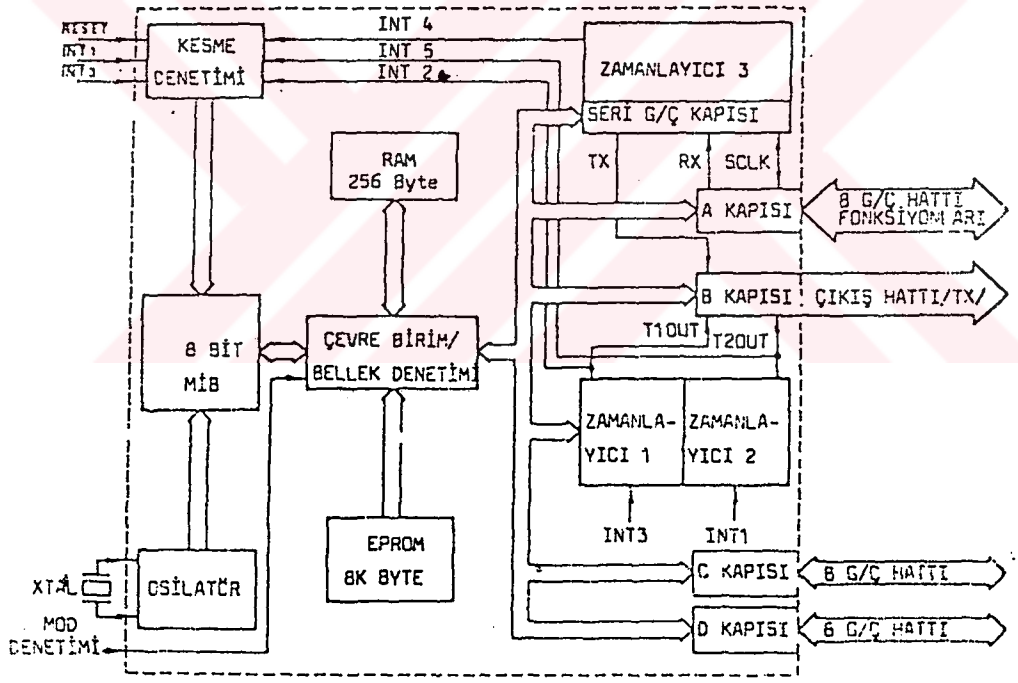
Tezin dördüncü bölümünde, emülatör kartının donanımı ile ilgili bilgi verilmiştir.

Sonuç bölümünde, kartın PC ile çalışmasını sağlayan işletim sistemi programı, gerçekleştirdiği işlevler ve nasıl kullanılacağı anlatılmıştır.

BÖLÜM 2

TMS77C82 MİMARİ YAPISI

Bir bilgisayar hangi boyutta olursa olsun temel olarak şu birimlerden oluşmuştur : giriş-çıkış, belleğe yazma, bellekten okuma, hesaplama ve karşılaştırma. Bu özellikleri sağlayan makineye bilgisayar denilir ve TMS77C82 mikrobilgisayarı tüm bu özellikleri içermektedir. Şekil 2.1'de bu mikrobilgisayara ait işlevsel blok diyagram görülmektedir.[4]



Şekil 2.1 TMS77C82 mikrobilgisayarına ait blok diyagram

Kullanılacağı uygulamaya göre dört farklı modda çalıştırılabilen mikrobilgisayarın uçları, her modda değişik işlevler gerçeklemektedir. Hatta, istenildiğinde mikrobilgisayar 8K-byte'lık bir eprom gibi kullanılabilir. Tablo 2.1'de çeşitli modlarda uçların hangi işlevleri gerçekleştirdiği gösterilmiştir.

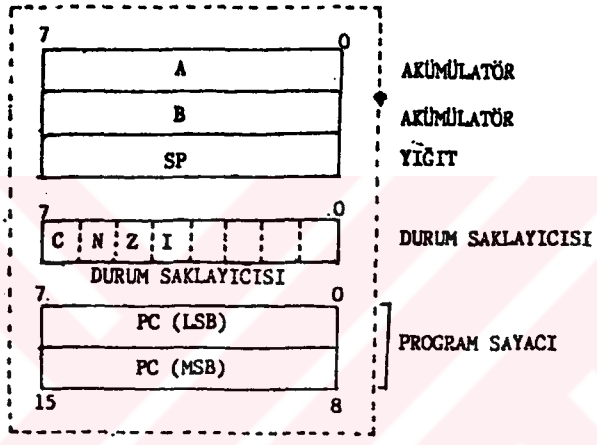
TABLO 2.1. TMS77C82 'nin bacak tanımlamaları

pin descriptions

OPERATION MODES					EPROM MODE		
SIGNAL	PIN NO. LCC	PIN NO. DIP	I/O	DESCRIPTION	SIGNAL	I/O	DESCRIPTION
A0 (LSb)	7	8	I/O	A0-A7 are general-purpose bidirectional pins.	A7	I	A3-A7, A12 are address lines.
A1	8	7	I/O		A6	I	
A2	9	8	I/O		A5	I	
A3	10	9	I/O		A4	I	
A4/SCLK	11	10	I/O	Data I/O/Serial port clock	A3	I	
A5/RXD	18	16	I/O	Data I/O/Serial port receiver	A12	I	
A6/EC2	16	15	I/O	Data I/O/Timer 2 event counter	$\overline{PGM}$	I	Program
A7/EC1	12	11	I/O	Data I/O/Timer 1 event counter	$\overline{G}$	I	Output enable
B0/T2OUT	3	3	O	B0-B3 are outputs. B4-B7 are outputs in Single-Chip mode and memory interface pins in all other modes. B0 and B1 are outputs for Timer 2 and Timer 1.			
B1/T1OUT	4	4	O				
B2	5	5	O				
B3/TXD	41	37	O	Data output/Serial port transmitter			
B4/ALATCH	42	38	O	Data output/Memory interface address latch strobe			
B5/R/$\overline{W}$	1	1	O	Data output/Memory read/write signal			
B6/$\overline{ENABLE}$	43	39	O	Data output/Memory interface enable strobe			
B7/CLKOUT	2	2	O	Data output/Internal clockout			
C0	31	28	I/O	Port C is a bidirectional data port. In Microprocessor, Peripheral-Expansion, and Full-Expansion modes, Port C is a multiplexed low address and data bus.	Q1	I/O	Q1-Q8 are bidirectional data lines.
C1	32	29	I/O		Q2	I/O	
C2	33	30	I/O		Q3	I/O	
C3	34	31	I/O		Q4	I/O	
C4	35	32	I/O		Q5	I/O	
C5	36	33	I/O		Q6	I/O	
C6	37	34	I/O		Q7	I/O	
C7	38	35	I/O		Q8	I/O	
D0	30	27	I/O	Port D is a bidirectional data port. In Microprocessor and Full-Expansion modes, it is the high address bus.	A8	I	A0-A2 and A8-A11 are address lines.
D1	29	26	I/O		A9	I	
D2	27	24	I/O		A11	I	
D3	26	23	I/O		A10	I	
D4	25	22	I/O		$\overline{E}$	I	Chip enable
D5	24	21	I/O		A0	I	
D6	22	20	I/O		A1	I	
D7	21	19	I/O		A2	I	
$\overline{INT1}$	14	13	I	Highest priority maskable external interrupt			
$\overline{INT3}$	13	12	I	Lowest priority maskable external interrupt			
$\overline{RESET}$	15	14	I	Reset	GND		V_{SS} for EPROM mode
MC	40	36	I	Mode control pin, V_{CC} for Microprocessor mode	V_{PP}		Program enable 12.5 V to program (0 V to verify)
XTAL2/CLKIN	19	17	I	Crystal input for control of internal oscillator	GND		V_{SS} for EPROM mode
XTAL1	20	18	O	Crystal output for control of internal oscillator			
V_{CC}	28	25		Supply voltage (positive)	V_{CC}		Supply voltage (6 V)
V_{SS}	23 39 44	40		Ground reference	GND		Ground reference

2.1. Yonga-Içi Bellek ve Saklayıcılar

En fazla 64 K-byte adres alanına sahip olan mikrobilgisayar, seçilen çalışma moduna göre, yonga-içi ve yonga dışı bellekleri kullanabilir. Diğer mikroişlemcilere göre az sayıda MIB saklayıcısı olan TMS77C82'nin, çok kullanışlı saklayıcı alanı ve giriş-çıkış alanı vardır. Şekil 2.2'de MIB saklayıcıları görülmektedir.



Şekil 2.2 MIB saklayıcıları

2.1.1. Saklayıcı Alanı (RF)

TMS77C82 mikrobilgisayarının içindeki, 256 byte'lık bellek (RAM), saklayıcı alanı olarak adlandırılır. Ilk iki saklayıcı, R0 ve R1,sırasıyla AKUMÜLATOR A ve AKUMÜLATOR B'yi gösterir. Birçok komutta,kapalı olarak A veya B akümülatörü kaynak veya varış saklayıcısı olarak kullanılır. Orneğin, STSP komutu yığın göstergesini B akümülatörüne yükler. Diğer komutlarda icra hızını arttırmak için A veya B akümülatörü kullanılabilir. Aksi söylenmedikçe, herhangi bir saklayıcı kaynak veya varış olarak kullanılabilir.[4]

2.1.2. Giriş-Çıkış Alanı (PF)

Giriş-çıkış alanı , P0-P255 saklayıcılarından

oluşmaktadır. Bellekte, >0100 ile >01FF adresleri arasına yerleştirilen bu saklayıcılar, kesme kontrolu, paralel G/Ç iskeleleri, zamanlayıcı kontrolu, bellek-genişleme kontrolu ve seri iskele kontrolu için kullanılır (Tablo 2.2). G/Ç saklayıcıları, kendi özel komutlarıyla kullanılırlar. Tek-yonga modu hariç, diğer modlarda, G/Ç saklayıcılarının hepsi kullanılamaz.[2]

Tablo 2.2. Giriş-çıkış saklayıcıları

i) P8,P9 ve P6 nın yüksek anlamlı dört biti Çevre birim genişletme, tam genişletme ve mikroişlemci modunda kullanılamaz.

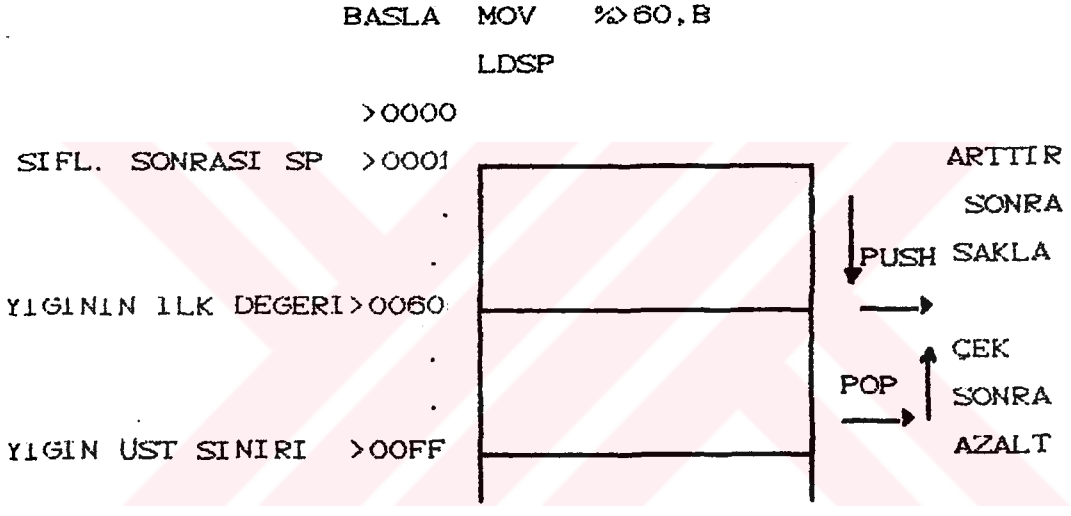
ii) P10 ve P11 tam genişletme ve mikroişlemci modlarında kullanılamaz.

REGISTER	ADDRESS	NAME	NOTE	FUNCTION
P0	>0100	IOCNT0	3	Interrupts 1, 2, and 3, expansion mode control
P1	>0101	IOCNT2		Polarity and edge/level control for $\overline{INT}$1 and $\overline{INT}$3
P2	>0102	IOCNT1	3	Interrupts 4 and 5
P3	>0103	–		Reserved
P4	>0104	APORT		A port data value
P5	>0105	ADDR		A port direction register
P6	>0106	BPORT	1	B port data value
P7	>0107	–		Reserved
P8	>0108	CPORT	1	C port data value
P9	>0109	CDDR	1	C port direction register
P10	>010A	DPORT	2	D port data value
P11	>010B	DDDR	2	D port direction register
P12	>010C	T1MSDATA	3	Timer 1 MSB reload register/MSB readout latch
P13	>010D	T1LSDATA	3	Timer 1 LSB reload register/LSB decrementer value
P14	>010E	T1CTL1	3	Timer 1 control register 1/MSB readout latch
P15	>010F	T1CTL0	3	Timer 1 control register 0/LSB capture latch value
P16	>0110	T2MSDATA	3	Timer 2 MSB reload register/MSB readout latch
P17	>0111	T2LSDATA	3	Timer 2 LSB reload register/LSB decrementer value
P18	>0112	T2CTL1	3	Timer 2 control register 1/MSB readout latch
P19	>0113	T2CTL0	3	Timer 2 control register 0/LSB capture latch value
P20	>0114	SMODE		Serial port mode control register
P21	>0115	SCTL0		Serial port control register 0
P22	>0116	SSTAT		Serial port status register
P23	>0117	T3DATA	3	Timer 3 reload register/decrementer value
P24	>0118	SCTL1		Serial port control register 1
P25	>0119	RXBUF		Receiver buffer
P26	>011A	TXBUF		Transmitter buffer
P27	>011B	–		Reserved
P28 P255	>011C->01FF	–		Peripheral expansion

2.1.3. Yığın Göstergesi (SP)

Yığın göstergesi, daima yığının en üst adresini gösteren 8 bitlik MIB saklayıcısıdır. Fiziksel olarak saklayıcı alanına yerleştirilmiştir.

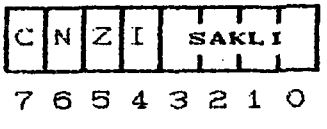
Yığın kullanıldığında, yığın göstergesi en son girilen veriyi gösterir. Sıfırlama sırasında, gösterge >01 ile yüklenir. Bu göstergenin değeri B akümülatörü (R1) ile LDSP komutu icra edilerek değiştirilir (Şekil 2.3). Bu özellik yığının saklayıcı alanı içinde istenen herhangi bir yere yerleştirilmesine imkan tanır. Yığın göstergesi B akümülatörüne STSP komutuyla yüklenebilir. Yığına veri atıldığında, göstergenin değeri otomatik olarak artar yığından veri çekildiğinde ise otomatik olarak azalır.



Şekil 2.3. Yığının oluşturulması

2.1.4. Durum Saklayıcısı (ST)

8-bitlik durum saklayıcısında, koşullu üç bit (elde: C, işaret:N, sıfır:Z) ve genel kesme izin biti vardır. C, N, Z bitleri aritmatik, bit öteleme ve koşullu dallanmalarda kullanılır.



Şekil 2.4. Durum saklayıcısı

Elde biti (Carry): Tüm aritmatik ve döndürme komutlarından etkilenir.

Negatif biti (Negative): Varış işleneninin en yüksek anlamlı biti ile yüklenir.

Sıfır biti (Zero): Komut icra edildiğinde varış işleneninin değeri sıfırsa bu bite 1 yüklenir.

Genel kesme izin biti (Global interrupt enable,I): Tüm kesmeleri yasaklar veya izin verir. EINT (Enable interrupts) komutu icra edildiğinde değeri 1 olur; DINT (Disable interrupts) komutu bu biti siler. Kesmelerin kabul edilebilmesi için, bu bitin 1 olması gerekir. Bu nedenle her kesmeye ait bayrak, bu bitin 1 ya da 0 olmasına göre belirlenir.

Koşullu sıçrama komutları, C, N ve Z durum bitleri ile, koşullu program akışlarına imkan tanır.

Sıfırlama sonrası durum saklayıcısındaki tüm bitler silinmiş olur. Diğer kesmelerde, durum saklayıcısı yığına itilir ve istendiğinde PUSHST, POPST komutlarıyla işlenebilir.

2.1.5 Program Sayacı (PC)

MIB saklayıcılarından olan program sayacı, iki tane 8-bitlik saklayıcıdan oluşur. Bu saklayıcılar, 16-bitlik adresin yüksek ve alçak anlamlı kısımlarını içirirler (PCH,PCL).

Her komut icra edildiğinde, program sayacı, 16-bitlik adres göstergesi gibi çalışarak, bellekteki o an icra edilen komut kodunu veya işleneni gösterir. Sıfırlama sırasında PC'nin değeri A (EAB) ve B (AAB) akümülatörlerine yüklenir.[4]

2.2. Yonga-İçi Genel Amaçlı G/Ç İskeleleri

Mikrobilgisayar, 8-bitlik paralel iskeleler şeklinde düzenlenmiş 32 G/Ç ucuna sahiptir. Bu iskeleler A B,C ve D olarak adlandırılır.[4]

İskele A: A0-A7 genel amaçlı giriş/çıkış uçlarıdır. Ayrıca, A5/RXD ve A4/SCLK uçları, seri haberleşme modu seçildiğinda seri yoldan veri alma ve saat giriş uçları olmaktadır. A7/EC1 ve A6/EC2 zamanlayıcı-1,2 için olay sayacı (Event Counter) olarak kullanılabilmektedir.

İskele B: B0-B3 çıkış, B4-B7 sadece tek-yonga modunda çıkış, diğer modlarda bellek arabirimi. Ayrıca,B0 ve B1, zamanlayıcı-1 ve zamanlayıcı-2 için çıkış uçlarıdır.

İskele C: Tek-yonga modunda genel amaçlı giriş/çıkış iskelesi, mikroişlemci ve genişletme modlarında, düşük anlamlı adresler ve veri yolu olarak kullanılır.

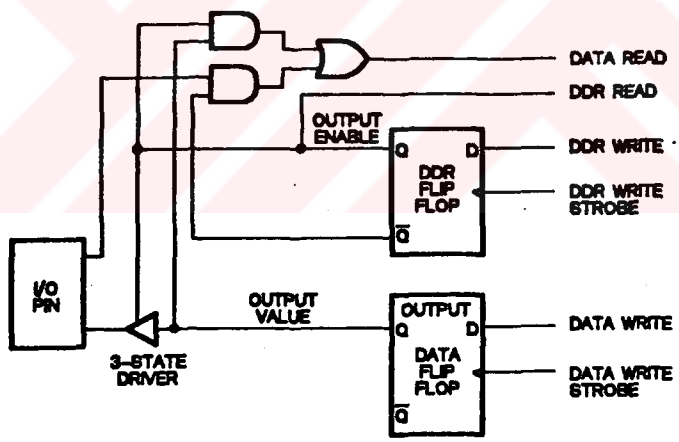
İskele D: Tek-yonga modunda genel amaçlı giriş/çıkış iskelesi, mikroişlemci ve tam genişletme modunda yüksek anlamlı adres yolu olarak kullanılır.

A, C ve D iskeleleri, G/Ç (giriş/çıkış) alanındaki *veri-yönlendirme* ve *veri* saklayıcıları ile kontrol edilir ve kullanılırlar. Sadece veri çıkışında kullanılan B iskelesinin, veri-yönlendirme saklayıcısı yoktur. Veri saklayıcısı, dış devreden gelen veya dış devreye gönderilecek sayısal veriyi içerir; Veri-yönlendirme saklayıcısı ise, o iskeleye ait her ucun giriş veya çıkış olduğunu tanımlar. Bu saklayıcıda, ilgili bite 0 veya 1 yazılarak, her uç kendi başına giriş ya da çıkış olarak belirlenmiş olur.

Veri-yönlendirme saklayıcısına yeni değerler yazmak

veri saklayıcısının içeriğini etkilemez. Böylece giriş ve çıkış olarak kullanılan tüm iskelelerde, veri akış yönü, sadece veri-yönlendirme saklayıcısının içeriği değiştirilerek belirlenebilir.

Donanım sıfırlaması sırasında, iskeleler yüksek empedanslı giriş durumuna gelecek şekilde, veri yönlendirme saklayıcıları silinir (0 yüklenir). Programlamada dış arabirimlere veri gönderme işlemi yapılırken önce veri saklayıcısının sonra da veri yönlendirme saklayıcısının istenen değerle yüklenmesi tavsiye edilir. Böylece daha önce veri saklayıcısına muhtemelen yüklenmiş olan istenmeyen verilerin, dış devrelere gönderilmesi önlenmiş olacaktır. Sıfırlama sırasında, B iskelesinin tüm bitleri 1 ile yüklenir. Şekil 2.5.'de her G/Ç ucunun lojik yapısı görülmektedir.



Şekil 2.5. G/Ç uçlarına ait lojik yapı

2.3 Bellek Modları

TMS77C82 mikrobilgisayarı 64 Kbyte'lık adres alanına sahiptir. Donanım ve yazılımla dört farklı bellek modu seçilebilir; tek-yonga, çevre birim genişletme, tam genişletme ve mikroişlemci modu. Mode kontrol (MC) giriş ucu Vcc'ye bağlandığında mikrobilgisayar mikroişlemci moduna geçer.[2][4]

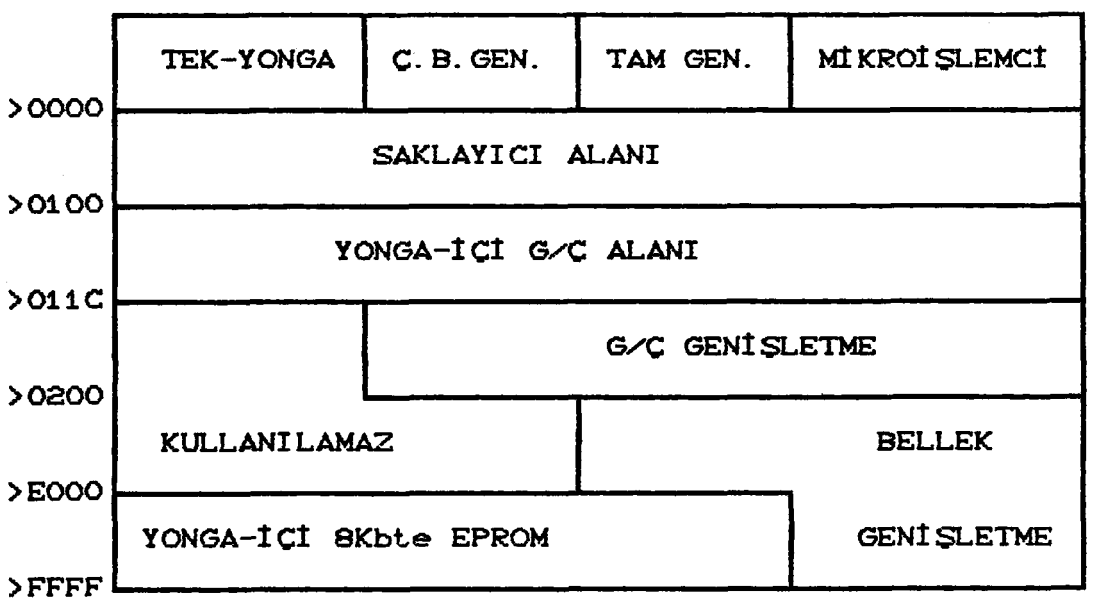
Bu uç Vss'ye bağlandığında diğer bellek modları, G/Ç alanındaki PO (IOCNTO) saklayıcısıyla seçilir (Tablo 2.3.).

MOD	MOD SEÇİM KOŞULLARI		
	MOD KONTROL UCU (MC)	IOCNTO SAKLAYICISI BİT 6 ve 7	
Tek-Yonga	Vss	0	0
G/Ç Genişletme	Vss	1	0
Tam-Genişletme	Vss	0	1
Mikroişlemci	Vcc	1	0

Not: X=1 veya 0 olabilir.

Tablo 2.3. Mod Seçim Koşulları

Sıfırlama sırasında, IOCNTO saklayıcısı 0 ile yüklenir. Tablo 2.4.'te modlara göre bellek haritası verilmiştir.



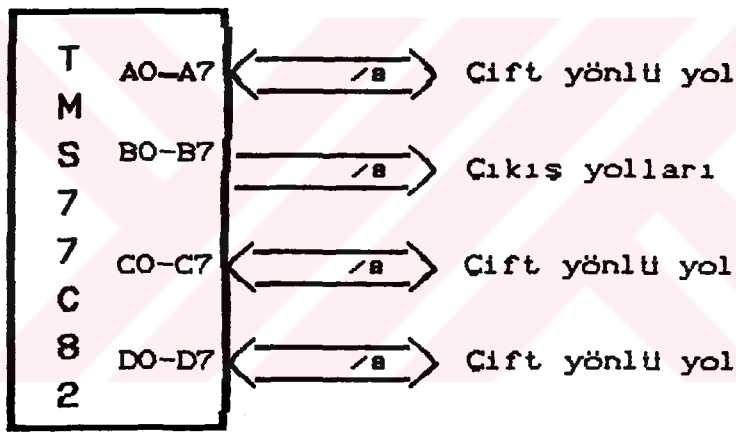
Tablo 2.4. Modlara göre bellek haritası

2.3.1 Tek-Yonga Modu

Bu modun seçilebilmesi için:

MC = Vss IOCNTO = OOXXXXXX

olmalıdır. Tek-yonga modunda mikrobilgisayar, miminimum ilave devre elemanına ihtiyaç gösterecek şekilde tasarlanmıştır. Programlamada uçucu bellek olarak saklayıcı alanı, sabit bellek olarak ise yonga-içi programlanabilir bellek (Eprom) kullanılır. TÜm G/Ç uçları çeşitli amaçlar için kullanılabilir. Örneğin: Tuş takımı okuma, göstergeleri sürme veya dış devrelerin kontrolu amacıyla kullanılabilir. G/Ç iskeleleri Şekil 2.6.'daki gibidir.



