

R 368
42

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEK YONGA MİKROBİLGİSAYAR UYGULAMASI

YÜKSEK LİSANS TEZİ

TUNCAY UZUN

ELEKTRONİK VE HABERLEŞME MÜH.

İSTANBUL 1987

YILDIZ ÜNİVERSİTESİ
GENEL KİTAPLIĞI
R 368

Kot : 42
Alındığı Yer : Fen Bilimleri Enst.
Tarih : 13/05/1991
Fatura :
Fiatı : 8000 TL
Ayniyat No : 1/3
Kayıt No : 47637
UDC : 001.64
Ek :

x

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Konu: Mikrobilgisayar ve tez çalışmaları. Süre: 1987.
Yazar: Tuncay Uzun ve Habibe Hanım. Danışman: Doç. Dr.
Yöneten: Sayın Doç. Dr. Saadettin Alkan'a teşekkür ederim.
Tarih: 1987.

TEK YONGA MİKROBİLGİSAYAR UYGULAMASI

YÜKSEK LİSANS TEZİ

TUNCAY UZUN

ELEKTRONİK VE HABERLEŞME MÜH.

İSTANBUL 1987

Mühendislik eğitimim ve tez çalışmalarım süresince değerli yardım ve katkılarını esirgemeyen hocam ve yürütücüm Sayın Doç.Dr. Sezgin ALSAN'a teşekkürü bir borç bilirim.

İÇİNDEKİLER

ÖZET	VII
SUMMARY	VIII
GİRİŞ	1
BİRİNCİ BÖLÜM	
HD63P01M1 CMOS TEK YONGA MIKROBİLGİSAYAR	
1.1 GİRİŞ	4
1.2 HD63P01M1'in Teknik Özellikleri	5
1.3 Bağlantı Uçlarının İşlevleri	7
1.3.1 Vcc, Vss	7
1.3.2 XTAL, EXTAL	7
1.3.3 $\overline{\text{STBY}}$ (Standby)	8
1.3.4 $\overline{\text{RES}}$ (Reset)	9
1.3.5 E (Enable)	11
1.3.6 $\overline{\text{NMI}}$ (Örtülemez Kesilme)	11
1.3.7 $\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$ (Kesilme isteği)	12
1.4 Tek Yonga Çalışmada Önemli Bağlantı Uçlarının İşlevleri	14
1.4.1 $\overline{\text{IS3}}$, SC1 (Giriş Belirteci)	14
1.4.2 $\overline{\text{OS3}}$, SC2 (Çıkış Belirteci)	14
1.5 Genişletilmiş Çalışmalarda Önemli Bağlantı Uçlarının İşlevleri	14
1.5.1 $\overline{\text{R/W}}$, SC2 (Oku/Yaz)	14
1.5.2 $\overline{\text{IOS}}$, SC1 (I/O Belirteci)	14
1.5.3 AS, SC1 (Adres Belirteci)	15

1.6	Fiziksel Bağlantı Birimleri	15
1.6.1	I/O Port 1	16
1.6.2	I/O Port 2	17
1.6.3	I/O Port 3	17
1.6.4	I/O Port 4	18
1.7	Tek Yonga Çalışma Şeklinde Port 3	20
1.7.1	I/O Port 3 Denetim/Durum Yazmacı	20
1.8	Çalışma Şekilleri	22
1.9	Bellek Haritaları	22
1.10	Çalışma Şeklinin Seçimi	22
1.10.1	Tek Yonga Çalışma Şekli (Mod 7)	33
1.10.2	Genişletilmiş Çokdüzeyli Çalışma Şekilleri (Mod 0,2,4,6)	33
1.10.3	Genişletilmiş Tekdüzeyli Çalışma Şekilleri (Mod 1,5)	35
1.11	Adres Yolunun Alt Baytını Tutmak	38
1.12	Programlanabilir Zamanlayıcı	38
1.12.1	Serbest Çalışan Sayıcı (FRC)	40
1.12.2	Çıkış Karşılaştırma Yazmacı (OCR) ..	41
1.12.3	Giriş Yakalama Yazmacı (ICR)	42
1.12.4	Zamanlayıcı Denetim/Durum Yazmacı ..	42
1.13	Seri Veri İletişim Arabirimi (SCI)	45
1.13.1	Uyanma Özelliği	45
1.13.2	Yazılım Özellikleri	46
1.13.3	Seri Veri İletişim Donanımı	46
1.13.4	Gönderme/Alma Denetim Durum Yazmacı (TRCSR)	47

1.13.5	Transfer Hızı/Modu Kontrol Yazmacı (RMCR)	50
1.13.6	İçeride Üretilen Saat	52
1.13.7	Dışarıda Üretilen Saat	52
1.13.8	Seri İletişimin Çalışması	52
1.13.9	Seri Veri Gönderme İşlemi	53
1.13.10	Seri Veri Alma İşlemi	54
1.14	RAM Kontrol Yazmacı	55
1.15	Ana İşlem Biriminin (CPU) İşlevleri	56
1.16	CPU Yazmaçları	57
1.16.1	Birikeçler (A,B,D)	67
1.16.2	Dizin Yazmacı (IX)	67
1.16.3	Yığın Gösterici (SP)	67
1.16.4	Program Sayıcı (PC)	67
1.16.5	Durum Kodu Yazmacı (CCR)	67
1.17	Adresleme Modları	68
1.17.1	Birikeç (ACCX) Adresleme	69
1.17.2	Hemen Adresleme	69
1.17.3	Doğrudan Adresleme	69
1.17.4	Genişletilmiş Doğrudan Adresleme	69
1.17.5	Dizinlenmiş Adresleme	69
1.17.6	İçerik Yoluyla Adresleme	70
1.17.7	Bağıl Adresleme	70
1.18	Yeni Komutlar	70
1.19	Ana İşlem Biriminin (CPU) Çalışması	71
1.20	Düşük Güçle Çalışma Modları	72
1.20.1	Sleep Modu	72
1.20.2	Standby Modu	75

1.21	Hata İşleme	78
1.21.1	Çalışma Kodu Hatası	78
1.21.2	Adres Hatası	78
1.22	HD6301V1'in HD63P01M1 ile Yapılan Geliştirmeye Uyarlanması	79
1.23	Yonga Üzerinde EPROM Kullanılırken Dikkat Edilmesi Gereken Durumlar	80
1.24	Osilasyon Devresi Baskılı Devre Üzerine Kurulurken Dikkat Edilmesi Gereken Durumlar ..	81
1.25	SCI'nin Alıcı Sınırları	82
İKİNCİ BÖLÜM		
HITACHI 6301 SERİSİ İÇİN YAPILMIŞ ÇEVİRİCİ		
2.1	GİRİŞ	85
2.2	HD63P01M1'in Çevirici Dili	86
2.2.1	Birikeç Adresleme	86
2.2.2	İçerik Yoluyla Adresleme	87
2.2.3	Hemen Adresleme	87
2.2.4	Doğrudan Adresleme	88
2.2.5	Dizinlenmiş Adresleme	92
2.2.6	Bağıl Adresleme	94
2.3	Çeviricinin Kullanımı	96
2.4	Kaynak Programın Biçimi	96
2.5	İfade Biçimi	96
2.5.1	Etiket Alanı	97
2.5.2	İşleyici Alanı	98
2.5.3	İşlem Alanı	98
2.5.4	Açıklama Alanı	98

2.6	Biçim Kontrolü	99
2.7	Ifadeler	99
2.8	Çevirici Çıkış Kütüğü	101
2.9	Çevirici Amaç Çıkış Kütüğü	101
2.10	Program Bölümleri	102
2.11	Çevirici Yönetim Komutları	103
2.12	6301 Geçişli Çevirici	107
2.13	İşlenenin Sözdizimi	107
2.14	LÉT Atama Deyimi	107
2.14.1	Anahtar Kelimeler ve İşleçler	108
2.14.2	Karışık Atama İşlemleri	108
2.15	IF Deyimi	110
2.16	_ENDS Deyimi	110
2.17	ON.ERROR ve ON.NOERROR Deyimleri	110
2.18	LEAVE ve RESTART Deyimleri	111
2.19	FOR Deyimi	111
2.20	REPEAT ve UNTIL Deyimleri	112
2.21	THRU Deyimi	112
2.22	WHILE Deyimi	113
2.23	BEGIN Deyimi	113
2.24	_CALLR Deyimi	113
2.25	_CALLS Deyimi	114
2.26	_RETURN Deyimi	114
2.27	SUBROUTINE Deyimi	114
2.28	ENTRYPOINT Deyimi	115
2.29	PROBE Deyimi	115
2.30	BEGINMODULE Deyimi	115

2.31 ENDMODULE Deyimi	115
2.32 Program örnekleri	116
ÜÇÜNCÜ BÖLÜM	
TEK YONGA MİKROBİLGİSAYAR UYGULAMASI	
3.1 GİRİŞ	136
3.2 HDEBPO1M1 ile Gerçekleştirilen 8 Hane X 8 Parçalı LED Göstergeyi Dinamik Sürme Uygulaması	137
3.3 Uygulamanın Donanım Özellikleri	137
3.4 Donanımın Çalışma Şekli	139
3.5 Yazılım Özellikleri	141
3.6 Program Kısımlarının İşlevleri	141
3.7 Programın çalışma şekli	143
3.7.1 Ana Program	143
3.7.2 TASI Altprogramı	147
3.7.3 Çıkış Karşılaştırma Kesilmesi Programı	148
EK A	155
KAYNAKÇA	163
DZGEÇMİŞ	164

ÖZET

Tek Yonga Mikrobilgisayar Uygulaması

Tek yonga mikrobilgisayarlar, akıllı denetim birimlerinde yaygın olarak kullanılan özel tümleşik devrelerdir. Bu tezde, HD63P01M1 CMOS 8-bit tek yonga mikrobilgisayar kullanılmıştır. Programlar, 6301 serisi için geçişli çevirici ile yazılmıştır. Tezde geçişli çevirici de anlatılmıştır.

HD63P01M1 tek yonga mikrobilgisayar ana işlem birimi (CPU), rasgele erişimli bellek (RAM), zamanlayıcı ve seri iletişim arabirimini içinde bulundurur. Bunlara ek olarak, mikrobilgisayarın paketi üstünde soketi olan bir EPROM vardır. CMOS teknolojisiyle üretildikleri için fazla güç harcamazlar. Bundan başka, bir kaç düşük güç harcayan şekli de vardır.

SUMMARY

Single Chip Microcomputer Application

Single chip microcomputers are special integrated circuits which are used extensively in intelligent controller units. HD63P01M1 CMOS 8-bit single chip microcomputer is used in this thesis. Programs are written by cross assembler for 6301 series. Also cross assembler is explained in thesis.

HD63P01M1 single chip microcomputer includes central processing unit (CPU), random access memory (RAM), timer and serial communication interface. In addition, there is an EPROM which has his socket on package of this microcomputer. They do not use much power being produced in CMOS technology. Besides there are several low consumption mode also.

GİRİŞ

İlk bilgisayarın yapılmasından bu yana az bir zaman geçmesine rağmen bilgisayarlardan faydalanmanın avantajları çok iyi anlaşılmıştır. Bu nedenle bilgisayarlar üzerinde yapılan araştırma ve geliştirme çalışmaları büyük önem kazanmıştır.

Tümdevre yapım tekniklerinin hızlı bir şekilde gelişmesiyle büyük çapta tümleştirme (LSI) ve çok büyük çapta tümleştirme (VLSI) gerçekleştirilmiştir. Bunun sonucunda büyük yer kaplayan bilgisayarlardan sonra mini bilgisayarlar, kişisel bilgisayarlar, mikrobilgisayarlar ve cep bilgisayarları yapılmıştır. Boyutunun küçülmesinin yanında daha bir çok yeni ve etkili yeteneklere sahip olmuşlardır.

Bu gelişme elektronik mühendisliğine de yansımıştır. Bir çok tümdevreden oluşan karışık bilgisayar donanımı bir kaç tümdevreden oluşan basit donanıma dönüşmüştür. Yazılım ve donanım olanakları kişisel bilgisayarlar için yapılmış çeşitli programlarla ve donanımlarla desteklenerek geliştirilmiştir.

Tek yonga mikrobilgisayar tümdevreleri mikroişleyici ve çevre birimlerini içinde bulundurmaktadır. Bu tümdevrelerin yazılım teknikleri donanım desteğiyle güçlendirilmiş ve kullanım kolaylığı sağlamıştır.

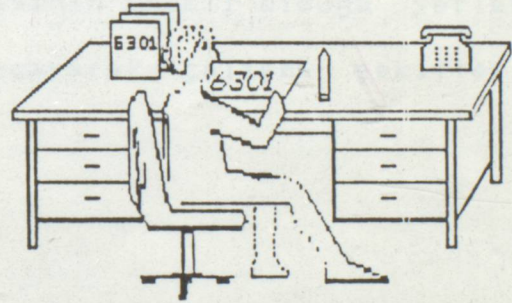
Tek yonga mikrobilgisayarların yaygın olarak kullanılmaya başlanmasıyla bu konudaki yazılım ve donanım ayrı bir önem kazanmıştır. Kişisel bilgisayarlar, mini

bilgisayarlar için yüksek seviyeli dillerden makine diline çevirici ve derleyiciler yapılmıştır. Bunlar sayesinde hızlı ve güvenilir program yazılması sağlanmıştır. Donanım ve yazılım geliştirmeleri için geliştirme setleri hazırlanmıştır. Ayrıca kişisel bilgisayarlar için donanım ve yazılım destekli tasarım programları bu tümdevreleri daha kullanışlı hale getirmiştir.

HD63P01M1 CMOS

TEK YONGA MİKROBİLGİSAYAR

BİRİNCİ BÖLÜM
HD63P01M1 CMOS
TEK YONGA MİKROBİLGİSAYAR



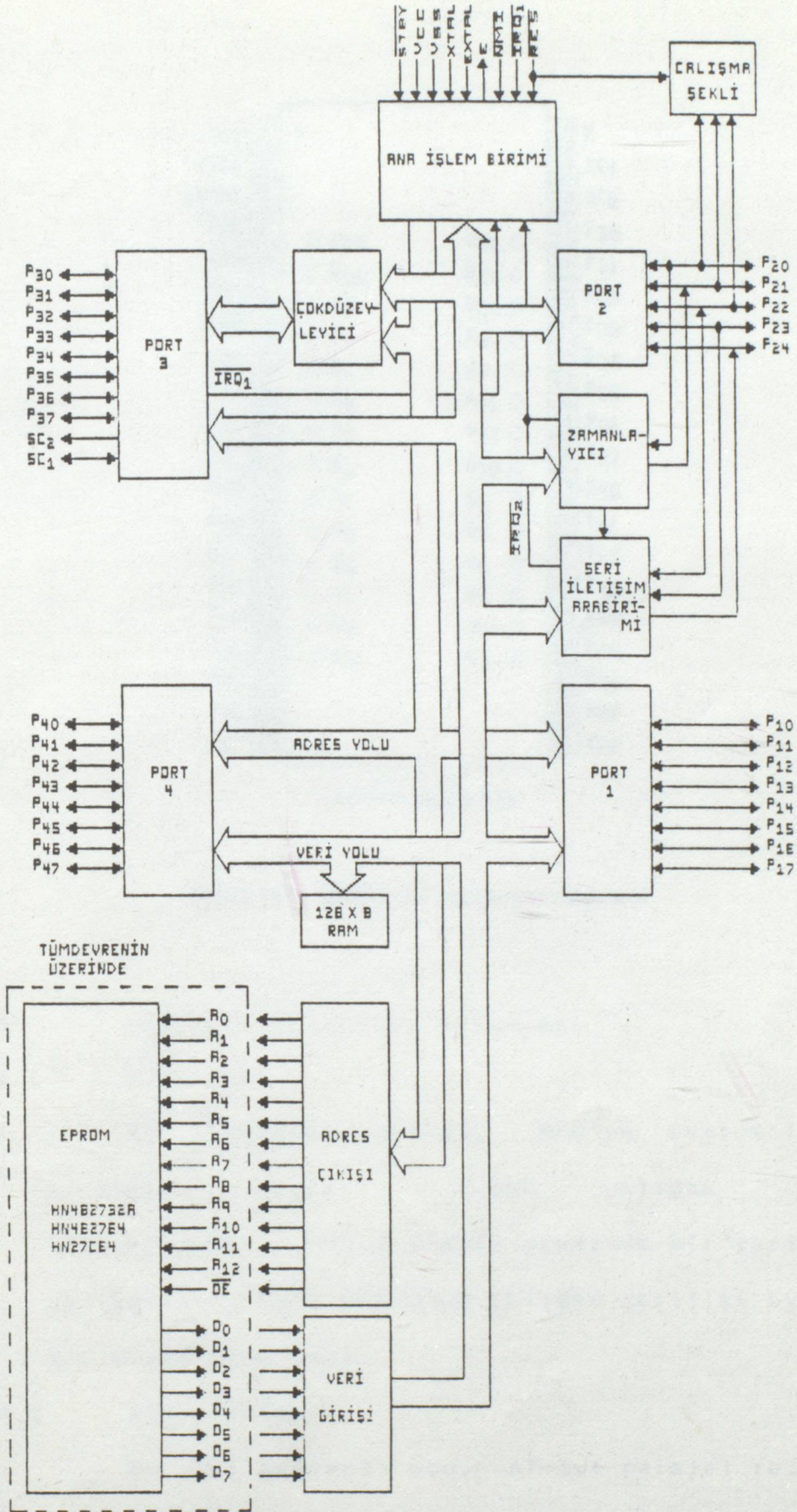
1.1 GİRİŞ

HD63P01M1, HD6301V1 ile tam uyumlu yazılıma sahip büyük çapta tümleştirilmiş (LSI) bir tek yonga mikrobilgisayar birimidir (MCU). Donanım açısından aralarındaki tek fark kullandıkları salt okunur bellek hacmi ve tipidir. HD63P01M1 4 veya 8 Kbayt'lık elektriksel olarak yazılabilen ultraviyole ile silinebilen belleği (EPROM) üzerine takılarak kullanılabilir. HD6301V1 ise 4 Kbayt'lık salt okunur belleği (ROM) içinde bulundurur. Bu nedenle yazılım ve donanım olanakları HD63P01M1 üzerinde kolaylıkla geliştirilebilir. Sonra üretim ve gizlilik aşamasında HD6301V1 kullanılabilir.

MCU içinde ana işlem birimi (CPU), 128 Bayt rasgele erişimli bellek (RAM), 16 Bit'lik bir zamanlayıcı (TIMER), seri iletişim arabirimi (SCI) ve 29 paralel giriş/çıkış bulundurur. CMOS MCU olduğu için fazla güç harcamaz ve güç kaynağının geniş gerilim sınırları içinde kullanılabilir. Ayrıca ortalama güç harcamasının önemli olduğu yerlerde etkili olan düşük güç harcayarak çalışma şekillerine sahiptir.

1.2 HD63P01'IN TEKNİK ÖZELLİKLERİ

- * HD6301V1 ile uyumlu bağlantı uçları
- * Yonga üzerindeki HD6301V1 ile uyumlu olan işlevler
 - 128 Bayt rasgele erişimli bellek
 - 29 Paralel giriş/çıkış
 - 16 Bit Programlanabilir zamanlayıcı
 - Seri iletişim arabirimi
- * Düşük güç harcayarak çalışma şekilleri
 - Sleep (uyku) çalışma şekli
 - Standby (durma) çalışma şekli
- * En düşük komut dönüş süresi
 - 1.0 us (f=1.0 MHz)
 - 0.67 us (f=1.5 MHz)
 - 0.5 us (f=2.0 MHz)
- * Bit işlemleri, Bit test komutları
- * Sistem kırılmasını önleme
 - Adres Trap (Adres hatasını yakalama)
 - Op-Code Trap (Çalışma kodu hatasını yakalama)
- * 65536 Bayt'a kadar adres boşluğu
- * 4 K veya 8 KBayt'lık EPROM kullanabilme
 - 4096 Bayt HN482732A
 - 8192 Bayt HN482764, HN27C64



Şekil 1.1 HD63P01M1 tek yonga mikrobilgisayarının blok diyagramı.

V _{SS}	1	0		40	E
XTAL	2			39	SC ₁
EXTAL	3			38	SC ₂
NMI	4	○ V _{CC}	V _{CC} ○	37	P30
IRQ1	5	○ A ₁₂	V _{CC} ○	36	P31
RES	6	○ A ₇	V _{CC} ○	35	P32
STBY	7	○ A ₆	A ₈ ○	34	P33
P20	8	○ A ₅	A ₉ ○	33	P34
P21	9	○ A ₄	A ₁₁ ○	32	P35
P22	10	○ A ₃	V _{SS} ○	31	P36
P23	11	○ A ₂	A ₁₀ ○	30	P37
P24	12	○ A ₁	CE ○	29	P40
P10	13	○ A ₀	07 ○	28	P41
P11	14	○ 0 ₀	06 ○	27	P42
P12	15	○ 0 ₁	05 ○	26	P43
P13	16	○ 0 ₂	04 ○	25	P44
P14	17	○ V _{SS}	03 ○	24	P45
P15	18			23	P46
P16	19			22	P47
P17	20			21	V _{CC}

(Üstten Görünüş)

Şekil 1.2 Bağlantı uçlarının düzeni

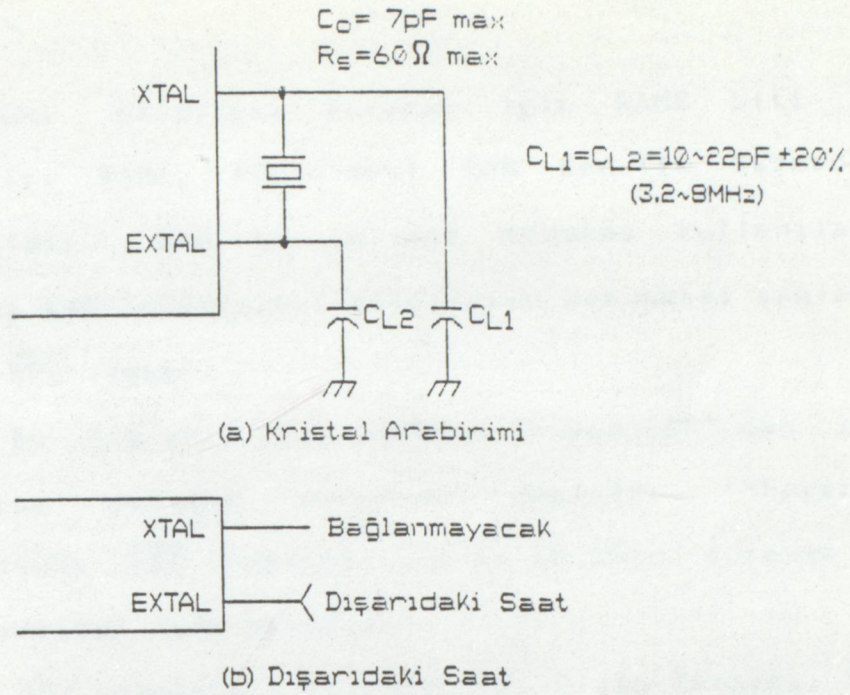
1.3 Bağlantı Uçlarının İşlevleri

1.3.1 V_{cc}, V_{ss}

Bu bağlantı uçları, MCU'ya $5V \pm 10\%$ 'luk güç kaynağını sağlar. Düşük çalışma hızında ($f_{max}=500kHz$), MCU 3 ile 6V arasında bir gerilim ile çalışabilir. Eğer EPROM'un çalışma gerilimi $5V \pm 5\%$ ise V_{cc} $5V \pm 5\%$ olmalıdır.

1.3.2 XTAL, EXTAL

Bu iki bağlantı ucu, AT-cut paralel rezonanslı kristal osilatörü oluşturur. Bu osilatörün frekansı



Şekil 1.3 Bağlantı Devresi

yonga içinde dörde bölünür. Böylece 4MHz'lik kristal kullanılması halinde sistem saat frekansı 1MHz olur. EXTAL ucu, XTAL ucu açıkken LSI'da üretilen saat frekansının dört katı ve %45-%55 darbe oranına sahip bir kare dalga ile dışarıdan sürülebilir. En büyük çalışma frekansı, sistem saat frekansının dört katını geçmemelidir. Şekil 1.3'de kristal osilatör bağlantısına bir örnek gösterilmiştir. Kristal, CL1 ve CL2 elemanları XTAL ve EXTAL bağlantı uçlarına mümkün olduğu kadar yakın bağlanmalıdır.