Şekil 2.6. Tek-Yonga Modu'nda G/Ç iskeleleri

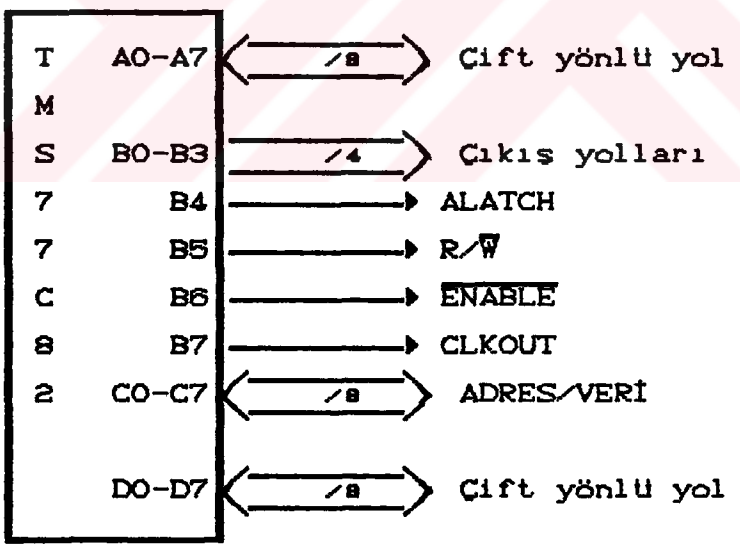
Tablo 2.4.'teki bellek haritasından görüldüğü gibi G/Ç saklayıcılarının bir kısmı ve ilave bellek birimleri adreslenemez. Eğer bu bölgelerden veri okunmaya kalkılırsa rasgele bir değerle karşılaşılır, bu bölgelere veri yazmak ise herhangi bir etki yaratmayacaktır.

2.3.2. Çevre Birim Genişletme Modu

Bu modun seçilebilmesi için:

MC = Vss IOCNTO = O1XXXXXX

olmalıdır. Bu modda, tek-yonga modunun G/Ç özellikleri tam-genişletme modunun bellek özellikleri görülmektedir. Yonga içi G/Ç alanı saklayıcıları kullanılmaz. Saklayıcıların bulunduğu bölgeye (>0100->01FF), yonga-dışı bellekler bağlanabilir. Mikrobilgisayar, ilave bellekler için gerekli kontrol çıkışlarını, B iskelesinin yüksek anlamlı 4 bitinden gönderir(Şekil 2.7.). G/Ç komutları icra edilirken, dış devreden veri alınmayacak olsa dahi iskele okunur (Örn: MOVP A,P6 da olduğu gibi). Eğer donanımın yapısı bu okumayı engelleyecek şekilde ise, G/Ç saklayıcılarının veri ile yüklenmesi için başka komutlar kullanılabilir (Örn: STA (Store A:Yaz A)). Yonga dışı adresleme yeteneği, diğer 8-bitlik mikroişlemciler için geliştirilmiş, yaygın kullanıma sahip G/Ç arabirimlerini destekler. G/Ç komutları ile bu arabirimler sanki yonga-içi'imiş gibi kullanılabilirler.[4]



Şekil 2.7. Çevre Birim genişletme modu

B iskelesi Şekil 2.7.'de görüldüğü gibi iki bölüme ayrılır: B0-B3 tek-yonga modunda olduğu gibi çıkış olarak ve B4-B7 uçları da ilave belleklerin kontrolu için kullanılır. Bu iskeleye ait veri saklayıcısı P6, tek

yonga modundaki işlevinden farklı çalışır. P6'ya bir veri yazıldığında, B0-B3 bu yeni veriyi çıkışa aktarır. B4-B7 uçları ise yeni verinin 4-bitlik anlamlı kısmını tanımaz, bellek kontrol işaretlerini göndermeğe devam eder. Dış bellek yazma çevrimi sırasında, 8-bitlik veri adresi >0106 olan dış belleğe de yazılır.

C iskelesinde adres/veri çoğullaması yapılır. Bu iskele 8-bitlik bir tutucu ile birlikte kullanılır. B4/ALATCH işareti, tutucunun G izin girişini sürecek şekilde bağlanır. B4/ALATCH lojik 1 seviyesinde olduğu zaman tutucunun D girişleri Q çıkışlarına aktarılır, lojik 0'a düştüğü zaman ise veriler tutulur. Bu şekilde adresler tutulduğunda, C iskelesi dış devreden veri alır ya da dış devreye veri yazar.

A ve D iskeleleri tek-yonga modunda olduğu gibi kullanılır.

2.3.3. Tam-Genişletme Modu

Bu modun seçilebilmesi için:

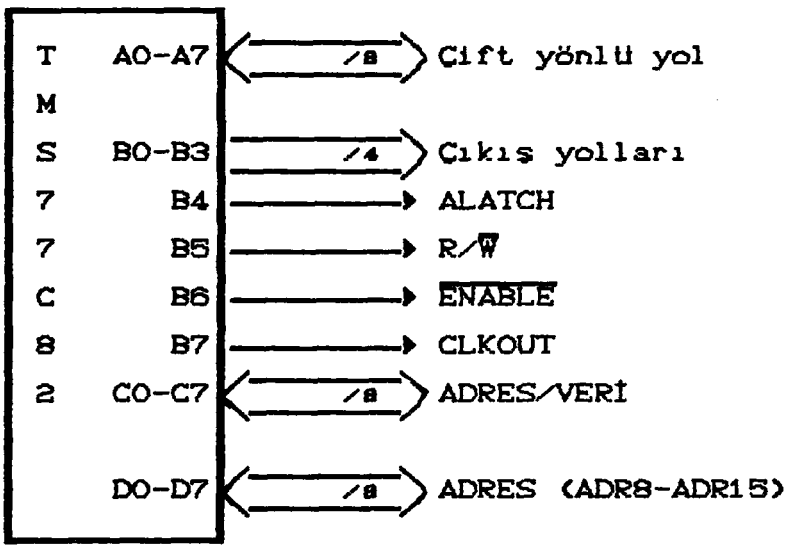
MC = Vss IOCNTO = 10XXXXXX

olmalıdır. Tam-genişleme modunda 64 Kbyte'lık bir alanı adresleyebilmek için 16 adres hattı kullanılır. Yonga dışı belleklere veri yazmak ve okumak için doğrudan, saklayıcı dolaylı ve sıralı adresleme formatlarını kullanan komutlar kullanılabilir. Bu mod G/Ç iskeleleri açısından çevre birim genişletme modu ile aynıdır; tek fark D iskelesinin yüksek anlamlı adresler için kullanılmasıdır. Böylece D iskelesinden G/Ç için faydalanılamaz (Şekil 2.8.). Çevre birim genişletme modunda olduğu gibi, G/Ç saklayıcı alanı artık direk yonga-içi değildir. Dolayısıyla G/Ç saklayıcılarına her erişimde bellek oku/yaz çıkışları aktif olur. Tablo 2.5.'te görüldüğü gibi D iskelesi veri saklayıcısı (DPORT) ve veri yönlendirme saklayıcısı (DDDR), bu modda yonga-dışı adreslerdedir.

D iskelesi giriş ya da çıkış işlevlerini gerçekleyebilecek şekilde >010A adresine bir tutucu bağlanarak çoğullama yapılabilir.

Tablo 2.5. G/Ç bellek haritası

			TEK-YONGA	Ç. B. GEN.	TAM GEN.	MİKROİŞ.
P0	>0100	İOCNT0	G/Ç Kontrol saklayıcısı 0			
P1	>0101	İOCNT2	G/Ç Kontrol saklayıcısı 1			
P2	>0102	İOCNT1	G/Ç Kontrol saklayıcısı 2			
P3	>0103	-	Saklı			
P4	>0104	APORT	A İskelesi veri sak.			
P5	>0105	ADDR	A İskelesi veri yönlendirme sak.			
P6	>0106	BPORT	B İs.veri			
P7	>0107	-	Saklı			
P8	>0108	CPORT	C İs.veri	G/Ç genişleme		
P9	>0109	CDDR	D İ.V.Y.S.			
P10	>010A	DPORT	D İskelesi veri			
P11	>010B	DDDR	D İs.ve.yönlen. sak			
P12	>010C	T1MSDATA	Zamanlayıcı1 sayaç tutma değeri (MSB)			
P13	>010D	T1LSDATA	Zamanlayıcı1 sayaç tutma değeri (LSB)			
P14	>010E	T1CTL1	Zamanlayıcı1 kontrol saklayıcısı1			
P15	>010F	T1CTL0	Zamanlayıcı1 kontrol saklayıcısı0			
P16	>0110	T2MSDATA	Zamanlayıcı2 sayaç tutma değeri (MSB)			
P17	>0111	T2LSDATA	Zamanlayıcı2 sayaç tutma değeri (LSB)			
P18	>0112	T2CTL1	Zamanlayıcı2 kontrol saklayıcısı1			
P19	>0113	T2CTL0	Zamanlayıcı2 kontrol saklayıcısı0			
P20	>0114	SMODE	Seri iskele mod kontrol saklayıcısı			
P21	>0115	SCTL0	Seri iskele kontrol saklayıcısı 0			
P22	>0116	SSTAT	Seri iskele durum saklayıcısı			
P23	>0117	T3DATA	Zamanlayıcı3 veri saklayıcısı			
P24	>0118	SCTL1	Seri iskele kontrol saklayıcısı 1			
P25	>0119	RXBUF	Alıcı tampon			
P26	>011A	TXBUF	Verici tamton			
P27	>011B	-	Saklı			
P28-P255	>011C->01FF		Çevre Birim Genişletme			



Şekil 2.8. Tam-Genişletme modunda G/Ç uçları

2.3.4. Mikroişlemci Modu

Bu modun seçilebilmesi için:

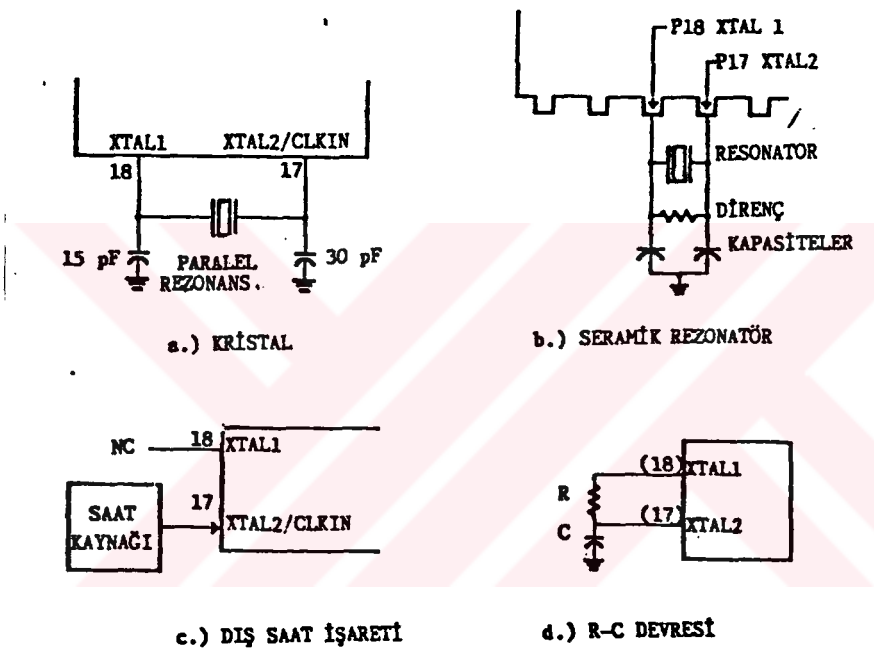
MC = Vcc IOCNTO = XXXXXXXX

olmalıdır. Mikroişlemci modu, yonga içi sabit bellekten istifade edilemeyen durumlar için düşünülmüştür. G/Ç uçlarının dağılımı tam-genişleme modundaki gibidir. Saklayıcı alanı ve G/Ç alanı haricinde tüm bellekler yonga dışı olarak adreslenir. Mikrobilgisayarın bu modda çalışabilmesi için MC ucu lojik 1'de tutulmalıdır. MC ucunun ne zaman lojik 1'e çekileceği hakkında herhangi bir sınırlama olmamakla beraber, bu değişimin reset sırasında yapılması tavsiye edilir. Eğer değişim, yonga-içi veya dışı okuma sırasında yapılırsa belirlenemeyen sonuçlar oluşabilir.[2]

2.4. Sistem Saati

Yonga-içi saat çevrim peryodu, $t_c(c)$ olarak adlandırılır ve, bir kristalden ya da diğer saat kaynaklarından elde edilir. TMS77C82 mikrobilgisayarı ayrıca R-C

devrelerle düşük güç-maskesi altında sürülebilir (Bölüm 2.4.1.). Bu durumda frekans hassasiyeti daha düşük olur. Yonga-içi saat, sürücü saat frekansını ikiye bölerek iç devrelerde kullanılacak saat frekansını oluşturur. Örneğin: 5 MHz'lik bir kristal yonga-içinde 2,5 MHz'lik frekans oluşturur. Şekil 2.9.'da sistem saat bağlantıları gösterilmiştir.



Şekil 2.9. Sistem saat bağlantıları

2.4.1. Düşük-Güç Maskesi

Mikrobilgisayar, düşük güç durma (Halt) modunda (Bkz. Bölüm 2.5) iken, saat maskelemesi özelliği ile farklı işlevlilik seviyeleri ve güç tüketiminin önlenmesi sağlanmıştır. Bu maskeler saat-açık (OSC-ON) ve saat-kapalı (OSC-OFF) olarak tanımlanır.

Saat-açık maskesi, düşük güç durma modunda iken sistem saatini açık tutmaya yarar. Mikrobilgisayar

sistem saati açık konumda olduğu için, bu moddan çıkarıldığında, hiçbir gecikme olmadan çalışmasına devam eder. Durma (HALT) modundan aktif moda geçişte gecikme istenmiyorsa bu maskeyi uygulamak yararlı olacaktır.

Eğer bu modda güç tüketiminin az olması çok önemli ise saat-kapalı maskesi kullanılabilir. Bu maske ile, durma moduna girildiğinde sistem saatinin çalışması iptal edilir. Böylece güç kaybı oldukça azalır. (ortalama 1 μA çekilir). Programlanabilir saat-kapalı maskesi, mikrobilgisayarın, R-C saat devreleriyle olduğu gibi, kristal ya da seramik rezonans saat kaynakları ile de sürülmesini destekler. Eğer bir R-C devresi bu maskeyle kullanılırsa durma moduna girer girmez salınıma başlar (Gecikme oluşmaz). Seramik rezonans ya da kristalli saat kaynakları sıfırlama cevabında düzenli salınımlar başlamadan önce 10 ms. kadar bir gecikmeye neden olur (Tablo 2.6.). Düşük güç modu, bu maske ve bu tür kaynaklar kullanıldığında salınımların düzenli hale gelebilmesi için gerekli 10 ms.'lik süre, sıfırlama sırasında mikrobilgisayara tanınmalıdır. (Durma-modu-saat-kapalı-maskesi'nden (HALT OSC-OFF) sadece sıfırlama ile çıkılabilir.)

Tablo 2.6. Düşük-güç maskeleri

MASKE	GÜÇ TÜKETİMİ	SAAT KAYNAĞI	SALINIM GECİKMESİ
SAAT KAPALI (OSC-OFF)	Çok	Seramik rezon., kristal	10 ms. gecikme
	Az	R-C devre, harici saat kaynakları	Gecikme yok
SAAT AÇIK	Az	Ser.rez., kristal veya diğer s.k.	Gecikme yok

2.5. Düşük-Güç Modları

TMS77C82 mikrobilgisayarının, az güç tüketimini öngören *uyandırma* (Wake-up) ve *durma* (Halt) modları vardır. Bu modlara IDLE komutu icra edilerek girilir. T1CTLO, T2CTLO ve SCTLO saklayıcılarındaki T1HALT, T2HALT ve SPH bitleri ile az güç tüketim mekanizması ayrı ayrı her zamanlayıcı ve seri iletişim birimi (UART) için çalıştırılabilir. Sıfırlama ($\overline{RESET}$) ya da uygun bir kesma ile bu modlardan çıkılabilir (Tablo 2.7.).

Tablo 2.7. Düşük güç modları

● Kesmelere çıkış için izin verilmeli.

●● IDLE komutu icra edilirken aktif olmayacak (inactive) şekilde programlanmamışsa, çıkılabilir.

MOD	BLOKLARIN İŞLEV DURUMLARI			MODA GİRİŞ	MODDAN ÇIKIŞ
	MIB	T1,T2,T3,UART	OSC		
UYANDIRMA	DURDURULMUŞ	T1,T2,veT3-UART tam aktif veya durdurulacağına dair programlanır	Aktif	IDLE	$\overline{RESET}$ $\overline{INT1}$● INT2●● $\overline{INT3}$ INT4●● INT5●●
DURMA SAAT ACIK	DURDURUL.	T1,T2 veT3-UART durdurulur	Aktif	IDLE	$\overline{RESET}$ $\overline{INT1}$ $\overline{INT3}$
DURMA SAAT KAPALI		T1,T2 veT3-UART durdurulur	Aktif değil	IDLE	$\overline{RESET}$

Aşağıdaki tabloda tüm bellek modlarına göre iskelelerin düşük güç modundaki durumları gösterilmiştir. Diğer tüm G/Ç uçları o anki durumlarını korurlar. Uyandırma moduna girildiğinde hiçbir çıkış yüksek empedans durumuna girmez.

Tablo 2.8. Düşük güç modlarında G/Ç iskeleleri

G/Ç UCU	TEK-YONGA	ÇEVRE BİRİM GENİŞLEME	TAM GENİŞLEME ve MİKROİŞ
CLKOUT(B7)	B7veri sak.de.	0	0
$\overline{\text{ENABLE}}$	B6veri sak.de.	1	1
R/$\overline{\text{W}}$	B5veri sak.de.	1	1
ALATCH	B4veri sak.de.	0	0
ADR/DATA (C)	G/Ç	X	X
HIGH ADR (D)	G/Ç	X	X

X = Belirsiz

Uyandırma modunda, T1HALT = 1 ve $\overline{\text{INT3}}$ ucu aktif ise zamanlayıcı-1 'in yakalama saklayıcısı (Capture latch) yeni bir değerle yüklenemez. T1HALT = 0 ve $\overline{\text{INT3}}$ aktif ise, INT3EN (Kesme-3 izin) bayrağına bakılmaksızın yakalama saklayıcısı yeni veri ile yüklenebilir.[2]

2.5. Kesmeler ve Sistem Sıfırlama

TMS77C82 mikrobilgisayarı, bir maskelenemez ve beş maskelenebilir olmak üzere altı kesmeye sahiptir (Tablo 2.7.). Her kesmenin kendine ait kesme önceliği vardır. Eğer iki kesme aynı anda oluşursa önceliği yüksek olanın servis programına dallanılır. Ayrıca, dış kesmelerin kabulü, (kenar/seviye ve polarite) yazılımla programlanabilir. Kesme-1 ve Kesme-3 dış devrelerden gelecek isteklere cevap verecek şekilde düzenlenmiştir. $\overline{\text{INT1}}$ ve $\overline{\text{INT3}}$ uçlarının aktif olmasıyla uygun programlara dallanma oluşur.

Tablo 2.7. Kesme tablosu

KESME	DIŞ/İÇ	KAYNAK	ÖNCELİK	VEKTÖR EA	VEKTÖR AA
$\overline{RESET}$	D	RESET ucu düşük	Hemen	>FFFE	>FFFF
$\overline{INT1}$	D	INT1 ucu aktif	Önc.1.	>FFFC	>FFFD
INT2	İ	Z-1 0'dan geçiş	Önc.2.	>FFFA	>FFFB
$\overline{INT3}$	D	INT3 ucu aktif	Önc.3.	>FFF8	>FFF9
INT4	İ	RX dolu,TX boş ve Z-3 0'dan geçiş	Önc.4.	>FFF6	>FFF7
INT5	İ	Z-2 0'dan geçiş	Önc.5.	>FFF4	>FFF5

2.5.1. Sistem Sıfırlama (Reset)

Sıfırlama, seviyesi 0 olan maskelenemeyen ve mikrobilgisayar tarafından hemen tanınan (hatta icra edilmekte olan bir komutun ortasında) bir kesmedir. Sıfırlama işlevinin icra edilebilmesi için, $\overline{RESET}$ ucunun en az 1,25 X t_c(c) kadar düşük seviyede (0 seviyesi) tutulması gerekir. Tablo 2.8.'de sistem sıfırlaması sonucunda iskelelerin ve MİB saklayıcılarının durumu görülmektedir.

Tablo 2.8.a. Sıfırlama sonrası MİB saklayıcıları

MİB SAKLAYICILARI	SIFIRLAMA SON.
DURUM SAKLAYICISI	Silinir
YIĞIN GÖSTERGESİ	>01 yüklenir
PROGRAM SAYACI	PC'nin eski değeri AveB PC'ye Sıfırlama vek.

Tablo 2.8.b. Sıfırlama sonrası G/Ç iskeleleri

PF SAK.	YER	SIFIRLAMA SON.
P5,P9P11	Veri Yön. Sak.	0 yüklenir (giriş)
P4,P8,P10	A,C,D Veri Sak.	Etkilenmez
P6	B İsk.Veri Sak.	1 yüklenir
P0,P1	IOCNT0,IOCNT2	0 yük. (7,6,3,2,0 bitleri ve IOCNT2 belirlenemez) INT4FLG,INT3FLG silinir
P2	IOCNT1	3,2,0 bitleri 0 yüklenir 7,6,5,4,1 bitleri etkilz. INT4FLG silinmez INT5FLG silinir
P21	SCTL0	7,2,1,0 bitleri 0 yüklen. 6 biti 1 yüklenir 5,4,3 etkilenmez
P22	SSTAT	6,1 bitleri 0 yüklenir 2,0 bitleri 1 yüklenir 7,5,4,3 bitleri eykilenmz
P24	SCTL1	6,5,4,3,2 bitleri 0 yük. 7,1,0 bitleri etkilenmez
P14	T1CTL1	6 biti 0 yüklenir diğer bitler etkilenmez
P18	T2CTL1	7,6 bitleri 0 yüklenir diğer bitler etkilenmez

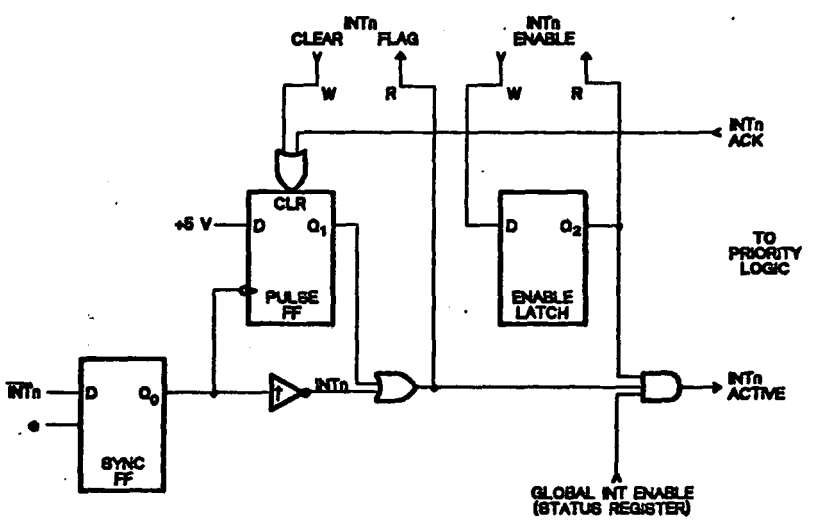
2.6.2. Kesme İşlemi

Dış kesme işaretinin mikrobilgisayar tarafından anlaşılabilmesi için şeklinin programlanması gerekir. İşarete ait formlar düşen kenar, düşen kenar seviye hissetmeli yükselen kenar, yükselen kenar seviye hissetmeli ve seviye hissetmeli olabilir. Bir kesmenin anlaşılması ve ilgili servis programına dallanma sırasıyla aşağıdaki işlem basamakları ile olur.[2][4]

a) Bir kesme oluştuğunda, yonga-içi saat tarafından seviyesi SYNC ikilisine girilir. İlgili kesmenin anlaşılabilmesi için, işaretin 1,25 X t_c (c) kadar aktif kalması gereklidir.

b) SYNC ikilisinin çıkışı darbe (PULSE) ikilisinin saat girişini oluşturur. Bu, kesme işaretinin kenar tespitinin yapıldığı ve darbe ikilisine sürekli 1 girildiği bir andır. Darbe ikilisinin çıkışları 1,25 saat çevrimi süresince tutulur. Eğer MİB tarafından tanınamadan kesme işareti kaldırılırsa, kesme oluştuğuna dair bilgi darbe ikilisinin Q1 çıkışında tutulur (Şekil 2.10.).

c) Kenar-hissetmeli kesmeler (Edge-sensetive interrupt) sadece darbe ikilisinin çıkışını sürer. Kesme geldiğinde (INTn aktif olduğunda), n. kesmeye ait izin biti ve genel kesme izin biti 1 ise uygun servis programına dallanılır. Şekil 2.10.'da dış kesmelerin mikrobilgisayar tarafından algılanmasına ait lojik diyagram verilmiştir.



Şekil 2.10. Yonga-içi kesme lojik arabirimi

d) Şekil 3.15.'te gösterildiği gibi, yonga-içi kesme lojiği, gelen n. seviyeye aktif kesme işaretini MİB'e gönderir. İcra edilen komut tamamlandığında, MİB aktif kesmeyi tanır ve INTn ACK (n. kesme tanındı) işaretini gönderir. Aynı yapı altında birden fazla kesme aynı anda geldiğinde, MİB gelen kesmeleri tanır fakat önceliği fazla olanın servis programına dallanılır. Örneğin; $\overline{INT2}$ ve $\overline{INT3}$ aynı anda aktif olsa $\overline{INT2}$ 'ye ait servis programına dallanılır.

e) MİB kesmeyi tanıdıktan sonra, INTn ACK işareti ile o kesmeye ait darbe ikilisini siler. Durum saklayıcısı ve program sayacının içerikleri yığına atılır. Durum saklayıcısı ve genel kesme izin biti silinir. MİB kesme öncelik lojiğinden kesme kodunu okuyarak, hizmet verilecek kesmeyi belirler. Sonra ilgili vektör, program sayacına yüklenir. Kesme vektörleri Tablo 2.7.'de verilmiştir.