1.3.3 STBY (Standby)

Bu bağlantı ucu, "MCU standby" çalışma modunu meydana getirir. "0" olduğu zaman osilatör durur ve MCU'nun reset durumunu dengeler. Standby'da RAM'ın

içindeki bilgileri korumak için RAME biti "0" yapılır. RAME, 40014'deki RAM Denetim Yazmacının 6.bitidir. RAM bu çalışma modunda kullanılamaz. Sadece RAM'in içindeki bilgilerin korunması sağlanır.

1.3.4 $\overline{\text{RES}}$ (Reset)

Bu bağlantı ucu, MCU'yu "Power-OFF" dan reset yaparak çalışma durumuna geçirir. "Power-ON" sırasında $\overline{\text{RES}}$ bağlantı ucu en az 20ms süreyle "0" seviyesinde tutulmalıdır.

CPU yazmaçları (Birikeçler, Dizin Yazmacı, Yığın Göstercisi, kesilmeyi örtme biti dışında Durum Kodu Yazmacı), RAM Denetim Yazmacı ve Port'ların (fiziksel bağlantı birimleri) veri yazmaçları reset işlemi sırasında set edilmemişlerdir. Bu nedenle içindeki bilgiler belirlenmemiştir.

Çalışma sırasında MCU'yu resetlemek için $\overline{\text{RES}}$ en az 3 sistem saati dönüşü süresince "0"da kalmalıdır. Resetleme işleminde, üçüncü dönüşte bütün adres yolları yüksek direnç gösterir. $\overline{\text{RES}}$ "0"da kaldığı sürece adres yolları bu durumunu korur. $\overline{\text{RES}}$ ucu "1" olduğunda, CPU yeniden çalışmasına başlar.

Reset işlemi sırasında bağlantı uçlarının durumu çizelge 1'de gösterilmiştir.

Resetleme işlemi sırayla aşağıdaki şekildedir.

(1) Port 2'deki P20,P21,P22 bağlantı uçlarından alınan bilgiler Program Denetim Yazmacının PC0,PC1,PC2 bitlerine aktarılır.

Çizelge 1 Reset sırasında bağlantı uçlarının durumu

Mod Bağlantı ucu	0	1	2,4	5	6	7
Port 1 P10~P17	yüksek direnç (giris)	←	←	←	←	←
Port 2 P20~P24	yüksek direnç (giris)	←	←	←	←	←
Port 3 P30~P37	\bar{E} : "1" Çıkış E : "1" Çıkış * (yüksek direnç)	yüksek direnç	\bar{E} : "1" Çıkış E : "1" Çıkış * (yüksek direnç)	yüksek direnç	\bar{E} : "1" Çıkış E : "1" Çıkış * (yüksek direnç)	yüksek direnç (giris)
Port 4 P40~P47	yüksek direnç (giris)	←	←	←	←	←
SC2	"1" Çıkış (Okuma)	←	←	←	←	"1" Çıkış
SC1	\bar{E} : "1" Çıkış E : yüksek direnç	←	←	"1" Çıkış	\bar{E} : "1" Çıkış E : yüksek direnç	yüksek direnç (giris)
A0~A12, CE	"1" Çıkış	←	←	←	←	←

* Not : Mod 0,2,4,6'da Port 3, E="1" olduğu sırada "1" çıkış olarak set edilir. Bu dışarıdaki bir bellek çıkışı ile çakışmaya neden olur. Bunun için aşağıda belirtilenlere dikkat edilmelidir.

- (1) Reset sırasında dışarıdaki belleği devre dışı bırakacak bir düzen kurun.
- (2) E="1" olduğu sırada SC1 bağlantı ucunu "0" yapan $4.7k\Omega$ aşağıya çekme direnci ekleyin.

Bu Port 3'u yüksek dirençli yapar.

(2) $\$FFFE$ ile $\$FFFF$ adreslerinin içindeki bilgiler başlangıç adresi olarak Program Sayacına yüklenir ve bu adresten itibaren yazılan program yürütülür.

(3) Kesilmeyi örtme biti set edilir. $\overline{IRQ1}$ ve $\overline{IRQ2}$ örtülebilir kesilmelerine izin verilmesi için bu bit "0" yapılmalıdır.

MCU, en çok 20ms için "Power-ON" yapıldıktan sonra saat normal osilasyona başlayıncaya kadar reset girişi kabul edemez. Geçiş zamanı sırasında MCU ile fiziksel bağlantı birimlerinin durumu belirsizdir. Sistemin yapılı düzeninde buna dikkat edilmelidir.

1.3.5 E (Enable)

Bu çıkış ucu, dışarıya sistem saati sağlar. Çıkış tek fazlı, TTL uyumludur ve frekansı kristal osilatörün frekansının 1/4'dür. Bu uç iki LS TTL yük ve 40pF kapasitansı sürebilir.

1.3.6 \overline{NMI} (Örtülemez Kesilme)

Giriş sinyalinin düşen kenarı bu bağlantı ucunda dedekte edilirken CPU, içeride örtülemez kesilme (\overline{NMI}) sırasını başlatır. $\overline{IRQ1}$ ve $\overline{IRQ2}$, mümkün olduğu kadar düşen kenarda dedeksiyonunu tamamlayan \overline{NMI} 'dan daha az önceliklidir. Durum Kodu Yazmacının (CCR) kesilmeyi örtme biti (1 biti) \overline{NMI} 'ya etki yapamaz.

Bir örtülemez kesilmeye cevapta Program Sayıcısının (PC), Dizin Yazmacının (IX), Birleşlerin (A,B) ile Durum Kodu Yazmacının (CCR) içindeki

bilgiler yığına atılarak saklanır. Sonra $\$FFFC$ ve $\$FFFD$ adreslerinin içerikleri Program Sayıcısına aktarılır ve CPU örtülemez kesilme programını yürütmeye başlar. RTI komutu ile örtülemez kesilmeden dönülür ve yığındaki bilgiler geri alınarak Program Sayıcısının kaldığı yerden programa devam edilir.

Reset sırasında düşen kenarın \overline{NMI} bağlantı ucuna giriş olarak vermeden önce Yığın Gösterici uygun bir bellek adresine set edilmelidir.

1.3.7 $\overline{IRQ1}$, $\overline{IRQ2}$ (Kesilme İsteği)

Bunlar Ana İşlem Birimine (CPU) iç kesilme önceliği soran seviye duyarlı bağlantı uçlarıdır. Kesilme isteğinde (IRQ), CPU bilgi istemeden önce ana işlemini tamamlayacaktır. Durum Kodu Yazmacının kesilmeyi örtme biti "1" olmadığı sürece CPU kesilmeyi kabul eder. Aksi halde kesilme dikkate alınmaz. İşleme başlamadan önce Program Sayıcısı (PC), Dizin Yazmacı (IX), Birikeçler (A,B) ve Durum Kodu Yazmacı (CCR) yığına atılarak saklanır. Kesilmeyi örtme biti set edilerek daha sonra gelebilecek bir örtülebilir kesilme engellenir. Son dönüş sırasında CPU $\overline{IRQ1}$ kesilmesi için $\$FFF8$ ve $\$FFF9$ adreslerinin içeriklerini Program Sayıcısına transfer ederek kesilme işlem programını yürütür. $\overline{IRQ2}$ 'de $\overline{IRQ1}$ gibi işlem görür fakat bu kesilmede $\$FFF0$ dan $\$FFF7$ 'ye kadar olan adreslerden uygun olan ikisi dikkate alınır. $\overline{IRQ1}$ donanıma bağlıdır. $\overline{IRQ2}$

ise zamanlayıcı ile ilgili olarak ICI,OCI,TOI ve seri iletişim arabirimine bağlıdır.

$\overline{IRQ1}$ ve $\overline{IRQ2}$ aynı anda üretilirse önceki sonrakinin önüne geçer. Durum Kodu Yazmacının kesilmeyi örtme biti set edilmiş ise her iki kesilmede dikkate alınmaz.

$\overline{IRQ1}$ korunmadığı ve aynı anda işlem gördüğü için örtüldükten sonra geçersizdir.

Adres veya çalışma kodu (Op-Code) hatasının meydana gelmesi halinde TRAP kesilmesi üretilir. Bu kesilme \overline{RES} 'e göre ikinci önceliğe sahiptir. Kesilmeyi örtme bitinin durumunun sistemden çıkartılması halinde CPU bir kesilme sırasını yürütmeye başlayacaktır. Bu kesilme için vektör \$FFEE ve \$FFEF adreslerinin içerikleri olacaktır.

Kesilmeler ile ilgili adresler Çizelge 2'de gösterilmiştir.

Çizelge 2 Kesilme Vektörlerinin Bellek Haritası

Yüksek Öncelik	Vektör		Kesilme
	MSB	LSB	
	FFFE	FFFF	\overline{RES}
	FFEE	FFEF	TRAP
	FFFC	FFFD	\overline{NMI}
	FFFA	FFFB	SWI (Yazılım ile kesilme)
	FFF8	FFF9	$\overline{IRQ1}$ (veya $\overline{IS3}$)
	FFF6	FFF7	ICF (Zamanlayıcı Giriş Yakalayıcı)
	FFF4	FFF5	OCF (Zamanlayıcı Çıkış Karşılaştırma)
	FFF2	FFF3	TOF (Zamanlayıcı Taftı)
Düşük Öncelik	FFF0	FFF1	SCI (RDRF+ORFE+TDRE)

1.4 Tek Yonga Çalışmada Önemli Bağlantı Uçlarının İşlevleri

1.4.1 $\overline{IS3}$, SC1 (Giriş Belirteci)

Bu işaret $\overline{IS3}$ kesilmesi ve Port 3'ün bilgi akışını denetler. Bu işaretin düşen kenarında Port 3 Denetim Durum Yazmacının ilgili bayrağı set edilir.

Port 3 Denetim Durum Yazmacı ile ilgili daha ayrıntılı bilgi ileride anlatılacaktır.

1.4.2 $\overline{OS3}$, SC2 (Çıkış Belirteci)

Bu işaret Port 3 giriş/çıkış bağlantı uçları üzerinde bilginin hazır olduğunu bilgiyi alacak cihaza bildirir. Bu bağlantı ucu ile ilgili zamanlama diyagramı EK A'daki Şekil-5'de gösterilmiştir.

1.5 Genişletilmiş Çalışmalarda Önemli Bağlantı Uçlarının İşlevleri

1.5.1 R/\overline{W} , SC2 (Oku/Yaz)

Bu TTL uyumlu çıkış işareti, çevre birimlerine ve belleklere Ana İşlem Biriminin okumama ("1") yoksa yazmama ("0") yaptığını bildirir. Normal standby durumunda okuma ("1") konumundadır. Bu çıkış bir TTL yük ve 90pF kapasitansı sürebilir.

1.5.2 \overline{IOS} , SC1 (I/O Belirteci)

Genişletilmiş tekdüzeyle çalışma Mod 5'de, A9'dan A15'e kadar "0" ve A8 "1" olduğu zaman \overline{IOS} "0" olur. Böylece dışarıda #0100 ile #01FF arasında bulunan 256 Bayt'lık bir belleğe erişilebilir. Bu bağlantı ucu ile ilgili zamanlama diyagramı EK A'daki

Şekil-2'de gösterilmiştir.

1.5.3 AS, SC1 (Adres Belirteci)

Genişletilmiş çokdüzeyleyilmiş çalışma şeklinde Adres Belirteci (AS) işareti bu bağlantı ucunda görünür. Port 3'de veri ile çokdüzeyleyilmiş alt 8-Bit adresi anahtarlama için kullanılır. Bu anahtarlama işlemi EK A'daki Şekil-9'da görülmektedir. Bu yüzden Port 3 E darbesi sırasında veri yolu olabilir. Bu işaretin zamanlama diyagramı EK A'daki Şekil-1'de gösterilmiştir.

AS, içerideki adresleme bölgesine geçişte dışarıya gönderilir.

1.6 Fiziksel Bağlantı Birimleri

HD63P01M1 CMOS tek yonga mikrobilgisayarında dört giriş/çıkış Port'u vardır. Bunlardan birisi 5 Bit'lik, diğer üçü ise 8 Bit'lidir. 8 Bit'lik bir Port'a 2 kontrol ucu bağlanmıştır. Her Port'un giriş/çıkış bağlantı uçlarının giriş veya çıkış olmasını sağlayan, birbirinden bağımsız, salt yazılabilen Veri Yönü Yazmacı vardır. Ayrıcalıklı bir durum olarak Port 2'nin 2. biti veri girişi veya zamanlayıcı çıkışı olabilir. Asla veri çıkışı olarak kullanılamaz.

Port'daki bir bağlantı ucunun çıkış olabilmesi için, Port'un Veri Yönü Yazmacının ilgili bitinin "1" yapılması gerekir. Giriş olarak programlanması için ise "0" yapılması gerekir.

Bu dört Port, Port 1, Port 2, Port 3 ve Port 4

şeklinde adlandırılır. Her Port ve birleşik Veri Yönü Yazmacının adresi Çizelge 3'de gösterilmiştir.

Çizelge 3 Port ve Veri Yönü Yazmaçlarının Adresleri

Portlar	Port Adresi	Veri Yönü
		Yazmaç Adresi
I/O Port 1	\$0002	\$0000
I/O Port 2	\$0003	\$0001
I/O Port 3	\$0006	\$0004
I/O Port 4	\$0007	\$0005

\overline{RES} Port'ların Veri Yazmaçlarına etki etmez. \overline{RES} 'den hemen sonra Veri Yazmaçlarının içerikleri belirsizdir. Veri Yönü Yazmaçları ise sıfırlanır.

1.6.1 I/O Port 1

8 Bit'lik bir I/O Port'udur. Her bağlantı ucu Port 1 Veri Yönü Yazmacının ilgili biti ile giriş veya çıkış olarak programlanabilir. Çıkış bağlantı uçları, giriş olarak kullanıldıklarında yüksek direnç gösterme özelliğine sahiptir. TTL (Tranzistor Tranzistor Lojik) uyumludurlar. Verinin doğru olarak okunabilmesi için bağlantı uçlarındaki, gerilimin "1" seviyesi için 2.0 V'dan büyük, "0" seviyesi için ise 0.8 V'dan küçük olması gerekir.

MCU resetlendikten sonra Mod 1 dışındaki çalışma şekillerinde bağlantı uçları giriş olarak şartlandırılır. Genişletilmiş tekdüzeyle çalışma şekli Mod 1'de Port 1 adres yolunun alt baytı (A0-A7) olacağından çıkış olarak şartlandırılır.

1.6.2 I/O Port 2

Port 2 Veri yönü Yazmacı ile şartlandırılabilen 5 giriş/çıkış bağlantı ucuna sahiptir. Çıkış bağlantı uçları giriş olarak kullanıldıklarında yüksek direnç gösterir. TTL uyumludur. Güvenli çalışma için veri gerilim seviyeleri "1" için 2.0 V'dan büyük ve "0" için 0.8 V'dan küçük olması gerekir.

MCU resetlendikten sonra bağlantı uçları giriş olarak şartlandırılır. Port 2'deki P20(8), P21(9) ve P22(10) bağlantı uçları, reset sırasında çalışma şeklini belirlemek amacı ile kullanılır. Bu üç bitin değeri, çalışma şeklinin seçimi bölümünde anlatılacak olan Port 2 Veri Yazmacının üst 3 bitine (bit7,6,5) aktarılır.

Bütün çalışma şekillerinde Port 2 giriş/çıkış Port'u olarak kullanılabilir. Bu Port aynı zamanda seri veri iletişim arabiriminin ve zamanlayıcının dış dünya ile bağlantısını sağlar. Bununla birlikte P21'in sadece veri girişi veya zamanlayıcı çıkışı olarak kullanılabilirdiği unutulmamalıdır.

1.6.3 I/O Port 3

Bağımsız giriş/çıkış bağlantı uçları, Veri grubu veya adres grubu ile çokdüzeylemiş veri grubu olarak kullanılabilen 8 bitlik bir Port'dur. Reset sırasında Port 2 aracılığı ile belirlenen çalışma şekline bağlı olarak şartlandırılır. Bu Port iki yönlü çalışabilir. Güvenli çalışma için gerilim seviyeleri, "1" için

2.0 V'dan büyük ve "0" için ise 0.8 V'dan küçük olması gerekir. TTL uyumlu 3-durumlu kuvvetlendirici bir TTL yük ve 90 pF kapasitansı sürebilir. Genişletilmiş çalışma şekillerinde resetden hemen sonra Veri Yönü Yazmacının kontrolü engellenerek R/\bar{W} 'in konumuna bağlı olur. Port 3'ün çalışması Bölüm 1.7'de ayrıntılı biçimde anlatılmıştır.

1.6.4 I/O Port 4

Seçilen çalışma şekline bağlı olarak 8 Bit paralel giriş/çıkış veya adres yolu olabilir. Güvenli veri iletim gerilim seviyeleri, "1" için 2.0 V'dan büyük ve "0" için 0.8 V'dan küçüktür. Her uç TTL uyumludur ve bir TTL yükü ve 90 pF kapasitansı sürebilir. Çeşitli çalışma şekillerinde Port 4'ün işlevi aşağıda anlatılmıştır.

Tek yonga çalışma şeklinde (Mod 7) Port4 kendi Veri Yönü Yazmacı ile set edilebilen bir 8 Bit paralel giriş/çıkış olur.

Genişletilmiş tekdüzeyle çalışma şeklinde (Mod 5) Port 4 kendi Veri Yönü Yazmacına "1" ler yazılarak adres yolunun alt baytını (A0-A7) oluşturur. Reset işleminden sonra giriş olarak şartlandırılır. Bu port'dan kullanım amacına göre paralel giriş veya çıkış yapılabilir.

Diğer bir genişletilmiş tekdüzeyle çalışma şeklinde (Mod 1) Port 4, Veri Yönü Yazmacı dikkate alınmaksızın adres yolunun üst baytını (A8-A15)

oluşturmak üzere çıkış olarak şartlandırılır.

Genişletilmiş çokdüzeyli çalışma şeklinde (Mod 6) Port 4, adres yolunun üst baytını (A8-A15) oluşturmak üzere çıkış olarak şartlandırılır. Reset işlemi sonunda giriş olarak set edilir. Port 4'den kullanım amacına göre giriş/çıkış yapmak mümkündür.

Diğer genişletilmiş çokdüzeyli çalışma şekillerinde (Mod 0,2,4) Port 4, Veri Yönü Yazmacı dikkate alınmadan, adres yolunun üst baytını (A8-A15) oluşturmak üzere çıkış olarak şartlandırılır.

Her çalışma şekli ile Port'lar arasındaki ilişki Çizelge 4'de gösterilmiştir.

Çizelge 4 Çalışma şekillerinde port ve denetim hatlarının işlevleri

Çalışma Şekli	PORT 1 Sekiz Hat	PORT 2 Beş Hat	PORT 3 Sekiz Hat	PORT 4 Sekiz Hat	SC ₁	SC ₂	
Tek Yonga (Mod 7)	I/O	I/O	I/O	I/O	$\overline{IS3}$ (I)	$\overline{OS3}$ (O)	
Genişletilmiş Çokdüzeyli (Mod 0,2,4,6)	I/O	I/O	Adres Yolu (A0~A7) Veri Yolu (D0~D7)	Adres Yolu* (A0~A15)	AS (O)	R/ \overline{W} (O)	
Genişletilmiş Tekdüzeyli	(Mod 5)	I/O	I/O	Veri Yolu (D0~D7)	Adres Yolu* (A0~A7)	\overline{IOS} (O)	R/ \overline{W} (O)
	(Mod 1)	Adres Yolu (A0~A7)	I/O	Veri Yolu (D0~D7)	Adres Yolu (A0~A15)	KM	R/ \overline{W} (O)

* Bu hatlar Mod 0,2,4 dışındaki çalışma şekillerinde salt giriş hattı olarak kullanılabilir.

I - Giriş
O - Çıkış
R/W - Oku/Yaz

IS3 - Giriş Belirteci
OS3 - Çıkış Belirteci
IOS - I/O Seçici

SC - Belirteç Denetim
AS - Adres Belirteci
KM - Kullanılmamış

1.7 Tek yonga çalışma şeklinde Port 3

Paralel giriş/çıkış bağlantı uçları Veri Yönü Yazmacına uygun olarak şartlandırılır.

Bu çalışma şeklinde ayrıca $\overline{IS3}$ (Input Strobe) ve $\overline{OS3}$ (Output Strobe) olarak adlandırılan iki kontrol ucu bulunur. Bunlar çevre birimleri ile anlaşmak amacıyla kullanılır ve I/O Port 3 Denetim/Durum Yazmacı tarafından kontrol edilir. Port 3'ün bu iki kontrol ucu aşağıdaki şekilde çalışır.

(1) Port 3'e girilen veri $\overline{IS3}$ (SC1) ile tutularak kullanılabilir.

(2) $\overline{OS3}$, Port 3'ün Veri Yazmacına okuma veya yazma yapıldığında CPU tarafından üretilir.

(3) $\overline{IRQ1}$ kesilmesi, $\overline{IS3}$ 'ün düşen kenarı ile üretilebilir.

Port 3 ile ilgili zamanlama diyagramları EK A'daki Şekil-5 ve Şekil-6'da gösterilmiştir.

1.7.1 I/O Port 3 Denetim/Durum Yazmacı

	7	6	5	4	3	2	1	0
		$\overline{IS3}$	X	$\overline{OS3}$		X	X	X
	$\overline{IS3}$	$\overline{IRQ1}$			TUTMA			
\$000F	BAYRAK	İZİNİ			İZİNİ			

Bit 0 Kullanılmaz

Bit 1 Kullanılmaz

Bit 2 Kullanılmaz

Bit 3 Tutma izni (Latch Enable)

Bu bit Port 3'e girilen veriyi tutmak için kullanılır. Eger bu bit "1" ise Port 3'e girilen veri $\overline{IS3}$ 'ün düşen kenarında tutulur. Tutulma MCU tarafından bırakılabilir ve yeni veri $\overline{IS3}$ 'ün düşen kenarında tekrar tutulabilir. Resetden sonra "0" olur. Bu durumda $\overline{IS3}$, Port 3 üzerinde etkili olamaz.

Bit 4 \overline{OSS} (Output Strobe Select)

Port 3'den okumayı yoksa yazmayı yapıldığını belirler. Eger Port 3'den okuma yapılıyorsa bu bit "0" olur. Yazma işleminde ise "1" olur. Resetden sonra "0" olur.

Bit 5 Kullanılmaz

Bit 6 $\overline{IS3}$ $\overline{IRQ1}$ izni

Eger bu bit "1" yapılırsa, $\overline{IRQ1}$ kesilmesine $\overline{IS3}$ bayrağı ile izin verilir. Aksi halde etkili olamaz. Resetden sonra "0" olur.

Bit 7 $\overline{IS3}$ Bayrağı

$\overline{IS3}$ 'ün (SC1) düşen kenarında set edilen ve salt okunabilen bir kontrol bitidir. Port 3 Denetim/Durum Yazmacının okunmasından sonra Port 3'e okuma/yazma yapılmasıyla "0" olur. Resetden sonra "0" olur.

Genişletilmiş tekdüzeyle çalışma şeklinde (Mod 1,5) Port 3 veri yolu (D0-D7) olur.

Genişletilmiş çokdüzeyli çalışma şeklinde (Mod 0,2,4,6) Port 3 veri yolu (D0-D7), aynı zamanda adres

yolunun alt baytı (A0-A7) olur. Bu iki seçeneği belirleyen kontrol ucu AS çıkış bağlantı ucudur. Bu bağlantı ucu "1" ise Port 3 adres yoludur. "0" ise veri yoludur.