2.6.3. Kesme Kontrolu

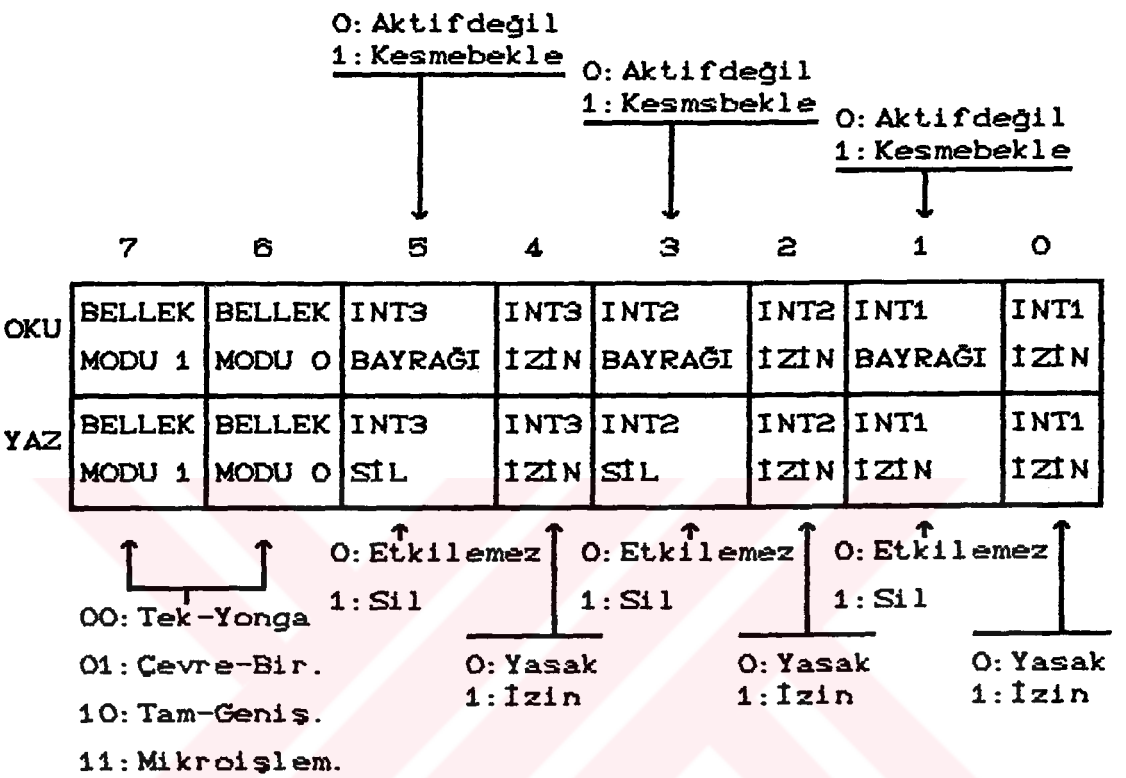
Giriş/Çıkış kontrol saklayıcıları (IOCNT0, IOCNT1, IOCNT2) kesme kontrol bitleri içerirler (Şekil 2.11.). Bu saklayıcılar okunurken ve yazılırkenfarklı değerler alırlar. Kontrol saklayıcılarında her kesme, bir bayrak biti ve bir de kesme izin biti ile temsil edilir. Ayrıca durum saklayıcısındaki genel kesme izin biti tüm kesmelere izin vermek ya da yasaklamak için kullanılabilir. Bir kesmenin MİB tarafından tanınması için üç koşul sağlanmalıdır:

a) IOCNT0, IOCNT1 ya da IOCNT2 saklayıcılarındaki izin bitlerinden ilgili olanına 1 yazılmış olmalı.

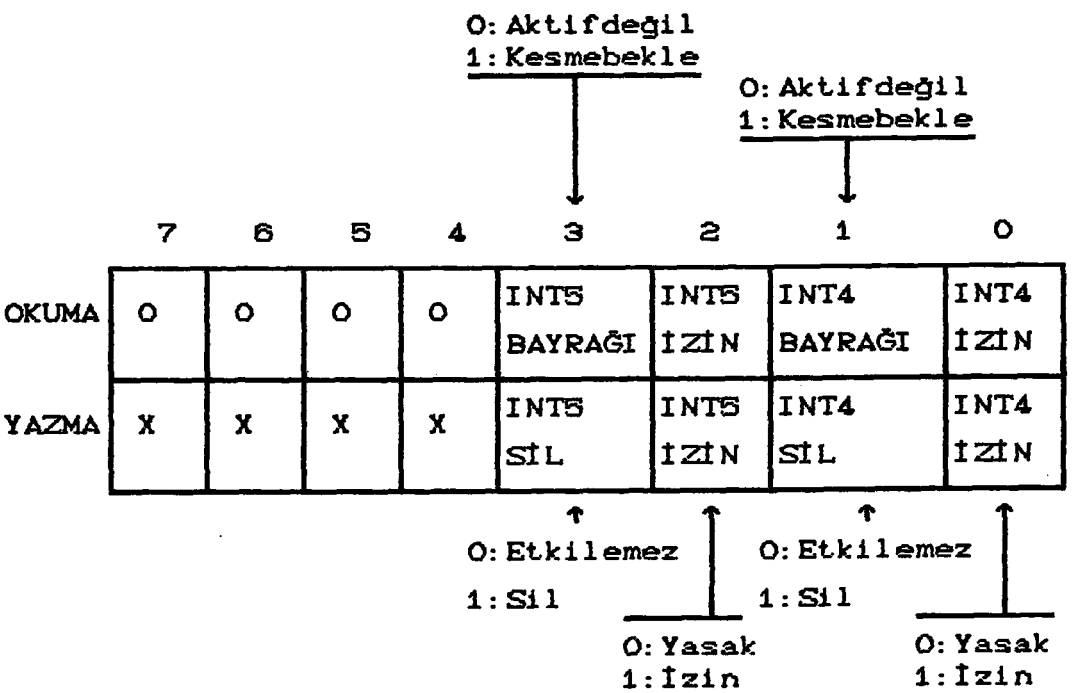
b) EINT (Enable Interrupt: Kesmelere İzin) komutu ile genel kesme izin biti 1 yapılmış olmalı.

c) Aynı anda gelen kesmeler içinde kabul edilmesi istenilenin önceliğinin en yüksek olması gerekir.

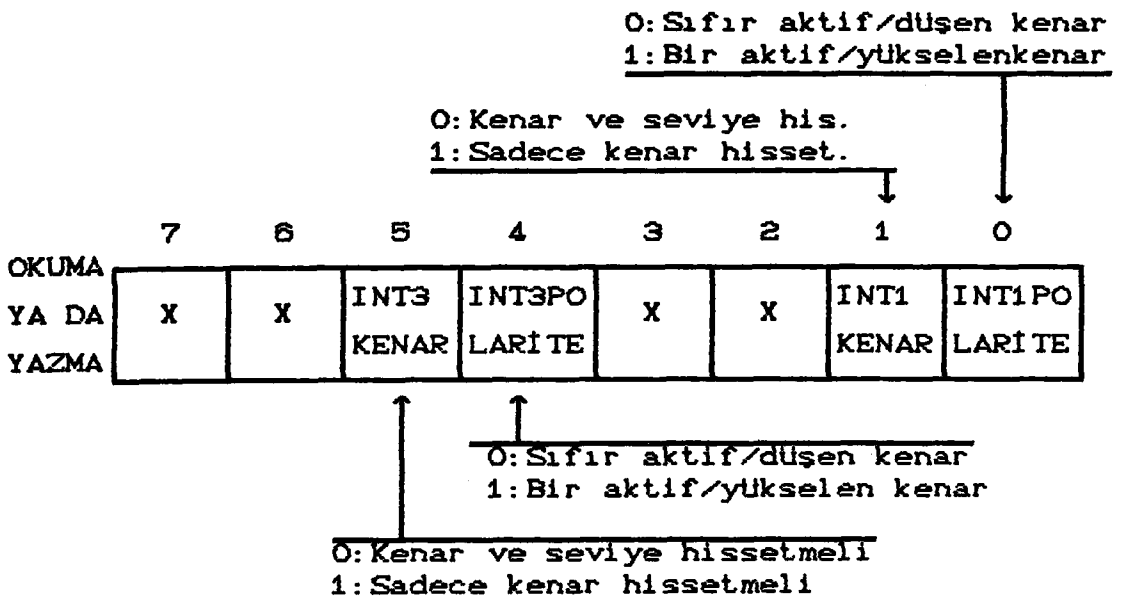
IOCNTO:



IOCNT1:



IOCNT2:



Şekil 2.11. Kesme kontrol saklayıcıları

Kesme bayrakları yazılımla da kontrol edilebilir. Örneğin aşağıdaki program parçası Kesme-1 oluşana kadar mikrobilgisayarın beklemesini sağlar.

```
BEKLE     BTJZP  %>02,PO,BEKLE
```

Böylece, kesme uçları çok kısa darbeleri (1,25Xt_c(c)) dahi hissedebilen giriş iskeleleri olarak kullanılabilir.

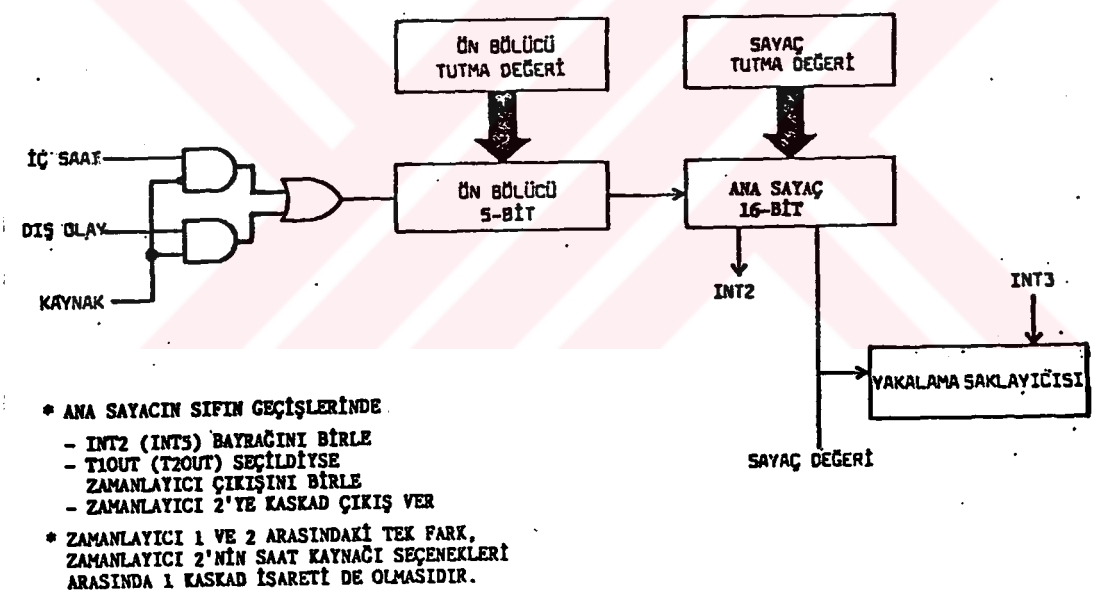
2.6.4. Çoklu Kesmeler

Bir kesme tanındığında, durum saklayıcısındaki kesme izin biti silinir ve servis programına dallanılır. Böylece, kesme servis programı icra edilinceye kadar diğer gelebilecek kesmeler MİB tarafından tanınmaz. Servis programı RETI komutu (Return from interrupt: Kesmeden dön) ile sonlandırılarak icra edildiğinde eski durum saklayıcısı yığından çekilir. Böylece genel kesme izin biti yine 1 olur ve artık gelen kesmeler tanınır.[4]

Kesme servis programı ile içiçe kesmelerin oluşmasına izin verilebilir. Bunun için, servis programı EINT (Enale Interrupt:Kesme izin) komutunu içermelidir.

2.7. Programlanabilir Zamanlayıcı/Olay Sayıcılar

TMS77C82 mikrobilgisayarı, yonga-içi, herbirine ait dur/başla kontrol biti olan, üç tane zamanlayıcıyla desteklenmiştir. Zamanlayıcı-1 ve zamanlayıcı-2 (Şekil 2.12.) 16-bitlik okunabilir sayaç, 16-bitlik değer yükleme saklayıcısı, 16-bitlik yakalama saklayıcısı ve 5-bitlik yüklenebilir ön bölücü ve saklayıcısından oluşmaktadır.



Şekil 2.12. Zamanlayıcı-1,2 işlevsel blok şeması

Zamanlayıcı-3, 8-bitlik okunabilir sayaç, 8-bitlik değer yükleme saklayıcısı, 2-bitlik ön bölücü ve saklayıcısından oluşur. Zamanlayıcı-3 genel amaçlı zamanlayıcı ya da seri iskele için veri akış hızı kaynağı (Baud rate generator) olarak kullanılabilir.

2.7.1. Zamanlayıcı Kontrol Saklayıcıları

Zamanlayıcı-1, 2'nin kontrolu ve önbölücü (Prescale) değerleri, T1CTL0 (P15), T2CTL0 (P19), T1CTL1 (P14) ve T2CTL1 (P15) kontrol saklayıcılarıyla belirlenir. Bu saklayıcılara sadece veri yazılabilir, yazılmış veya program akışı sırasında değişmiş değerler daha sonra okunamaz. Eğer, T1CTL0 okunmaya kalkılırsa, birinci zamanlayıcıya ait yakalama saklayıcısının (Capture latch) az anlamlı sekizlisi okunmuş olur. Aynı şekilde T2CTL0 okunduğunda ikinci zamanlayıcının yakalama saklayıcısı okunur. T1CTL1 ve T2CTL1 okunduğunda ise zamanlayıcı-1,ve 2'ye ait sayaç tutma saklayıcısının yüksek anlamlı sekizlisi okunmuş olur. Kontrol ve önbölücü saklayıcıları sadece yazılabilir oldukları için, ANDP, ORP ve XORP komutları bu saklayıcılarla kullanılamaz. Aşağıdaki komutlar kontrol saklayıcıları ile kullanılabilir.

MOVP	%>XX,Pn	STA	%>01xx
MOVP	A,Pn	STA	*Rn
MOVP	B,Pn	STA	>01xx(B)

%>XX = 8-bit hex ivedi veri
>01xx = 16-bit giriş-çıkış genişletme alanı için hex adres.
A = A akümülatörü
B = B akümülatörü
Pn = Giriş-Çıkış genişletme alanı saklayıcısı
Rn = Genel amaçlı saklayıcı

Aynı komutlar zamanlayıcılara ait veri saklayıcılarını yazmak için de kullanılabilir. Bu saklayıcılar T1LSDATA, T1MSDATA, T2LSDATA, T2MSDATA'dır. Kontrol ve veri saklayıcıları Şekil 2.13'te gösterilmiştir.

T1CTL1: P14

	7	6	5	4	3	2	1	0
OKUMA	SAYAÇ TUTMA DEGERİ (Yüksek anlamlı)							
YAZMA	X	T2OUT	X	X	X	X	X	X

T1CTL0: P15

7	6	5	4	3	2	1	0
SAYAÇ YAKALAMA DEGERİ (Az anlamlı)							
BASLA	KAYNAK	T1HALT	ÖNBÖLÜCÜ				

T2CTL1: P18

	7	6	5	4	3	2	1	0
OKUMA	SAYAÇ TUTMA DEGERİ (Yüksek anlamlı)							
YAZMA	KASKAT	T2OUT	X	X	X	X	X	X

T2CTL0: P19

	7	6	5	4	3	2	1	0
OKUMA	SAYAÇ YAKALAMA DEGERİ (Az anlamlı)							
YAZMA	BAŞLA	KAYNAK	T2HALT	ÖNBÖLÜCÜ				

Şekil 2.13. Zamanlayıcılara ait kontrol saklayıcıları

Zamanlayıcı kontrol saklayıcısı T1CTL0 veya T2CTL0'ın yedinci bitine (Başla biti) 0 yazıldığında zamanlayıcı devre dışı bırakılır ve o anki sayaç değeri dondurulmuş olur. Ayrıca zamanlayıcı-1'in İNT2 bayrağı veya zamanlayıcı-2'nin İNT5 bayrağı (hangisi seçilmişse) 0 ile yüklenir. BAŞLA bitine 1 yazıldığında ilgili tutuculara önbölücü ve sayaç tutma değerleri yazılır ve sayma başlar.

Sayacın ve önbölücünün birlikte her 0'dan geçişlerinde ilgili kesme bayrağı 1 ile yüklenir, önbölücü ve sayaç tutma değerleri tekrar saklayıcılara yazılır. Her iki zamanlayıcı 16-bitlik yakalama saklayıcıları ile bağlantılıdır. Bu saklayıcılar yakalama işareti aktif olduğunda o andaki sayaç değeri ile yüklenirler. Kesme-3 aktif olduğunda zamanlayıcı-1 için, kesme-1 aktif olduğunda ise zamanlayıcı-2 için yakalama işareti aktif hale getirilmiş olur. Durdurulmuş modda her iki saklayıcı devre dışı bırakılmış olur.[2][4]

Olay sayıcı (Event Counter): Zamanlayıcı-1 ve 2 olay sayıcı modunda ise, A7 ve A6 uçları sırasıyla birinci ve ikinci zamanlayıcıların darbe kaynakları olur. Bu uçlardan girilecek frekans $f_{osc}/4$ 'ten büyük olmamalıdır.

2.7.2. Zamanlayıcıların Kaskad Bağlanması

TMS77C82 mikrobilgisayarının içindeki iki zamanlayıcı daha büyük bir zamanlayıcı elde etmek amacıyla kaskad bağlanabilir. Bu durumda zamanlayıcı-2 için saat girişi A6/EC2 ucundan yapılır. T2CTL1 (P18) saklayıcısının yedinci biti KASKAD bitidir.

Bit 7 = 0: Bu bitin 0 olmasıyla kaskad bağlamadan vazgeçilmiş olunur. T2CTL0 saklayıcısının altıncı biti ile ikinci zamanlayıcı için kaynak seçimi yapılabilir.

Bit 7 = 1: Bu durumda zamanlayıcı-1'in çıkışı ikinci zamanlayıcının saat kaynağı durumuna gelir. Eğer bu bit 1 yapılmışsa ikinci zamanlayıcıya ait KAYNAK biti herhangi bir etki yaratmaz.

2.7.3. Zamanlayıcı ve Önbölücü İşlemi

Zamanlayıcı saat darbeleri yonga-içi veya yonga-dışı olsun, 5-bit modülo-N sayacı ile bölünebilir. Önbölücü

değeri kontrol saklayıcısının az anlamlı beş biti ile belirlenir. Sayaç azaltmalı çalışır ve her 0, >FF geçişinde alt taşma (underflow) oluşur. Zamanlayıcı ve önbölücü aşağıdaki işlem sırasına göre çalışır:

1. Zamanlayıcı çalışmadan önce sayaç ve önbölücü belirlenen değerlerle yüklenir.

2. Her darbede önbölücü bir azalır.

3. Önbölücünün sıfır geçişlerinde sayaç bir azaltılır. Eğer sayaç ve önbölücü birlikte sfırdan geçerlerse kesme oluşur.

4. Önbölücü ve sayaç sıfırdan geçtiklerinde tekrar belirlenen değerlerle yüklenirler.

2.7.4. Zamanlayıcı Kesmeleri

Önbölücü ve sayaç birlikte sıfırdan geçtiklerinde kesme oluşur. Önbölücü ve sayaç tekrar belirlenen değerlerle yüklenirler. Birinci zamanlayıcı ikinci kesme, ikinci zamanlayıcı için beşinci kesme ayrılmıştır. Oluşturulmak istenen iki kesme arasındaki zaman aşağıdaki formülle hesaplanabilir:

$$t_{int} = t_{clk} * (PR+1) * (TR+1)$$

t_{int} : İki kesme arasındaki zaman
t_{clk} : Zamanlayıcı giriş saatinin peryodu
PR : 5-bit önbölücü değeri
TR : 16-bit sayaç tutma değeri

Kesme-3 girişi düşen kenarlı bir işaretle aktif hale getirildiğinde birinci zamanlayıcıya ait 16-bitlik sayaç değeri yakalama saklayıcısına yüklenir. Aynı olay kesme-1 geldiğinde ikinci zamanlayıcı için oluşur.

2.7.5. Zamanlayıcı Çıkış İşlevi

Zamanlayıcı-1 ve 2 için sırasıyla B1 ve B0 uçları çıkış olarak kullanılabilir. Sayaç her sıfırdan geçişte çıkış bir önceki konumunu değiştirir (Sıfırsa bir, birse sıfır olur). Bu işlevin kontrolu T1CTL1 ve T2CTL1 saklayıcılarındaki T1OUT ve T2OUT bitleri ile yapılır.

Zamanlayıcı çıkış işlevine izin verildiğinde, B0 ve/veya B1 uçlarındaki lojik değerler, B iskelesi veri saklayıcısına herhangi bir-şey yazmakla değiştirilemez. B iskelesi okunduğunda, B0'daki değer görülebilir. Böylece o andaki zamanlayıcı çıkışı okunabilir.

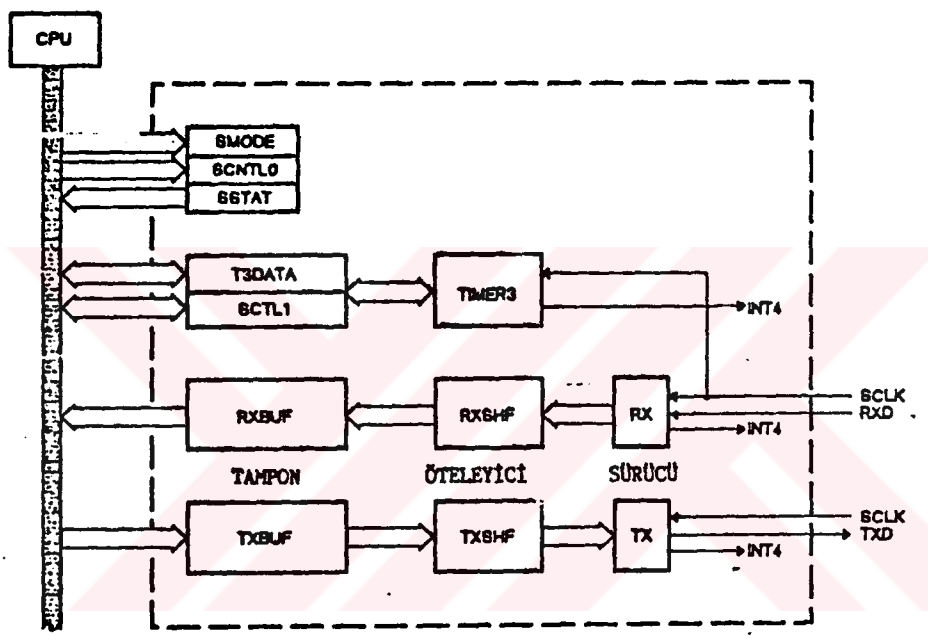
2.8. Seri İskele

TMS77C82 mikrobilgisayarı, G/Ç ve iletişim kapasitesini arttırmak amacıyla seri iskele ile donatılmıştır. Birçok mikroişlemci ya da bilgisayarla haberleşebilecek esnek yapıya sahip olan bu iskele, alma (RX), gönderme (TX) saklayıcıları ve veri akış hızı üreten zamanlayıcıdan (T3) oluşmaktadır. Seri iskelenin kontrolü ve aktif hale getirilmesi aşağıdaki saklayıcılarla olur:

Tablo 2.8. Seri iskele kontrol saklayıcıları

SAKLAYICI	ADRES	ADI	TİPİ	İŞLEVİ
P20	>0114	SMODE	Y/O	Seri iskele modu
P21	>0115	SCTL0	Y/O	Seri iskele kontrol 0
P22	>0116	SSTAT	OKU	Seri iskele durum
P23	>0117	T3DATA	O/Y	Zamanlayıcı-3 veri
P24	>0118	SCTL1	Y/O	Seri iskele kontrol 1
P25	>0119	RXBUF	OKU	Alıcı tampon
P26	>011A	TXBUF	YAZ	Gönderme tamponu

Seri iskelenin arzu edilen iletişim formunda olması için yazılımla hazır duruma getirilmelidir. İlgili kontrol kelimeleri ile, veri hızı, karakter uzunluğu, tek/çift/eşlik yok, stop biti sayısı v.b. seçilebilir. Şekil 2.14., seri iskele işlevsel blok diyagramını göstermektedir.



* UART çift yönlü (full duplex) çalışabilir.

Şekil 2.14. Seri iskele işlevsel blok diyagramı

2.8.1. Seri İskele Saklayıcıları

Seri iskele, G/Ç genişletme alanındaki kontrol saklayıcıları ile yönlendirilir. Bu saklayıcılar Tablo 2.8. de görülmektedir.

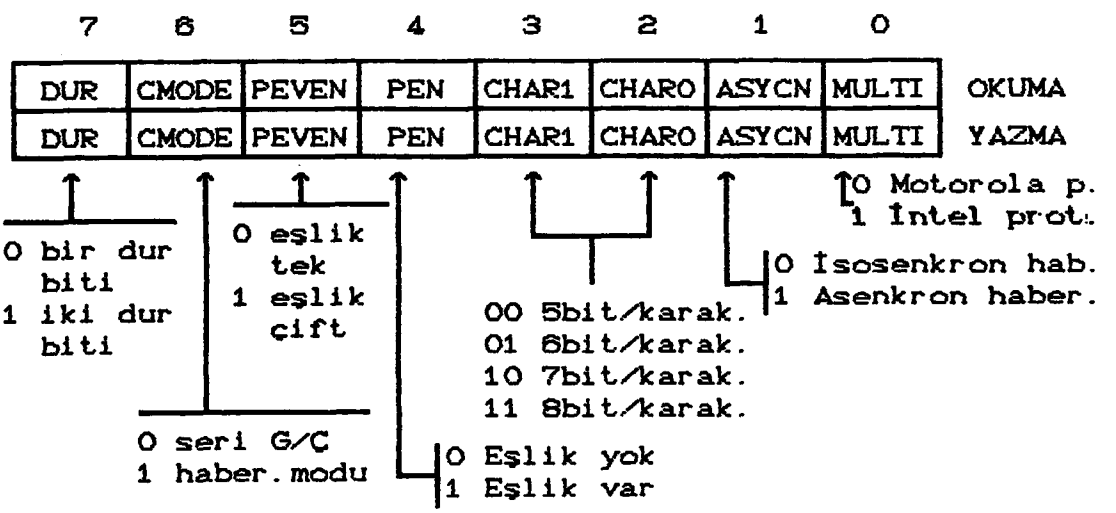
Seri Mod Saklayıcısı, SMODE, iletişim tipini belirlemek için kullanılır. İletişim tipleri şunlardır: Asenkron, isosenkron, seri G/Ç.