1.8 Çalışma Şekilleri

- Mod 0 : Çokdüzeylemiş test modu
 Mod 1 : Tekdüzeyleli/parçalı çözümlenmiş mod
 Mod 2,4 : Çokdüzeylemiş/RAM modu
 Mod 5 : Tekdüzeyleli/parçalı çözümlenmiş mod
 Mod 6 : Çokdüzeylemiş/parçalı çözümlenmiş mod
 Mod 7 : Tek yonga çalışma modu

1.9 Bellek Haritaları

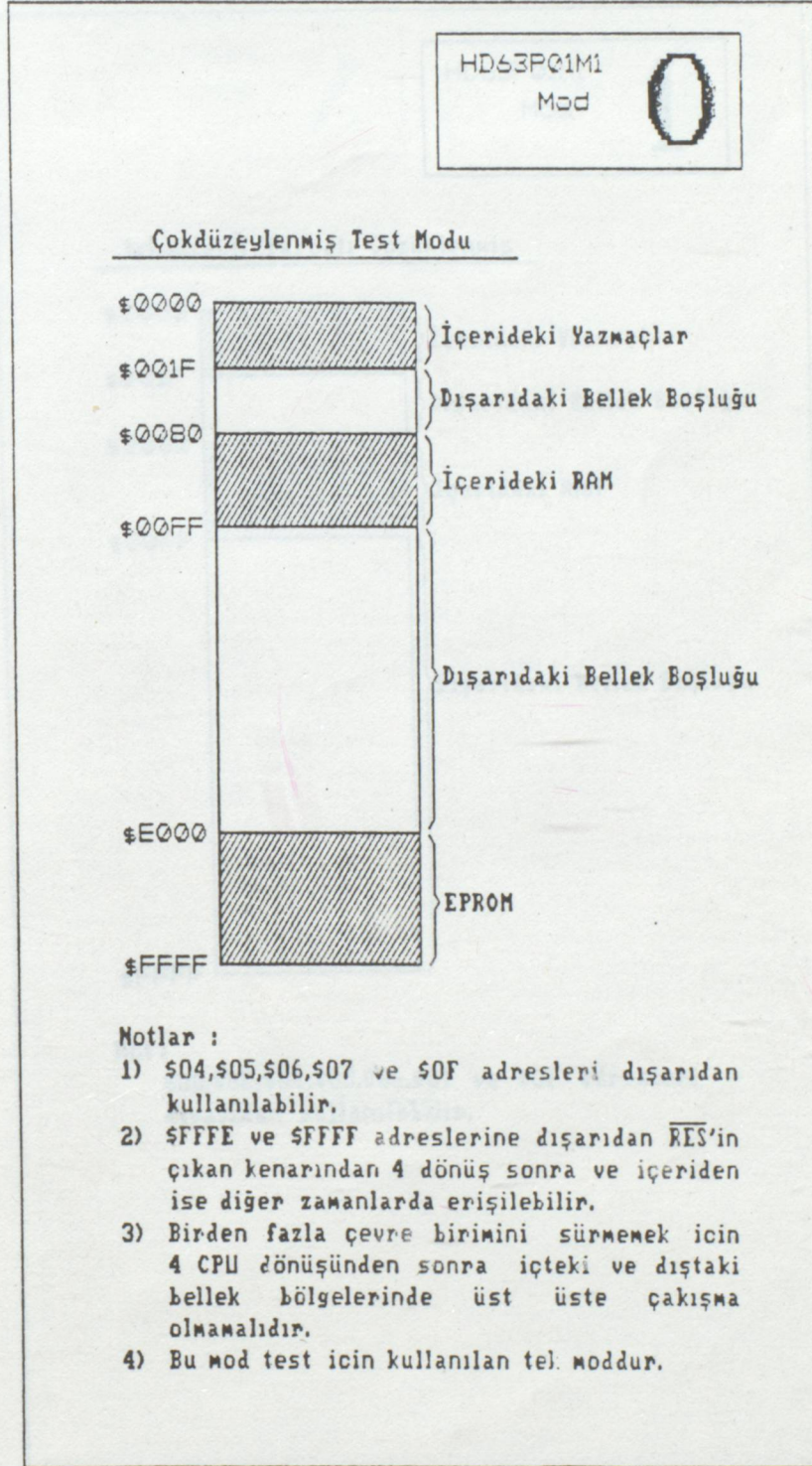
HD63P01M1, çalışma moduna bağlı olarak 64 KBayt'a kadar bellek adresleyebilir. Şekil 1.4'de her çalışma moduna ait bellek haritaları verilmiştir. İçteki yazmaçlar Çizelge 5'de #00'dan #1F'e kadar olan adresleri kullanır.

1.10 Çalışma Şeklinin Seçimi

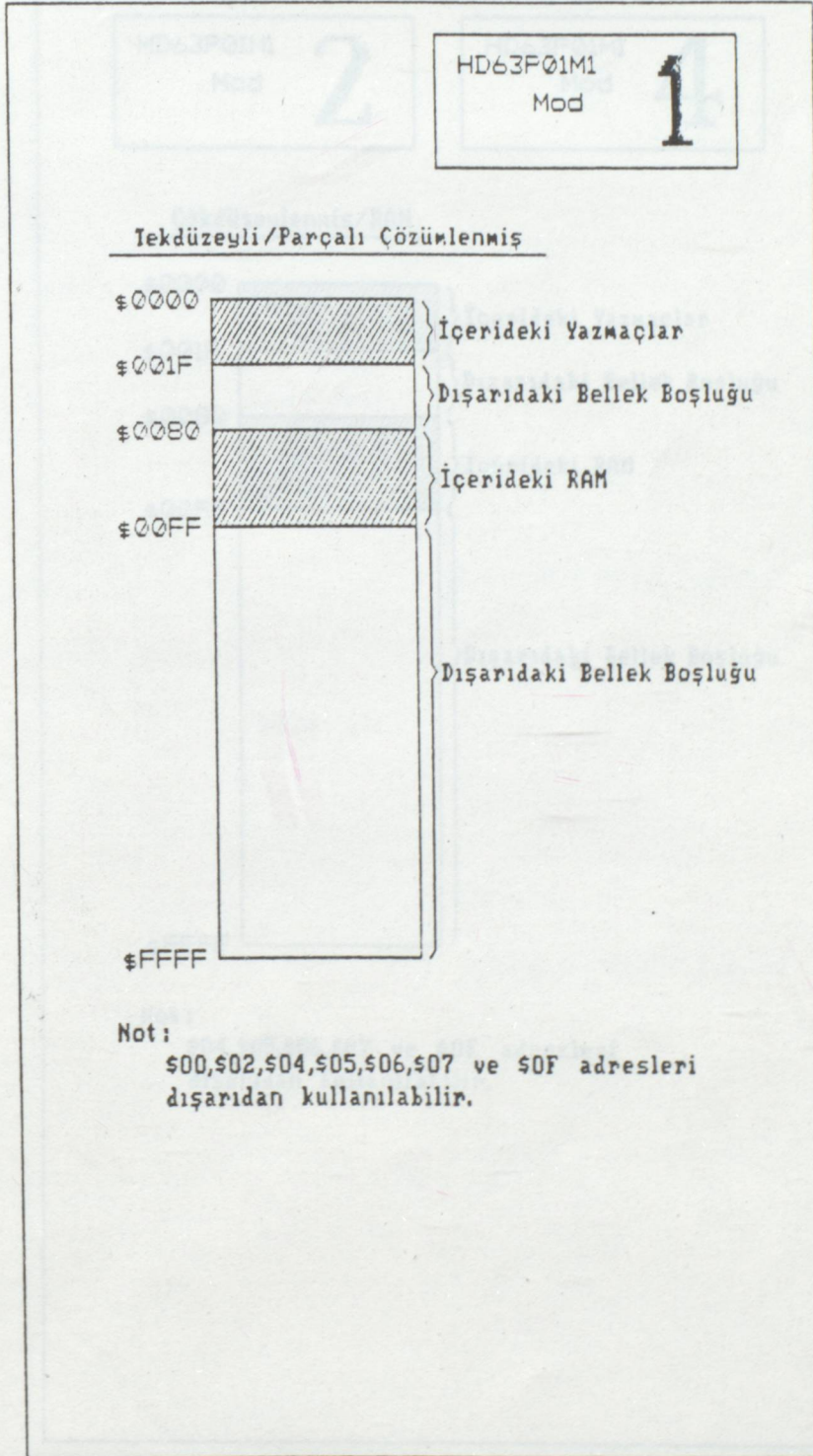
Çalışma şekli, reset işlemi sonunda 8,9 ve 10 nolu bağlantı uçlarında kullanıcı tarafından belirtilmelidir. Bu üç bağlantı ucu reset işaretinin çıkan kenarında Port 2'nin Veri Yazmacındaki PC0,PC1 ve PC2 kontrol bitlerine transfer edilir. Port 2 Veri Yazmacı aşağıdaki biçimdedir.

Port 2 Veri Yazmacı

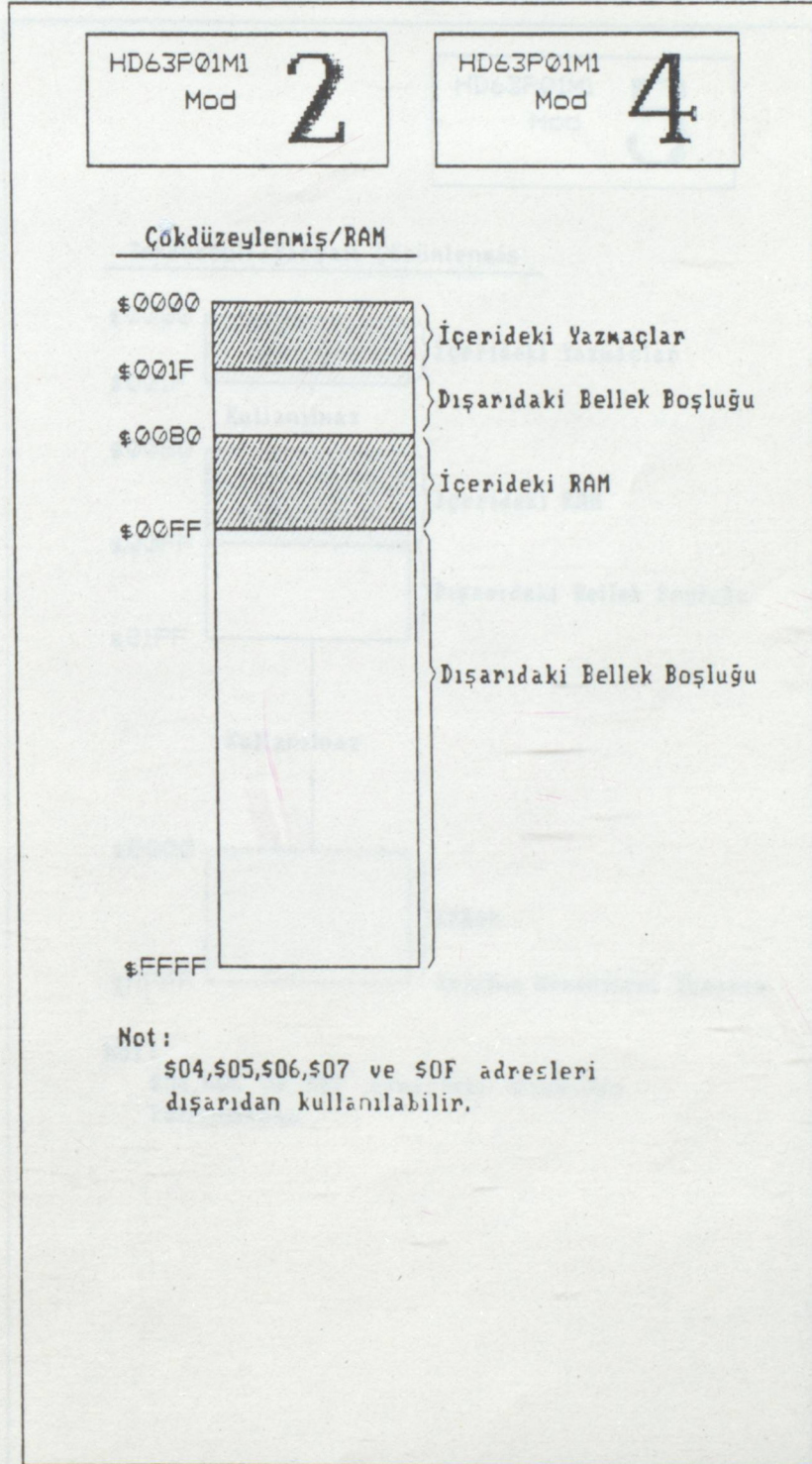
#0003	PC2	PC1	PC0	I/O 4	I/O 3	I/O 2	I/O 1	I/O 0
-------	-----	-----	-----	-------	-------	-------	-------	-------



Şekil 1.4a HD63P01M1 Bellek Haritaları



Şekil 1.4b HD63F01M1 Bellek Haritaları

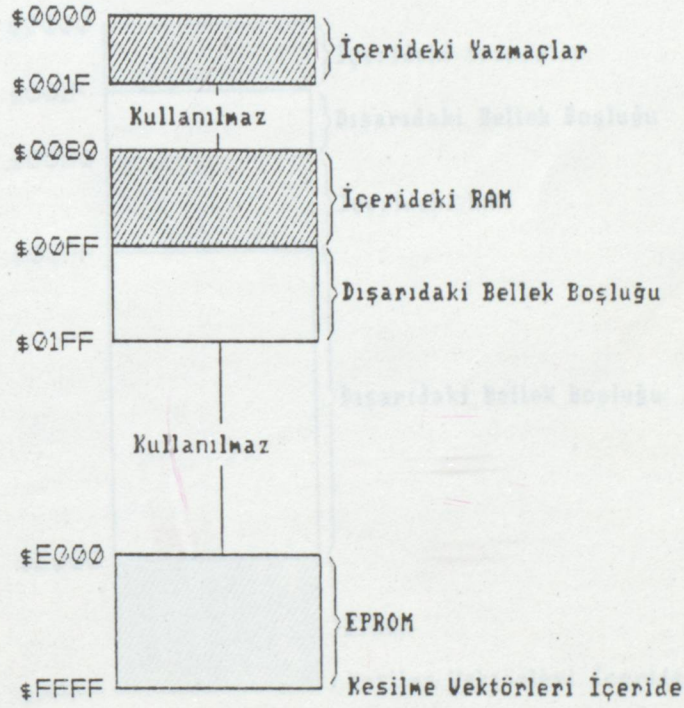


Şekil 1.4c HD63P01M1 Bellek Haritaları

HD63F01M1
Mod

5

Tekdüzeyli/Parçalı Çözümlemiş



Not:

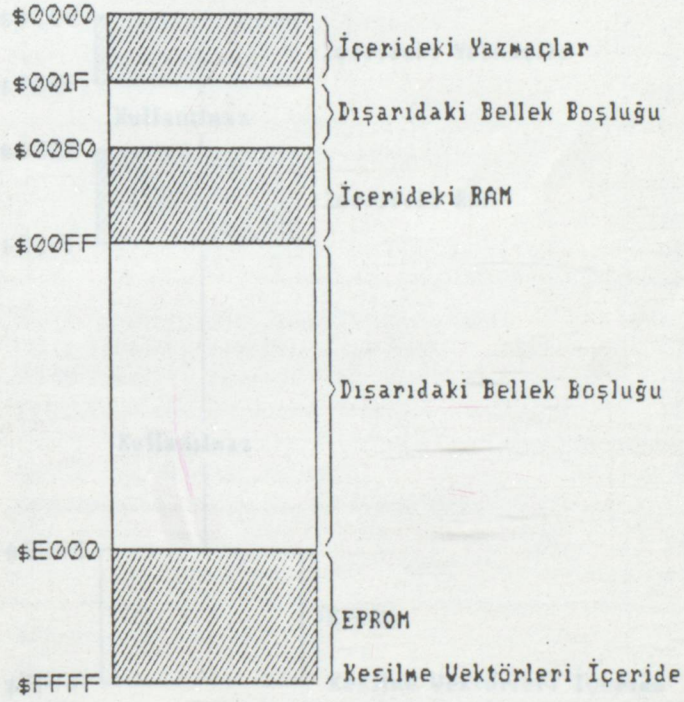
\$04, \$06 ve \$0F adresleri dışarıdan kullanılamaz.

Şekil 1.4d HD63F01M1 Bellek Haritaları

HD63P01M1
Mod

6

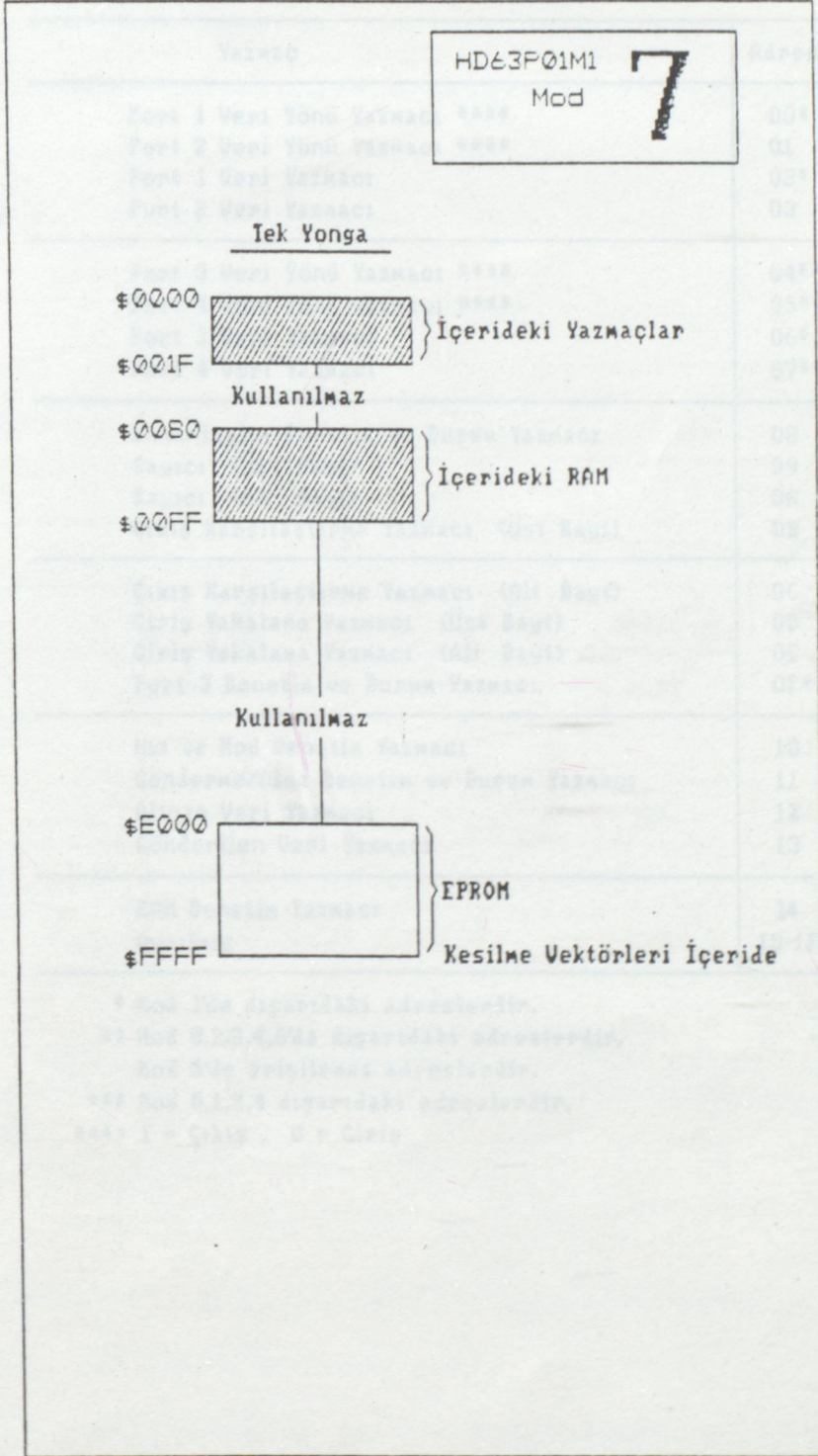
Çokdüzeylenmiş/Parçalı Çözümlemiş



Not :

\$04,\$06 ve \$0F adresleri dışarıdan kullanılabilir.

Şekil 1.4e HD63P01M1 Bellek Haritaları



Şekil 1.4f HD63P01M1 Bellek Haritaları

Çizelge 5 İçerideki Yazmaç Bölgesi

Yazmaç	Adres
Port 1 Veri Yönü Yazmacı ****	00*
Port 2 Veri Yönü Yazmacı ****	01
Port 1 Veri Yazmacı	02*
Port 2 Veri Yazmacı	03
Port 3 Veri Yönü Yazmacı ****	04**
Port 4 Veri Yönü Yazmacı ****	05***
Port 3 Veri Yazmacı	06**
Port 4 Veri Yazmacı	07***
Zamanlayıcı Denetim ve Durum Yazmacı	08
Sayıcı (Üst Bayt)	09
Sayıcı (Alt Bayt)	0A
Çıkış Karşılaştırma Yazmacı (Üst Bayt)	0B
Çıkış Karşılaştırma Yazmacı (Alt Bayt)	0C
Giriş Yakalama Yazmacı (Üst Bayt)	0D
Giriş Yakalama Yazmacı (Alt Bayt)	0E
Port 3 Denetim ve Durum Yazmacı	0F**
Hız ve Mod Denetim Yazmacı	10
Gönderme/Alma Denetim ve Durum Yazmacı	11
Alınan Veri Yazmacı	12
Gönderilen Veri Yazmacı	13
RAM Denetim Yazmacı	14
Ayrılmış	15-1F

* Mod 1'de dışarıdaki adreslerdir.

** Mod 0,1,2,4,6'da dışarıdaki adreslerdir.

Mod 5'de erişilemez adreslerdir.

*** Mod 0,1,2,4 dışarıdaki adreslerdir.

**** 1 = Çıkış , 0 = Giriş

Şekil 1.5'de çalışma şeklinin seçiminde kullanılacak donanıma ait bir örnek gösterilmiştir. 4053B, reset sırasında MCU'nun çevre birimlerinden ayrılarak çalışma şekli seçimi bilgisine bağlanmasını sağlar. Hiç bir çalışma şekli yazılım ile değiştirilemez. Çünkü Port 2 Veri Yazmacının 5,6 ve 7.bitleri salt okunur bitlerdir. Çalışma şekli seçimi Çizelge 6'da verilmiştir.

Çizelge 6 Çalışma Şekillerinin Seçimi

Mod	P22 (PC2)	P21 (PC1)	P20 (PC0)	ROM	RAM	Kesilme Vektörleri	Adres ve Veri Yolunun Şekli	Çalışma Şekli
7	H	H	H	I	I	I	I	Tek Yonga
6	H	H	L	I	I	I	MUX(3)	Çokdüzeyli/Parçalı Çöz.
5	H	L	H	I	I	I	NMUX(3)	Tekdüzeyli/Parçalı Çöz.
4	H	L	L	E(1)	I	E	MUX	Çokdüzeyli/RAM
3	L	H	H	-	-	-	-	Kullanılmamış
2	L	H	L	E(1)	I	E	MUX	Çokdüzeyli/RAM
1	L	L	H	E(1)	I	E	NMUX	Tekdüzeylemiş
0	L	L	L	I	I	I(2)	MUX	Çokdüzeylemiş Test

I - İçeride

E - Dışarıda

MUX - Çokdüzeylemiş

NMUX - Tekdüzeylemiş

L - Lojik "0"

H - Lojik "1"

1) İçerideki ROM devre dışı bırakılmış.

2) Reset vektörü RES'in "1"e gitmesinden 4 dönüş sonra dışarıdadır.

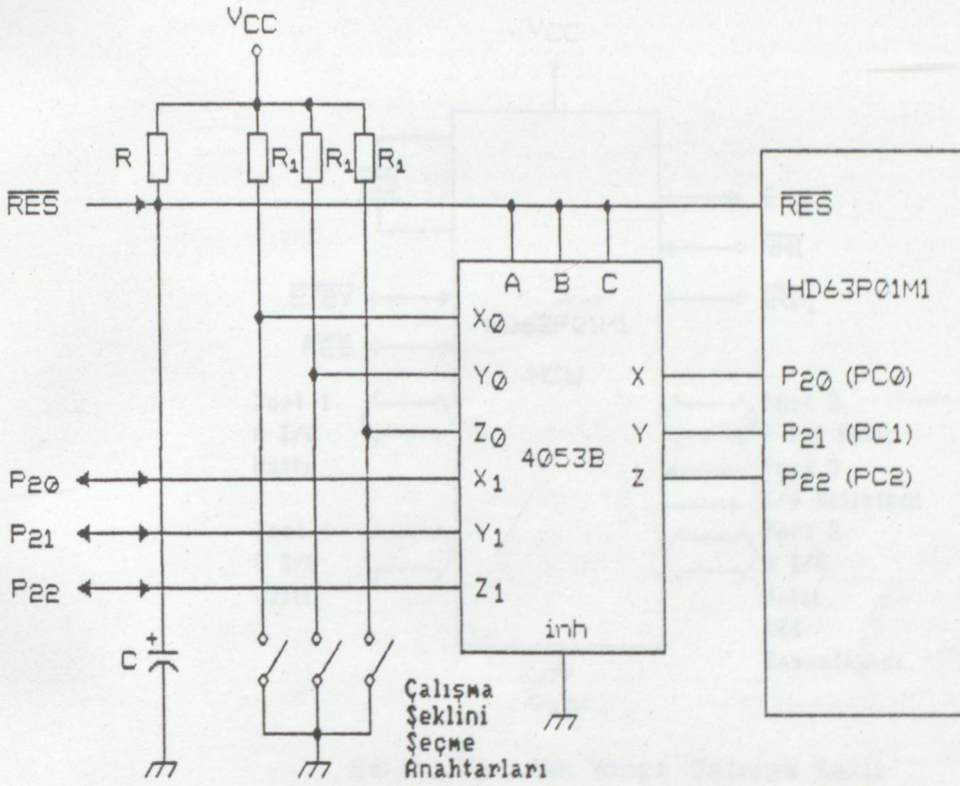
3) Port 4 adres çıkışlarından kullanılmayanlar giriş portu yapılabilir.

HD63P01M1, üç temel çalışma şeklinde çalışır.

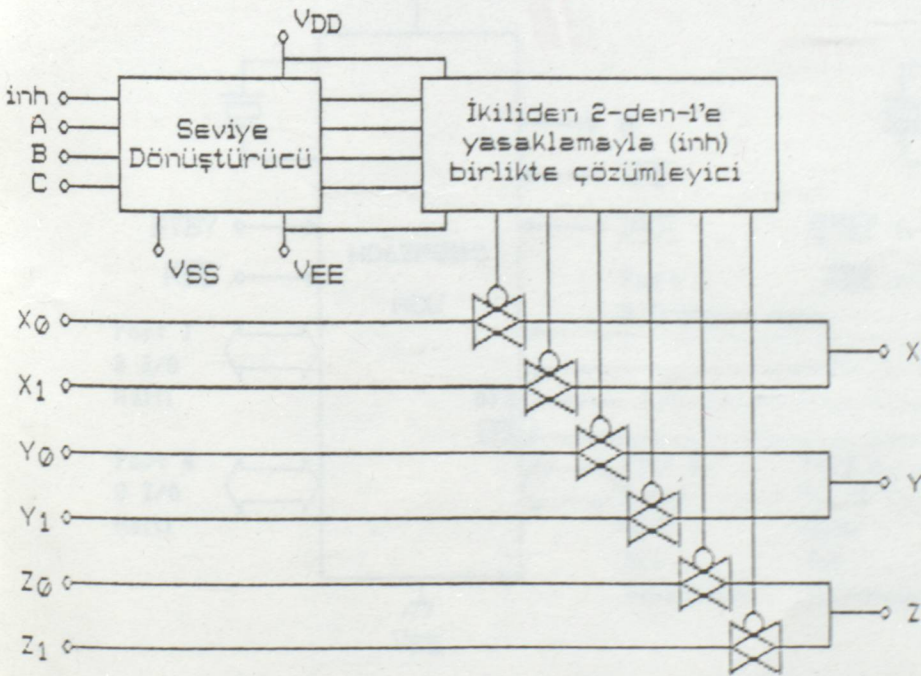
(1) Tek yonga çalışma şekli

(2) Genişletilmiş tekdüzeyli çalışma şekli

(3) Genişletilmiş çokdüzeyli çalışma şekli



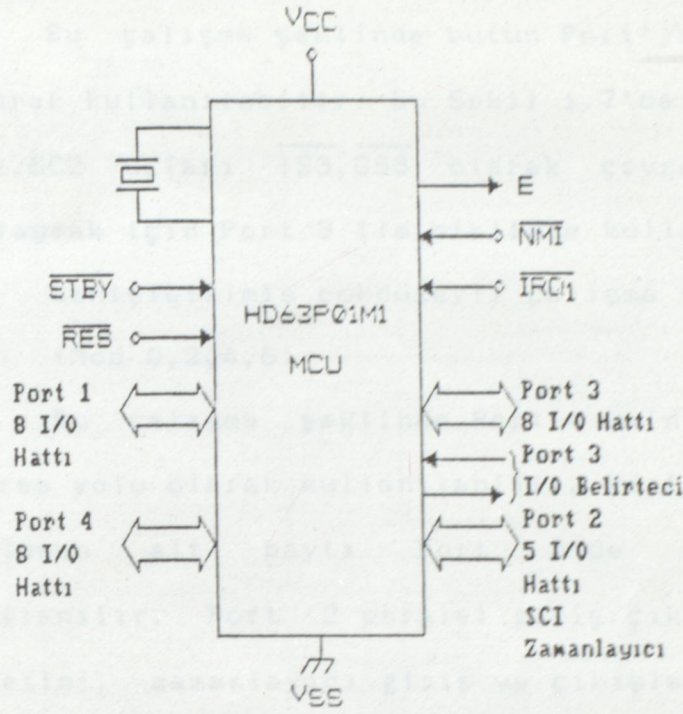
Şekil 1.5 Çalışma şekli seçimi için önerilen devre



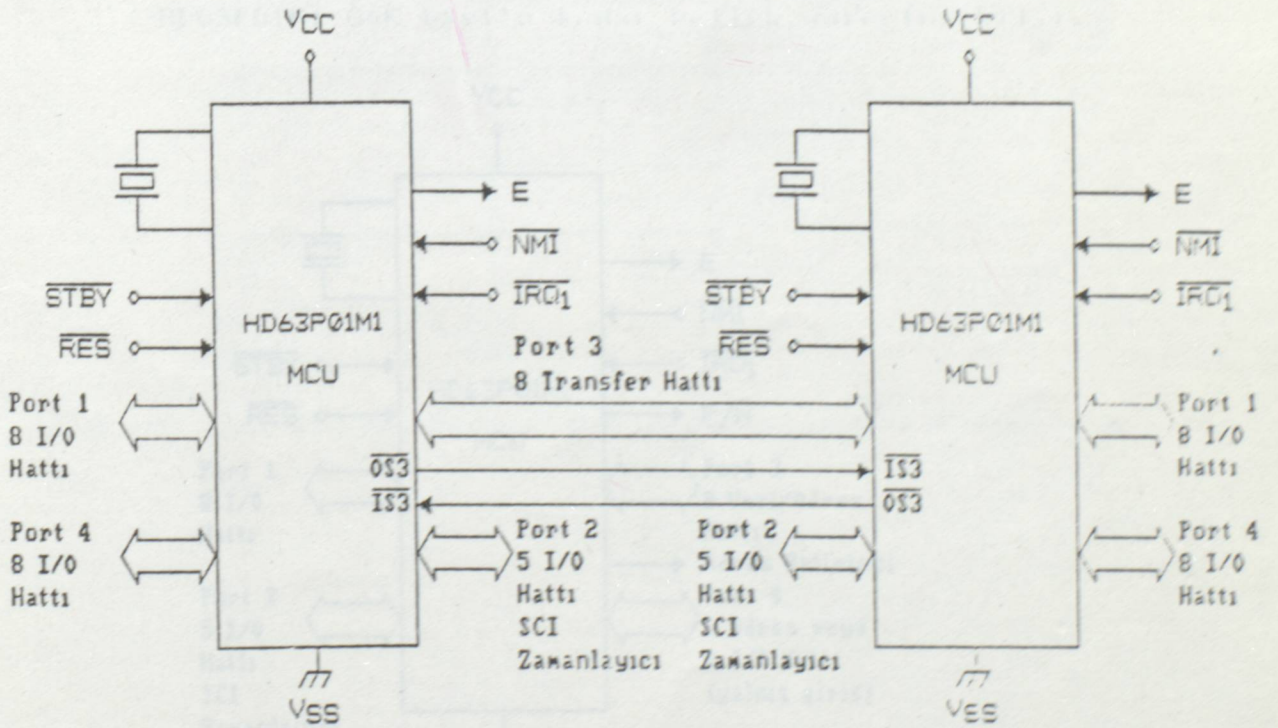
Doğruluk Tablosu

Girişler		Çıkışlar				
inh	C	B	A	Z	Y	X
0	0	0	0	Z ₀	Y ₀	X ₀
0	0	0	1	Z ₀	Y ₀	X ₁
0	0	1	0	Z ₀	Y ₁	X ₀
0	0	1	1	Z ₀	Y ₁	X ₁
0	1	0	0	Z ₁	Y ₀	X ₀
0	1	0	1	Z ₁	Y ₀	X ₁
0	1	1	0	Z ₁	Y ₁	X ₀
0	1	1	1	Z ₁	Y ₁	X ₁
1	x	x	x			

Şekil 1.6 4053B Çokdüzeyleyici/Tekdüzeyleyici



Şekil 1.7a Tek Yonga Çalışma Şekli



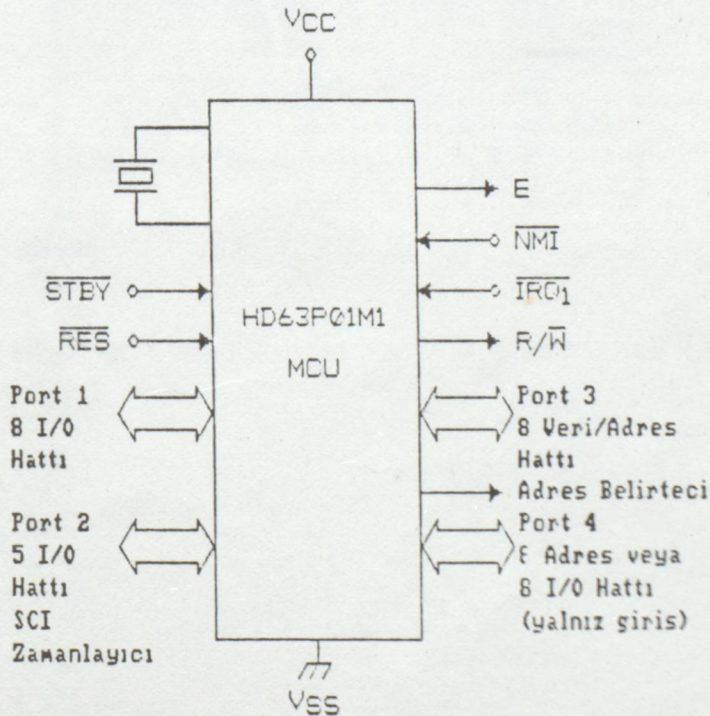
Şekil 1.7b Tek Yonga Çift İşleyicili Düzen

1.10.1 Tek yonga çalışma şekli (Mod 7)

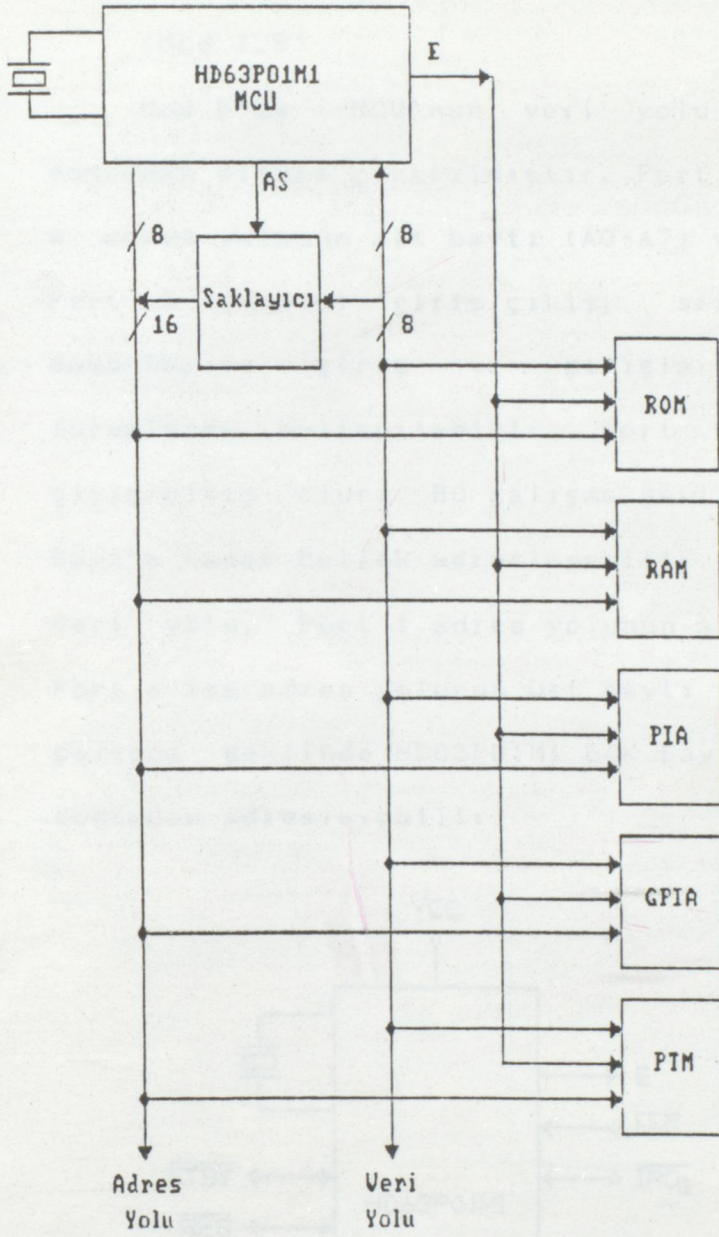
Bu çalışma şeklinde bütün Port'lar giriş/çıkış olarak kullanılabilir. Bu Şekil 1.7'de gösterilmiştir. SC1, SC2 uçları $\overline{IS3}$, $\overline{OS3}$ olarak çevre birimleriyle anlaşmak için Port 3 ile birlikte kullanılır.

1.10.2 Genişletilmiş çokdüzeyli çalışma şekilleri (Mod 0, 2, 4, 6)

Bu çalışma şeklinde Port 4 yalnız giriş veya adres yolu olarak kullanılabilir. Veri yolu ile adres yolunun alt baytı Port 3'de anahtarlansarak kullanılır. Port 2 paralel giriş/çıkış, seri veri iletimi, zamanlayıcı giriş ve çıkışları gibi çeşitli durumlarda kullanılabilir. Port 1'den 8 Bit paralel giriş/çıkış yapılabilir. Bu çalışma şeklinde HD63F01M1 64K bayt'a kadar bellek adresleyebilir.



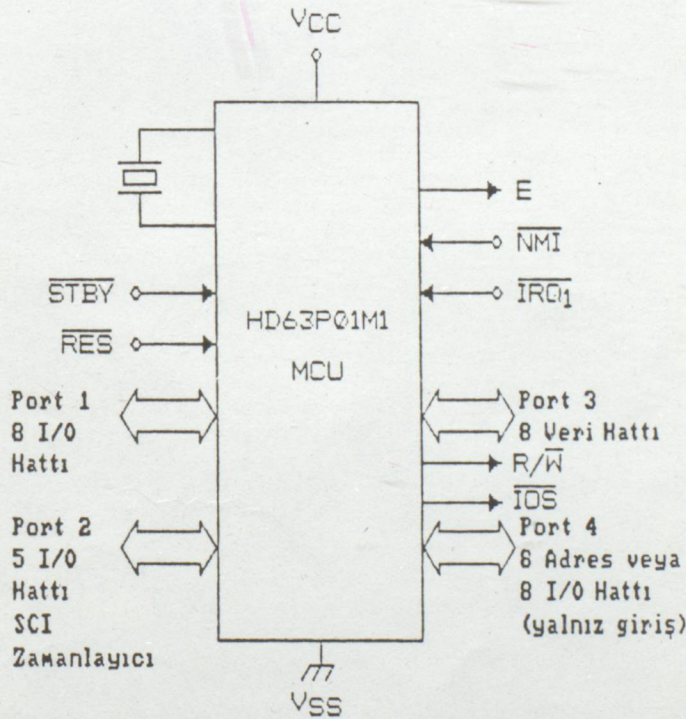
Şekil 1.8a Genişletilmiş Çokdüzeyli Çalışma Şekli



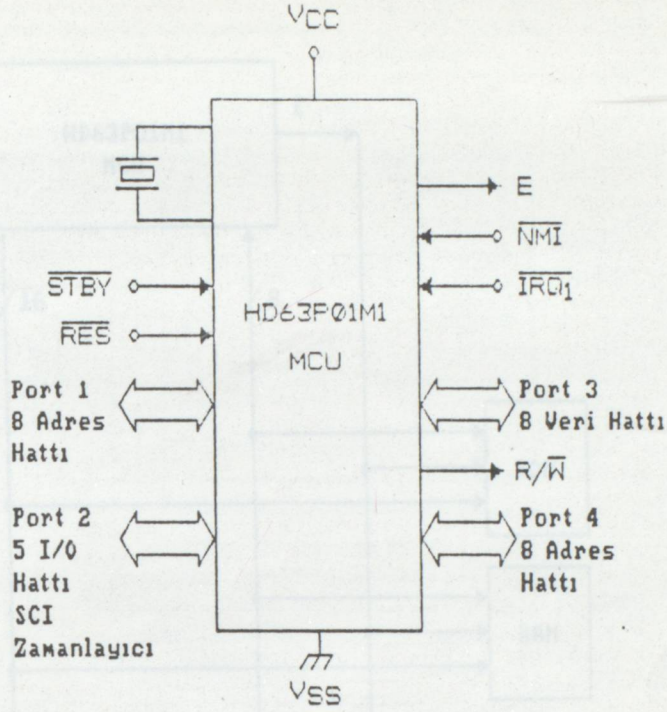
Şekil 1.8b HD63P01M1 MCU Genişletilmiş Çokdüzeyli Çalışma (Mod 2,4 ve 6)

1.10.3 Genişletilmiş tekdüzeyle çalışma şekilleri
(Mod 1,5)

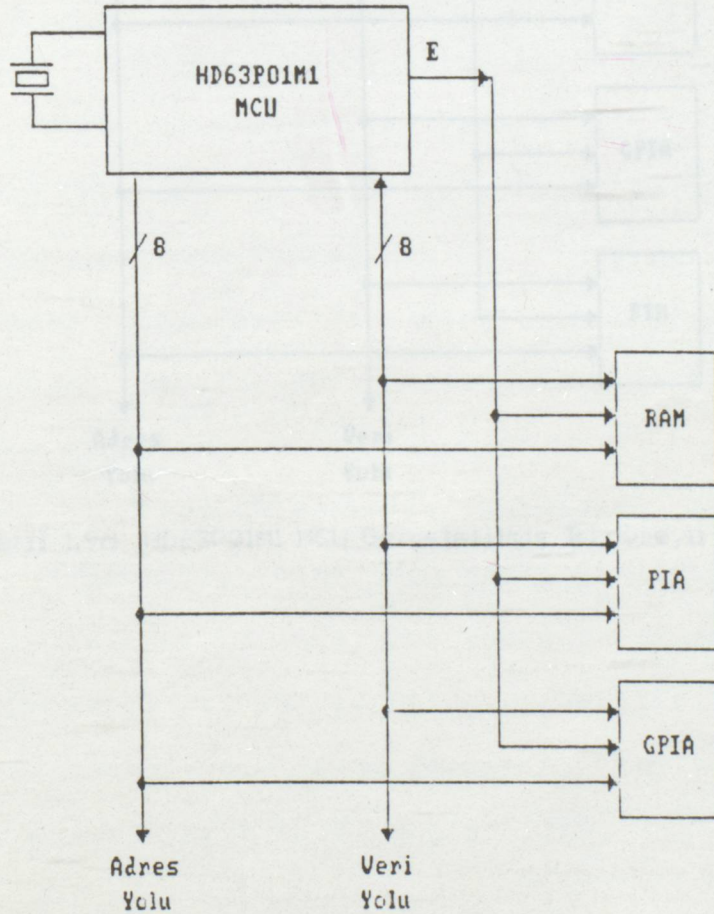
Mod 5'de MCU'nun veri yolu ve adres yolu doğrudan dışarı çıkarılmıştır. Port 3 veri yolu, Port 4 adres yolunun alt baytı (A0-A7) veya giriş olur. Port 2 paralel giriş/çıkış, seri veri iletimi, zamanlayıcı giriş ve çıkışları gibi çeşitli durumlarda kullanılabilir. Port 1 yalnız paralel giriş/çıkış olur. Bu çalışma şeklinde HD63P01M1 256 Bayt'a kadar bellek adresleyebilir. Mod 1'de Port 3 veri yolu, Port 1 adres yolunun alt baytı (A0-A7), Port 4 ise adres yolunun üst baytı (A8-A15) olur. Bu çalışma şeklinde HD63P01M1 64K Bayt'a kadar belleği doğrudan adresleyebilir.



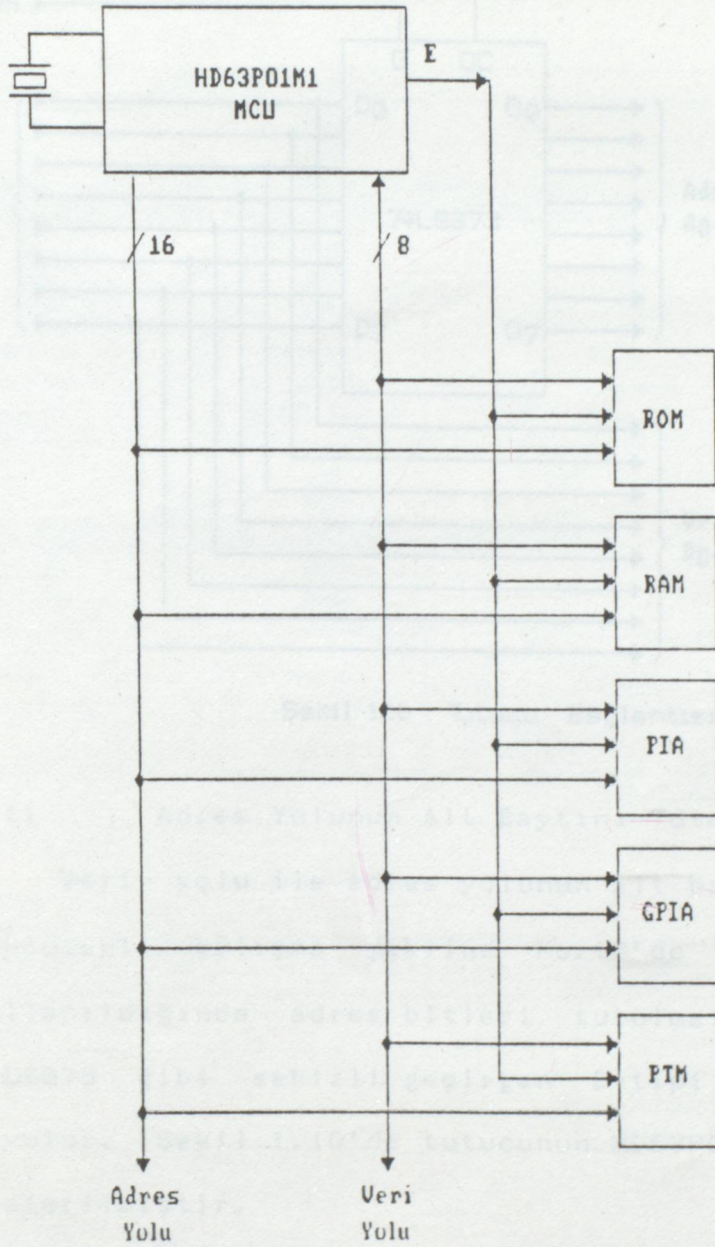
Şekil 1.9a Genişletilmiş Tekdüzeyle Çalışma Mod 5



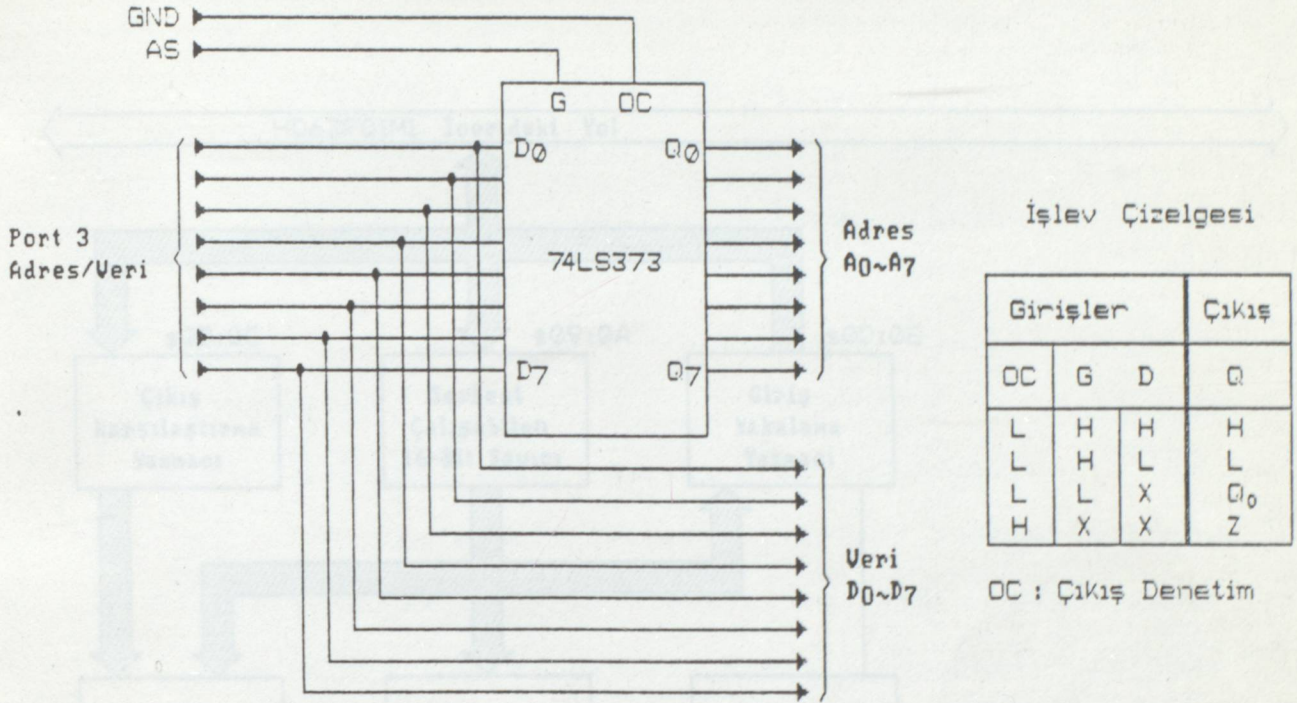
Şekil 1.9b Genişletilmiş Tekdüzeyli Çalışma Mod 1