Seri İskele Kontrol-0 Saklayıcısı, SCTL0, veri gönderme/veri alma izin (TX/RX enable), hata bayraklarının silinmesi gibi işlevlerinin kontrolu için kullanılır. Durum saklayıcısı, SSTAT, sadece okunabilir ve seri iskelenin durumunu bildirir. T3DATA, üçüncü zamanlayıcıya ait veri saklayıcısıdır. RXBUF, alınan veriyi içeren sadece okunabilir bir saklayıcıdır. Bu saklayıcı devre-içi öteleme saklayıcısı RXSHF ile iki defa tamponlanmıştır. Böylece MİB, yeni veri yazılmadan eski veriyi okuyabilir. TXBUF ise sadece yazılabilir bir saklayıcıdır ve MİB'teki veriyi göndermeye yarar. RXBUF gibi iki defa tamponlanmıştır.

Mikrobilgisayar üzerindeki TXD ve RXD uçları (B3/TXD, A5/RXD), seri iskele kullanılmadığında G/Ç iskelesi (paralel) olarak kullanılabilir.

2.8.1.1. Seri Mod Saklayıcısı (SMODE)

Seri Mod Saklayıcısı, SMODE, veri alma ve gönderme kontrolu amacıyla kullanılır. Karakter formu, iletişim tipi (Asenkron, İsosenkron, Seri G/Ç) bu saklayıcının içeriği ile belirlenir (Şekil 2.15.).



Şekil 2.15. Seri Mod Saklayıcısı (SMODE)

2.8.1.2. Seri Kontrol Saklayıcısı-0 (SCLTO)

Seri iskeleden veri alma ve veri gönderme izni, hata bayraklarının silinmesi ve yazılımla sıfırlama kontrolu bu saklayıcı ile yapılır (Şekil 2.16.).

7	6	5	4	3	2	1	0	
SPH	UR	0	0	0	RXEN	SCLKEN	TXEN	OKUMA
SPH	UR	X	ER	X	RXEN	SCLKEN	TXEN	YAZMA

0 gönderme yasak
1 gönderme izin

0 hata bayrakla. silinmeyecek
1 hata bayrakla. silinecek

0 A4 G/Ç ucu
1 A4 seri saat giriş

0 seri is. izin
1 sıfırla

0 alıcı devre dışı A5 ucu G/Ç
1 alıcı çalışmada

0 durdurulmuş modda seri iskele ve zam.3 aktif
1 aktif değil

Şekil 2.16. Seri İskele Kontrol Saklayıcısı-0

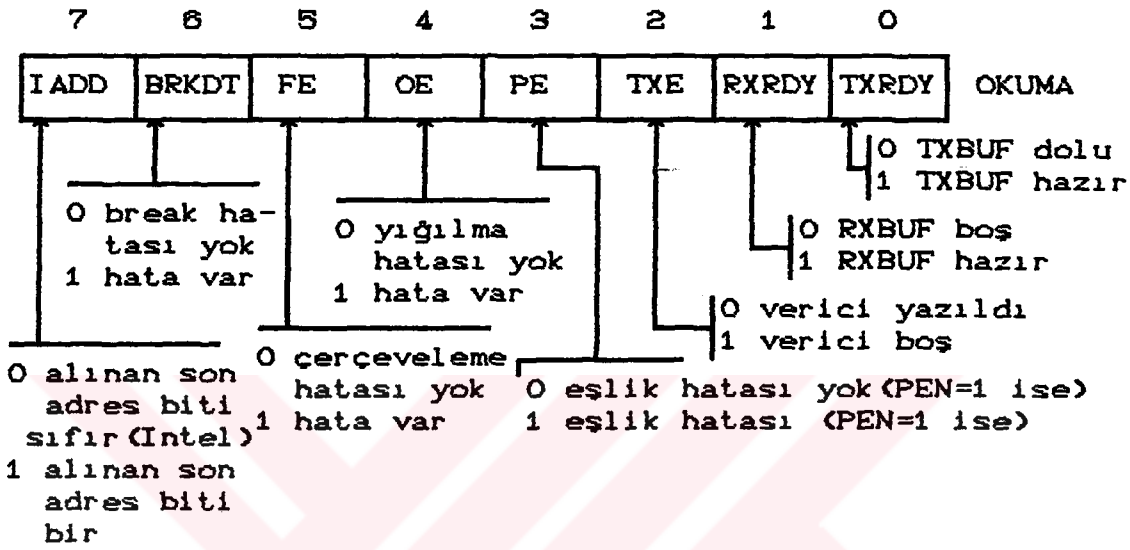
Hata sıfırlama biti (Error Reset: ER), seri iskele çalıştırıldığında hata bayraklarını silmek için kullanır. Eğer bu bit 0 ise hata bayrakları hiçbir durumdan etkilenmez, 1 ise SSTAT saklayıcısındaki PE, OE ve FE hata bayrakları silinir.

Yazılımla seri iskeleyi sıfırlamak (Reset) mümkündür. Bu amaçla altıncı bite (UR) 1 yazmak yeterlidir. UR biti 0 yapılıncaya kadar ilgili tüm lojik yapı sıfırlama durumunda kalır. Sistem sıfırlama sırasında bu bite otomatik olarak 1 yüklenir.

Yedinci bit, seri iskele durdurma izin (Serial Port Halt Enable: SPH) biti, seri iskelenin ve üçüncü zamanlayıcının durdurulmuş modda (IDLE state) aktif olup olmayacağını belirlemek için kullanılır.

2.8.1.3. Seri İskele Durum Saklayıcısı (SSTAT)

Seri iskele durum saklayıcısı (Serial Port Status Register: SSTAT), iskelenin durumu hakkında bilgiler içeren sadece okunabilir bir saklayıcıdır (Şekil 2.17.).



Şekil 2.17. Seri İskele Durum Saklayıcısı (SSTAT)

Her bit tek tek aşağıda açıklanmıştır:

Verici hazır (TXRDY): Bu bit, vericinin bir sonraki karakterin gönderilmesine hazır olduğunu gösterir. Karakter yüklendiğinde otomatik olarak sıfırlanır. Eğer dörtüncü kesmeye izin verilmişse, kesme oluştuğunda bu bit 1 ile yüklenir.

Alıcı hazır (RXRDY): Alıcının yeni veri ile yüklendiğini gösterir ve her okumada sıfırlanır. Kesmelere izin verilmişse, dördüncü kesme oluştuğunda bu bit otomatik olarak 1 ile yüklenir.

Alıcı boş (TXE): Ötelemeli alıcı saklayıcısı (TXSHF) ve TXBUF (Şekil 2.14.) boş olduklarında bu bit 1 olur.

TXBUF'a yazıldığında sıfırlanır. Sistem sıfırlaması sırasında bu bite 1 yüklenir.

Eşlik hatası (Parity Error: PE): Eşlik hatası oluşunca bu bit 1 ile yüklenir. Bu bit SCTLO'daki ER biti ile sıfırlanabilir.

Yığılma hatası (Overrun Error: OE): Alıcı okunmadan yeni bir karakterle yüklenmişse bu bit 1 ile yüklenir. Önceki karakter kaybolmuş olur. Bu bitte SCTLO saklayıcısındaki ER biti ile sıfırlanır.

Çerçeveleme hatası (Framing Error:FE): Stop biti olmayan karakter alındığında bu bit 1 ile yüklenir. Böylece başla biti ile başlayan senkronizasyon bozulmuş ve yanlış çerçeveleme oluşmuştur. SCTLO'daki ER biti ile sıfırlanabilir.

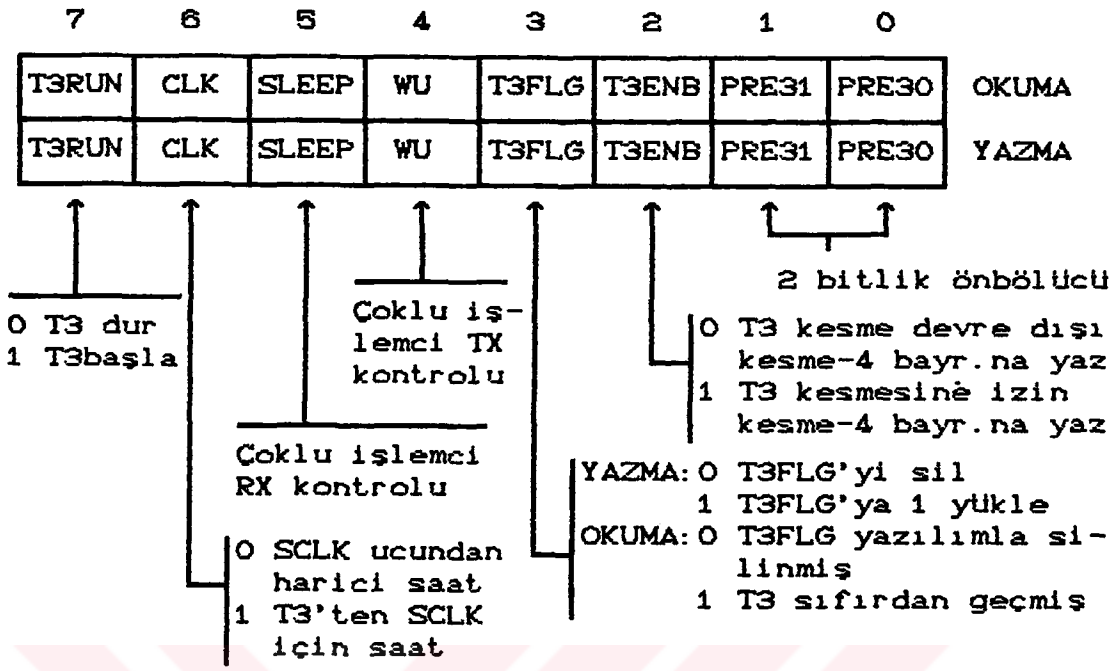
Break Detect(BRKDT): Alıcı hat (RXD Line), on bit veya daha fazla bir süre sürekli düşük seviyede kalırsa bu bit 1 yüklenir. Break sona erdiğinde hemen sıfırlanır. Seri G/Ç modunda bu bit sıfırlanır.

Intel adres (IADD): Bu bit Intel protokolu kullanıldığında alınan verinin son biti gösterir.

2.8.1.4. Seri İskele Kontrol Saklayıcısı-1 (SCTL1)

Seri iskele kontrol saklayıcısı-1, zamanlayıcı-3 dur/başla işlevi, seri saat kaynağı, çoklu işlemci iletişimi, zamanlayıcı-3 kesmeleri ve zamanlayıcı-3 önbölücü değeri belirleme için kullanılır (Şekil 2.18.).

Zamanlayıcı-3 önbölücü saklayıcısı (PRE31, PRE30): Bu iki bit zamanlayıcı-3 için önbölücü görevini görür.



Şekil 2.18. Seri İskele Kontrol-1 Saklayıcısı

Zamanlayıcı-3 kesme izin (T3ENB): Bu bit 1 yapıldığında, T3FLG 1 ile yüklenirse, INT4FLG'da 1 ile yüklenir. T3ENB biti SCTLO'daki UR biti ile değilde sistem sıfırlaması ile silinebilir. Böylece üçüncü zamanlayıcı seri iskeleden bağımsız olarak kullanılabilir.[4]

Zamanlayıcı-3 kesme bayrağı (T3LG): Bu bit, üçüncü sayaç ve önbölücüsü birlikte sıfırdan geçtiklerinde 1 ile yüklenir. T3FLG seri iskeleden gelen kesmeleri gösterir. İlgili servis programında bu bit silinmelidir. Ayrıca bu bitin SCTLO'daki UR biti ile değilde sistem sıfırlaması ile silinmesi üçüncü zamanlayıcının seri iskeleden bağımsız olarak çalışmasını sağlamıştır.

Uyandırma (Wake-Up: WU): Bu bit çoklu işlemcili ile-iletişim modunda veri verme kontrolu için kullanılır.

Uyuma (Sleep: SLEEP): Çoklu işlemcili iletişim modunda veri alma kontrolu için kullanılır.

Seri saat kaynağı (Serial Clock Source: CLK): Seri saat kaynağını belirler. Sistem sıfırlaması sırasında silinir ve UR biti silinene kadar 1 ile yüklenemez.

Zamanlayıcı-3 başla (Timer 3 satart: START): Üçüncü zamanlayıcıyı durdurmak ve başlatmak için kullanılır.

2.8.1.5. Zamanlayıcı-3 Veri Saklayıcısı (T3DATA)

Üçüncü zamanlayıcıya ait veri saklayıcısı, T3DATA, sayaç değerini tutan 8-bitlik okunabilir ve yazılabilir bir saklayıcıdır.

2.8.1.6. Alıcı ve Verici Tamponlar (RXBUF, TXBUF)

Alıcı tampon, RXBUF, sadece okunabilir bir saklayıcıdır ve seri iskeleden alınan veriyi içerir. Bu saklayıcıya veri yazmak herhangi bir etki yaratmayacaktır (Şekil 2.19.).

7	6	5	4	3	2	1	0
			ALINAN VERİ				
0	0	0	← 5	VERİ	BİTİ		→
0	0	←	6	VERİ	BİTİ		→
0	←		7	VERİ	BİTİ		→
←			8	VERİ	BİTİ		→

Şekil 2.19. Alıcı tampon (RXBUF)

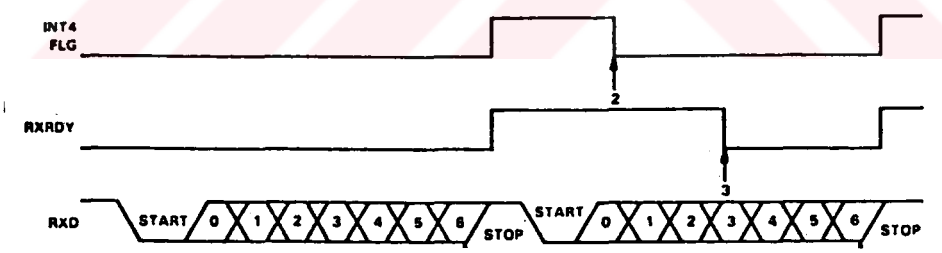
Verici tampon, TXBUF, gönderilecek verinin yazıldığı sadece yazılabilir bir saklayıcıdır (Şekil 2.20.).

			GÖNDERİLECEK	VERİ			
X	X	X	←	5	VERİ	BİTİ	→
X	X	←	6	VERİ	BİTİ		→
X	←		7	VERİ	BİTİ		→
←			8	VERİ	BİTİ		→

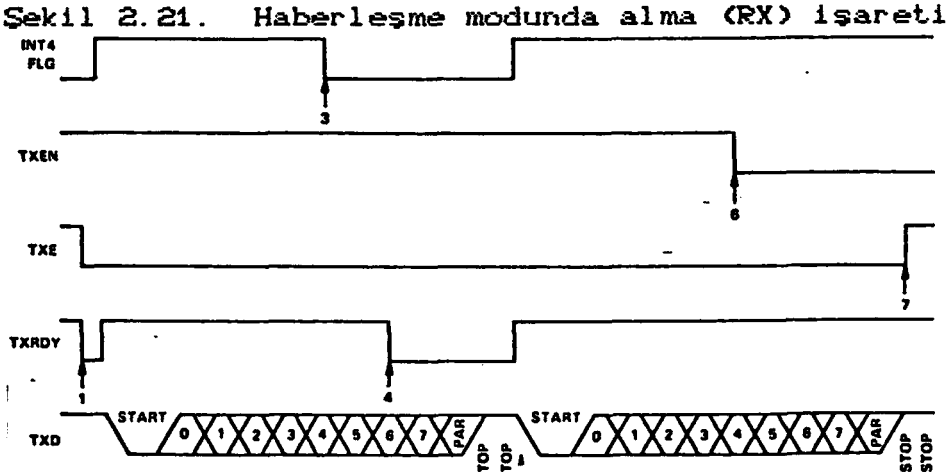
Şekil 2.20. Verici Tampon (TXBUF)

2.8.1.7. Haberleşme Modunda RX ve TX İşareti

Aşağıda haberleşme sırasında kontrol ve veri bitlerinin zamanla nasıl değiştikleri gösterilmiştir. Veri alma (RX) işaretinin incelenmesinde verinin; (başla + yedi veri biti + dur biti) şeklinde, veri göndermede ise; (başla +sekiz veri biti +eşlik biti + iki dur biti) şeklinde olduğu varsayılmıştır. Ayrıca seri saat darbelerinin sürekli var olduğu düşünülerek gösterilmemiştir.

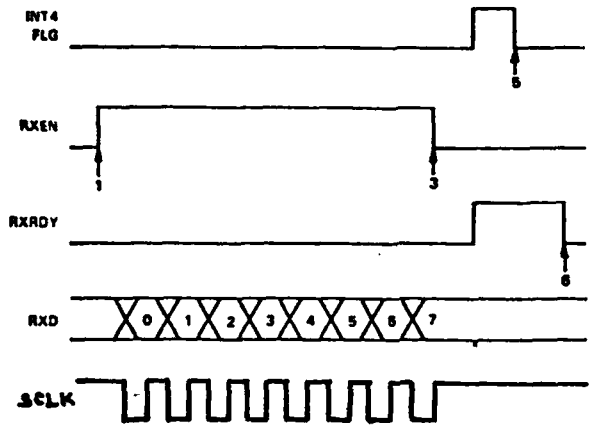


Şekil 2.21. Haberleşme modunda alma (RX) işareti

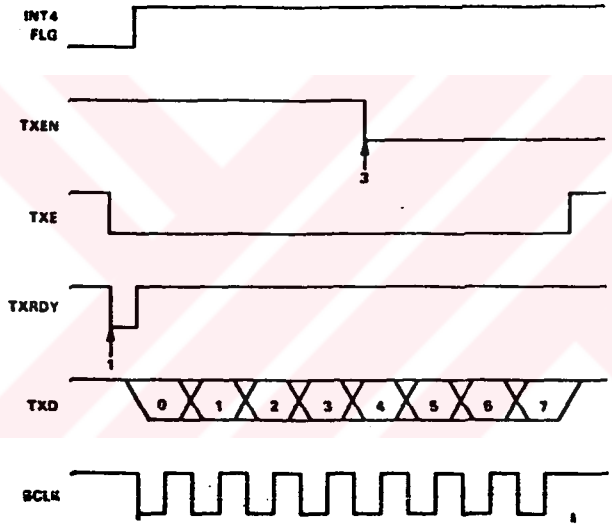


Şekil 2.22. Haberleşme modunda verme (TX) işareti

2.8.1.8. Giriş/Çıkış Modunda RX ve TX İşareti



Şekil 2.23. Giriş/Çıkış modunda alma (RX) işareti



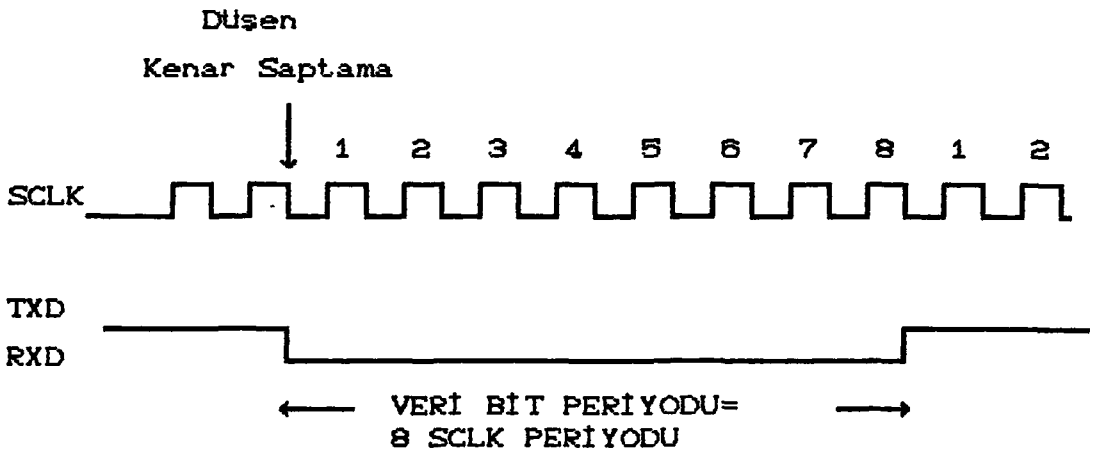
Şekil 2.24. Giriş/Çıkış modunda verme (TX) işareti

2.8.2. Seri Haberleşme Modları

Veri alma/verme üç modda yapılır: İkisi haberleşme modu, asenkron ve isosenkron diğeri ise seri G/Ç.

2.8.2.1. Asenkron Haberleşme Modu

Asenkron haberleşme modunda, çerçeve başla biti, beş-ten sekize kadar veri biti, tek/çift/eşlik yok biti ve bir ya da iki dur biti şeklindedir. Her bitin uzunluğu sekiz seri saat darbesi uzunluğuna eşittir (Şekil 2.25.).

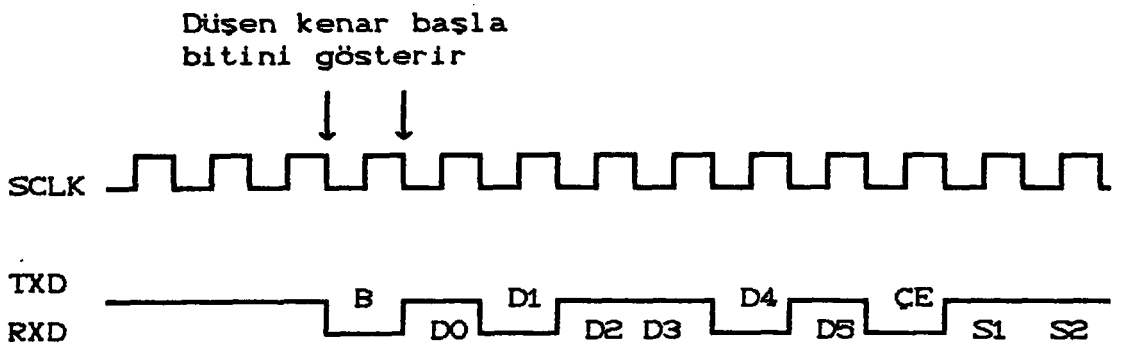


Şekil 2.25. Asenkron iletişim formu

Anlamlı başla biti alındığında, seri yoldan veri alma başlar. Anlamlı başla biti, birbiri ardına gelen darbelere yapılan üç örneklemeden ikisinin sıfır olması ile tanınır. Veri alma işlevi (RX) kendi kendini senkronize ettiği için, alıcı ve verici tarafta aynı saat darbelerine (SCLK) gerek yoktur.[5]

2.8.2.2. İsosenkron Haberleşme Modu

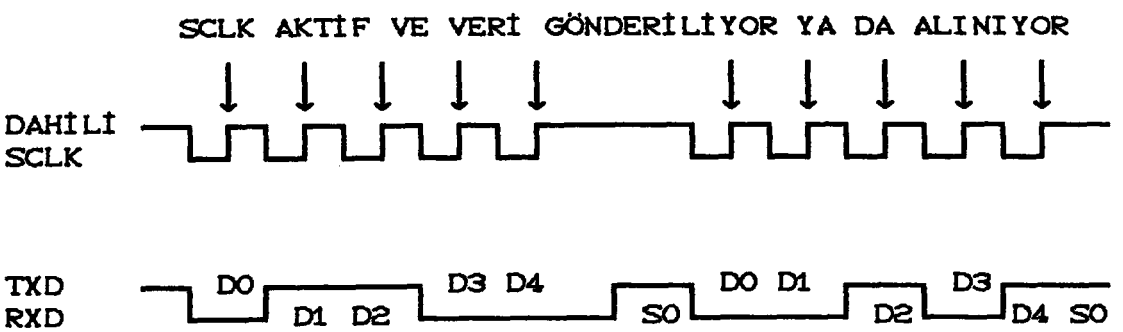
İsosenkron haberleşme modu, asenkron ve seri G/Ç modlarının özelliklerini içeren karma bir protokole sahiptir. İsosenkron modda çerçeve şekli asenkron modunda olduğu gibidir. Bununla birlikte, her saat darbesi ile bir veri biti alınır. Oysa asenkron modda sekiz saat darbesi süresince sadece bir veri biti okunmaktaydı. Bu asenkron moda göre daha hızlı veri transferini mümkün kılmaktadır. Seri G/Ç'da olduğu gibi bu modda veri biti periyodu seri saat darbesi periyoduna eşittir (Şekil 2.26.). Veri alma olayı kendi kendini senkronize edemediği için her iki tarafta aynı saat kaynağı kullanılmalıdır. Asenkron ve isosenkron modda bir çerçeve tam olarak alındığında, RXBUF saklayıcısı RXSHF ve RXRDY saklayıcılarından yüklenir ve kesme-4 bayrağı birlenir. MİB'in RXBUF'u her okuyuşunda RXRDY sıfırlanır.



Şekil 2.26 İsosenkron iletişim şekli

2.8.2.3. Seri Giriş/Çıkış Modu

Seri G/Ç modunda çerçeve formu beş ile sekiz arasında veri biti ve bir dur bitinden oluşur. Dahili veya harici senkronize saat girişi ya zamanlayıcı-3'ten, ya da devre dışından sağlanmalıdır. Bit periyodu SCLK periyoduna eşittir. TXRDY 1 iken, TXBUF'a veri yazılarak veri gönderme işlemi başlatılmış olur (Şekil 2.27.). Veri alma işlemi ise RXEN'e 1 yazılarak başlatılır. Veri alma izin biti (RXEN), veri alındığında otomatik olarak silinir. Veri almada olduğu gibi, TXEN biti veri gönderildikten sonra otomatik olarak silinmez. Alıcı ve vericiyi aynı anda çalıştırmak için, önce gönderilecek veri TXBUF'a yazılmalı sonra, RXEN ve TXEN'e bir komutla 1 yüklenmelidir.



Şekil 2.27. Seri Giriş/Çıkış haberleşme şekli

2.8.3. Çoklu-İşlemcili Haberleşme

Seri iskele asenkron ya da isosenkron haberleşme modunda iken, çoklu-işlemci haberleşme formu kullanılabilir. Böylece, aynı seri hat ile birçok mikroişlemci arasında aktif veri transferi yapılabilir. Veriler çerçeve blokları şeklinde kaynaktan varış yerlerine transfer edilirler. Seri iskele veri bloğunun başlangıcını tanıma özelliğine sahiptir. Alıcıdan (RX'ten) alınan kesme ve durum bilgileri blok başlangıcı tanımlanıncaya kadar saklanır.[4]

Her iki çoklu işlemci modunda işlem sırası şöyledir:

1. Seri iskele uyandırılır (Wake-up) ve bloğun ilk çerçevevesi okunur (Varış adresini içerir).

2. Gelen veriyi kontrol edecek ve bellekte uygun yere yerleştirecek program girilir.

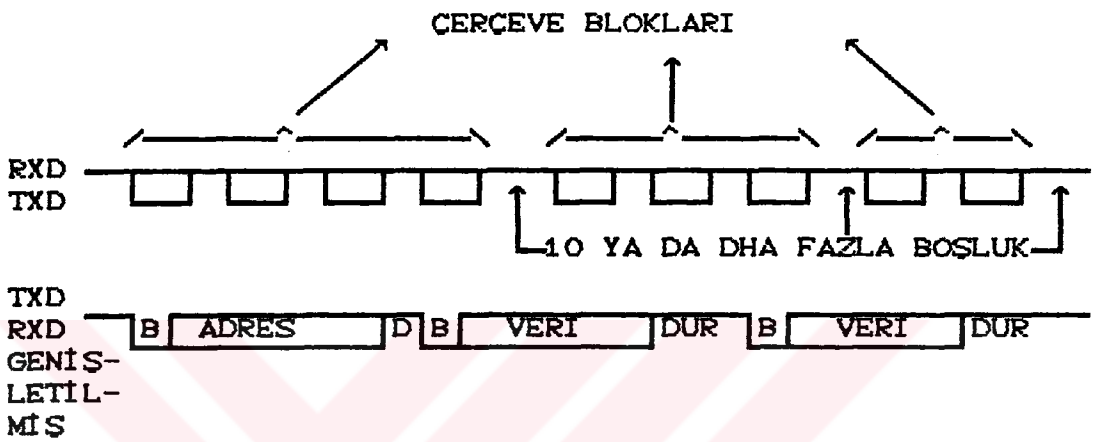
3. Hangi mikroişlemci seçilmişse blokların diğerlerini okumaya devam eder. Seçilmeyen mikroişlemciler yazılımla tekrar uyuma (Sleep) moduna getirilirler.

Seri hatta tüm işlemcilerin SLEEP bitleri 1 dir. Böylece veri yolundaki adres akışı 1 olduğunda sadece kesme ile uyandırılabilir. Aktif blok adresini alan tüm işlemciler bunu kendi adresleriyle karşılaştırdıklarında, seçilen işlemcilerin SLEEP bitleri 0 ile yüklenir ve kalan bloklar okunur.

Daha fazla esneklik sağlayabilmek için seri iskele temel iki işlemci; Motorola ve Intel protokolü için desteklenmiştir. Motorola protokolü MC6801 işlemcisi ile, Intel protokolü ise I8051 protokolü ile uyumludur. Çoklu işlemci modu SMODE saklayıcısındaki MULTI biti ile seçilir.

2.8.3.1. Motorola (MC6801) Protokolü

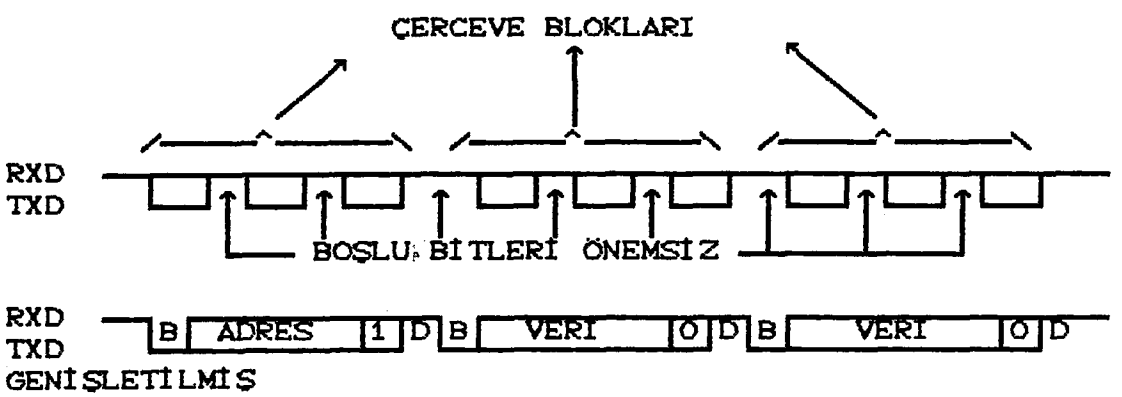
Bu protokolde, bloklar ve çerçeveler uzunca zaman boşluklarıyla birbirinden ayrılmışlardır. Bir çerçeveden sonraki on ya da daha fazla boşluk biti yeni blok başlangıcını gösterir (Şekil 2.28.).



Şekil 2.28. Motorola çoklu-işlemci iletişim formatı

2.8.3.2. Intel (I8051) Protokolü

Intel protokolünda çerçeve, eşlik bitinden hemen önce adres biti içerir. İlk çerçevedeki adres bitinin bir olmasıyla bloklar ayırdedilir. Diğer çerçevelerin adres bitleri sıfırdır. Boşluk süresi önemli değildir (Şekil 2.29.).



Şekil 2.29 Intel çoklu-işlemci iletişim formatı

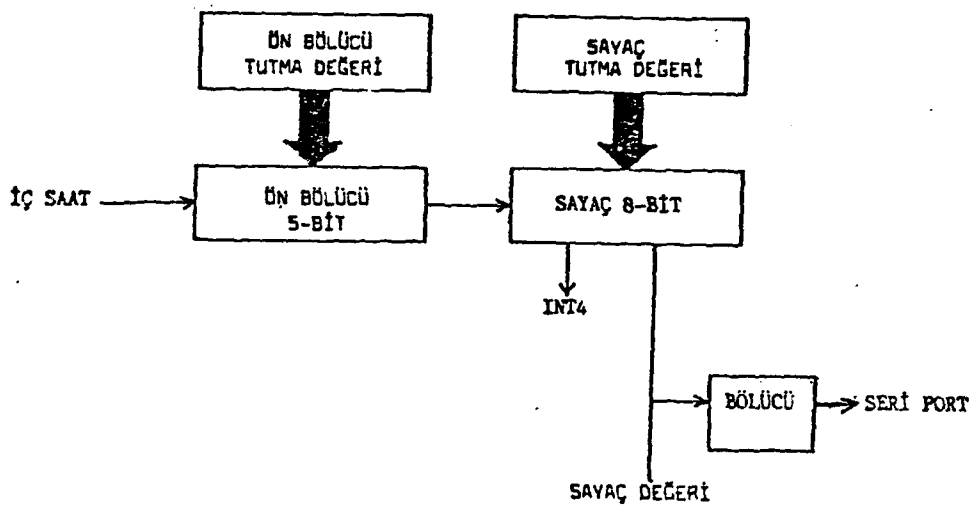
2.8.4. Seri İskele Koşullandırma

Seri iskele kullanılmadan önce koşullandırılmalıdır; ancak böylece G/Ç genişletme alanı saklayıcıları, okunup yazılıyormuş gibi kullanılabilir. Seri iskeleyi etkileyecek tüm saklayıcılara programla uygun veriler yüklenmelidir. Bir kere seri iskele koşullandırıldıktan sonra, seçilen çalışma modlarında sürekli kullanılabilir. Veri göndermek bir tampona veri yazmakla, veri alma ise bir tamponun okunmasıyla olur. Yalnız gönderme ve almaya izin veren kontrol bitlerinin uygun şekilde yazılmış olması gereklidir. Seri iskele üç şekilde sıfırlanabilir: donanım sıfırlaması (Reset tuşu ile), yazılım sıfırlaması (SCTLO'daki UR biti ile) ve SMODE saklayıcısını yazarak.

2.8.5. Zamanlayıcı-3 (Timer-3)

Üçüncü zamanlayıcı genel amaçlı zamanlayıcı ya da seri iskele için saat kaynağı olarak kullanılabilmektedir. Zamanlayıcı-3 T3DATA ve SCTL1 saklayıcıları ile aktif hale getirilir. Üçüncü zamanlayıcı iki bitlik ön bölücü ve sekiz bitlik sayaçtan oluşur (Şekil 2.30.). Bu saklayıcılar her sıfır oluşlarında belirlenen değerlerle yüklenirler.

T3RUN biti sıfır iken, ön bölücü ve sayaç değerleri , tutuculardan (Reload register) önbölücü ve sayaç saklayıcılarına yüklenir. BU zamanlayıcıda, diğer ikisinden farklı olarak, bir bit (T3RUN) sıfırlandığında sayacın o anki değeri tutulmaz. Sayaç, T3RUN biti 0'dan 1'e geçtiğinde saymaya başlar. Sistem sıfırlamasında bu bit sıfır ile yüklenir. Eğer T3ENB biti (SCTL1'de ikinci bit), bir ise, sayacın sıfırdan geçişlerinde, üçüncü zamanlayıcı bayrağı ve kesme-4 bayrağı bir ile yüklenirler. Zamanlayıcı-3'ün bayrakları seri iskele sıfırlamasından (UART Reset) etkilenmezler. Böylece, üçüncü zamanlayıcı seri iskeleden bağımsız olarak kullanılabilir.



Şekil 2.30. Zamanlayıcı-3 İşlevsel Blok Diyagramı

Zamanlayıcı-3, seri iskele saatı olarak da kullanılabilir. Veri akış hızı, önbölücü ve sayaç değeri ile programlanabilir. Aşağıdaki eşitlikler asenkron ve isosenkron haberleşme için çıkarılmıştır:

$$\text{Asenkron Veri Ak. H.} = \frac{f_{osc}}{64*(PRR+1)*(TRR+1)} = \frac{SCLK}{8}$$

$$\text{Isosenkron Veri Ak. H.} = \frac{f_{osc}}{8*(PRR+1)*(TRR+1)} = SCLK$$

f_{osc} = Kristal frekansı ya da harici saat frekansı
PRR = Zamanlayıcı-3 sayaç değeri (Reload reg., P23)
TRR = Zamanlayıcı-3 önbölücü değeri (Prescalar, P24)
SCLK= Seri saat

Seri G/Ç modu için veri akış hızı isosenkron'da olduğu gibi hesaplanır.

2.8.6. Seri İskele Kesmeleri

Kesme-4 seri iskeleye ayrılmıştır. Bu kesmenin üç farklı kaynağı şunlardır:

1. Verici (Transmitter: TX)
2. Alıcı (Receiver: RX)
3. Zamanlayıcı-3 (Timer3: T3)

MİB kesmeyi tanıdığında, RXRDY, TXRDY ve T3FLG bayraklarına bakılarak kesme kaynağı öğrenilebilir. Uygulama yerine göre, kesme servis programında bu saptanabilir.

BÖLÜM 3

TMS77C82 SİMGESEL DİLİ VE KOMUT KÜMESİ

Bir mikrobilgisayarın tasarımından bildiği sadece komut kümesidir. Kullanıcı bu komut kümesini kullanarak, bilgisayara isteklerini anlatabilir. Bilgisayarın komutlarını kullanarak program yazmaya Simgesel Dilde Program yazma denir. Simgesel dil sembolik komutlardan ve direktiflerden oluşur. Simgesel direktifler, yazılan programın makine diline çevrilme (ikili sayı düzeni) işlemini kontrol eder, program içindeki verileri yerleştirir ve program içinde kullanılan sembollere değer atanmasını sağlarlar. Simgesel dilin, makina diline göre en büyük avantajı, sembolik komutların, makina dilindeki 1 ve 0' lara göre daha kolay hatırlanabilmesidir.

Her mikroişlemcinin komut kümesi farklı olduğundan farklı bilgisayarın simgesel dildeki programları da farklıdır. Daha önce de belirtildiği gibi makine dilinde komutlar ikili sayılar biçiminde görünür, ancak simgesel dilde bunlara karşılık düşürülmüş olan kısaltmalar (Mnemonik) kullanılır.

Bu bölümde, TMS77C82'ye ait simgesel dil ve bu simgesel dilin kullandığı adresleme yöntemleri ele alınmış, her komutun bayrakları nasıl etkilediği belirtilmiştir.

3.1. Simgesel Dilde Kaynak Program Formu

Kaynak Program:

Simgesel dildeki bir kaynak program, simgesel direktifler, makina komutları, sembolik komutlar veya açıklamaları içeren kaynak ifadelerden oluşur. Bir kaynak ifade sırasıyla dört alan içerebilir; etiket, komut, işlenen ve açıklama. İlk karakteri "*" olan kaynak ifadeleri dikkate alınmaz. Şekil 3.1'de örnek bir program görülmektedir. Kaynak ifadelerinin yazılım şekli aşağıdaki gibidir:

[<etiket>] <komut> [<işlenen>] [<açıklama>]

- Etiket ve açıklama alanları isteğe bağlıdır.
- Alanlar birbirinden bir veya daha fazla boşlukla ayrılır.
- İfadeler boşluk ya da etiketle başlamalıdır.

```
*-----------------------------------------------*
*        BU BİR ÖRNEK KAYNAK PROGRAMIDIR        *
*-----------------------------------------------*

         IDT    'ÖRNEK'
         CLR    B
ETİKET1  MOVP   P4,A
         BTJZ   %01,A,ETİKET1
         END
```

Şekil 3.1. Kaynak ifadeleri şekli

Şimdi kaynak ifadelerinde yer alan alanları biraz daha açıklayalım:

Etiket Alanı:

Etiket alanı, birçok komut ve semboller için zorunlu değildir. Etiket kullanılmadığında kaynak ifadesinin ilk karakteri boşluk olmalıdır. Kaynak ifadesinin ilk karakterinden başlayıp ilk boşluğa kadar süren etiket alanı, en fazla altı alfanümerik karakter içerir ve ilk karakter alfabetik olmalıdır.

Sadece etiket alanı içeren kaynak ifadeleri de vardır. Aşağıdaki ifade buna bir örnektir ve aktif bölgesel sayacın değeri bir etikete eşitlenmektedir.

```
<etiket>  EQU  $
```

Komut Alanı:

Etiket alanı bitimindeki boşluktan sonra komut alanı başlar. Bu alanda, bir ya da daha fazla boşlukla sonlandırılır. Eğer etiket kullanılmamışsa, komut ikinci karakterden başlayabilir. Komut alanı aşağıda belirtilen fadeleri içerebilir:

- Komutlar (Mnemonik)
- Kullanıcının tanımladığı komutlar
- Simgesel yönlendiriciler

İşlenen Alanı:

İşlenen alanı, komut alanından sonundaki boşluktan sonra başlar ve açıklamaların olduğu alana kadar devam eder. İşlenen alanı birbirinden virgülle ayrılmış sabitlerden veya ifadelerden oluşabilir. Bir veya daha fazla boşlukla bu alan sonlandırılır.

Açıklamalar Alanı:

İşlenen alanının sonundaki en az bir boşluktan sonra (ya da işlenen yoksa komut alnından sonra) açıklamalar alanı başlar ve satır sonuna kadar devam edebilir. Burada yazılanlar program üzerinde hiçbir etki yaratmazlar.

3.1.1. Kaynak Programlarda Kullanılan Sabitler

Simgesel dildeki programın tanıdığı sabitler aşağıda verilmiştir:

- Onluk tamsayılar
- İkilik tamsayılar
- Onaltılık tamsayılar
- Alfanümerik sabitler
- Simgesel-zaman sabitleri

Onluk Sayılar:

Onluk tamsayı sabitleri, onlu haneler şeklinde -32786 ile +65535 arasında olabilir. Negatif onluk sayılar, 32768 ile 65535 arasındaki pozitif tamsayıların ikiye tümleyeni alınarak elde edilir. Onluk sayılara örnekler aşağıda verilmiştir.

1000	sabiti 1000 veya >3E8'e eşittir.
-32768	sabiti -32768 veya >8000'e eşittir.
25	sabiti 25 veya >19'a eşittir.
65535	sabiti 65535 veya >FFFF'e eşittir.

İkilik Sayılar:

İkilik sayılar, ikilik sayı düzeninde onaltı haneye kadar yazılabilir. Yazılan sayının ikilik düzende olduğunu göstermek için, sol baş tarafına soru işareti "?" konur. Aşağıda örnekler verilmiştir:

?00010011 sabiti 19 veya >13'e eşittir.
?0111111111111111 sabiti 32768 veya >7FFF'e eşittir.
?11110 sabiti 30 veya >001E'ye eşittir.

Onaltılık Sayılar:

Onaltılık sayılar en fazla dört haneye kadar, önüne büyüktür ">" işareti konularak yazılır. Onaltılık sayılar 0'dan 9'a kadar onluk ve A'dan F'e kadar alfanümerik karakterlerden oluşur.

>78 sabiti 120'ye eşittir.
>F sabiti 15'e eşittir.
>37AC sabiti 14252'ye eşittir.

Alfanümerik Sabitler:

Alfanümerik sabitler bir ya da iki alfabetik karakterin tek tırnak içinde yazılmasıyla ifade edilirler. Karakter dizisi içinde tek tırnak varsa, iki tek tırnak arka arkaya basılmalıdır. Her karakter, 8-bitlik ASCII değere karşı düşürülür. Sadece iki tek tırnak (içinde karakter olmayan) >0000'a eşittir. Aşağıdaki örnekler kaynak programlarında kullanılabilir:

'AB' >4142'yi gösterir.
'C' >43 veya >0043'ü gösterir.
'N' >4E veya >004E'yi gösterir.
'''D' >2C44'yi gösterir.

Simgesel Zaman Sabitleri:

Simgesel zaman sabitleri, eşitlik "EQU" direktifi ile, kaynak programlarında değer atanmasında kullanılan sembollerdir. Sembol değerleri, kaynak program makina dili programına dönüştürülürken belirlenir.[2]

3.1.2. Kaynak Programlarda Kullanılan Semboller

Semboller, etiket ya da işlenen alanında kullanırlar. Alfanümerik karakterlerden (A-Z, 0-9, $) oluşan bir sembolün ilk karakteri, A-Z ve $ olabilir. Ayrıca ilk karakter boşluk olamaz. Sembollerde karakter miktarı açısından bir sınırlama olmamasına rağmen,ilk altı karakter program için anlamlıdır. Bu nedenle sembolün ilk altı karakteri tek olmalıdır. Kullanıcının tanımladığı semboller sadece kaynak programında geçerlidir.

Etiket alanında tanımlanan semboller, sembolik adreslerdir. Bu semboller program içindeki bölgelerle ilgilidir ve diğer ifadelerde kullanılmamalıdır. Komutlar ve simgesel direktifler, sembol olarak etiket alanında kullanılabilir.

İşlenen alanında kullanılan semboller, daha önce tanımlanmış olmalıdır. Aşağıda bazı örnekler verilmiştir:

```
START
ADD
OPERATION
```

Bu sembollerin herbirine, etiket olarak bulunduğu bölgenin adresi atanmıştır. OPERATION etiketi kesilerek OPERAT olarak kabul edilir.

Belirlenmiş Semboller:

Dolar işareti ($), saklayıcı (Rn) ve iskele (Pn) sembolleri daha önceden belirlenmişlerdir. Dolar işareti o anki yerel sayacın değerini gösterir. ST (Status Register) sembolü tekrar tanımlanamaz.

R0	Saklayıcı 0
P22	G/Ç saklayıcısı 22

Terimler:

Terimler, komut kodlarının ve simgesel direktiflerin işlenen kısımlarında kullanılır. Terim, kaynak programda kullanılan herhangi bir sabit olabilir.

Karakter Dizisi:

Birçok simgesel direktif, karakter dizisini işlenen olarak kullanır. Bir karakter dizisi tek tırnak içine yazılmış karakterlerden oluşur. Karakter dizisi içinde tek tırnak gösterilmek istendiğinde arka arkaya iki tek tırnak koymak gereklidir. Karakter dizisi kullanılacak her direktif için, dizinin en fazla uzunluğu belirtilmelidir. Aşağıda karakter dizilerine bazı örnekler verilmiştir:

'ORNEK PROGRAM'	14 karakterli olan ORNEK PROGRAM dizisini gösterir.
'PLAN ''C'''	8 karakterli PLAN 'C' yi gösterir.

3.1.3. Kaynak Programlarda Kullanılan İfadeler

Kaynak proramlarındaki ifadeler, komutların ve simgesel direktiflerin işlenen alanlarında (operand field) kullanılırlar. Bir ifade, sabit veya sembol, sabit veya semboller dizisi, veya aritmatik işlemci ile ayrılmış sabit veya sembollerden oluşan diziler şeklinde olur. Her sabit veya sembolün önüne eksi (-), artı (+) veya tümleme (#) işareti gelebilir. "#" işareti hangi sabit ya da sembolün önüne gelirse, onun lojik tümleyenini aldırır. Bir ifade boşluk içermeyebilir. Dış referanslara göre tanımlanmış semboller, aşağıda tanımlanan belirli sınırlar içinde aritmatik komutların işlenenleri olabilirler.

İfadelerdeki Aritmatik İşlemciler:

İfadelerde kullanılan aritmatik işlemciler şunlardır:

+ Toplama
- Çıkarma
* Çarpma
/ Bölme
Lojik tümleme

Bir ifade kaynak programında değerlendirilirken, önce önünde eksi işareti olan sabit ya da sembollerin negatifi alınır, sonra aritmatik işlem soldan sağa doğru icra edilir. İşlem sırası; önce toplama ve çıkarma sonra çarpma şeklindedir ve tüm işlemler tamsayılarla yapılır. Bölme işleminde oluşabilecek virgül sonrası haneler atılır.

Örneğin; 4+5*2, 18 Olarak sonuçlandırılır. 7+1/2 ifadesi 4, 1/2+7 ifadesi ise 7 ile sonuçlandırılacağına dikkat ediniz.

İfadelerdeki Lojik İşlemciler:

Eğer bir sayı ya da ifadenin önünde "#" işareti varsa o sayı veya ifadenin lojik tümleyeni alınır. Eğer parantez içine alınmadıysa, tüm diğer aritmatik işlemler lojik tümlemeye göre önceliklidir.

İfadelerdeki Parantezler:

İfadelerdeki icra sırasını değiştirmek için parantez kullanılır. Parantezli ifadeler içiçe, sekiz seviyeli olabilir. Önce en içteki parantez icra edilir sonra sırasıyla dış parantezlere çıkılır, en sonunda ise solda sağa doğru işlem tamamlanır. Aşağıdaki örnekte icra sırası gösterilmiştir:

```
LAB1+((4+3)*7)
```

1. 4 ile 3 toplanır
2. 7, 7 ile çarpılır

3. LAB1'in değeri ile 49 toplanır.

İyi-Tanımlanmış (Well-Defined) İfadeler:

Bazı simgesel direktifler için işlenen alanında iyi-tanımlanmış ifadeler kullanmak gereklidir. Bu ifadeler sadece simgesel zaman sabitleri ve sembollerden oluşurlar ve karakter sabiti içermezler.

İfadelerdeki Değişebilir (Relocatable) Semboller:

İşlenen alanındaki değişebilir sembol ve sabitler içeren ifadeler, eğer bir bölme ya da çarpma işlemcisini hemen takip ediyorsa, geçerli değildir. Birkaç bölümden oluşan ifadelerde ifadenin o anki değeri, değişebilir bölümüne bağlı olarak değişebilir ise, ifadenin değer bölümündeki sembol kapsanmayabilir. Aşağıda bazı örnekler rilmiştir:

BLUE+1	Sonuç BLUE sembolünün değeri artı birdir.
2*16+RED	Sonuç RED sembolünün değeri ve iki çarpı onaltının toplamıdır.
440/2-RED	Sonuç 440'ın ikiye bölünmesinden sonra RED'in çıkartılmasıyla elde edilendir.

3.1.4. Kaynak Programlarda kullanılan Simgesel Direktifler

Bu bölümde TMS77C82 mikrobilgisayarında kullanılan simgesel direktifler tablo halinde verilmiştir.

TABLO 3.1 Simgesel direktiflerin özet tablosu

YEREL SAYACI ETKİLEYEN DİREKTİFLER	
KOMUT	DİREKTİF
AORG	Sabit başlangıç
BES	Sembolle blok sonlandırma
BSS	Sembolle blok başlangıcı
CEND	Ortak segment sonu
CSEG	Ortak segment

DEND	Veri segmenti
DORG	Dummy origin
DSEG	Veri segmenti
EVEN	Çift sınır (Even baundary)
PEND	Program segmenti sonu
PSEG	Program segmenti
RORG	Değişebilir başlangıç (Relocatable origin)
SIMGESEL DILDE PROGRAM ÇIKIŞLARINI ETKILEYECEK DIREK.	
IDT	Program belirleyici (Identifier)
LIST	Kaynak listelemeyi tekrar başlatır
OPTION	Isteğe bağlı çıkış ilaveleri
PAGE	Sayfa geçme
TITL	Sayfa başlığı
UNL	Kaynak listelemeyi durdurma
SABITLERE ILK DEGER ATAMAYA YARAYAN DIREKTIFLER	
BYTE	Sekizli tanımlama
DATA	Iki sekizli tanımlama
EQU	Simgesel-zaman sabiti tanımlama
TEXT	Text tanımlama
PROGRAM BAGLAMAYA (LINK) ETMEYE YARAYAN DIREKTIFLER	
DEF	Dış tanımlama
LOAD	Kuvvet yükleme (Force load)
REF	Dış referans
SREF	Ikincil dış referans
ÇEŞITLI DIREKTIFLER	
COPY	Kaynak kopyalama
END	Program sonu
MLIB	Genel kütüphane (Macro library) tanımlama

3.2. TMS7000 Ailesi Simgesel Dil Komut Kümesi

TMS7000 mikroişlemci ailesi, giriş, çıkış, veri işleme veri karşılaştırma ve program akışını düzenleyen 61 komutluk bir komut kümesine sahiptir. Bu küme kendi içinde sekiz işlevsel bölüme ayrılabilir:

- Aritmatik komutlar
- Dallanma komutları
- Karşılaştırma komutları
- Kontrol komutları
- Yükleme ve aktarma komutları
- Lojik komutlar
- Öteleme komutları
- G/Ç komutları

3.2.1. Tanımlar

Komut kümesinde kullanılan kullanılan bazı semboller Tablo 3.2.'de gösterilmiştir.