Şekil 1.9c HD63P01M1 MCU Genişletilmiş Tekdüzeyli Çalışma (Mod 5)



Şekil 1.9d HD63F01M1 MCU Genişletilmiş Tekdüze/yılı Çalışma (Mod 1)



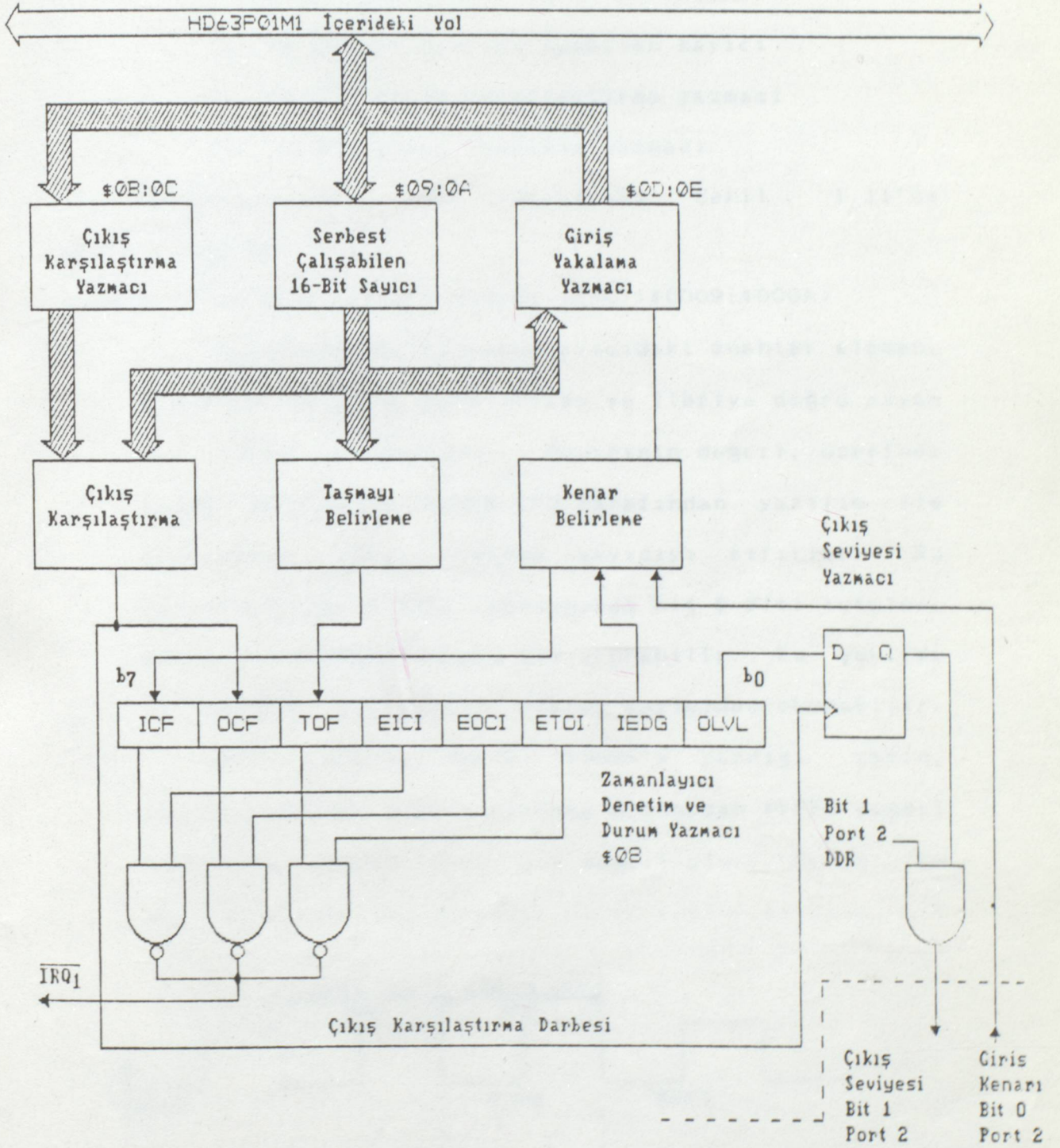
Şekil 1.10 Tutucu Bağlantısı

1.11 Adres Yolunun Alt Baytını Tutmak

Veri yolu ile adres yolunun alt baytı genişletilmiş çokdüzeyli çalışma şekline Port3'de çokdüzeyli olarak kullanıldığında adres bitleri tutulmalıdır. Bunun için 74LS373 gibi sekizli geçirgen D-tipi tutucuya ihtiyaç duyulur. Şekil 1.10'da tutucunun HD63P01M1 ile bağlantısı gösterilmiştir.

1.12 Programlanabilir Zamanlayıcı (TIMER)

HD63P01M1, dışarıdan giriş yapılabilen 16 bitlik bir zamanlayıcıyı içinde bulundurur. Buna ek olarak bir zamanlayıcı çıkış işaretide üretebilir. Her iki giriş ve dalga şekli için darbe genişliği, birkaç mikrosaniyeden bir çok saniyeye kadar değişebilir.



Şekil 1.11 Programlanabilir Zamanlayıcının Elck Diyagramı

Zamanlayıcı donanımı aşağıdakilerden oluşur.

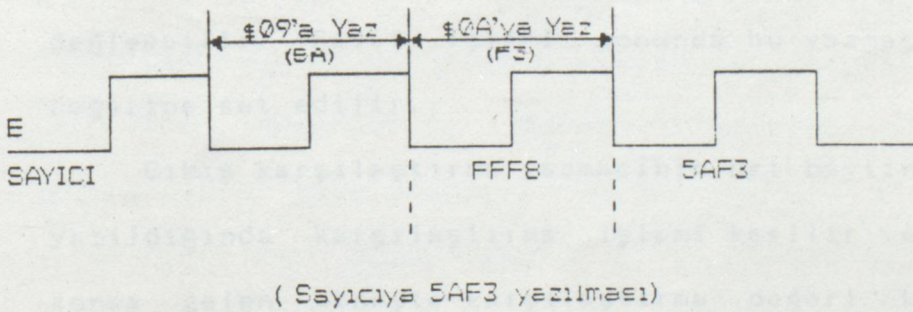
- * bir 8 Bit denetim ve durum yazmacı
- * bir 16 Bit serbest çalışabilen sayıcı
- * bir 16 Bit çıkış karşılaştırma yazmacı
- * bir 16 Bit giriş yakalama yazmacı

Zamanlayıcının blok diyagramı Şekil 1.11'de gösterilmiştir.

1.12.1 Serbest Çalışan Sayıcı (FRC)(\$0009:\$000A)

Programlanabilir zamanlayıcıdaki anahtar eleman, E sistem saati ile sürülebilir ve iletilebilir doğru sayan bir 16 Bit'lik sayıcıdır. Sayıcının değeri, üzerinde işlem yapılmadığı zaman CPU tarafından yazılım ile okunabilir. Reset işlemi sayıcıyı sıfırlar. Bu sayıcının üst 8 Biti okunduğunda alt 8 Biti tutulur. Sonra gelen okuma komutu ile okunabilir. Bu şekilde iki baytlık veri güvenli olarak sayıcıdan okunabilir.

CPU kararlı veriyi \$0009'a yazdığı zaman, yazılan verinin değeri dikkate alınmadan \$FFF8 değeri sayıcının (\$0009:\$000A) ilk değeri olur. Sonra CPU kararlı veriyi \$000A'ya yazdır ve veri sayıcının alt



Şekil 1.12 Sayıcı Yazma Zamanlaması

baytını (4000A) set eder. Ayrıca 40009'a yazılan veri sayıcısının üst baytını set eder.

Sayıciya veri yazılacağı zaman, iki baytı birden yazabilen komutlar (ör:STD) kullanılmalıdır. Eğer sadece verinin üst baytı yazılırsa sayıcı 4FFF8'e set olur.

Seri iletişim arabirimi sistem saatinden yararlanıyorsa sayıcıya yazmak, SCI'nin çalışmasını bozabilir.

1.12.2 Çıkış Karşılaştırma Yazmacı (OCR) (4000E:4000C)

Çıkış işaretini denetlemek amacıyla kullanılan okunabilir/yazılabilir 16 Bit'lik yazmaçtır. Bu yazmacın içinde sayıcısının (FRC) değeri ile devamlı bir karşılaştırma vardır. İçeriği FRC'nin değerine eşit olduğunda zamanlayıcı denetim/durum yazmacındaki (TCSR), çıkış karşılaştırma bayrağı (OCF) "1" olur ve TCSR'deki çıkış seviyesi bitinin (OLVL) değeri, Port 2'nin 1.bitine aktarılır. Port 2 veri yönü yazmacının 1.bitini "1" ise OLVL, Port 2'nin 1.bitinde görünür. Aksi halde görülemez. Daha sonra çıkış seviyesinin değeri ikinci karşılaştırma için değişebilir. Reset işleri sonunda bu yazmaç 4FFFF değerine set edilir.

Çıkış karşılaştırma yazmacının üst baytına veri yazıldığında karşılaştırma işlemi kesilir ve hemen sonra gelen dönüşte karşılaştırma değeri korunur. Aynı şekilde FRC'ye yazma işlemi yapıldığında

karşılaştırma kesilir.

Çıkış karşılaştırma yazmacına, iki baytı birden yazabilen komutlar (ör.STD) kullanılarak yazılır.

1.12.3 Giriş Yakalama Yazmacı (ICR)(\$000D:\$000E)

Giriş yakalama yazmacı (ICR), dışarıdan bir giriş işareti uygulandığında, bu işaretin değişimi sırasında FRC'nin değerini saklayan salt okunur 16 bitlik yazmaçtır.

Sayıcı geçişinin tetiklenmesi istendiği zaman, giriş transfer değişikliği, giriş kenarı biti (IEDG) ile kontrol edilebilir. Dışarıdan girilen işaretin kenar yakalama devresine girebilmesi için Port2'nin veri yönü yazmacının 0.biti "0" yapılmalıdır.

Bütün durumlarda girişin yakalanabilmesi için en az 2 sistem saati dönüş süresi geçmelidir.

1.12.4 Zamanlayıcı Denetim/Durum Yazmacı (TCSR)(\$0008)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	\$0008

Alt 5 biti (bit4-bit0) okunur/yazılır, üst 3 biti ise salt okunur olan 8 bitlik bir yazmaçtır. Üst 3 bit zamanlayıcı durum bilgisini aşağıda gösterildiği gibi bildirir.

(1) Bir uygun geçiş giriş bağlantı ucunda yakalanmıştır. (ICF)

(2) Serbest çalışan sayıcının ve çıkış karşılaştırma yazmacının içerikleri eşit olmuştur. (OCF)

(3) \$0000'a kadar sayılmıştır. (TOF)

IRQ2 kesilmesine izin verilip verilmeyeceğini belirleyen Zamanlayıcı Kontrol/Durum Yazmacında (TSCR) her bayrağın bir kontrol biti bulunur. Eğer Durum Kodu Yazmacının (CCR), "1" biti "0" yapılmış ise her bayrak için öncelikli bir kesilme vektör adresi üretilir. Her bitin işlevi aşağıda verilmiştir.

Bit 0 Çıkış Seviyesi (OLVL)

Sayıcı (FRC) ile çıkış karşılaştırma yazmacı (OCR) eşit olduğunda bu bit Port2 veri yazmacının 1.bitine transfer edilir. Eğer Port2 veri yönü yazmacı uygun şekilde şartlandırılmış ise çıkış dışarıya ulaşabilir.

Bit 1 Giriş Kenarı (IEDG)

Port2'nin 0.bitinin geçiş kenarını kontrol eden bu bit, giriş yakalama yazmacından (ICR) serbest sayıcıya (FRC) veri transferini tetikleyecektir. Dışarıdan işaret girilebilmesi için Port2 veri yönü yazmacının 0.bitini "0" yapılmalıdır. IEDG "0" ise düşen kenarda, "1" ise çıkan kenarda tetikleme yapılır.

Bit 2 Zamanlayıcı Taştı Kesilmesine İzin (ETOI)

Bu bit "1" yapılırsa, TOF bayrağının sebep olabileceği $\overline{\text{IRQ2}}$ kesilmesine izin verilir. "0" yapılırsa kesilmeye izin verilmez.

Bit 3 Çıkış Karşılaştırma Kesilmesine İzin (EOCI)

Bu bit "1" yapılırsa, OCF bayrağının neden olabileceği $\overline{\text{IRQ2}}$ kesilmesine izin verilir. "0" yapılırsa kesilmeye izin verilmez.

Bit 4 Giriş Yakalama Kesilmesine İzin (EICI)

Bu bit "1" yapılırsa, ICF bayrağının neden olabileceği $\overline{\text{IRQ2}}$ kesilmesine izin verilir. "0" yapılırsa kesilmeye izin verilmez.

Bit 5 Zamanlayıcı Taştı Bayrağı (TOF)

Bu salt okunur bit, serbest sayıcının (FRC) \$FFFF'den \$0000'a geçmesi ile "1" olur. Ana işlem biriminin (CPU) TCSR'yi (TOF "1") okumasından sonra sayıcıyı (\$0009) okumasıyla "0" olur.

Bit 6 Çıkış Karşılaştırma Bayrağı (OCF)

Bu salt okunur bit, sayıcı ile çıkış karşılaştırma yazmacının içerikleri eşitlendiği zaman "1" olur. Ana işlem biriminin (CPU) TCSR'yi (OCF "1") okumasından sonra çıkış karşılaştırma yazmacına (\$000B veya \$000C) yazmasıyla "0" olur.

Bit 7 Giriş Yakalama Bayrağı (ICF)

Bu salt okunur bit, önceden belirlenen kenar geçişi sırasında "1" olur. Ana işlem biriminin (CPU) TCSR'nin (ICF "1") okumasından sonra giriş yakalama

yazmacını (\$000D) okumasıyla "0" olur.

Reset işlemi sonunda TCSR'nin bütün bitleri "0" olur.

1.13 Seri Veri İletişim Arabirimi (SCI)

HD63P01M1, tümüyle ikili (full-duplex) asenkron seri veri iletişim arabirimini (SCI) içinde bulundurur. SCI veri iletim hızı yazılım ile bir kaç çeşit seçilebilir. Aynı ayrı çalışan fakat veri biçimi ve hızı aynı olan bir veri gönderici ve alıcıdan oluşur. Veri alma ve gönderme Port2'nin P22, P23 ve P24 bağlantı uçlarıyla sağlanabilir. Ana işlem birimiyle (CPU) bilgi iletişimi yapabilir. Seri veri iletişim donanımı ve yazılımı ile ilgili ayrıntılı bilgi aşağıda verilmiştir.

1.13.1 Uyanma Özelliği (Wake-Up)

Tipik çok kullanıcı çalışmalarda yazılım yönetimi, genellikle ilk olarak bir işaret baytı alacaktır. Bunun amacı (Wake-Up) seçilmeyen MCU'nun mesajın kalan kısmına karışmamasını sağlamaktır. Böylece seçilmediği belirlenen MCU yeni mesaj gelene kadar bekletilir ve bütün ileri kesilme bilgileri durdurulabilir.

Uyanma özelliğine veri iletim hattından 10 tane "1" verisi gönderilerek tekrar izin verilebilir. Bu yüzden yazılım sıralaması, iki mesaj arasındaki boş bir periyoda konulmalı ve mesaj içinde bu durum korunmalıdır.

Seçilmemiş MCU, donanım veya ikinci bir mesaj ile tekrar uyanma özelliği kazanabilir.

1.13.2 Yazılım Özellikleri

HD63P01M1 aşağıda belirtilen programlama özelliklerine sahiptir.

Veri iletim şekli, standart darbe/boşluk biçimindedir. (NZR)

Saat kaynağı kullanım ihtiyacına göre içeriden veya dışarıdan sağlanabilir.

Veri iletim hızı, E sistem saati ile birlikte verilen 4 seçenektен birisi veya dışarıdan girilen saatin frekansının 1/4'ü oranındadır.

Uyanma özelliğine izin verilebilir veya verilmez.

Verici ve alıcı için ayrı ayrı örtülebilir veya izin verilebilir kesilme istekleri vardır.

İçerideki saatin Port2'deki P22 bağlantı ucundan dışarı çıkmasına kullanıcı isteğine göre izin verilebilir veya verilmeyebilir

Port2'deki P23,P24 bağlantı uçları seri giriş/çıkış için kullanılabilir.

1.13.3 Seri Veri İletişim Donanımı

Seri veri iletişim donanımı Şekil 1.13'de gösterildiği gibi 4 yazmaç tarafından kontrol edilir. Bu yazmaçlar aşağıda belirtilmiştir.

Bir 8 bit kontrol/durum yazmacı

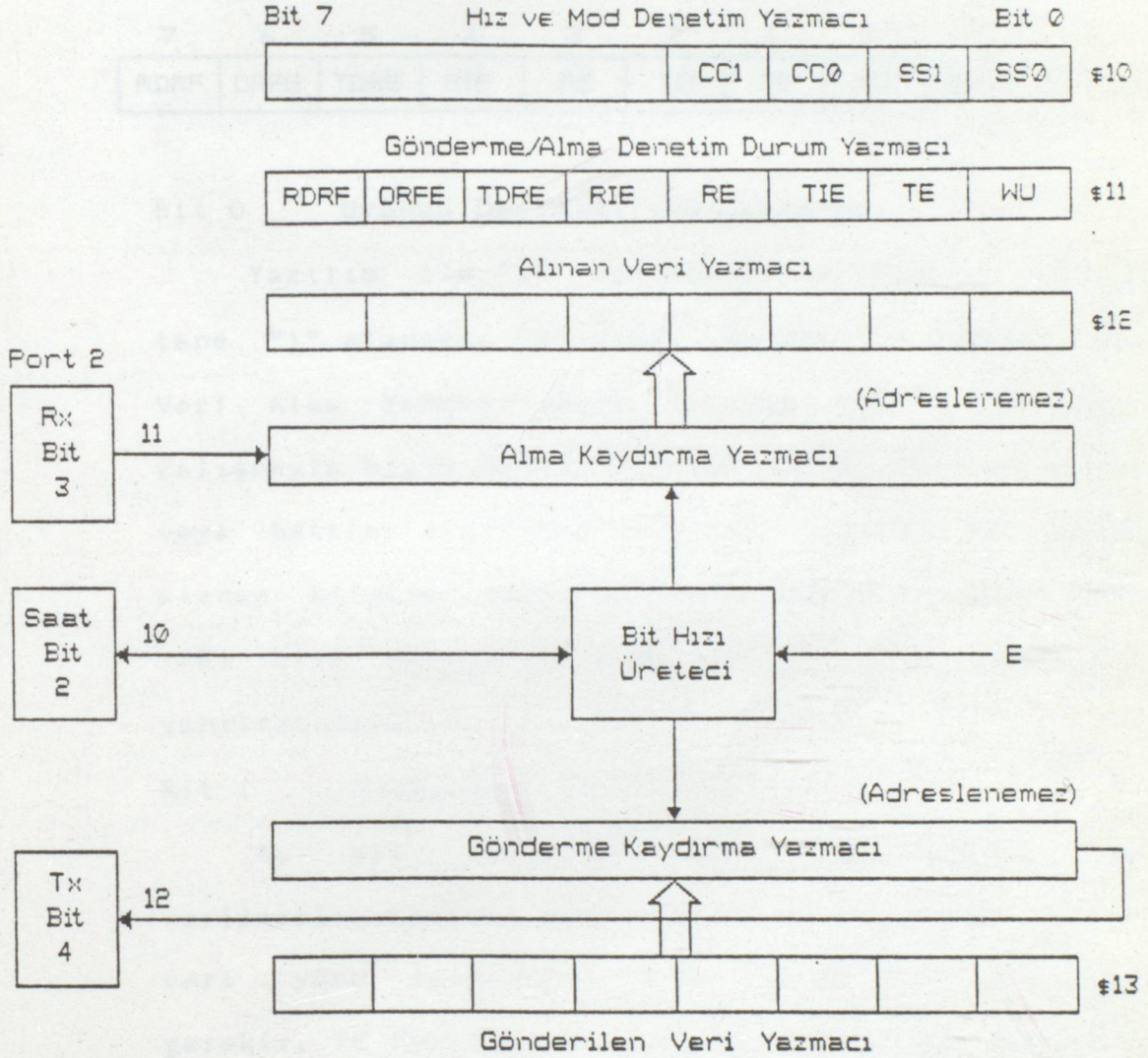
Bir 4 bit hız/mod kontrol yazmacı (salt yazılır)

Bir 8 bit veri alma yazmacı (salt okunur)

Bir 8 bit veri gönderme yazmacı (salt yazılır)

Bu 4 yazmaç dışında Port 2'nin P22,P23 ve P24

bağlantı uçlarıyla birlikte veri yazmacı kullanılır.



Şekil 1.13 Seri Giriş/Çıkış Yazmacı

1.13.4 Gönderme/Alma Denetim Durum Yazmacı

(TRCSR) (\$0011)

Gönderme/Alma denetim durum yazmacı (TRCSR), 0,1,2,3 ve 4. biti yazılabilen, bütün bitleri okunabilen 8 bitlik yazmaçtır. Bu yazmaç reset (\overline{RES}) işlemi sonunda \$20'ye set edilir. Bu yazmaç açık

olarak aşağıda gösterilmiştir.

Gönderme/Alma Denetim Durum Yazmacı

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	0011

Bit 0 Uyanma Özelliği (WU)(Wake-Up)

Yazılım ile "1" yapılabilir ve donanım ile 10 tane "1" alınırsa "0" olur. Bu bit "1" olduğu zaman Veri Alma Yazmacı Dolu bayrağı (RDRF) ve Aşırı Çalışmayla biçim Hatası bayrağı (ORFE) veriler alınsa veya hatalar algılansa bile "1" olamaz. Bu yüzden alınan bilgiler doğru bilgiler değildir. Alma izini (RE) biti uyanma özelliği bitinden (WU) önce "1" yapılmalıdır.

Bit 1 Göndermeye İzin (TE)

Bu bit vericinin veri göndermesine izin verilmesini kontrol eder. TE "1" olduğu zaman, Port 2 veri yönü yazmacının 4.bitinde "1" yapılması gerekir. TE "0" olsa bile bu bit "1" olarak kalır. TE "1" yapıldıktan sonra 10 tane "1" verisi gönderilirse verici bilgi göndermeye hazır duruma gelir. "0" yapılırsa gönderme örtülür ve seri giriş/çıkış Port 2'deki P24 bağlantı ucu üzerinde etkili olamaz.

Bit 2 Gönderme Kesilmesine İzin (TIE)

TIE biti "1" olduğu zaman Veri Gönderme Yazmacı Boş bayrağı (TDRE) bir $\overline{IRQ2}$ kesilmesine neden olabilir. "0" yapılırsa TDRE bayrağının neden

olabileceği IRQ2 kesilmesi örtülür.

Bit 3 Alma izini (RE)

Bu bit "1" yapıldığı zaman, Port 2'deki P23 bağlantı ucu veri yönü yazmacı dikkate alınmadan bir alıcı girişi gibi kullanılır. "0" yapılırsa alıcı örtülür.

Bit 4 Alma Kesilmesine İzin (RIE)

RIE biti "1" yapıldığında RDRF veya ORFE bayraklarının sebep olabileceği bir IRQ2 kesilmesine izin verilir. "0" yapıldığında ise kesilme örtülür.

Bit 5 Veri Gönderme Yazmacı Boş (TDRE)

Veri gönderme yazmacından çıkış öteleme yazmacına veri gönderildiği zaman bu bit donanım tarafından "1" yapılır. TDRE bayrağı, denetim durum yazmacının okunmasından sonra veri yazmacına bir sonraki verinin yazılmasıyla "0" olur. TDRE biti reset işlemi sonunda "0" başlangıç değerini alır.

Bit 6 Aşırı Çalışmayla Biçim Hatası (ORFE)

Aşırı çalışma, biçim hatası meydana geldiği zaman (sadece alma durumunda), bu bit donanım tarafından "1" yapılır. Aşırı çalışma hatası, RDRF biti "1" durumundayken, Alıcı Veri Yazmacına yeni veri yazılmak isteniyorsa meydana gelir. Biçim hatası ise bit sayıcı, alınan bayt'daki bit düzeni ile uygun olmadığı zaman meydana gelir. Biçim hatası meydana geldiği zaman RDRF "1" olmaz. Bu yüzden biçim hatası aşırı çalışma hatasından farkedilebilir. Bunun anlamı

eğer ORFE ile RDRF "1" ise aşırı çalışma hatası algılanmıştır. "0" ise biçim hatası algılanmıştır. ORFE biti, denetim durum yazmacı okunduktan sonra veri alma yazmacının okunmasıyla veya reset işlemi sonunda "0" olur.

Bit 7 Veri Alma Yazmacı Dolu (RDRF)

Veri alma yazmacı dolu biti (RDRF), verinin öteleme yazmacından alıcı veri yazmacına gönderilmesi sırasında donanım ile "1" yapılır. RDRF biti, denetim durum yazmacı okunduktan sonra veri alma yazmacının okunmasıyla veya reset işlemi sonunda "0" olur.

1.13.5 Tranfer Hızı/Modu Kontrol Yazmacı (RMCR)

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	#0010

Bu yazmaç aşağıdaki seri giriş çıkış işlevlerini içerir.

- veri iletim hızı
- veri biçimi
- saat kaynağı
- Port 2'deki P22 bağlantı ucunun belirlenmesi

RMCR salt yazılır, reset işlemi sonunda "0" olan 4 bitlik bir yazmaçtır. 4 bitlik alan ayrı ayrı iki tane 2 bitlik bölge olarak düşünülmüştür. Alt 2 bit içerideki saat tarafından bit işlem hızını belirlerken, üst 2 bit ise veri biçimini ve saat

kaynağının seçimini kontrol eder.

Bit 0, Bit 1 Seçilen Hız (SS0)(SS1)

Bu bitler, sistem saatinin değeri ile orantılı veri hızını seçer. Seçenekler, CPU'nun saat frekansının (E) 1/16, 1/128, 1/1024, 1/4096 oranlarıdır. Bunlar Çizelge 7'de gösterilmiştir.

Çizelge 7 SCI Bit Zamanları ve Transfer Hızları

SS1	SS0	XTAL	2.4576 MHz	4.0 MHz	4.9152 MHz
		E	614.4 kHz	1.0 MHz	1.2288 MHz
0	0	E ÷ 16	26µs/38,400 Baud	16µs/62,500 Baud	13µs/76,800 Baud
0	1	E ÷ 128	208µs/4,800 Baud	128µs/7812.5 Baud	104.2µs/9,600 Baud
1	0	E ÷ 1024	1.67ms/600 Baud	1.024ms/976.6 Baud	833.3µs/1,200 Baud
1	1	E ÷ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud	3.333ms/300 Baud

Bit 2, Bit 3 Saat Denetimi/Veri Biçimi (CC0)(CC1)

Bu bitler veri şekli ile saat kaynağını kontrol eder. Çizelge 8'de ayrıntılı olarak anlatılmışlardır.

Çizelge 8 SCI Veri Biçimi ve Saat Kaynağı Denetimi

CC1	CC0	Biçim	Saat Kaynağı	Port 2 Bit 2	Port 2 Bit3	Port 2 Bit4
0	0	—	—	—	—	—
0	1	NRZ	İçeride	Kullanılmaz***	**	**
1	0	NRZ	İçeride	Çıkış*	**	**
1	1	NRZ	Dışarıda	Giriş	**	**

* Saat çıkışı RE ve TE bitlerinin değerine bakılmaksızın vardır.

** Bit 3 TRCS'deki RE="1" ise seri veri girişi olarak kullanılır.

Bit 4 TRCS'deki TE="1" ise seri veri çıkışı olarak kullanılır.

*** Bu bağlantı ucu Giriş/Çıkış Port'u olarak kullanılabilir.

1.13.6 İçeride Üretilen Saat

Eğer kullanıcı seri giriş/çıkış için içerideki saati kullanacak olursa aşağıda belirtilen isteklere uyulmalıdır.

- CC1 CCO "10" olmalıdır.
- En yüksek saat hızı E/16 olmalıdır.
- Saat hızı bit hızına eşit olmalıdır.
- RE ve TE bitlerinin etkisi yoktur.

1.13.7 Dışarıda Üretilen Saat

Eğer kullanıcı, seri giriş/çıkış için gerekli saati dışarıda üretmek isterse aşağıda belirtilen şartlara uymak zorundadır.

- CC1 CCO "11"e set edilmelidir.
- Dışarıdaki saatin frekansı, istenen veri transfer hızının 8 katına eşit ve E/2'den küçük olmalıdır.

1.13.8 Seri İletişimin Çalışması

Seri giriş/çıkış donanımı çalıştırılmadan önce yazılım ile bazı ön şartların yerine getirilmesi gerekir. Yazılım sıralaması aşağıda belirtildiği şekilde olmalıdır.

- İstenen çalışma şekline göre veri transfer hızı ve mod kontrol yazmacının kontrol bitleri yazılır.
- İstenen çalışma şekline göre Gönderme/Alma Denetim Durum yazmacının (TRCSR) kontrol bitleri yazılır.

Eger Port 2'deki P23 ve P24 bağlantı uçları seri giriş/çıkış için kullanılırsa TE ve RE bitleri "1"de tutulur. TE ve RE bitleri, SCI'nın çalışması sırasında "0" ve sonra "1" yapılırsa, en az bir bit veri hızı süresince "0"da tutulmalıdır. Eger TE,RE bir bit veri hızı süresince "1"de tutulursa verici ve alıcı ilk koşulları doğrultusunda doğru olarak çalışamaz.

1.13.9 Seri Veri Gönderme İşlemi

Veri iletimine TRCSR'nin TE kontrol biti ile izin verilir. Bu bit "1" yapıldığı zaman Verici Öteleme Yazmacının çıkışı Port 2 Veri Yönü Yazmacı dikkate alınmaksızın çıkış olarak şartlandırılan Port 2'deki P24 bağlantı ucundan dışarı çıkar.

Reset işlemi sonunda kullanıcı istenen çalışma şekli için TRCSR yazmacı ile RMC yazmacının her ikisinde ilk koşullandırılması yapılmalıdır. TE bitinin "1" yapılması 10 tane "1" gönderilmesine neden olur. Bunu takiben iç uygunluk sağlanır ve verici çalışmaya hazır olur. Sonra aşağıda belirtilen iki durumdan birisi meydana gelir.

(1) Eger verici veri yazmacı boş ise (TDRE="1") devamlı "1" gönderilir.

(2) Eger verici veri yazmacı dolu ise (TDRE="0") veri veri öteleme yazmacına transfer edilir ve veri iletimi başlar.

Veri iletimi sırasında, önce başlangıç biti ("0") gönderilir. İkinci olarak 8 bit veri 0.bit'ten başlanarak gönderilir. Son olarak durma biti ("1") gönderilir. Veri gönderme yazmacının içindekiler, çıkış öteleme yazmacına gönderildiği zaman, donanım TDRE bayrağını "1" yapar. Eğer ana işlem birimi (CPU), uygun zamanda bayrağa cevap vermezse TDRE "1"de tutulur ve veri gönderme yazmacına veri yazılincaya kadar devamlı "1" gönderilir.

1.13.10 Seri Veri Alma İşlemi

Seri veri alma işlemine TRCSR'nin RE kontrol biti tarafından izin verilir. Seri giriş Port 2'deki P23 bağlantı ucundan içeri alınabilir. Alma işlemi, RMC ve TRCSR yazmaçları tarafından kontrol edilir. Alınan bit dizisi başlangıç bitiyle ("0") uygunlaştırılır. On (10) bit zamanı sırasında veri yaklaşık olarak her bitin merkezinden alınır. Eğer 10.bit "1" (durma biti) değilse, biçim hatası meydana gelir ve ORFE bayrağı "1" olur.

Eğer 10.bit "1" ise veri Alıcı Veri Yazmacına gönderilir ve RDRF bayrağı "1" olur. Eğer ikinci verinin 10.bit'i alınır ve RDRF hala "1" ise o zaman bir aşırı çalışma hatasının oluştuğunu gösteren ORFE bayrağı "1" olur.

Ver alma yazmacının ana işlem birimi (CPU) tarafından okunmasını takiben RDRF veya ORFE bayraklarına bir cevap olarak durum yazmacının

okunmasından sonra RDRF ve ORFE bayrakları "0" olur.

1.14 RAM Kontrol Yazmacı

Bu yazmaç \$0014 adresine yerleştirilmiştir ve standby RAM belleğin durumu hakkında bilgi verir. Bu yazmaç aşağıda gösterildiği şekildedir.

RAM Denetim Yazmacı

7	6	5	4	3	2	1	0	
STBY PWR	RAME	X	X	X	X	X	X	\$0014

- Bit 0 Kullanılmaz
- Bit 1 Kullanılmaz
- Bit 2 Kullanılmaz
- Bit 3 Kullanılmaz
- Bit 4 Kullanılmaz
- Bit 5 Kullanılmaz
- Bit 6 RAM'e İzin Verme (RAME)

Kullanıcı bu kontrol bitini kullanarak yonga üzerindeki RAM'i devre dışı bırakabilir. RAM'e izin verme biti $\overline{\text{RES}}$ 'in çıkan kenarında "1" yapılır ve RAM'e izin verilir. Yazılım ile "1" veya "0" yapılabilir. Eğer RAME biti "0" yapılırsa içteki RAM'in bulunduğu adresler dışarıdaki adresler olur ve CPU bu adreslerden dışarıdaki bir bellekten veri okuyabilir. Bu bit yonga üzerinde bulunan RAM'deki bilginin korunabilmesi için standby mod başında "0" olmalıdır.

Bit 7 Standby Biti

Bu bit kullanıcı programı tarafından okunabilir veya yazılabilir. Vcc gerilimi değiştiği zaman "0" olur. Standby moda gitmeden önce bu bit normalde program tarafından "1" yapılır. CPU standby moddan çıktığı zaman bu bit kontrol edilmelidir. Eğer "1" ise RAM'deki bilgi standby moda korunmuştur ve geçerlidir.

1.15 Ana İşlem Biriminin (CPU) İşlevleri

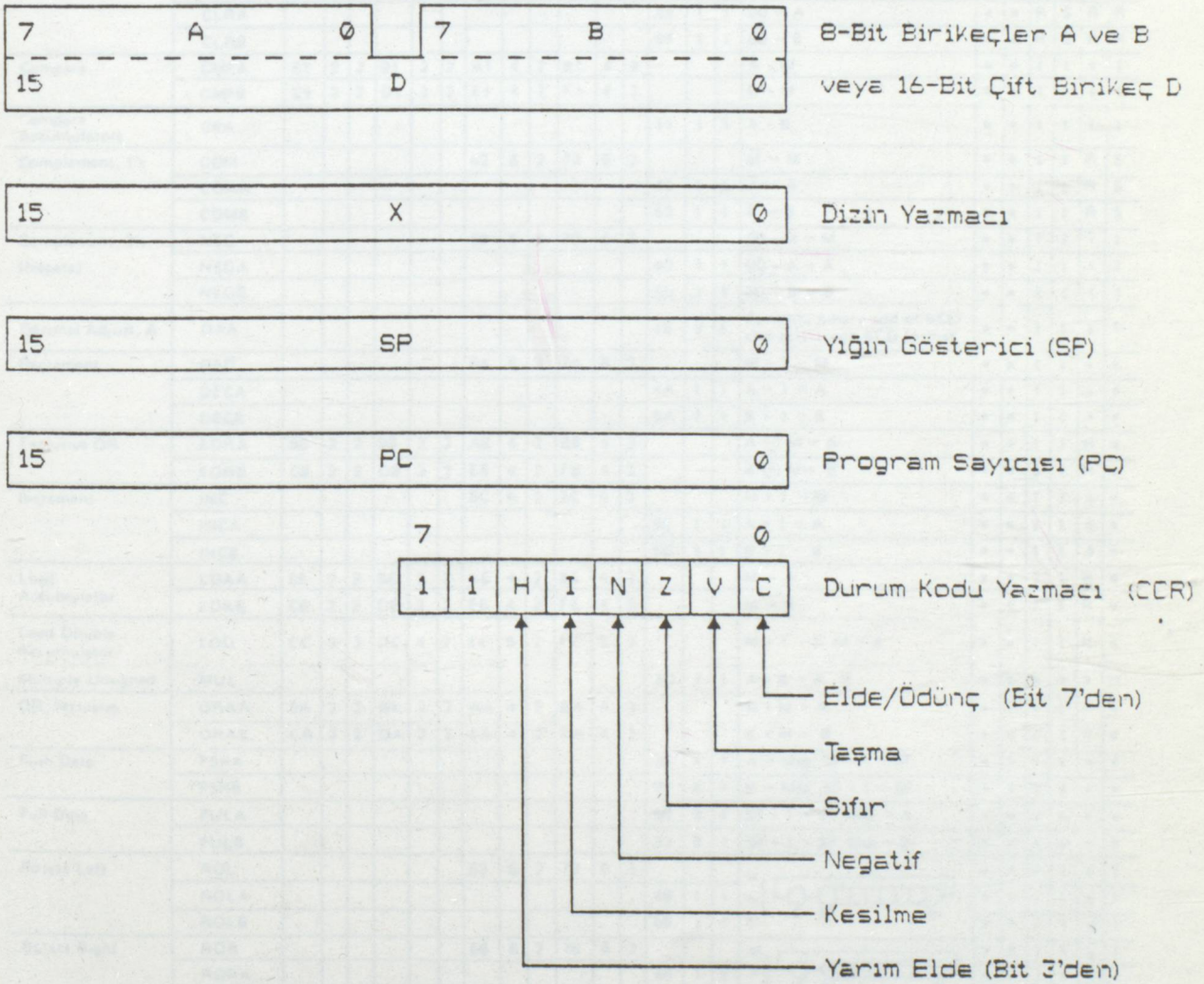
HD63P01M1, HMCS6800 ailesinden HD6801'in bütün komutlarını ve geliştirilmiş yeni komutları içeren bir komut kümesine sahiptir. Anahtar komutun işleme zamanı sistemin işlem hızını yükseltmek için azaltılmıştır. Buna ek olarak bit işlem komutları, dizin yazmacı ile birikeç arasındaki iki yönlü değişme komutu, sleep komutu eklenmiştir. Bu kısım özet olarak aşağıdaki konuları içerir.

- CPU yazmaçları
(Şekil 1.14'e bakınız)
- Birikeç ve bellek işlemleriyle ilgili komutlar
(Çizelge 9'a bakınız)
- Yeni komutlar
- Dizin yazmacı ve Yığın işlemleriyle ilgili komutlar (Çizelge 10'a bakınız)
- Sıçrama ve dallanma komutları
(Çizelge 11'e bakınız)

- Durum kodu yazmacıyla ilgili komutlar
(Çizelge 12'ye bakınız)
- Çalışma kodu (Op-Code) haritası
(Çizelge 13'e bakınız)
- Dönüş-Dönüş (Cycle-by-Cycle) çalışma
(Çizelge 14'e bakınız)

1.16 CPU Yazmaçları

HD63P01M1'deki CPU'da üç tane 16 Bit ve üç tane 8 Bit yazmaç vardır.



Şekil 1.14 Ana İşlem Birimi (CPU) Yazmaçları

Çizelge 9 Birikeç ve bellek işlemleriyle ilgili komutlar

Operations	Mnemonic	Addressing Modes										Boolean/ Arithmetic Operation	Condition Code Register								
		IMMED		DIRECT		INDEX		EXTEND		IMPLIED			5	4	3	2	1	0			
		OP	#	OP	#	OP	#	OP	#	OP	#			H	I	N	Z	V	C		
Add	ADDA	B8	2	2	98	3	2	AB	4	2	BB	4	3	$A + M - A$	1	0	1	1	1	1	
	ADDB	C8	2	2	DB	3	2	EB	4	2	FB	4	3	$B + M - B$	1	0	1	1	1	1	
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3	$A + B + M, M + 1 - A, B$	0	0	1	1	1	1	
Add Accumulators	ABA											1B	1	1	$A + B - A$	1	0	1	1	1	
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3	$A + M + C - A$	1	0	1	1	1	1	
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3	$B + M + C - B$	1	0	1	1	1	1	
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3	$A \cdot M - A$	0	0	1	1	R	0	
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3	$B \cdot M - B$	0	0	1	1	R	0	
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3	$A \cdot M$	0	0	1	1	R	0	
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3	$B \cdot M$	0	0	1	1	R	0	
Clear	CLR							6F	5	2	7F	5	3	$00 - M$	0	0	R	S	R	R	
	CLRA											4F	1	1	$00 - A$	0	0	R	S	R	R
	CLRB											5F	1	1	$00 - B$	0	0	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3	$A - M$	0	0	1	1	1	1	
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3	$B - M$	0	0	1	1	1	1	
Compare Accumulators	CBA												11	1	1	$A - B$	0	0	1	1	1
Complement, 1's	COM							63	6	2	73	6	3	$\bar{M} - M$	0	0	1	1	R	S	
	COMA											43	1	1	$\bar{A} - A$	0	0	1	1	R	S
	COMB											53	1	1	$\bar{B} - B$	0	0	1	1	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3	$00 - M - M$	0	0	1	1	1	2	
	NEGA											40	1	1	$00 - A - A$	0	0	1	1	1	2
	NEGB											50	1	1	$00 - B - B$	0	0	1	1	1	2
Decimal Adjust, A	DAA											19	2	1	Converts binary add of BCD characters into BCD format	0	0	1	1	1	3
Decrement	DEC							6A	6	2	7A	6	3	$M - 1 - M$	0	0	1	1	4	0	
	DECA											4A	1	1	$A - 1 - A$	0	0	1	1	4	0
	DECB											5A	1	1	$B - 1 - B$	0	0	1	1	4	0
Exclusive OR	EORA	B8	2	2	98	3	2	AB	4	2	BB	4	3	$A \oplus M - A$	0	0	1	1	R	0	
	EORB	C8	2	2	DB	3	2	EB	4	2	FB	4	3	$B \oplus M - B$	0	0	1	1	R	0	
Increment	INC							6C	6	2	7C	6	3	$M + 1 - M$	0	0	1	1	5	0	
	INCA											4C	1	1	$A + 1 - A$	0	0	1	1	5	0
	INCB											5C	1	1	$B + 1 - B$	0	0	1	1	5	0
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3	$M - A$	0	0	1	1	R	0	
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3	$M - B$	0	0	1	1	R	0	
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3	$M + 1 - B, M - A$	0	0	1	1	H	0	
Multiply Unsigned	MUL												3D	7	1	$A \times B - A, B$	0	0	0	0	11
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3	$A + M - A$	0	0	1	1	R	0	
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3	$B + M - B$	0	0	1	1	R	0	
Push Data	PSHA												36	4	1	$A - M_{sp}, SP - 1 - SP$	0	0	0	0	0
	PSHB												37	4	1	$B - M_{sp}, SP - 1 - SP$	0	0	0	0	0
Pull Data	PULA												32	3	1	$SP + 1 - SP, M_{sp} - A$	0	0	0	0	0
	PULB												33	3	1	$SP + 1 - SP, M_{sp} - B$	0	0	0	0	0
Rotate Left	ROL							69	6	2	79	6	3					1	1	1	1
	ROLA												49	1	1			1	1	1	1
	ROLB												59	1	1			1	1	1	1
Rotate Right	ROR							66	6	2	76	6	3					1	1	1	1
	RORA												46	1	1			1	1	1	1
	RORB												56	1	1			1	1	1	1

Note) Condition Code Register will be explained in Note of Table 11.

(to be continued)

Çizelge 9 Birikeç ve bellek işlemleriyle ilgili koutlar

Operations	Mnemonic	Addressing Modes										Boolean/ Arithmetic Operation	Condition Code Register																
		IMMED		DIRECT		INDEX		EXTEND		IMPLIED			5	4	3	2	1	0											
		OP	#	OP	#	OP	#	OP	#	OP	#		H	I	N	Z	V	C											
Shift Left Arithmetic	ASL					68	6	2	78	6	3								*	*	1	1	0	1					
	ASLA										48	1	1								*	*	1	1	0	1			
	ASLB											58	1	1								*	*	1	1	0	1		
Double Shift Left, Arithmetic	ASLD												05	1	1		*	*	1	1	0	1							
Shift Right Arithmetic	ASR					67	6	2	77	6	3								*	*	1	1	0	1					
	ASRA											47	1	1								*	*	1	1	0	1		
	ASRB												57	1	1								*	*	1	1	0	1	
Shift Right Logical	LSR					64	6	2	74	6	3								*	*	R	1	0	1					
	LSRA											44	1	1								*	*	R	1	0	1		
	LSRB												54	1	1								*	*	R	1	0	1	
Double Shift Right Logical	LSRD												04	1	1		*	*	R	1	0	1							
Store Accumulator	STAA			97	3	2	A7	4	2	B7	4	3							A - M	*	*	1	1	R	*				
	STAB			D7	3	2	E7	4	2	F7	4	3							B - M	*	*	1	1	R	*				
Store Double Accumulator	STD			DD	4	2	ED	5	2	FD	5	3							A - M B - M + 1	*	*	1	1	R	*				
Subtract	SUBA	B0	2	2	90	3	2	A0	4	2	B0	4	3							A - M - A	*	*	1	1	1	1			
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3							B - M - B	*	*	1	1	1	1			
Double Subtract	SUBD	B3	3	3	93	4	2	A3	5	2	B3	5	3							A - B - M M + 1 - A B	*	*	1	1	1	1			
Subtract Accumulators	SBA												10	1	1							A - B - A	*	*	1	1	1	1	
Subtract With Carry	SBCA	B2	2	2	92	3	2	A2	4	2	B2	4	3							A - M - C - A	*	*	1	1	1	1			
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3							B - M - C - B	*	*	1	1	1	1			
Transfer Accumulators	TAB												16	1	1							A - B	*	*	1	1	R	*	
	TBA													17	1	1							B - A	*	*	1	1	R	*
Test Zero or Minus	TST					6D	4	2	7D	4	3							M - 00	*	*	1	1	R	R					
	TSTA													4D	1	1							A - 00	*	*	1	1	R	R
	TSTB														5D	1	1							B - 00	*	*	1	1	R
And Immediate	AIM			71	6	3	61	7	3										M - IMM - M	*	*	:	:	R	*				
OR Immediate	OIM			72	6	3	62	7	3										M + IMM - M	*	*	:	:	R	*				
EOR Immediate	EIM			75	6	3	65	7	3										M ⊕ IMM - M	*	*	:	:	R	*				
Test Immediate	TIM					7B	4	3	6B	5	3							M - IMM	*	*	:	:	R	*					

Note) Condition Code Register will be explained in Note of Table 11.

Çizelge 10 Dizin Yazmacı, Yağın işlemleriyle ilgili komutlar

Pointer Operations	Mnemonic	Addressing Modes										Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED		DIRECT		INDEX		EXTEND		IMPLIED			5	4	3	2	1	0
		OP	~ #	OP	~ #	OP	~ #	OP	~ #	OP	~ #		H	I	N	Z	V	C
Compare Index Reg	CPX	BC	3 3	9C	4 2	AC	5 2	BC	5 3			X - M M + 1	•	•	•	•	•	•
Decrement Index Reg	DEX									09	1 1	X - 1 - X	•	•	•	•	•	•
Decrement Stack Ptr	DES									34	1 1	SP - 1 - SP	•	•	•	•	•	•
Increment Index Reg	INX									08	1 1	X + 1 - X	•	•	•	•	•	•
Increment Stack Ptr	INS									31	1 1	SP + 1 - SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3 3	DE	4 2	EE	5 2	FE	5 3			M - X _H , (M + 1) - X _L	•	•	•	•	•	•
Load Stack Ptr	LDS	BE	3 3	9E	4 2	AE	5 2	BE	5 3			M - SP _H , (M + 1) - SP _L	•	•	•	•	•	•
Store Index Reg	STX			DF	4 2	EF	5 2	FF	5 3			X _H - M, X _L - (M + 1)	•	•	•	•	•	•
Store Stack Ptr	STS			9F	4 2	AF	5 2	BF	5 3			SP _H - M, SP _L - (M + 1)	•	•	•	•	•	•
Index Reg - Stack Ptr	TXS									35	1 1	X - 1 - SP	•	•	•	•	•	•
Stack Ptr - Index Reg	TSX									30	1 1	SP + 1 - X	•	•	•	•	•	•
Add	ABX									3A	1 1	B + X - X	•	•	•	•	•	•
Push Data	PSHX									3C	5 1	X _L - M _{sp} , SP - 1 - SP X _H - M _{sp} , SP - 1 - SP	•	•	•	•	•	•
Pull Data	PULX									38	4 1	SP + 1 - SP, M _{sp} - X _H SP + 1 - SP, M _{sp} - X _L	•	•	•	•	•	•
Exchange	XGDX									18	2 1	ACCU - IX	•	•	•	•	•	•

Note) Condition Code Register will be explained in Note of Table 11.