Tablo 3.2. TMS7000 mikroişlemci ailesi sembol tablosu

SEMBOL	TANIMI
A,B	A,B saklayıcıları saklayıcı alanında R0,R1
Rn	Saklayıcı alanındaki n. saklayıcı
Pn	G/Ç genişletme alanındaki n. saklayıcı
s	Kaynak işleneni
d	Varış işleneni
Rs	Saklayıcı alanındaki kaynak sak. 0=<s=<255
Ps	G/Ç genişletme alanındaki kay. sak. 0=<s=<255
Rd	Saklayıcı alanındaki varış saklayıcısı
Pd	G/Ç genişletme alanındaki varış saklayıcısı
Rp	Saklayıcı çifti
iop	İvedi işlenen
ST	Durum saklayıcısı
SP	Yığın göstergesi
PC	Program sayacı
pcn	Bir sonraki komutun konumu
$	Program sayacının o anki değeri
b	Bit sayısı 0=<b=<7
offset	Bağıl adres
ta	Varış adresi
@	Adres ya da etiket gösterir
%	İvedi işleneni gösterir
*	Dolaylı saklayıcı alanı adresleme modu
<XADDR>	Dolaylı adresleme işlenenini gösterir
?	İkili sayı
>	Onaltılı sayı
MSB	En Yüksek anlamlı byte
LSB	En Az anlamlı byte
MSb	En Yüksek anlamlı bit
LSb	En Az anlamlı bit
cnd	Koşul
()	İçerik
→	'e atamak
←	'a eşit olmak
[ ]	İsteğe bağlı (optional) ilaveler
< >	Birşeylerin yazılması gerektiğini gösterir

3.2.2. Adresleme Modları

TMS7000 ailesi, Tablo 3.3.'de gösterilen sekiz adresleme modunu desteklemektedir.

Tablo 3.3. TMS7000 mikroişlemci ailesi adresleme modları

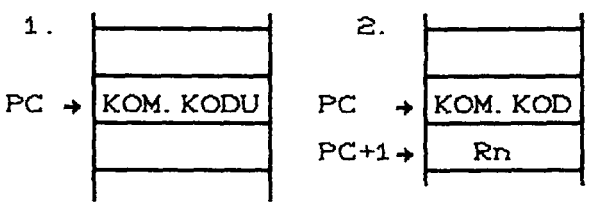
ADRESLEME MODU	ÖRNEK		
Tek saklayıcı	LABEL	DEC INC CLR	B R45 R23
Çift saklayıcı	LABEL	MOV ADD CMP	B,A A,R17 R32,R73
G/Ç genişletme	LABEL	XORP MOVP	A,P17 P42,B
İvedi	LABEL	AND ANDP BTJO	%>C5,R55 %VALUE,P32 %>D6,R80,LABEL
Program sayacına bağıl	LABEL1	JMP DJNZ BTJO BTJOP	LABEL A,LABEL %>16,R12,LABEL B,PY,LABEL
Doğrudan	LABEL	LDA CMPA	@>F3D4 @LABEL
Saklayıcı dolaylı	LABEL	STA	*R43
Sıralı	LABEL2	BR	@LABEL(B)

3.2.2.1. Tek Saklayıcı Adresleme

Tek saklayıcı adreslemeli komutlar, işlenen olarak 8-bitlik bir saklayıcı kullanırlar. Saklayıcı Rn olarak adlandırılabilir (0=<n=<255). A ve B akümülatörleri R0 ve R1 olarak da kullanılabilir.

Durum 1: <KOMUT> A
<KOMUT> B

Durum 2: <KOMUT> Rn



Şekil 3.2. Tek saklayıcı adreslemede komutların ve işlenenlerin (opcode) bellekteki konumları

3.2.2.2. İki Saklayıcılı Adresleme

İki saklayıcılı adresleme komutlarında, kaynak ve varış işleneni olarak sekizer bitlik iki saklayıcı kullanılır. Simgesel dil yazım kurallarına göre, önce kaynak saklayıcısı ve daha sonra varış saklayıcısı gelmelidir. Şekil 3.3.'de bu adresleme modunda kullanılan komutların ve işlenenlerin bellekte ne kadar yer (sekizli olarak) kapladıkları gösterilmiştir.

KAYNAK \ VARIŞ	A	B	Rd
A	2	1	2
B	1	2	2
iop	2	2	3
Rs	2	2	3

Aktarma komutları için

KAYNAK \ VARIŞ	A	B	Rd
A	2	2	3
B	1	2	3
iop	2	2	3
Rs	2	2	3

Diğer komutlar için

Şekil 3.3. İki saklayıcı adreslemede komutların ve işlenenlerin sekizli (byte) sayısı

3.2.2.3. Giriş/Çıkış Genişletme Alanı Adresleme

Bu adresleme modunda, komutlar giriş/çıkış için kullanılırlar. Bu alana ait her saklayıcı Pn olarak adlandırılır. Bu adresleme modunda dört komut kullanılmaktadır:

- MOVP,
- ANDP,
- ORP ve
- XORP.

Bu komutlarda A veya B saklayıcıları kaynak, Pn saklayıcıları ise varış olacak şekilde kullanılabilir. Aktarma

komutlarından MOVP 'de Pn kaynak , A ve B saklayıcıları ise varış olarak kullanılabilir. BTJOP ve BTJZP komutları da aynı adresleme yöntemini kullanmakla beraber şekli (formatı) biraz farklıdır. Şekil 3.4.'de komutların vakaynak ve varış dağişkenlerine göre ne kadar yer (sekizli) kapladıkları gösterilmiştir.

VARIŞ

KAYNAK	A	B	Pd
A			2
B			2
iop			3
Pd	2	2	

ANDP, ORP ve MOVP komutları için

VARIŞ

KAYNAK	Pd
A	3
B	3
iop	4

BTJOP ve BTJZP komutları için

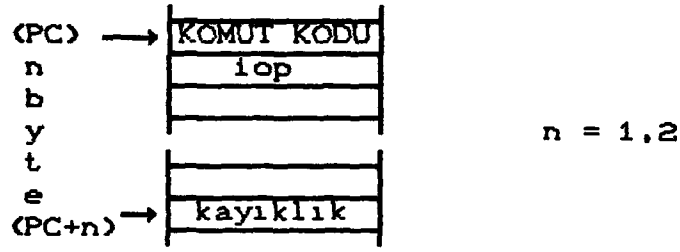
Şekil 3.4. İki saklayıcılı adresleme modunda komutların kapladıkları alan (Sekizli olarak)

3.2.2.4. İvedi Adresleme

Bu moda ait komutlarda, 8 bitlik ivedi işlenenler kullanılır. İvedi veri sabit bir değer veya önüne "%" işareti konmuş bir etiket olabilir. MOVD komutu her iki şekildeki, 16-bitlik işlenenleri kullanabilir.

3.2.2.5. Program Sayacına Bağıl Adresleme

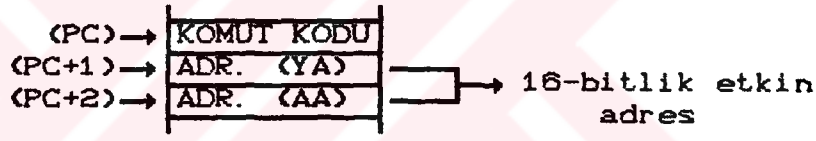
Tüm sıçrama komutları, program sayacına bağıl adresleme modunu kullanır. Simgesel dil programındaki kaynak ifadeleri, daima sıçrama komutları için ulaşılacak adres (ta) bilgisini içerirler. Mikrobilgisayar ulaşılacak adresi bulmak için kayıklığı (offset) şu şekilde hesaplar: *kayıklık* = *ta-pcn*, pcn bir sonraki komutun konumunu gösterir. Bağıl adres ya da kayıklık, $-128 \leq kayıklık \leq 127$ olarak sınırlandırılmıştır. Şekil 3.5.'de bir sıçrama komutunun bellekteki durumu görülmektedir.



Şekil 3.5. Sıçrama komutunun bellekteki durumu

3.2.2.6. Direk Adresleme

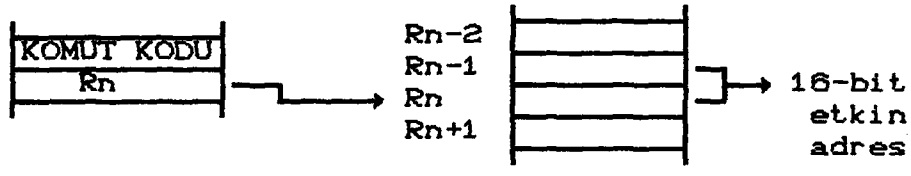
Direk adresleme modunda komutlar, işlenene erişmek için 16-bitlik adres kullanırlar. Direk ya da sembol olarak yazılmış adresin önüne "@" işareti konulmalıdır.



Şekil 3.6. Direk adresleme

3.2.2.7. Saklayıcı Alanı Dolaylı Adresleme Modu

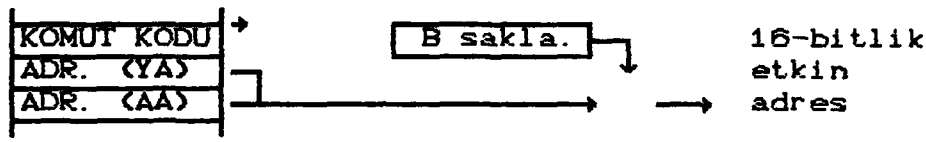
Bu adresleme modunu kullanan komutlarda, etkin adres olarak bir çift saklayıcının içeriği kullanılır. Dolaylı adres, saklayıcı adı (Rn) ve önüne asteriks, "*" işareti konularak belirtilir, yani *Rn gibi. Adresin düşük anlamlı kısmı Rn'de, yüksek anlamlı kısmı ise Rn-1'de saklanır (Şekil 3.7.).



Şekil 3.7. Saklayıcı alanı dolaylı adresleme

3.2.2.8. Sıralı Adresleme

Bu adreslemede, B saklayıcısının içeriği ile 16-bitlik adres toplanır ve bellek adresi hesaplanır. Simgesel dildeki ifade, önünde "@" işareti olan 16-bitlik sabit değer veya etiket ve ardından parantez içinde B; @ETİKET(B) şeklindedir. Toplamada elde oluşmuşsa, otomatik olarak yüksek anlamlı sekizliye aktarılır. Şekil 3.8.'de aktif adresin hesaplanması gösterilmiştir.



Şekil 3.8. Sıralı adreslemede adres hesabı

3.3. Komut Kümesi

TMS7000 mikroişlemci ailesine ait komut kümesi, komut kodları, komut uzunlukları, işlenen tipleri ve durum bitlerinin ne şekilde etkilendiği aşağıda verilmiştir. Kullanılan bazı kısaltmalar şunlardır:

A = A akümülatörü.
B = B akümülatörü.
k = Kaynak (source).
v = Varış (destination).
e = Elde (carry).
Rn = Saklayıcı alanındaki n. saklayıcı.
Pn = G/Ç genişletme alanındaki n. saklayıcı.
iop= İvedi işlenen (immediate operand).
ofs= Kayıklık (offset).
X = Durum biti etkilenmez.
S = Durum biti işlem sonucuna göre 1 veya 0 olur.
B = Kodun sekizli olarak bellekte kapladığı alan.
CNZI= Durum saklayıcısındaki bayraklar; elde, negatif, sıfır ve genel kesme izin bayrağı

Tablo 3.4. TMS77c82 Mikrobilgisayarı Komut Kümesi

	KOMUT	KODU	B	CNZI	AÇIKLAMA
ADC	B,A	69	1	SSSX	(k)+(v)+(e)→ (v)
	Rk,A	19	2		
	Rk,B	39	2		
	Rk,Rv	49	3		
	%iop,A	29	2		
	%iop,B	59	2		
	%iop,Rv	79	3		
ADD	B,A	68	1	SSSX	(k)+(v)→ (v)
	Rk,A	18	2		
	Rk,B	38	2		
	Rk,Rv	48	3		
	%iop,A	28	2		
	%iop,B	58	2		
	%iop,Rv	78	3		
AND	B,A	63	1	OSSX	(k).VE.(v)→ (v)
	Rk,A	13	2		
	Rk,B	33	2		
	Rk,Rv	43	3		
	%iop,A	23	2		
	%iop,B	53	2		
	%iop,Rv	73	3		
ANDP	A,Pv	83	2	OSSX	(k).VE.(Pv)→ (Pv)
	B,Pv	93	2		
	%iop,Pv	A3	3		
BTJO	B,A,ofs	66	2	OSSX	(k).VE.(V)≠ 0 ise
	Rn,A,ofs	16	3		(PC)+ofs→ (PC)
	Rn,B,ofs	36	3		
	Rn,Rv,ofs	46	4		değilse (PC)+1→(PC)
	%iop,A,ofs	26	3		
	%iop,B,ofs	56	3		
	%iop,Rn,ofs	76	4		
BTJOP	A,Pn,ofs	86	3	OSSX	(k).VE.(Pn)≠ 0 ise
	B,Pn,ofs	96	3		(PC)+ofs→ (PC)
	%>iop,Pn,ofs	A6	4		değilse (PC)+1→ (PC)
BTJZ	B,A,ofs	67	2	OSSX	(k).VE DEĞİL.(Pn)≠0
	Rn,A,ofs	17	3		ise (PC)+ofs→ (PC)
	Rn,B,ofs	37	3		
	Rn,Rf,ofs	47	4		değilse (PC)+1→ (PC)
	%>iop,A,ofs	27	3		
	%>iop,B,ofs	57	3		
	%<iop,Rn,ofs	77	4		
BTJZP	A,Pn,ofs	87	3	OSSX	(k).VE DEĞİL.(Pn)≠0
	B,Pn,ofs	97	3		ise (PC)+ofs→ (PC)
	%<iop,Pn,ofs	A7	4		değilse (PC)+1→ (PC)
BR	@ETİKET	8C	3	XXXX	(v) → (PC)
	@ETİKET(B)	AC	3		
	*Rn	9C	2		

KOMUT		KODU	B	CNZI	AÇIKLAMA
CALL	@ETİKET	8E	3	XXXX	(SP)+1→(SP)
	@ETİKET(B)	AE	3		(PCy)→ ((SP))
	*Rn	9E	2		(SP)+1→(SP)
					(PCa)→ ((SP))
					İşlenen adresi→(PC)
CLR	A	B5	1	001X	0→ (v)
	B	C5	1		
	Rv	D5	2		
CLRC		B0	1	0SSX	0→ (C)
CMP	B,A	6D	1	SSSX	(v)-(k) hesaplanır
	Rn,A	1D	2		sonuçtan sadece
	Rn,B	3D	2		bayraklar etkilenir
	Rn,Rn	4D	3		
	%iop,A	2D	2		
	%iop,B	5D	2		
	%iop,Rn	7D	3		
CMPA	@ETİKET	8D	3	SSSX	(A)-(k) hesaplanır
	@ETİKET(B)	AD	3		sonuçtan sadece
	*Rn	9D	2		bayraklar etkilenir
DAC	B,A	6E	1	SSSX	(k)+(v)+(e)→(v) (BCD)
	Rk,A	1E	2		
	Rk,B	3E	2		
	Rk,Rv	4E	3		
	%>iop,A	2E	2		
	%>iop,B	5E	2		
	%>iop,Rd	7E	3		
DEC	A	B2	1	SSSX	(v)-1→ (v)
	B	C2	1		
	Rv	D2	2		
DECD	A	BB	1	SSSX	(s.çifti)-1→(s.çifti)
	B	CB	1		0-FFFF geçişlerinde
	Rp	DB	2		C=0 olur
DINT		06	1	0000	0→(genelkesmeiz.biti)
DJNZ	A,ofs	BA	2	XXXX	(v)-1→ (v) , eğer
	B,ofs	CA	2		(v)≠0 ise
	Rv,ofs	DA	3		(PC)+ofs→ (PC)
DSB	B,A	6F	1	SSSX	(v)-(k)-1+(e)→(v) (BCD)
	Rk,A	1F	2		
	Rk,B	3F	2		
	Rk,Rv	4F	3		
	%>iop,A	2F	2		
	%>iop,B	5F	2		
	%>iop,Rv	7F	3		
EINT		05	1	1111	1→(genelkesmeiz.biti)
IDLE		01	1	XXXX	(PC)→(PC)kesmeye kadar
					(PC)+1→ (PC) kesmeden
					sonra

KOMUT		KODU	B	CNZI	AÇIKLAMA
INC	A	B3	1	SSSX	(v)+1→ (v)
	B	C3	1		
	Rv	D3	2		
INV	A	B4	1	OSSX	DEĞİL (v)→ (v)
	B	C4	1		
	Rv	D4	2		
JMP	ofs	E0	2	XXXX	(PC)+ofs→(PC)
JC	ofs	E3	2	XXXX	Koşul sağlanmışsa
JEQ	ofs	E2	2		
JHS	ofs	E3	2		(PC)+ofs→(PC)
JL	ofs	E7	2		sağlanmamışsa
JN	ofs	E1	2		
JNC	ofs	E7	2		(PC)+1→ (PC)
JNE	ofs	E6	2		
JNZ	ofs	E6	2		
JP	ofs	E4	2		
JPZ	ofs	E5	2		
JZ	ofs	E2	2		
LDA	@ETİKET	8A	3	OSSX	
	@ETİKET(B)	AA	3		(k)→ (A)
	*Rn	9A	2		
LDSP		0D	1	XXXX	(B) → (SP)
MOV	A,B	C0	1	OSSX	
	A,Rv	D0	2		(k) → (v)
	B,A	62	1		
	B,Rv	D1	2		
	Rk,A	12	2		
	Rk,B	32	2		
	Rk,Rv	42	3		
	%>iop,A	22	2		
	%>iop,B	52	2		
	%>iop,Rv	72	3		
MOVD	%>iop,Rp	88	4	OSSX	(saklayıcı çifti)→
	%>iop(B),Rp	A8	4		(saklayıcı çifti)
	Rp,Rp	98	3		
MOVP	A,Pv	82	2	OSSX	
	B,Pv	92	2		(k) → (v)
	%>iop,Pv	A2	3		
	Pk,A	80	2		
	Pk,B	91	2		
MPY	B,A	6C	1	OSSX	
	Rk,A	1C	2		(k)*(v)→ (A,B)
	Rk,B	3C	2		
	Rn,Rn	4C	3		
	%>iop,A	2C	2		
	%>iop,B	5C	2		
	%>iop,Rn	7C	3		

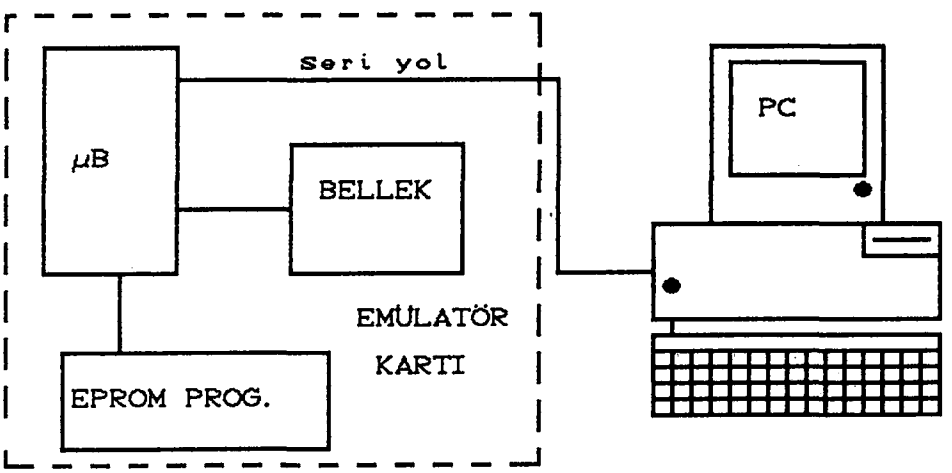
KOMUT		KODU	B	CNZI	AÇIKLAMA
NOP		00	1	XXXX	(PC)+1→ (PC)
OR	B,A	64	1	OSSX	
	Rk,A	14	2		(k).VEYA.(v)→ (v)
	Rk,B	34	2		
	Rk,Rv	44	3		
	%>iop,A	24	2		
	%>iop,B	54	2		
	%>iop,Rv	74	3		
ORP	A,Pv	84	2	OSSX	
	B,Pv	94	2		(k).VEYA.(v)→ (v)
	%>iop,Pv	A4	3		
POP	A	B9	1	OSSX	((SP))→ (v)
	B	C9	1		(SP)-1→ (SP)
	Rv	D9	2		
POP	ST	08	1	YI-ĞIN-DAN	((SP))→ (ST)
					(SP)-1→ (SP)
PUSH	A	B8	1	XXXX	(SP)+1→ (SP)
	B	C8	1		(k)→ (SP)
	Rk	D8	2		
PUSH	ST	OE	1	XXXX	(SP)+1→ (SP)
					(Durum sak.)→ ((SP))
RETI		OB	1	YI-ĞIN-DAN	(SP) → (PC) (YA)
					(SP)-1→ (SP)
					(SP) → (PC) (AA)
					(SP)-1→ (SP)
					(SP) → Durum sak.
					(SP)-1→ (SP)
RETS		OA	1	XXXX	(SP) → (PC) (AA)
					(SP)-1→ (SP)
					(SP) → (PC) (YA)
					(SP)-1→ (SP)
RL	A	BE	1	bSSX	Bit(n) → Bit(n+1)
	B	CE	1	7	Bit(7) → Bit(0) ve (e)
	Rv	DE	2		
RLC	A	BF	1	bSSX	Bit(n) → Bit(n+1)
	B	CF	1	7	Bit(7) → Bit(0) ve (e)
	Rv	DF	2		
RR	A	BC	1	bSSX	Bit(n+1) → Bit(n)
	B	CC	1	0	Bit(0) → Bit(7) ve (e)
	Rv	DC	2		
RRC	A	BD	1	bSSX	Bit(n+1) → Bit(n)
	B	CD	1	0	(e) → Bit(7)
	Rv	DD	2		Bit(0) → (e)

KOMUT		KODU	B	CNZI	AÇIKLAMA
SBB	B,A	6B	1	SSSX	(v)-(k)-1 (e)→ (v)
	Rk,A	1B	2		
	Rk,B	3B	2		
	Rk,Rv	4B	3		
	%iop,A	2B	2		
	%iop,B	5B	2		
	%iop,Rv	7B	3		
SETC		07	1	101X	1 → (e)
STA	@ETİKET	8B	3	0SSX	(A) → (v)
	@ETİKET(B)	AB	3		
	*Rv	9B	2		
STSP		09	1	XXXX	(SP) → (B)
SUB	B,A	6A	1	SSSX	(v)-(k) → (v)
	Rk,A	1A	2		
	Rk,B	3A	2		
	Rk,Rv	4A	3		
	%iop,A	2A	2		
	%iop,B	5A	2		
	%iop,Rv	7A	3		
SWAP	A	B7	1	SSSX	v(Hn,Ln)→ v(Ln,Hn)
	B	C7	1		
	Rn	D7	2		
TRAP	0-23	E8-FF	1	XXXX	(SP)+1 → (SP) (PC)(YA)→ (SP) (SP)+1 → (SP) (PC)(AA)→ (SP) (Vektör) → (PC)
TSTA		B0	1	0SSX	0 → (c)
TSTB		C1	1	0SSX	0 → (c)
XCHB	A	B6	1	0SSX	B ↔ (v)
	Rn	D6	2		
XOR	B,A	65	1	0SSX	(k).XOR.(v) → (v)
	Rk,A	15	2		
	Rk,B	35	2		
	Rk,Rv	45	3		
	%iop,A	25	2		
	%iop,B	55	2		
	%iop,Rv	75	3		
XORP	A,Pv	85	2	0SSX	(k).XOR.(Pn) → (Pn)
	B,Pv	95	2		
	%iop,Pv	A5	3		

BÖLÜM 4
TMS77C82 EMÜLATÖR KARTI

Araştırma geliştirme bilgisayarı olarak gerçeklenen sistem, genel olarak iki bölümden oluşmaktadır. Bu bölümler, mikrobilgisayar ve çevre birimlerin bulunduğu KART (devre), ve kendisinden bir terminal gibi faydalanılan PC 'dir (Şekil 4.1.) . PC'de yazılan programlar, yine PC'de makina diline çevrilip karta yüklenebilmektedir. Kart ile PC arasındaki haberleşme seri yoldan asenkron iletişim protokolüne uygun olarak yapılmaktadır. Kart üzerinde kullanıcı programları için 8 K-byte'lık bellek bulunmaktadır. Ayrıca, bu bölgedeki programlar, kart üzerinde gerçeklenen EPROM programlayıcı sayesinde, TMS77C82 mikrobilgisayarının EPROM'una aktarılabilmektedir. Sistem tarafından kullanılmayan G/Ç iskeleleri, ve zamanlayıcılar dış devrelerde (uygulama devrelerinde) kullanılabilir.

Bu bölümde emülatör kartının yapısı hakkında bilgiler verilmiştir. Sistemin bir bölümünü oluşturan PC, bir terminal gibi düşünülmüş ve fazla değinilmemiştir.