Çizelge 11 Sığrama, Dallanma komutları

Operations	Mnemonic	Addressing Modes										Branch Test	Condition Code Register					
		RELATIVE		DIRECT		INDEX		EXTEND		IMPLIED			5	4	3	2	1	0
		OP	~ #	OP	~ #	OP	~ #	OP	~ #	OP	~ #		H	I	N	Z	V	C
Branch Always	BRA	20	3 2									None	•	•	•	•	•	•
Branch Never	BRN	21	3 2									None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3 2									C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3 2									C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3 2									Z = 1	•	•	•	•	•	•
Branch If > Zero	BGE	2C	3 2									$N \odot V = 0$	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3 2									$Z + (N \odot V) = 0$	•	•	•	•	•	•
Branch If Higher	BHI	22	3 2									C + Z = 0	•	•	•	•	•	•
Branch If < Zero	BLE	2F	3 2									$Z + (N \odot V) = 1$	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3 2									C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3 2									$N \odot V = 1$	•	•	•	•	•	•
Branch If Minus	BMI	2B	3 2									N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3 2									Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3 2									V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3 2									V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3 2									N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5 2										•	•	•	•	•	•
Jump	JMP					6E	3 2	7E	3 3				•	•	•	•	•	•
Jump To Subroutine	JSR			9D	5 2	AD	5 2	BD	6 3				•	•	•	•	•	•
No Operation	NOP									01	1 1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI									3B	10 1		•	•	•	•	•	•
Return From Subroutine	RTS									39	5 1		•	•	•	•	•	•
Software Interrupt	SWI									3F	12 1		•	•	•	•	•	•
Wait for Interrupt*	WAI									3E	9 1		•	•	•	•	•	•
Sleep	SLP									1A	4 1		•	•	•	•	•	•

Note) *WAI puts R/ \bar{W} high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 11.

Çizelge 12 Durum Kodu Yaznaçıyla ilgili komutlar

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code Register											
		IMPLIED				5	4	3	2	1	0						
		OP	~	#								H	I	N	Z	V	C
Clear Carry	CLC	0C	1	1	0 - C	R	
Clear Interrupt Mask	CLI	0E	1	1	0 - I	.	R	
Clear Overflow	CLV	0A	1	1	0 - V	R	.	
Set Carry	SEC	0D	1	1	1 - C	S	
Set Interrupt Mask	SEI	0F	1	1	1 - I	.	S	
Set Overflow	SEV	0B	1	1	1 - V	S	.	.	
Accumulator A → CCR	TAP	06	1	1	A → CCR	-----						-----					
CCR → Accumulator A	TPA	07	1	1	CCR → A	

[NOTE 1] Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

- ① (Bit V) Test: Result = 10000000?
- ② (Bit C) Test: Result ≠ 00000000?
- ③ (Bit C) Test: BCD Character of high-order byte greater than 9? (Not cleared if previously set)
- ④ (Bit V) Test: Operand = 10000000 prior to execution?
- ⑤ (Bit V) Test: Operand = 01111111 prior to execution?
- ⑥ (Bit V) Test: Set equal to N=C=1 after the execution of instructions
- ⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
- ⑧ (All Bit) Load Condition Code Register from Stack.
- ⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
- ⑩ (All Bit) Set according to the contents of Accumulator A.
- ⑪ (Bit C) Result of Multiplication Bit 7=1 of ACCB?


[NOTE 2] CLI instructions and interrupt.

If interrupt mask-bit is set ($I=1$) and interrupt is requested ($\overline{IRQ_1} = "0"$ or $\overline{IRQ_2} = "0"$), and then CLI instruction is executed, the CPU responds as follows.

- 1 the next instruction of CLI is one-machine cycle instruction.
Subsequent two instructions are executed before the interrupt is responded.
That is, the next and the next of the next instruction are executed.
- 2 the next instruction of CLI is two-machine cycle (or more) instruction.
Only the next instruction is executed and then the CPU jump to the interrupt routine.
Even if TAP instruction is used, instead of CLI, the same thing occurs.

Çizelge 13 Çalışma Kodu Haritası

OP CODE	i				ACC A	ACC B	IND	EXT DIR	ACCA or SP				ACCB or X					
	HI	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110		1111
LO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0000	0	SBA	BRV	TSX	NEG				SUB								0	
0001	1	NOP	CBA	BRN	INS	AIM				CMP								1
0010	2			BHI	PULA	OIM				SBC								2
0011	3			BLS	PULB	COM				SUBD				ADD				3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS	EIM				BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR				STA				STA				7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI	TIM				ADD								B
1100	C	CLC		BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC		BLT	MUL	TST				BSR	JSR					STD		D
1110	E	CLI		BGT	WAI	JMP				LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR				STS				STX				F

UNDEFINED OP CODE 

* Only for instructions of AIM, OIM, EIM, TIM

Çizelge 14 Dönüş-Dönüş Çalışma

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R· \bar{W}	Data Bus
IMMEDIATE					
ADC ADD	2	1	Op Code Address+1	1	Operand Data
AND BIT		2	Op Code Address+2	1	Next Op Code
CMP EOR					
LDA ORA					
SBC SUB					
ADDD CPX	3	1	Op Code Address+1	1	Operand Data (MSB)
LDD LDS		2	Op Code Address+2	1	Operand Data (LSB)
LDX SUBD		3	Op Code Address+3	1	Next Op Code
DIRECT					
ADC ADD	3	1	Op Code Address+1	1	Address of Operand (LSB)
AND BIT		2	Address of Operand	1	Operand Data
CMP EOR		3	Op Code Address+2	1	Next Op Code
LDA ORA					
SBC SUB					
STA	3	1	Op Code Address+1	1	Destination Address
		2	Destination Address	0	Accumulator Data
		3	Op Code Address+2	1	Next Op Code
ADDD CPX	4	1	Op Code Address+1	1	Address of Operand (LSB)
LDD LDS		2	Address of Operand	1	Operand Data (MSB)
LDX SUBD		3	Address of Operand+1	1	Operand Data (LSB)
		4	Op Code Address+2	1	Next Op Code
STD STS	4	1	Op Code Address+1	1	Destination Address (LSB)
STX		2	Destination Address	0	Register Data (MSB)
		3	Destination Address+1	0	Register Data (LSB)
		4	Op Code Address+2	1	Next Op Code
JSR	5	1	Op Code Address+1	1	Jump Address (LSB)
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer-1	0	Return Address (MSB)
		5	Jump Address	1	First Subroutine Op Code
TIM	4	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Op Code Address+3	1	Next Op Code
AIM EIM OIM	6	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	Address of Operand	0	New Operand Data
		6	Op Code Address+3	1	Next Op Code

- Continued -

Çizelge 14 Dönüş-Dönüş Çalışma

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/W	Data Bus
INDEXED					
JMP	3	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Jump Address	1	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	Offset
2		FFFF	1	Restart Address (LSB)	
3		IX+Offset	1	Operand Data	
4		Op Code Address+2	1	Next Op Code	
STA	4	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	0	Accumulator Data
		4	Op Code Address+2	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	Offset
2		FFFF	1	Restart Address (LSB)	
3		IX+Offset	1	Operand Data (MSB)	
4		IX+Offset+1	1	Operand Data (LSB)	
5		Op Code Address+2	1	Next Op Code	
STD STS STX	5	1	Op Code Address+1	1	Offset
2		FFFF	1	Restart Address (LSB)	
3		IX+Offset	0	Register Data (MSB)	
4		IX+Offset+1	0	Register Data (LSB)	
5		Op Code Address+2	1	Next Op Code	
JSR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer-1	0	Return Address (MSB)
		5	IX+Offset	1	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	Offset
2		FFFF	1	Restart Address (LSB)	
3		IX+Offset	1	Operand Data	
4		FFFF	1	Restart Address (LSB)	
5		IX+Offset	0	New Operand Data	
6		Op Code Address+2	1	Next Op Code	
TIM	5	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Offset
		3	FFFF	1	Restart Address (LSB)
		4	IX+Offset	1	Operand Data
		5	Op Code Address+3	1	Next Op Code
CLR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	IX+Offset	0	00
		5	Op Code Address+2	1	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	Immediate Data
2		Op Code Address+2	1	Offset	
3		FFFF	1	Restart Address (LSB)	
4		IX+Offset	1	Operand Data	
5		FFFF	1	Restart Address (LSB)	
6		IX+Offset	0	New Operand Data	
7		Op Code Address+3	1	Next Op Code	

- Continued -

Çizelge 14 Dönüş-Dönüş Çalışma

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/ \bar{W}	Data Bus
EXTEND					
JMP	3	1	Op Code Address+1	1	Jump Address (MSB)
		2	Op Code Address+2	1	Jump Address (LSB)
		3	Jump Address	1	Next Op Code
ADC ADD TST	4	1	Op Code Address+1	1	Address of Operand (MSB)
AND BIT		2	Op Code Address+2	1	Address of Operand (LSB)
CMP EOR		3	Address of Operand	1	Operand Data
LDA ORA SBC SUB		4	Op Code Address+3	1	Next Op Code
STA	4	1	Op Code Address+1	1	Destination Address (MSB)
		2	Op Code Address+2	1	Destination Address (LSB)
		3	Destination Address	0	Accumulator Data
		4	Op Code Address+3	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data (MSB)
		4	Address of Operand+1	1	Operand Data (LSB)
		5	Op Code Address+3	1	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	Destination Address (MSB)
		2	Op Code Address+2	1	Destination Address (LSB)
		3	Destination Address	0	Register Data (MSB)
		4	Destination Address+1	0	Register Data (LSB)
		5	Op Code Address+3	1	Next Op Code
JSR	6	1	Op Code Address+1	1	Jump Address (MSB)
		2	Op Code Address+2	1	Jump Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	Stack Pointer	0	Return Address (LSB)
		5	Stack Pointer-1	0	Return Address (MSB)
		6	Jump Address	1	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	Address of Operand	0	New Operand Data
		6	Op Code Address+3	1	Next Op Code
CLR	5	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Address of Operand	0	00
		5	Op Code Address+3	1	Next Op Code

- Continued -

Çizelge 14 Dönüş-Dönüş Çalışma

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/ \bar{W}	Data Bus
IMPLIED					
ABA ABX	1	1	Op Code Address + 1	1	Next Op Code
ASL ASLD		2	FFFF	1	Restart Address (LSB)
ASR CBA		3	Stack Pointer - 2	0	Index Register (LSB)
CLC CLI		4	Stack Pointer - 1	0	Index Register (MSB)
CLR CLV		5	Stack Pointer	0	Accumulator A
COM DEC		6	Stack Pointer + 1	0	Accumulator B
DES DEX		7	Stack Pointer + 2	0	Accumulator C
INC INS		8	Stack Pointer + 3	0	Accumulator D
INX LSR		9	Op Code Address + 1	1	Conditional Code Register
LSRD ROL		10	Op Code Address + 1	1	Next Op Code
ROR NOP		11	Stack Pointer + 1	1	Restart Address (LSB)
SBA SEC		12	Stack Pointer + 2	1	Conditional Code Register
SEI SEV		13	Stack Pointer + 3	1	Accumulator A
TAB TAP		14	Stack Pointer	1	Index Register (LSB)
TBA TPA		15	Stack Pointer - 1	1	Index Register (MSB)
TST TSX		16	Stack Pointer - 2	1	Next Op Code
DAA XGDX	2	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Accumulator Data
		4	Op Code Address + 1	1	Next Op Code
PULX	4	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	Data from Stack (MSB)
		4	Stack Pointer + 2	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Index Register (LSB)
		4	Stack Pointer - 1	0	Index Register (MSB)
		5	Op Code Address + 1	1	Next Op Code
RTS	5	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	Return Address (MSB)
		4	Stack Pointer + 2	1	Return Address (LSB)
		5	Return Address	1	First Op Code of Return Routine
MUL	7	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	FFFF	1	Restart Address (LSB)
		5	FFFF	1	Restart Address (LSB)
		6	FFFF	1	Restart Address (LSB)
		7	FFFF	1	Restart Address (LSB)

- Continued -

Çizelge 14 Dönüş-Dönüş Çalışması

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/ \bar{W}	Data Bus
IMPLIED					
WAI	9	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer - 1	0	Return Address (MSB)
		5	Stack Pointer - 2	0	Index Register (LSB)
		6	Stack Pointer - 3	0	Index Register (MSB)
		7	Stack Pointer - 4	0	Accumulator A
		8	Stack Pointer - 5	0	Accumulator B
		9	Stack Pointer - 6	0	Conditional Code Register
RTI	10	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	Conditional Code Register
		4	Stack Pointer + 2	1	Accumulator B
		5	Stack Pointer + 3	1	Accumulator A
		6	Stack Pointer + 4	1	Index Register (MSB)
		7	Stack Pointer + 5	1	Index Register (LSB)
		8	Stack Pointer + 6	1	Return Address (MSB)
		9	Stack Pointer + 7	1	Return Address (LSB)
		10	Return Address	1	First Op Code of Return Routine
SWI	12	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer - 1	0	Return Address (MSB)
		5	Stack Pointer - 2	0	Index Register (LSB)
		6	Stack Pointer - 3	0	Index Register (MSB)
		7	Stack Pointer - 4	0	Accumulator A
		8	Stack Pointer - 5	0	Accumulator B
		9	Stack Pointer - 6	0	Conditional Code Register
		10	Vector Address FFFA	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	First Op Code of SWI Routine
SLP	4	1	Op Code Address + 1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		Sleep			High Impedance-Non MPX Mode Address Bus -MPX Mode
		3	FFFF		Restart Address (LSB)
		4	Op Code Address + 1		Next Op Code

RELATIVE

BCC	BCS	3	1	Op Code Address + 1	1	Branch Offset
BEQ	BGE		2	FFFF	1	Restart Address (LSB)
BGT	BHI		3	{ Branch Address Test = "1" { Op Code Address + 1 Test = "0"	1	First Op Code of Branch Routine Next Op Code
BLE	BLS					
BLT	BMT					
BNE	BPL					
BRA	BRN					
BVC	BVS					
BSR		5	1	Op Code Address + 1	1	Offset
			2	FFFF	1	Restart Address (LSB)
			3	Stack Pointer	0	Return Address (LSB)
			4	Stack Pointer - 1	0	Return Address (MSB)
			5	Branch Address	1	First Op Code of Subroutine

1.16.1 Birikeçler (A,E,D)

İki tane 8 bit yazmaç (ACCA ve ACCB), veri, aritmetik ve mantıksal işlemlerin sonuçlarını saklar. İki yazmaç birleştiği zaman 16 bitlik işlemler için kullanılabilen 16 bit birikeç (ACCD) meydana getirir. ACCA ile ACCB'nin içerikleri bir ACCD çalışması sonunda bozulur.

1.16.2 Dizin Yazmacı (IX)

Dizinli adreslemelerde kullanılmak üzere 16 Bitlik veriyi veya doğrudan 16 bitlik veriyi saklayabilen 16 bitlik bir yazmaçtır.

1.16.3 Yığın Gösterici (SP)

Yığın işleminin yapıldığı adresi saklayan 16 bitlik bir yazmaçtır. Bu yazmaç 16 bitlik veriyi saklamak için kullanılabilir.

1.16.4 Program Sayıcı (PC)

Yürütülmekte olan programın adresini belirleyen 16 Bit'lik bir yazmaçtır. Yazılım ile bu yazmaca erişilemez.

1.16.5 Durum Kodu Yazmacı (CCR)

Bu yazmaç 6. ve 7. bitleri kullanılmayan 16 Bit'lik bir yazmaçtır. Elde (C:carry), taşma (V:overflow), Sıfır (Z:zero), Eksi (N:negatif), kesilmeyi örtme (I:interrupt) ve yarım elde (H:half carry) bitlerinden meydana gelmiştir. Bu bitler, bir komut işlendikten sonra, işlem sonucuna bağlı olarak durumlarını değiştirirler. Çeşitli komutlarla

1.17 degerlendirilebilir ve kontrol edilebilirler.

Bit 5 Yarım Elde (H)

Bir ADD, ABA veya ADC komutu işlendiğinde sonucun 3. bitinden 4.bitine taşma olursa "1" olur. Aksi halde "0" olur.

Bit 4 Kesilmeyi Örtme (I)

Bu bit "1" yapılırsa herhangi bir örtülebilir kesilme ($\overline{IRQ1}, \overline{IRQ2}$) dikkate alınmaz.

Bit 3 Eksi (N)

Bir komut işlendikten sonra sonucun MSB biti "1" ise bu bit "1" olur. Eger MSB "0" ise "0" olur.

Bit 2 Sıfır (Z)

Bir komut işlendikten sonra işlemin sonucu sıfır ise bu bit "1" olur. Aksi halde "0" olur.

Bit 1 Taşma (V)

Bir komut işlendikten sonra, işlemin sonucu 2'nin tamamlayıcısı bir taşma veriyorsa bu bit "1" olur. Taşma yoksa "0" olur.

Bit 0 Elde (C)

Bir komut işlendikten sonra eger MSB'den bir ödünç veya elde alınırsa bu bit "1" olur. Aksi halde "0" olur.

1.17 Adresleme Modları

HD63P01M1'in 7 adresleme modu vardır. Bu adresleme modlarının en iyi şekilde kullanılabilmesi etkili bir programlamayı gerektirir.

1.17.1 Birikeç (ACCX) Adresleme

Sadece bir birikeç adreslenir ve birikeç A veya B seçilir. Bu adresleme modundaki komut 1 Bayt'dır.

1.17.2 Hemen (immediate) Adresleme

Bu adresleme modunda komutun ikinci baytındaki veri değerlendirilir. Özel bir durum olarak LDX ve LDS komutlarında, komutun ikinci ve üçüncü baytlarındaki veri alınır. CPU bu adresleme modunda içine adres bilgisi gönderir. Bu adresleme modundaki komut 2 veya 3 Bayt'dır.

1.17.3 Doğrudan Adresleme

Doğrudan adresleme modunda komutun ikinci baytı verinin bulunduğu belleğin adresi gösterir. 256 Bayt (0'dan 255'e kadar) bellek doğrudan adreslenebilir. Bu nedenle tasarlanan sistemlerde bu bölgenin RAM seçilmesi önerilir. Bu adresleme modundaki AIM, OIM, EIM ve TIM komutları 3 Bayt'lık komutlardır. Diğer komutlar ise 2 Bayt'lıktır.

1.17.4 Genişletilmiş Doğrudan Adresleme

Bu adresleme modundaki komutun ikinci baytı verinin bulunduğu adresin üst baytını (A8-A15), üçüncü bayt ise adresin alt baytını (A0-A7) gösterir. Bu adresleme modunda komut 3 Bayt'lıktır.

1.17.5 Dizinlenmiş Adresleme

Bu adresleme modunda komutun ikinci baytı ile dizin yazmacının alt baytı toplanır. Özel bir durum olarak AIM, OIM, EIM ve TIM komutlarında komutun

üçüncü baytı ile dizin yazmacının alt baytı toplanır. Sonuç, dizin yazmacının üst baytı ile birlikte dikkate alınır ve bellek adresi belirlenir. Şekillenen adres geçici adres yazmacında saklanarak dizin yazmacının içeriği bozulmaz. Bu adresleme modunda AIM, OIM, EIM ve TIM 3 Bayt'lık, bunların dışındaki komutlar ise 2 Bayt'lık komutlardır.

1.17.6 İçerik (implied) Yoluyla Adresleme

Bu adresleme modundaki komut, kendisi adresi belirleyerek çalışır. Bu adresleme modunda komut 1 Bayt'lıktır.

1.17.7 Bağlı Adresleme

Bu adresleme modunda komutun ikinci baytı program sayacının alt baytı ile toplanır. Elde veya ödünç dikkate alınarak program sayacının üst baytına etki ettirilir. Böylece komutun bulunduğu adresten 126 gerideki veya 129 ilerideki adrese gidilebilir. Bu adresleme modundaki komut 2 Bayt'lıktır.

1.18 Yeni Komutlar

HD6801'in komut tablosuna ek olarak HD63P01M1'in komut tablosunda aşağıda açıklanan komutlar bulunur.

AIM (M)•(imm)→(M)

Immediate veri ile bellek içeriğini "AND" işlemine sokar ve işlem sonucunu yine aynı belleğe kaydeder.

OIM (M)+(imm)→(M)

Immediate veri ile bellek içeriğini "OR"

işlemine sokar ve işlem sonucunu yine aynı belleğe kaydeder.

EIM (M)⊕(imm)→(M)

Immediate veri ile bellek içeriğini "EXOR" işlemine sokar ve işlem sonucunu yine aynı belleğe kaydeder.

TIM (M)·(imm)

Immediate veri ile bellek içeriğini "AND" işlemine sokar ve işlem sonucunu durum kodu yazmacının ilgili bitini değiştirerek bildirir.

Bu üç baytlık komutlarda birinci bayt çalışma kodunu, ikinci bayt immediate veriyi ve üçüncü bayt ise adresi belirler.

XGDX (ACCD)↔(IX)

Dizin yazmacı ile birikecin içeriklerini iki yönlü olarak değiştirir.

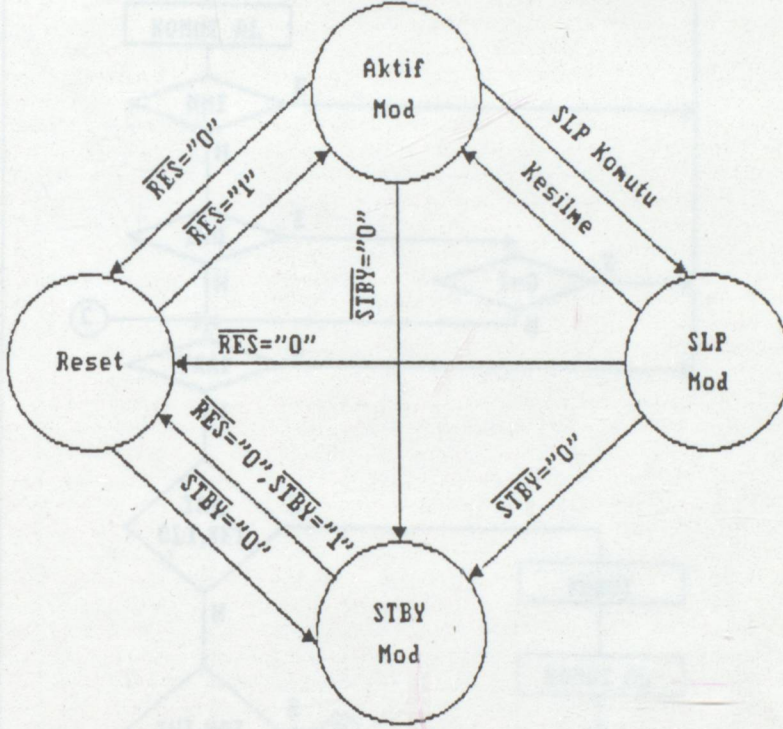
SLP Sleep

Bu komut ileride ayrıntılı bir şekilde anlatılacaktır.

1.19 Ana İşlem Biriminin (CPU) Çalışması

CPU bellekten bir komut alır ve istenen çalışmayı sağlar. Bu işlem sırası reset (RES) ile başlar ve özel bir kontrol işareti tarafından kesilmediği sürece çalışmasına sınırsız olarak devam eder. SWI, RTI, WAI ve SLP komutları, NMI, IRQ1, IRQ2 ve STBY donanım kontrolleri CPU'nun çalışmasını etkiler. Şekil 1.15'de CPU çalışma modu geçiş

diyagramı ve Şekil 1.16'da ise sistem akış diyagramı gösterilmiştir.