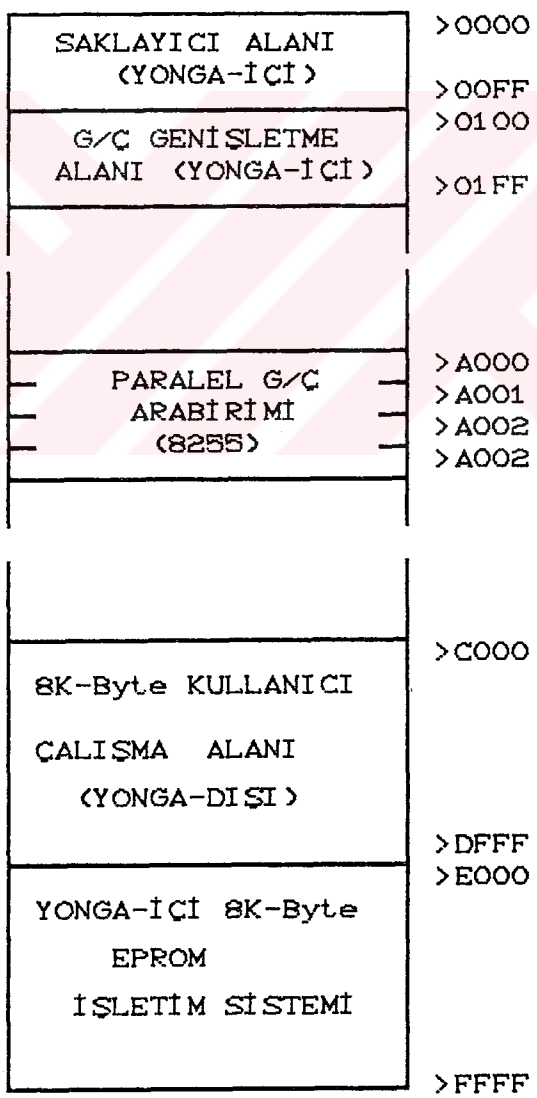
Şekil 4.1. Sistem blok şeması

4.1. Emülatör Kartının Özellikleri

Gerçeklenen kartın yapısal özelliklerini, şu temel ögeler etrafında toplamak mümkündür:

a) Kartın bellek düzeni,
b) Zamanlayıcı ve giriş/çıkış iskeleleri,
c) Kart ile PC arasında haberleşme,
d) Kartın EPROM programlayıcı bölümü.

4.1.1. Kartın Bellek Düzeni



Şekil 4.1. TMS77C82 Emülatör Kartı bellek haritası

Emülatör kartında kullanılan TMS77C82 mikrobilgisayarı, daha önce anlatılan bellek modlarından, Tam Genişleme modunda koşullandırılmıştır. Bu modda, 64 K-Byte adresleme alanına sahip olan mikrobilgisayar, bellek bölgelerinin bir kısmını yonga-içi, bir kısmını da yonga-dışı olarak kullanır (Şekil 4.1.).

Saklayıcı alanı, G/Ç genişletme alanı ve >E000'dan başlayıp >FFFF'e kadar devam eden EPROM yonga-içi olarak kullanılır. Kart için geliştirilen işletim sistemi programı da yonga-içi eprom içine yerleştirilmiştir. Kullanıcı programları için sistemde, 8 K-Byte bellek (Statik RAM) bulunmaktadır. İstenildiğinde yonga-içi EPROM dışındaki tüm adreslere bu tür bellekler bağlanarak kapasite arttırılabilir. Sisteme ait bellek haritası Şekil 4.1. 'de görülmektedir.

4.1.2. Zamanlayıcı ve Giriş/Çıkış İskeleleri

Geliştirme bilgisayarlarının sahip olması gereken önemli özelliklerden birinin G/Ç yeteneği olduğu daha önce söylenmişti. Bu nedenle gerçeklenen emülatör kartının, kullanıcıya kalan G/Ç iskeleleri ve zamanlayıcıları (bir kısmı sistem tarafından kullanılır) bu bölümde anlatılmıştır. Bellek modlarından Tam Genişleme modunda koşullandırılan mikrobilgisayarın kart üzerinde kullanıcıya kalan iskeleleri, Tablo 4.1'de gösterilmiştir.

Bunun dışında, EPROM programlayıcı için devreye ilave edilen paralel iletişim arabirimi (8255), üç sekizli (24 uç) G/Ç iskelesi daha karta kazandırmıştır. Kart üzerindeki EPROM programlayıcı kullanılmadığı sürece bu iskelelerin bir kısmı, sistemden bağımsız olarak kullanılabilir. Paralel arabirim >A000 adresine yerleştirilmiştir. Arabirimin sistem tarafından kullanılan ve kullanıcıya bırakılan iskeleleri Tablo 4.2.'de gösterilmiştir. Ayrıca, 8255 paralel iletişim arabirimi hakkında

daha ayrıntılı bilgi [8] nolu kaynaktan edinilebilir.

Tablo 4.1. Emülatör kartının G/Ç iskeleleri

İSKELE	İŞLEVİ	S	K
A İSKELESİ	A0: Eprom prog.da Ğ işareti	X	-
	A1: Eprom prog.da PGM işareti	X	-
	A2: Paralel iskele	-	X
	A3: Paralel iskele	-	X
	A4/SCLK: Par.isk./seri saat girişi	-	X
	A5/RXD: Seri veri girişi	X	-
	A6/EC2: Par.isk./olay sayacı2	-	X
	A7/EC1: Par.isk/olay sayacı11	-	X
B İSKELESİ	B0/T2OUT: Par.isk./Zam.2 çıkış	-	X
	B1/T1OUT: Par.isk./Zam.1 çıkış	-	X
	B2: Paralel iskele	-	X
	B3/TXD: Par.isk./Seri birim veri gön	X	-
	B4/ALATCH: yonga-dışı bellek	X	-
	B5/R/W:	X	-
	B6/ENABLE: kontrolu	X	-
	B7/CLKOUT:	-	X
C İSKELESİ	Veri/Adres (AA)	X	-
D İSKELESİ	Adres (YA)	X	-

S: Sistem tarafından kullanılıyor.
K: Kullanıcıya bırakılmış.

Tablo 4.2. Eprom Programlayıcı bölümünün G/Ç iskelesi olarak kullanılması

İSKELE	İŞLEVİ	S	K
A İSKELESİ	PA0-PA4: Eprom p. adres. (YA)	X	X
	PA5: Paralel iskele	-	X
	PA6-PA7: Eprom p. E ve Vpp işareti	X	-
B İSKELESİ	Eprom p. adres (AA)/Par. iskele	X	X
C İSKELESİ	Eprom p. veri /par. iskele	X	X

S: Sistem tarafından kullanılıyor
K: Kullanıcıya bırakılmış

Emülatör kartında, mikrobilgisayarın yonga-içi zamanlayıcılarından biri kullanılıp, diğer ikisi kullanıcıya bırakılmıştır. Sistem tarafından kullanılan üçüncü zamanlayıcı, seri haberleşmede veri akış hızı kaynağı (Baud rate generator) olarak kullanılmaktadır.

Tamamıyla kullanıcıya bırakılan zamanlayıcı 1 ve 2 istenildiğinde olay sayıcı (event counter) olarak da kullanılabilir.

4.1.3. Kart ile PC Arsında Haberleşme

Kişisel bilgisayarların artık birçok yerde kullanılıyor olması ve bu tezde gerçeklenen geliştirme bilgisayarının ucuza mal edilebilmesi için, ön çalışmalarda PC'den faydalanılması kararına varılmıştır. Karta tuş takımı ve gösterge sistemi yapmak yerine PC ekranından ve tuş takımından faydalanılmıştır. Böylece, maliyetin düşürülmesi yanında PC'nin sağladığı avantajlardan da faydalanılmıştır. Emülatör kartı ile çalışma sırasında, PC terminal gibi davranır. Bunun için PC'de bir program (VT 220) koşturulur. Veri iletişimi seri yoldan yapılır.

Simgesel dildeki programlar PC'de herhangi bir dosyada (editör) bulunur. Daha sonra, bu dosyayı makina diline çeviren bir program (XASM7000.EXE) çalıştırılır. Yazılan sigesel dildeki program dosyası, çalıştırılan programın giriş dosyasını oluşturur. Bu programın iki çıkış dosyası sırasıyla; <dosya adı>.LST ve <dosya adı>.MAP'dir. MAP dosyaları simgesel dildeki programın makina diline (object code) dönüştürülmüş halidir ve bu dosya seri yoldan karttaki belirlenen bellek bölgesine yüklenir.

PC ile kart arasında seri bağlantı yapıldıktan sonra, veri transferi asenkron seri iletişim protokolüne uygun olarak yapılır. Kartın haberleşme protokol parametreleri aşağıdaki gibidir:

İletişim hızı (Baud Rate)	:	300 bit/s.
Veri uzunluğu	:	8 bit
Dur biti sayısı	:	2
Eşlik biti (Parity Bit)	:	Tek

Seri haberleşme arabirimi, mikrobilgisayarın içinde (yonga-içi) gerçeklendiği için, kart üzerindeki eleman sayısı oldukça azalmıştır.

4.1.3. Kartın EPROM Programlayıcı Bölümü

Sürekli kullanılan kullanıcı programları, silinip kaybolmaması için EPROM'a aktarılır. Bu tezde kullanılan mikrobilgisayar, aynı yonga içinde 8 K-byte'lık EPROM içerdiğinden, kullanıcı proramlarının bu tür EPROM'lara yazılmasını sağlayan devre, kart üzerinde gerçeklenmiştir.

Başka EPROM programlayıcılar kullanıldığında, kullanıcı programının makina kodu karşılıkları uygun formatta yazılmalıdır .

Mikrobilgisayar üzerindeki EPROM (8 K-byte), birçok uygulamaya yetecek kapasitededir. Geliştirme işlemi tamamlanmış kullanıcı programlarını yonga-içi EPROM'a yüklemek, uygulama devresinin, hem daha basit ve hem de daha ucuz olmasını sağlayacaktır. Bu yoldan hareketle, emülatör kartına EPROM programlayıcı ilave edilmiştir. Kart üzerindeki soket TMS77C82 mikrobilgisayarı için uyumludur. İstenildiği taktirde, uç değişim soketi kullanılarak, 8 K-byte'lık CMOS (V_{pp} = 12,5 V.) EPROM'lar da aynı kart üzerinde programlanabilir. Programlama için zamanlama diyagramı eklerde verilmiştir.

BÖLÜM 5
İŞLETİM SİSTEMİ PROGRAMI

Bir bilgisayar yalnız donanım olarak var ise bunun çalışmayacağı, ve çalışabilmesi için programlanması gerektiği açıktır. Ancak, komutlardan başka bir şey bilmeyen bir bilgisayar nasıl programlanacaktır? Böyle bir bilgisayara kullanıcı isteklerini nasıl aktaracaktır? [3]

Bir bilgisayara program yükleme ve bilgisayarı kullanıcının emrine hazır etmek için de bir program gerekir. Bu amaçla yazılmış olan programa, İşletim Sistemi Programı denir. İşletim sistemi doğal olarak bir başka ortamda hazırlanır ve kullanılacak bilgisayara yerleştirilir. [6]

İşletim sistemleri, sağladığı olanaklar ve yeteneklerine göre farklı uzunluklarda olabilirler. Özellikle, araştırma ve eğitim amacıyla kullanılan, kısıtlı yeteneği bulunan, geliştirme mikrobilgisayarlarında, işletim sistemi makina dilinde program girebilmeye olanak sağlar. Bu boydaki işletim sistemlerine, çoğunlukla Monitör Programı denir. Monitör programı kullanıcıya aşağıdaki olanakları sağlamalıdır. [6]

1. Tuş takımı ve bir gösterge birimi aracılığı ile kullanıcıya bilgisayarı kullanma olanağı sağlamalıdır.

2. MİB'nin içi, yani tüm saklayıcılar ve bellek gözleri gözlenebilmeli ve istendiğinde değiştirilebilmelidir.

3. Araya sokma ve aradan çıkarma işlemi yapılabilmelidir.

4. Bağıl adresleme olanağı sağlamalıdır.

5. Belli bir başlangıç noktasından, programı çalıştırmak mümkün olabilmelidir.

Daha yetenekli mikrobilgisayar kartlarında monitör programlarından daha gelişmiş programlar vardır ve bunlar işletim sistemi olarak anılırlar. Ayrıca işletim sistemlerinin de çeşitlileri vardır. Bir bilgisayar amaca göre farklı işletim sistemleri ile çalıştırılabilir. İşletim sistemi, kullanıcıya aşağıdaki olanakları sağlamalıdır.

1. Program yazmaya ve düzeltmeye olanak sağlamalıdır.

2. Yazılmış olan kaynak programı, simgesel dil derleyicisi ya da yüksek düzey dil derleyicisi kullanarak makina diline çevirebilmelidir.

Tablo 5.1. İşletim sistemi komutları

İŞLETİM SİST. KOMUTU		İŞLEVİ
M: XXXX YYYY		XXXX-YYYY arasındaki bellek içeriği gör.
W: XXXX		XXXX adresindeki veri görüme
	/	XXXX adresindeki veriyi değiştirme
	F	İleri adres arama
	B	Geri adres arama
	.	Bağıl adres hesabı
R		Saklayıcı içeriği görme
	/	Saklayıcı içeriği değiştirme
L: XXXX		XXXX adresinden itibaren programı yükle
E: XXXX YYYY		XXXX adresinden başlayan program eproma YYYY adresinden itibaren yüklenir
	B	YYYY adresinden itibaren eprom boş mu?
	P	YYYY adresinden itibaren programlama
	V	XXXX adresindeki prog.la epromda YYYY adresindeki prog. karşılaştırılır
G: XXXX		XXXX adresindeki programı icra ettirir

Bu tezde gerçeklenen sistemde, kaynak programlar PC'de yazılıp, derlenip makine diline çevrilir. Daha sonra, işletim sistemi, PC'de makine diline çevrilmiş dosyaların karta aktarılmasını sağlar. Tablo 5.1'de bu işletim sisteminin gerçeklediği işlevler gösterilmiştir.

Bilgisayar açıldığında ya da sıfırlandığında işletim sistemi programı kendiliğinden işlemeye başlar ve ekrana ">" işaretini göndererek kullanıcının komutlarını bekler.

5.1. Bellek İçeriği Görme İşlevi (M)

Yonga-içi veya yonga-dışı bellek gözlerinin gözlenebilmesini sağlar. Bu işlevin seçilmesi ve çalıştırılması aşağıdaki şekilde yapılır:

1. "M" (büyük M) tuşuna basılarak bu işlev seçilir. Ekranda ">M:" görülecektir.

2. Görülmek istenen bellek bölgesinin başlangıç adresi onaltılık sayı olarak girilir (4 hane). Örneğin >C000 girilmiş olsun, ekranda ">M:C000 " görülecektir.

3. Son adres onaltılı sayı sisteminde girilir. Mikrobilgisayar bu iki adres arasındaki verileri göndermeye başlar. Veriler, bir satırda 16 byte olacak şekilde ve onaltılı sayı düzeninde ekrana gönderilir. Son adres >C008 olsun. Bu durumda bilgisayar çıkışı aşağıdaki gibi olur.

```
>M:C000 C008
C000:01 A2 45 67 83 55 BC 5C F7 FF 21 00 01 88 32 CC
 >
```

Mikrobilgisayar, bellek dökümünü ekrana verdikten sonra, ekrana ">" karakterini göndererek bir sonraki komutu

bekler. Bu işlevden, işlem tamamlanmadan çıkılmak istenirse "X" tuşuna basılır.

5.2. Bellek İçeriği Değiştirme İşlevi (W)

Bellek içeriği değiştirme ve bağıl adres hesabı için aşağıdaki yol izlenir.

1. "W" tuwuna basılır, ekranda ">W:" görülecektir.

2. İçeriği değiştirilecek ya da bağıl adres hesabı yapılacak ilk adres girilir (4 hane). Eğer girilen adres eprom bölgesi ise, işletim sistemi kullanıcıyı işlev dışına çıkarır. Bu durumda bilgisayar tekrar komut bekler. Adres eprom dışı bir bölge ise, o adresteki veri görülür, örneğin adres >C000 olsun:

```
>W:C000 01
```

3. Şimdi bilgisayar ikinci komutu beklemektedir. Bu komutlar ileri adres tarama "F", geri adres tarama "B", veri değiştirme "/" ve bağıl adres hesabı "." dır.

3.a. "F" tuşuna basılarak adres 1 arttırılır ve hemen bellek içeriği de yanına yazılır. Yukardaki durumda "F" tuşuna basılırsa ekrandaki son durum aşağıdaki gibi olur.

```
>W:C000 01
C001 A2
```

3.b. "B" tuşuna basılırsa yukarıdaki işlemin tersi yapılır.

3.c. "/" tuşun basıldığında bilgisayar iki haneli (1 byte) veri bekler. Girilen yeni veri, belirlenen adrese yazılmış olur. Yeni veri >BB olsun.
Ekrandaki son durum aşağıdaki gibidir.

```
>W:C000 01
C001 A2/BB
C001 BB
```

3.d. Bulunulan adresten herhangi bir adrese olan uzaklık hesaplanmak istenirse "." tuşuna basılır. Aradaki farkın hesaplanması istenen son adres, >C045 olsun bu durumda ekranda aşağıdakiler görülür.

```
>W:C000 01 . C045 0044
>
```

5.3. Saklayıcı İçeriği Görme ve Değiştirme (R)

MIB saklayıcıları olan, akümülatörlerin (A,B), durum saklayıcısının (ST), yığın göstergesinin (SP) ve program sayacının (PC) içerikleri kullanıcı tarafından görülebilir ve yığın göstergesi dışında değiştirilebilir. Yığın göstergesi, program içinde bir simgesel dil komutu ile istenen değere getirilebilir. Bunun dışında saklayıcı içeriği değiştirilirken yığın göstergesinin değiştirilmesi, ana programa geri dönüşü engelliyeceğinden izin verilmemiştir.

Saklayıcı içeriklerinin görülmesi için "R" tuşuna basılır. Ekrandaki durum aşağıdaki gibidir.

```
>A:05 B:DC F:E0 S:67 P:C000
```

Burada;
- A: A akümülatörü
- B: B akümülatörü
- F: Durum saklayıcısı (ST)
- S: Yığın göstergesi (PS)
- P: Program sayacı (PC) 'dır.

Saklayıcı içeriği değiştirilmek istendiğinde "/"

tuşuna basılır ve hangi saklayıcı değiştirilecekse, etiketi, arkasından da yeni değeri girilir. A akümülatörü değiştirilecek olsun, ekranda aşağıdakiler görülür.

```
>A:05 B:DC F:E0 S:67 P:C000
/A:FF
A:FF B:DC F:E0 S:67 P:C000
```

5.4. Program Yükleme İşlevi (L)

Simgesel dildeki kullanıcı programları PC'de yazılıp derlendikten, makina diline çevrildikten sonra, emülatör kartına yüklenir. Program karta yüklenmeden önce, işletim sistemi tüm bellek gözelerini (RAM bölgesini) >FF ile doldurur. Sonra, PC'den gelen makina kodlarını belirlenen başlangıç adresinden itibaren yerleştirir.

Program yükleme işlevi, "L" tuşuna basılarak ve programın başlangıç adresi girilerek seçilir. >C000 adresine yüklenmek istenen bir program için işletim sistemi komutu aşağıda verilmiştir.

```
>L:C000
```

5.5. İcra Etme İşlevi (G)

Kullanıcı programlarının koşturulması için, "G" tuşuna basılıp, kullanıcı programının başlangıç adresi girilir. >C000'da başlayan bir program için örnek komut aşağıdaki gibidir.

```
>G:C000
```

5.6. EPROM Programlayıcı İşlevi (E)

Üzerinde çalışmaların tamamlandığı kullanıcı programlarının aynı türden mikrobilgisayarların EPROM'una

kaydedilebilmesi için, emülatör kartına EPROM programlayıcı ilave edilmiştir. Kart üzerindeki EPROM programlacı ile, EPROM'un boş olup olmadığı kontrol edilebilir, EPROM programlanabilir ve bellekteki veri ile EPROM'daki karşılaştırılabilir. EPROM'un boş olup olmadığı kontrolu ve programlama, başlangıç adresi girilerek yapılmaktadır. Dolayısıyla, silinmeden, başlangıç adresi değiştirilerek aynı EPROM'a birden fazla program yüklenebilir.

Bu işlevin çalıştırılması aşağıdaki şeklide olur.

1. "E" tuşuna basılarak bu işlev seçilir. Ekranda ">E:" görülür.

2. Kullanıcı belleği (>C000->DFFF) içindeki, EPROM'a aktarılması düşünülen programın başlangıç adresi girilir. Örneğin bellek başlangıç adresi >C00A olsun. Ekrandaki durum aşağıdaki gibidir.

>E:C00A

3. Epromun başlangıç adresi girilir. Bu adres >0000 ile >1FFF arasındadır. Eprom başlangıç adresi >0006 olsun bu durumda:

>E:C000 0006

4. Şimdi hangi işin yapılacağı makinaya girilmelidir. Bu işler;

a) EPROM'un boş olup olmadığı "B",
b) Kopyalama "P",
c) Karşılaştırma "V"'dır.

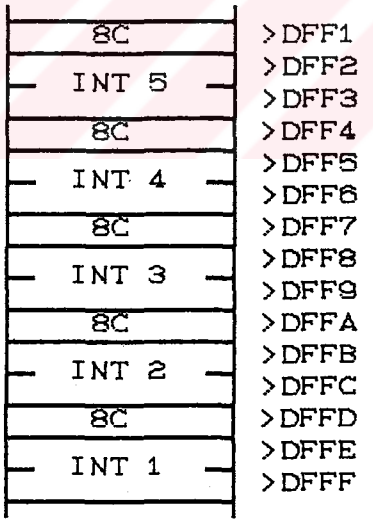
EPROM boş değilse, veya karşılaştırma sonucunda bellekteki program ile EPROM'a yazılan program farklı ise işletim sistemi hata mesajı verir.

5.7. Kullanıcı Programları

İşletim sistemi, simgesel dilde yazılan kullanıcı programlarına bazı sınırlamalar getirmektedir. Bu sınırlamalar aşağıda maddeler halinde verilmiştir.

1. Saklayıcı alanındaki 26 saklayıcı (R2-R27), işletim sistemi tarafından kullanılmaktadır. Kullanıcı programlarında bu saklayıcılar kullanılmamalıdır.

2. Sıfırlama hariç tüm kesmeler kullanıcıya bırakılmıştır. Bu kesmelere ait vektörler, kullanıcı belleğinin aşağıda belirtilen adreslerine yerleştirilmelidir. Ayrıca, her vektörden önce, dallanma (BR) komutunun kodu olan >8C girilmelidir. Aşağıdaki şekilde kullanıcı kesme vektörleri ve adreslerinin bellekteki durumu gösterilmiştir.



Şekil 5.1. Kullanıcı kesme vektörlerinin bellekteki durumu

3. Bellekteki kullanıcı programları, mikrobilgisayar içindeki EPROM'a aktarılmak istendiğinde vektörler düzenlenmelidir.

Vektörlerin tekrar yazılması, işletim sisteminin sağladığı bellek yazma işlevi ile yapılabileceği gibi, kaynak program içinde ikinci bir başlangıç adresi seçilerek de yapılabilir. Aşağıda bir örnek verilmiştir.

```
        AORG    >C000
        .
        .
******* SERVİS PROGRAMI ****************
INT1    .
        .
        .
        AORG    >DFFD
        BYTE    >8C
        DATA    INT1
        END
```

Eğer bu program EPROMa aktarılacaksa ikinci kısım şu şekilde değiştirilmelidir.

```
        AORG    >FFFC
        DATA    INT1
```

4. Bir program içinde en fazla iki kere AORG komutu kullanılabilir.

5. Her kullanıcı programının son komutu BR >E006 olmalıdır. Böylece program icra edildikten sonra işletim sistemine dönülür.

SONUÇLAR VE ÖNERİLER

Bu çalışmada Texas İnstruments firmasının ürettiği, TMS77C82 mikrobilgisayarı kullanılarak,bir araştırma geliştirme bilgisayarı gerçeklenmiştir. Gerçekleme, TMS 77C82 Emülatör Kartı olarak adlandırılmış ve mikrobilgisayarın özellikleri anlatılmıştır.

Günümüz teknojisi ile sabit ve silinebilir belleklerin ve diğer arabirimlerin aynı yonga üzerinde gerçeklenmesi bilgisayar boyutlarını küçültmüş, fiyatlarını düşürmüştür. TMS77C82 mikrobilgisayarı, yonga içi bellekleri (RAM,ROM), zamanlayıcıları ve seri arabirimi ile oldukça gelişmiş 8-bitlik bir mikrobilgisayardır. Yonga içi saklayıcı alanı ve giriş/çıkış genişletme alanı saklayıcıları çok amaçlı olarak kullanılabilmektedir. Tek komutla bu saklayıcılarla işlem yapmak mümkündür. Ayrıca, mikrobilgisayar uygulama yerine göre farklı dört bellek modunda koşullandırılabilmektedir. Mikrobilgisayar içindeki üç zamanlayıcıdan ikisi tamamıyla kullanıcıya bırakılmış, birisi seri haberleşmede kullanılmıştır. Mikrobilgisayar, sıfırlama ile birlikte altı kesme seviyesine sahiptir. Bu kesmelerin üçü dış devrelere, diğer üçü de yonga-içi arabirimlere hizmet verir.

Çok yönlü kullanılabilen mikrobilgisayarın özellikleri, her bellek modunda değişmektedir. Tezde, tam genişleme modunda koşullandırılan mikrobilgisayar, bu moda ait özellikleri emüle edebilmektedir. Karta, diğer modlarda çalışması düşünülen programlar da yüklenebilir ve bu programlar EPROMa yazılabilir.

Bu sistemin en büyük üstünlüğü, program geliştirme ve EPROMa yazma işlevlerinin, başka bir cihaz gerektirmeden aynı kart üzerinde yapılabilmesidir. EPROM

programlayıcı TMS77C82 mikrobilgisayarının EPROM'unu programlayacak şekilde tasarlanmıştır. Uçlar uygun şekilde değiştirilerek, 8 K-byte'lık CMOS EPROM'lar aynı sistemde programlanabilir.

Kartın EPROM programlayıcı bölümünde kullanılan paralel giriş çıkış iskeleleri, başka amaçlar için de kullanılabilir. Ancak bu durumda, EPROM programlayıcı kullanılamaz.

İşletim sistemi programı, bellek içeriği görme/değiştirme, saklayıcı içeriği görme/değiştirme, program yükleme, program icra etme ve EPROM programlama gibi işlevleri yerine getirmektedir. Bu program sistemde kullanılan mikrobilgisayarın epromuna kaydedilerek, kart üzerinde yerden ve malzemeden tasarruf edilmiştir.

Gerçeklenen sistemde eksik görülen ya da bundan sonraki bir çalışmada tamamlanabilecek unsurlar şunlardır.

1. İşletim sisteminin çok gelişmiş hata mesajları vermemesi, ve birçok yerde yazılanı düzeltmeye imkan tanımaması istenmeyen bir durumdur.

2. Sistemde, kullanıcı programları için duraksama noktası koyma ve adım adım icra işlevleri gerçeklenmemiştir. Program hatalarının bulunmasında oldukça kolaylık sağlayan bu işlevler bir araştırma geliştirme bilgisayarı için eksikliktir.

MİB saklayıcılarından durum saklayıcısında, adım adım icra kontrol bayrağının (Trace Flag) olmaması bu işlevin gerçeklenmesi için, diğer mikroişlemcilerden farklı bir yol izlenmesini gerektirecektir. Böyle bir işlev, uygun kesmelerden birinin sürekli aktif hale getirilmesi ile yapılabilir. Böylece mikrobilgisayar her komut icra edildikten sonra ilgili servis programına dallanır.

KAYNAKLAR

[1] TOKHEIM Roger L., Microprocessor Fundamentals, Schaum's Outline Series, McGraw-Hill 1983

[2] TMS7000 Family Data Manual, Texas Instruments, 1989

[3] ADALI Eşref, Mikroişlemciler Ders Notu, İ.T.Ü.

[4] TMS77C82 8 Bit CMOS Eprom Microcomputer, Advance Information, 1988

[5] ZAKS R., LESEA A., Microprocessor Interfacing Techniques, Sybex, 1979

[6] DONOVAN S.E., MADNICK J.J., Operating Systems, Mc Graw-Hill, 1982

[7] TMS7000 Assembly Language Programmer's Guide, Texas Instruments, 1983

[8] Link Edior Users's Guide, Texas Instruments, 1985

Ek A TMS77C82 Mikrobilgisayarının Elektriksel Özellikleri

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC} (see Note 6) ..... −0.3 to 7 V
Supply voltge, V_{PP} (MC pin) ..... −0.3 V to 14 V
Input voltage range ..... −0.3 V to V_{CC} +0.3 V
Output voltage range ..... −0.3 V to V_{CC} +0.3 V
Maximum I/O buffer current (per pin) ..... ±10 mA
Maximum supply current, I_{CC} ..... 60 mA
Maximum supply current, I_{SS} ..... −60 mA
Storage temperature range ..... −55°C to 150°C
Operating free air temperature ..... −40°C to 85°C

†Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
NOTE 6: All voltage values in this data sheet are with respect to V_{SS}.

recommended operating conditions

		MIN	TYP	MAX	UNIT
Supply voltage		3		6	V
Programming supply voltage (MC pin)		12	12.5	13	V
INT1, INT3, RESET, X'TAL PIN					
High level input voltage	5.5V ≤ V_{CC} ≤ 6V	V_{CC} − 1.0		V_{CC}	V
	4.5V ≤ V_{CC} ≤ 5.5V	V_{CC} − 0.7		V_{CC}	V
	3.5V ≤ V_{CC} ≤ 4.5V	V_{CC} − 0.5		V_{CC}	V
	2.5V ≤ V_{CC} ≤ 3.5V	V_{CC} − 0.35		V_{CC}	V
Low level input voltage	5.5V ≤ V_{CC} ≤ 6V	0		1.00	V
	4.5V ≤ V_{CC} ≤ 5.5V	0		0.70	V
	3.5V ≤ V_{CC} ≤ 4.5V	0		0.50	V
	2.5V ≤ V_{CC} ≤ 3.5V	0		0.35	V
MC PIN					
High level input voltage, VIH	5V ≤ V_{CC} ≤ 6V	V_{CC} − 0.5		V_{CC}	V
	4V ≤ V_{CC} ≤ 5V	V_{CC} − 0.4		V_{CC}	V
	3V ≤ V_{CC} ≤ 4V	V_{CC} − 0.3		V_{CC}	V
	2.5V ≤ V_{CC} ≤ 3V	V_{CC} − 0.2		V_{CC}	V
Low level input voltage, VIL	5V ≤ V_{CC} ≤ 6V	0		0.5	V
	4V ≤ V_{CC} ≤ 5V	0		0.4	V
	3V ≤ V_{CC} ≤ 4V	0		0.3	V
	2.5V ≤ V_{CC} ≤ 3V	0		0.2	V
PORT (EXCEPT INT1, INT3, RESET, X'TAL, MC					
High level input voltage, VIH	5V ≤ V_{CC} ≤ 6V	V_{CC} − 1.3		V_{CC}	V
	4V ≤ V_{CC} ≤ 5V	V_{CC} − 1.0		V_{CC}	V
	3V ≤ V_{CC} ≤ 4V	V_{CC} − 0.7		V_{CC}	V
	2.5V ≤ V_{CC} ≤ 3V	V_{CC} − 0.4		V_{CC}	V
Low level input voltage, VIL	5V ≤ V_{CC} ≤ 6V	0		1.5	V
	4V ≤ V_{CC} ≤ 5V	0		1.1	V
	3V ≤ V_{CC} ≤ 4V	0		0.7	V
	2.5V ≤ V_{CC} ≤ 3V	0		0.3	V

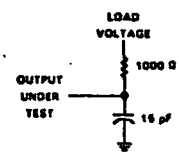
electrical characteristics over full range of operating conditions (V_{CC} = 4.5 V to 5.5 V)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
ICC Supply current	Operating mode	V_{CC} = 5.0 V, f = 6.0 Mhz		16.1	24.8	mA
		V_{CC} = 5.0 V, f = 4.0 Mhz		14.4	22.2	mA
		V_{CC} = 5.0 V, f = 1.0 Mhz		11.2	17.2	mA
	Wake-up modes 1 & 5 (see Note 1)	< Freq = F Mhz >		1.2 * F + 0.3	2.0 * F + 0.5	mA
		< Freq = 6.0 Mhz >		7.5	12.5	mA
	Wake up mode 2 (see Note 1)	< Freq = F Mhz >		0.86 * F + 0.3	1.44 * F + 0.5	mA
		< Freq = 6.0 Mhz >		5.5	9.1	mA
	Wake-up modes 3 & 4 (see Note 1)	< Freq = F MHz >		0.52 * F + 0.3	0.8 * F + 0.5	mA
		< Freq = 6 MHz >		3.4	5.8	mA
	Halt-off mode	V_{CC} = 5 V			20	μA
V_{OH} High-level output voltage		V_{CC} = 5.0 V, I_{OH} = −1.0 mA	2.50	4.5		mA
		V_{CC} = 5.0 V, I_{OH} = 0.3 mA	4.50	4.8		V
V_{OL} Low-level output voltage		V_{CC} = 5.0, I_{OL} = 1.7 mA		0.3	0.4	V
I_{OH} Output source current (V_{HO} = V_{CC} − 0.5 V)		V_{CC} = 3.0 V	−100	−700		μA
		V_{CC} = 4.0 V	−0.2	1.0		mA
		V_{CC} = 5.0 V	−0.3	−1.2		mA
	V_{OH} = 2.5 V	V_{CC} = 5.0 V	−1.0	−5.0		mA
I_{OL} Output sink current (V_{OL} = 0.4 V)		V_{CC} = 3.0 V	0.7	2.0		mA
		V_{CC} = 4.0 V	1.02	0.4		mA
		V_{CC} = 5.0 V	1.7	2.0		mA
I_I Input leakage current	MC	V_I = V_{SS} or V_{CC}		±0.1	±5	μA
	All others	V_I = V_{SS} to V_{CC}		±0.1	±5	μA
C_I Input capacitance				5		pF

All inputs = V_{CC} or V_{SS} (except XTAL2). All I/O and output pins are open circuit.

Note 1:

MODE	TIMER-1		TIMER-2	UART
WAKE UP1	USE		USE	USE
WAKE UP2	TIMER-1	OR	USE	USE
WAKE-UP3	OFF		OFF	USE
WAKE-UP4	TIMER-1	OR	USE	OFF
WAKE UP5	USE		USE	OFF
HALT-OFF	OFF		OFF	OFF

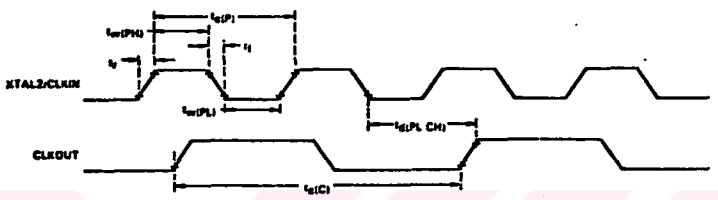


Ek B TMS77C82 Saat ve Zamanlama Özellikleri

recommended crystal operating conditions over full operating range

PARAMETER			MIN	TYP	MAX	UNIT
f_{osc}	Crystal frequency	V_{CC} = 3.0 V	0.5		1.0	MHz
		V_{CC} = 4.0 V	0.5		3.0	MHz
		V_{CC} = 5.0 V ± 10%	0.5		6.0	MHz
		V_{CC} = 6.0 V	0.5		6.0	MHz
	CLKIN duty cycle		45		55	%
$t_c(P)$	Crystal cycle time	V_{CC} = 2.5 V	1000		2000	ns
		V_{CC} = 4.0 V	333		2000	ns
		V_{CC} = 5.0 V	167		2000	ns
		V_{CC} = 6.0 V	167		2000	ns
$t_c(C)$	Internal state cycle time	V_{CC} = 2.5 V	2000		4000	ns
		V_{CC} = 4.0 V	666		4000	ns
		V_{CC} = 5.0 V	333		4000	ns
		V_{CC} = 6.0 V	333		4000	ns
$t_w(PH)$	CLKIN pulse duration high		50			ns
$t_w(PL)$	CLKIN pulse duration low		50			ns
t_r	CLKIN rise time				30	ns
t_f	CLKIN fall time				30	ns
$t_d(PL\text{-}CH)$	CLKIN fall to CLKOUT rise delay time			140	250	ns

clock timing



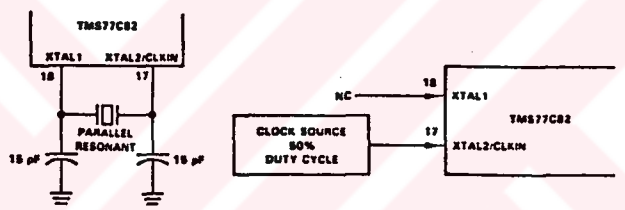


FIGURE 27. RECOMMENDED CLOCK CONNECTIONS

memory interface timing as a function of frequency

In the table below, $t_{c(C)}$ is the period of the internal clock, and $t_{c(C)} = 2/f_{osc}$. At 6 MHz $t_{c(C)}$ would be 333 ns. Minimum and maximum times may be calculated by using the formulas below with the appropriate clock period.

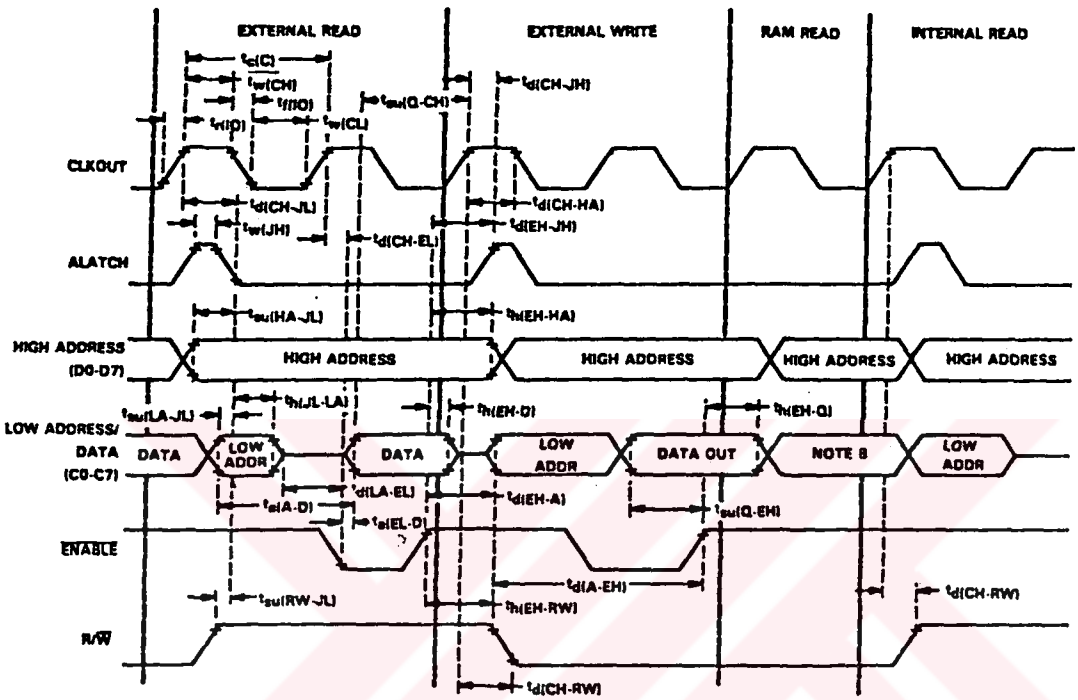
PARAMETER		MIN	TYP	MAX	UNIT
$t_c(C)$	CLKOUT cycle time (see Note 7)	333		2000	ns
$t_w(CH)$	CLKOUT high pulse duration	$0.5t_{c(C)} - 50$	$0.5t_{c(C)}$	$0.5t_{c(C)} + 50$	ns
$t_w(CL)$	CLKOUT low pulse duration	$0.5t_{c(C)} - 50$	$0.5t_{c(C)}$	$0.5t_{c(C)} + 50$	ns
$t_d(CH\text{-}AL)$	Delay time, CLKOUT rise to ALATCH fall	$0.5t_{c(C)} - 50$	$0.5t_{c(C)}$		ns
$t_d(CH\text{-}AH)$	Delay time, CLKOUT rise to ALATCH rise	$0.25t_{c(C)} - 40$	$0.25t_{c(C)}$		ns
$t_d(CH\text{-}HA)$	Delay time, CLKOUT rise to high address valid	$0.25t_{c(C)} - 40$	$0.25t_{c(C)}$		ns
$t_w(AH)$	ALATCH high pulse duration	$0.25t_{c(C)} - 40$	$0.25t_{c(C)}$		ns
$t_{su}(HA\text{-}AL)$	Setup time, high address valid before ALATCH fall	$0.25t_{c(C)} - 40$	$0.25t_{c(C)}$		ns
$t_{su}(LA\text{-}AL)$	Setup time, low address valid before ALATCH fall	$0.25t_{c(C)} - 55$	$0.25t_{c(C)}$		ns
$t_h(AL\text{-}LA)$	Hold time, low address valid after ALATCH fall	$0.25t_{c(C)} - 50$	$0.25t_{c(C)}$		ns
$t_{su}(RW\text{-}AL)$	Setup time, R/$\overline{W}$ valid before ALATCH fall	$0.25t_{c(C)} - 60$	$0.25t_{c(C)}$		ns
$t_h(EH\text{-}RW)$	Hold time, R/$\overline{W}$ valid after $\overline{ENABLE}$ rise	$0.5t_{c(C)} - 60$	$0.5t_{c(C)}$		ns
$t_h(EH\text{-}HA)$	Hold time, high address valid after $\overline{ENABLE}$ rise	$0.5t_{c(C)} - 60$	$0.5t_{c(C)}$		ns
$t_{su}(D\text{-}EH)$	Setup time, data output valid before $\overline{ENABLE}$ rise	$0.5t_{c(C)} - 35$	$0.5t_{c(C)}$		ns
$t_h(EH\text{-}D)$	Hold time, data output valid after $\overline{ENABLE}$ rise	$0.5t_{c(C)} - 50$	$0.5t_{c(C)}$		ns
$t_d(LA\text{-}EL)$	Delay time, low address high impedance to $\overline{ENABLE}$ fall	$0.33t_{c(C)} - 35$	$0.25t_{c(C)}$		ns
$t_d(EH\text{-}A)$	Delay time, $\overline{ENABLE}$ rise to next address drive	$0.5t_{c(C)} - 60$	$0.5t_{c(C)}$		ns
$t_a(EL\text{-}D)$	Access time, data input valid after $\overline{ENABLE}$ fall	$0.75t_{c(C)} - 130$	$0.5t_{c(C)} - 20$		ns
$t_a(A\text{-}D)$	Access time, address valid to data input valid	$1.5t_{c(C)} - 175$	$1.5t_{c(C)} - 100$		ns
$t_d(A\text{-}EH)$	Delay time, address valid to $\overline{ENABLE}$ rise	$1.5t_{c(C)} - 50$	$1.5t_{c(C)}$		ns
$t_h(EH\text{-}D)$	Hold time, data input valid after $\overline{ENABLE}$ rise	0			ns
$t_d(EH\text{-}AH)$	Delay time, $\overline{ENABLE}$ rise to ALATCH rise	$0.5t_{c(C)} - 60$	$0.5t_{c(C)}$		ns
$t_d(CH\text{-}EL)$	Delay CLKOUT rise to $\overline{ENABLE}$ fall		30		ns

NOTE 7: $t_{c(C)}$ is defined to be $2/f_{osc}$ and may be referred to as a machine state or simply a state.

As an example, consider calculating the minimum data out hold time from $\overline{ENABLE}$ rising [$t_{h(EH\text{-}Q)}$]. At 6 MHz this would give:

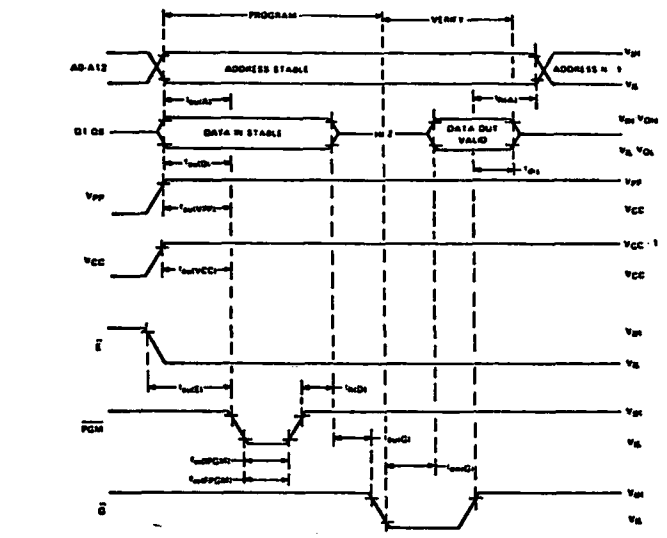
$$t_{h(EH\text{-}Q)} = 0.5t_{c(C)} - 50 \text{ ns}$$
$$= 0.5(333 \text{ ns}) - 50 \text{ ns}$$
$$= 166.5 \text{ ns} - 50 \text{ ns}$$
$$\therefore t_{h(EH\text{-}Q)} = 116.5 \text{ ns}$$

Ek C Bellek Okuma ve Yazma Çevrimleri

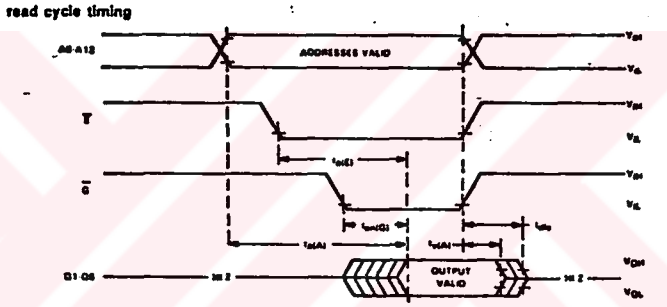


Ek D TMS77C82 Mikrobilgisayarı Eprom Programlama Çevrimleri ve Mikrobilgisayar/Eprom Uç Değişimi

program cycle timing



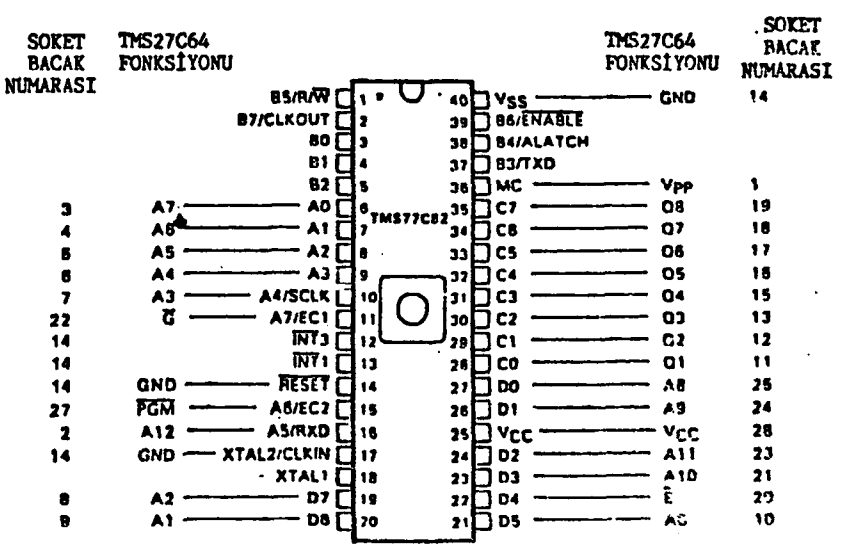
read cycle timing



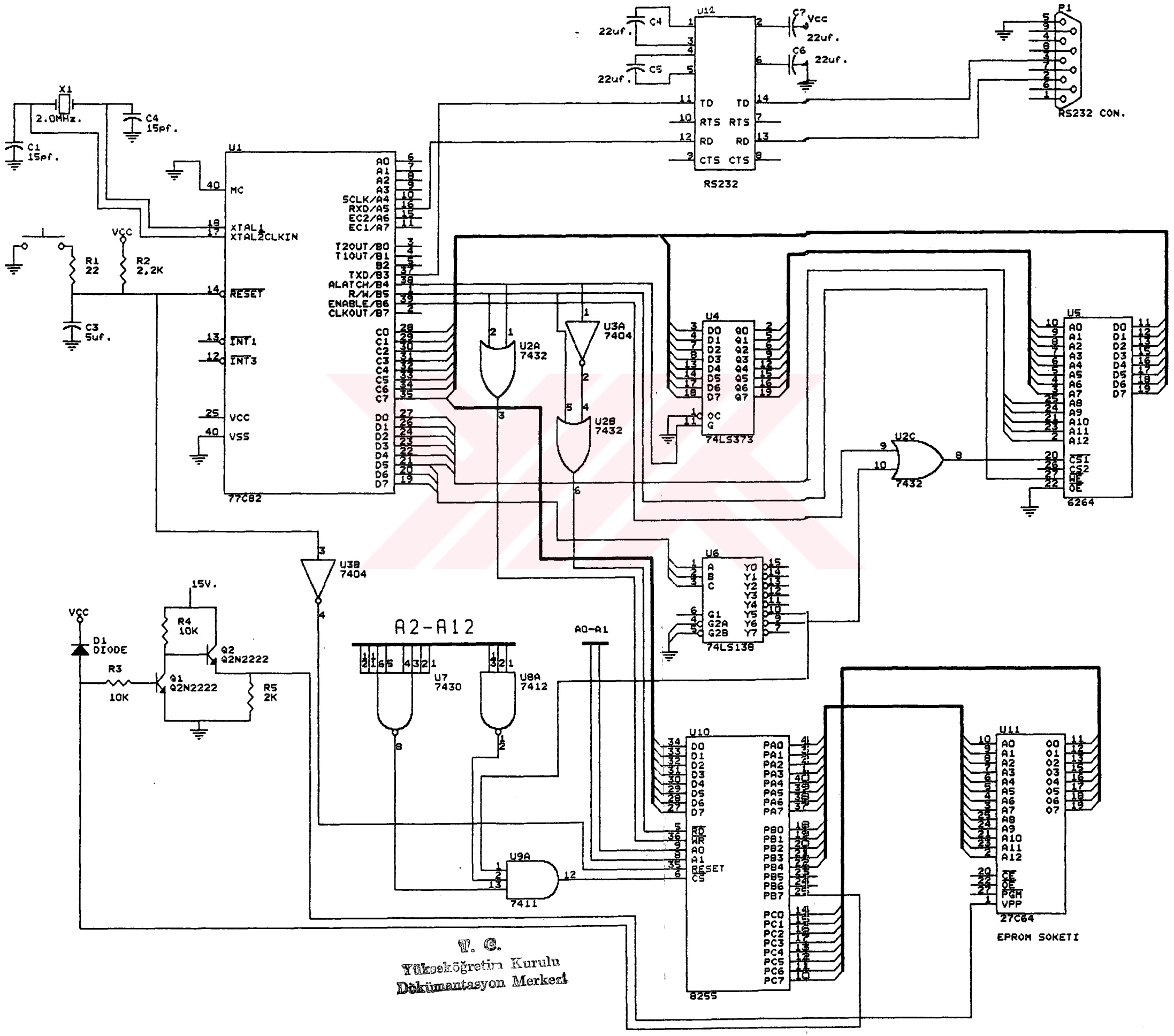
NOTE: The timing measurement reference levels for inputs and outputs are 0.8 V and 2 V.

* 27C64 UYUMLU

* YAZILIMI KORUMA ÖZELLİĞİ (R-Bit)



Ek E TMS77C82 Emülatör Kartı Devre Şeması



T. C.
Yükseköğretim Kurulu
Dokümantasyon Merkezi

ÖZGEÇMİŞ

Ahmet ÖZMEN, 1966 yılında Adapazarı'nda doğmuştur. 1983 yılında İzmit Teknik Lise'sinden mezun olarak aynı yıl İ.T.Ü. Elektrik-Elektronik Fakültesi'nin Elektronik ve Haberleşme Bölümüne girmiştir. 1987 yılında Elektronik Mühendisi ünvanını alarak aynı Fakülte'de Yüksek Lisans öğrenimine başlamıştır.

T. C.
Yükseköğretim Kurulu
Dokümantasyon Merkezi