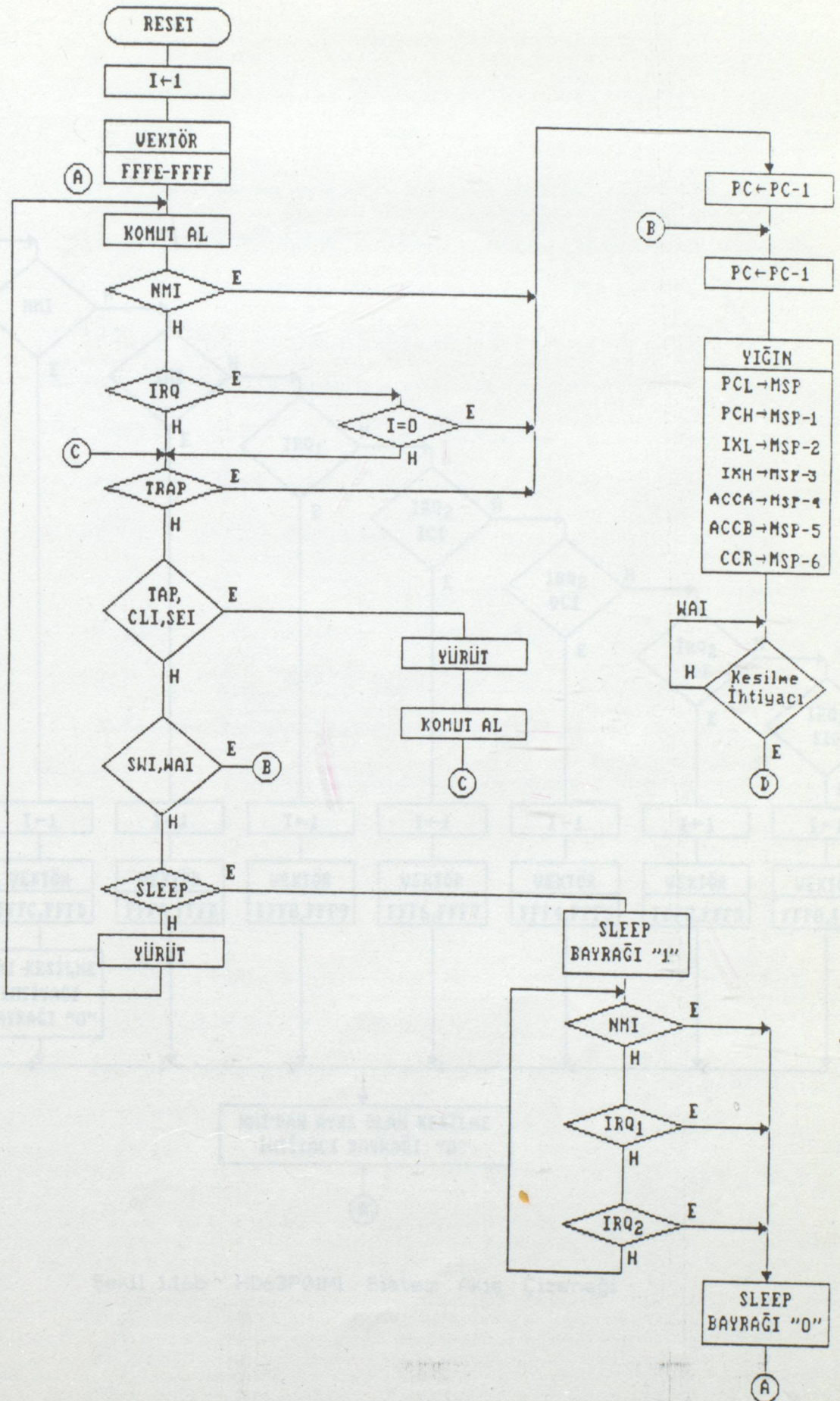
Şekil 1.15 Aktif mod, Standby mod, Sleep mod ve Reset arasındaki geçişler.

1.20 Düşük Güçle Çalışma Modları

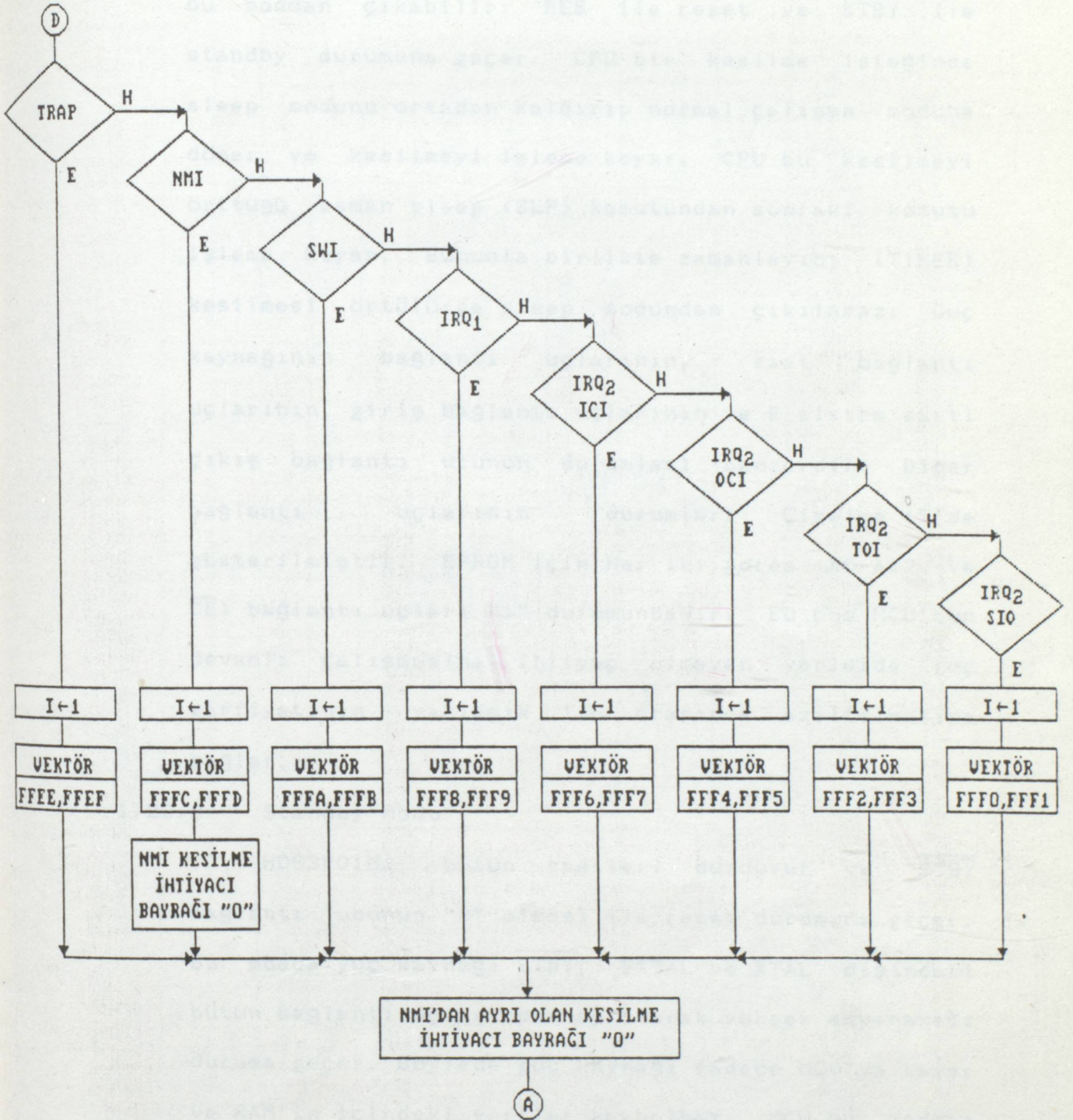
HD63P01M1 sleep ve standby modları şeklinde adlandırılan iki düşük güçle çalışma şekline sahiptir.

1.20.1 Sleep Modu

SLP komutunun işlenmesiyle MCU sleep moduna girer. Sleep modunda yazmaçların içerikleri aynı kalırken CPU normal çalışmasını durdurur. Bu modda SCI ve TIMER gibi CPU birimleri dışındaki çevre birimleri çalışmalarına devam eder.



Şekil 1.16a HD63P01M1 Sistem Akış Çizeneği



Şekil 1.16b HD63P01M1 Sistem Akış Çizeneği

MCU reset (\overline{RES}), \overline{STBY} veya bir kesilme etkisiyle bu moddan çıkabilir. \overline{RES} ile reset ve \overline{STBY} ile standby durumuna geçer. CPU bir kesilme isteğinde sleep modunu ortadan kaldırıp normal çalışma moduna döner ve kesilmeyi işleme koyar. CPU bu kesilmeyi örttüğü zaman sleep (SLP) komutundan sonraki komutu işleme koyar. Bununla birlikte zamanlayıcı (TIMER) kesilmesi örtülürse sleep modundan çıkılamaz. Güç kaynağının bağlantı uçlarının, saat bağlantı uçlarının, giriş bağlantı uçlarının ve E sistem saati çıkış bağlantı ucunun durumları benzerdir. Diğer bağlantı uçlarının durumları Çizelge 15'de gösterilmiştir. EPROM için her iki adres (A0-A12 ve \overline{CE}) bağlantı uçları "1" durumundadır. Bu mod MCU'nun devamlı çalışmasına ihtiyaç olmayan yerlerde güç sarfiyatının yaklaşık 1/6 oranında azaltılmasını sağlar.

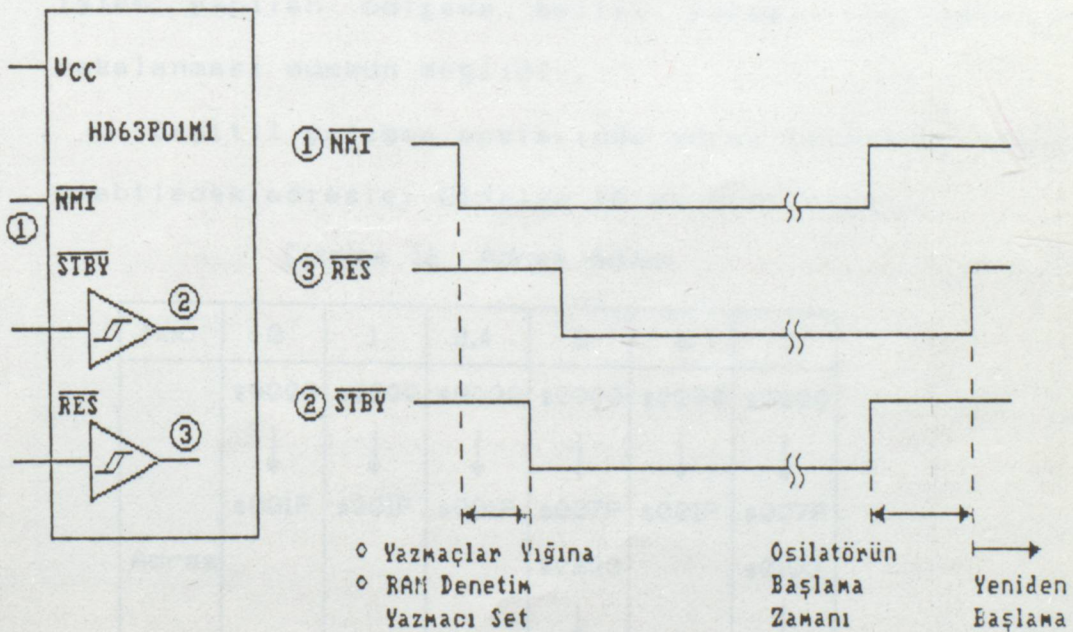
1.20.2 Standby Modu

HD63P01M1 bütün saatleri durdurur ve \overline{STBY} bağlantı ucunun "0" olması ile reset durumuna geçer. Bu modda güç kaynağı \overline{STBY} , EXTAL ve XTAL dışındaki bütün bağlantı uçlarından ayrılarak yüksek empedanslı duruma geçer. Böylece güç kaynağı sadece MCU'ya kalır ve RAM'in içindeki veriler kaybolmaz. MCU bu moddan sadece reset ile çıkabilir.

Çizelge 15 Sleep modunda bağlantı uçlarının durumu

Bağlantı ucu	Mod	0	1	2,4	5	6	7
Port 1 P10~P17	İşlev	I/O Port	Alt Adres Yolu	I/O Port	←	←	←
	Durum	Öncekini korur	Çıkış "1"	Öncekini korur	←	←	←
Port 2 P20~P24	İşlev	I/O Port	←	←	←	←	←
	Durum	Öncekini korur	←	←	←	←	←
Port 3 P30~P37	İşlev	\bar{E} : Alt Adres Yolu E : Veri Yolu	Veri Yolu	\bar{E} : Alt Adres Yolu E : Veri Yolu	Veri Yolu	\bar{E} : Alt Adres Yolu E : Veri Yolu	I/O Port
	Durum	\bar{E} : Çıkış "1" E : Yüksek Direnç	Yüksek Direnç	\bar{E} : Çıkış "1" E : Yüksek Direnç	Yüksek Direnç	\bar{E} : Çıkış "1" E : Yüksek Direnç	Öncekini korur
Port 4 P40~P40	İşlev	Üst Adres Yolu	←	←	Alt Adres Yolu veya Giriş Portu	Üst Adres Yolu veya Giriş Portu	I/O Port
	Durum	Çıkış "1"	←	←	Adres Yolu : Çıkış "1" Port : Öncekini korur	←	Öncekini korur
SC2		Çıkış "1" (Okuma durumu)	←	←	←	←	Çıkış "1"
SC1		Adres Çıkışı Belirteci	←	←	Çıkış "1"	Adres Çıkışı Belirteci	Giriş Bağlantı ucu

İlk olarak CPU bilgisiyle RAM üzerindeki SP'nin içindeki bilgiler NMI aracılığıyla korunur. Sonra RAM kontrol yazmacının RAME biti "0" yapılır ve STBY power biti "1" yapılarak standby moda girilir. Eğer restartda STBY power biti hala "1" ise gücün MCU'ya beslenmiştir ve RAM'in içindeki veriler kaybolmamıştır. Böylece sistem ilk bilgilerini SP aracılığıyla CPU'ya döndürerek kendisini yeniler. Bu moda sadece güç kaynağı ve STBY bağlantı uçları aktiftir. XTAL bağlantı ucunun durumu içeride belirlenmiştir. Böylece MCU bağlantı uçlarının durumlarından etkilenmez. XTAL çıkışta "1" olur. Diğer bütün bağlantı uçları yüksek empedans gösterir. EPROM için adres (A0-A12) ve CE bağlantı uçları "1" olur. Standby mod zamanlaması Şekil 1.17'de gösterilmiştir.



Şekil 1.17 Standby Mod Zamanlaması

1.21 Hata İşleme

CPU bellek olmayan bir bölgede işlem yaptığında veya belirsiz bir çalışma kodu aldığı anda en büyük öncelikli kesilmeyi (TRAP) işleme koyar. TRAP, sistemi program hatalarından veya gürültünün neden olabileceği bozulmalardan korur.

1.21.1 Çalışma Kodu Hatası

Belirsiz bir çalışma kodu alındığında CPU normal bir kesilme kadar iyi olan TRAP kesilmesini işletir. Önce CPU yazmaçları korunur ve sonra \$FFEE, \$FFEF adreslerinden bir vektör alınır.

1.21.2 Adres Hatası

Sistemdeki veya dışarıdaki bellek bölgesinin dışında işlem yapan bir komut alındığında, MCU çalışma kodu hatasında olduğu gibi kesilme üretir. Bununla birlikte çalışma modu ile belirlendiği halde işlem yapılan bölgede bellek yoksa, bu hatanın yakalanması mümkün değildir.

Çeşitli çalışma modlarında adres hatasına neden olabilecek adresler Çizelge 16'da verilmiştir.

Çizelge 16 Adres Hatası

Mod	0	1	2,4	5	6	7
	\$0000	\$0000	\$0000	\$0000	\$0000	\$0000
	↓	↓	↓	↓	↓	↓
Adres	\$001F	\$001F	\$001F	\$007F	\$001F	\$007F
				\$0200		\$0100
				↓		↓
				\$DFFF		\$DFFF

1.22 HD6301V1'in HD63P01M1 ile Yapılan Geliştirmeye Uyarlatabilmesi için Dikkat Edilmesi Gereken Durumlar

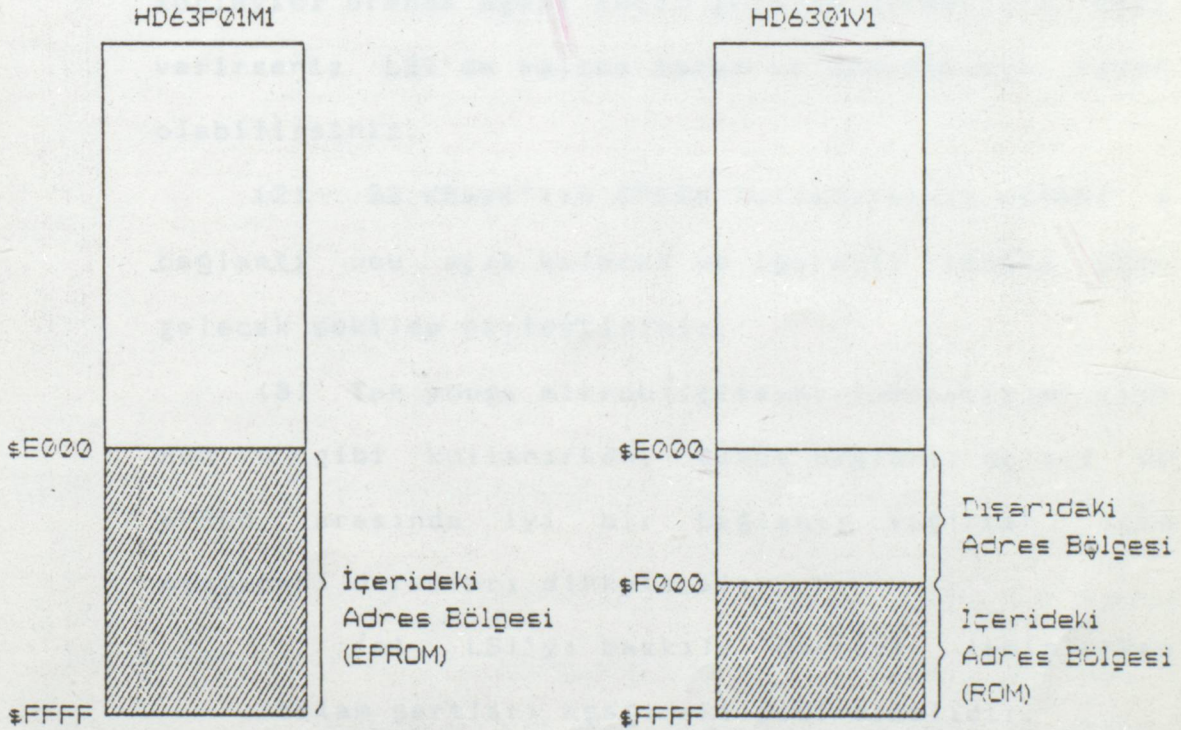
HD63P01M1'deki EPROM bellek haritasında \$E000 ile \$FFFF arasına yerleştirilen 8 KBayt'lık bir yer kaplar. HD6301V1'deki 4 KBayt'lık ROM'un HD63P01 ile rekabet edebilmesi için aşağıdaki şartlara dikkat edilmelidir.

(1) Mod 5 (Genişletilmiş Tekdüzeyle Çalışma) ve Mod 7 (Tek Yonga Çalışma)

\$F000 ile \$FFFF arasına yerleştirilen EPROM'u kullanınız.

(2) Mod 6 (Genişletilmiş Çokdüzeyle Çalışma)

EPROM için yine \$F000 ile \$FFFF arasındaki bellek bölgesini kullanınız. Bununla birlikte \$E000



Şekil 1.18 Mod 6'da Bellek haritası

ile #EFFF arasındaki bellek bölgesini kullanmayınız. Çünkü bu bellek bölgesi HD63P01M1'de — ieride, HD6301V1'de ise dıřarıda adreslenmiřtir. Bu durum Őekil 1.18'de gsterilmiřtir.

(3) Mod 1,2,4

ROM bu modlarda dıřarıda adreslendiđi iin dikkate alınacak bir durum yoktur.

1.23 Yonga zerinde EPROM Kullanılırken Dikkat Edilmesi Gereken Durumlar

Yonga zerinde kullanılan EPROM'un takılabileceđi zel bir soket yapısı olduđundan ařađıda belirtile durumlara dikkat edilmelidir.

(1) LSI bađlantı uđları kadar iyi olan soket bađlantı uđlarına kesinlikle deđiřen gerilim veya zorlayıcı oranda ařırı sabit gerilim vermeyiniz. Eger verirseniz LSI'da kalıcı zarar ve bozulmalara neden olabilirsiniz.

(2) 32 KBayt'lık EPROM kullanırsanız steki 4 bađlantı ucu aık kalacak ve iřaretili kenarı ste gelecek Őekilde yerleřtiriniz.

(3) Tek yonga mikrobilgisayar tmdevresini mask ROM'lu gibi kullanırken, soket bađlantı uđları ve EPROM arasında iyi bir bađlantı sađlamak iin ařađıdaki durumları dikkate alın.

(a) LSI'yı baskılı devreye lehimlerken ortam Őartları ařađıdaki gibi olmalıdır.

Sıcaklık 250 C'den az olmalıdır.

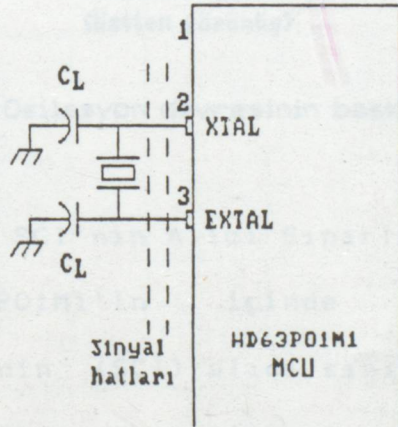
Zaman 10 saniye içinde kalmalıdır.

(b) Lehimleme işleminden sonra baskılı devreyi çeşitli maddelerle kaplarken veya yıkarken soketin içine bağlantıyı etkilecek şekilde sıvı kaçmamasına dikkat ediniz.

(c) Titreşimli yerlerde, titreşimi önleyecek önlemler alınız.

(d) Tekrar tekrar takılıp sökülme işlemleri soketin bağlantı uçlarında gevşeklik meydana getirir. Böyle durumlarda yeni bir tane kullanılması önerilir.

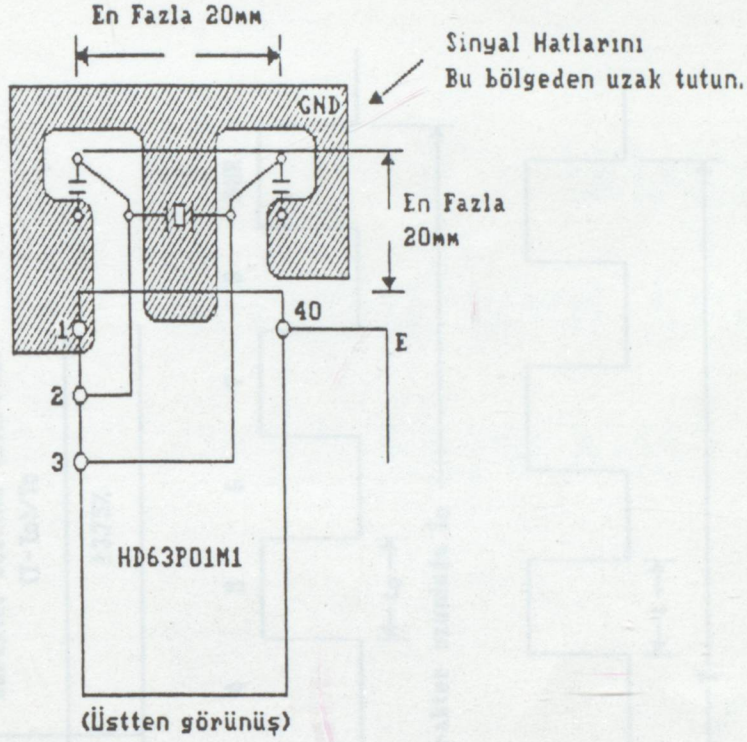
1.24 Osilasyon Devresinin Baskılı Devre Üzerine Kurulurken Dikkat Edilmesi Gereken Durumlar



Şekil 1.19 Osilasyon devresinin baskılı devresini bu şekilde tasarlamayınız.

Şekil 1.19'da gösterildiği gibi işaret hatları osilasyon devresi yoluna sokulursa, normal osilasyon bozulabilir. Kristal ile CL, HD63P01M1'e mümkün olduğu kadar yakın bağlanmalıdır.

Sekil 1.20'de osilatör devresi için bir baskılı devre örneği verilmiştir.



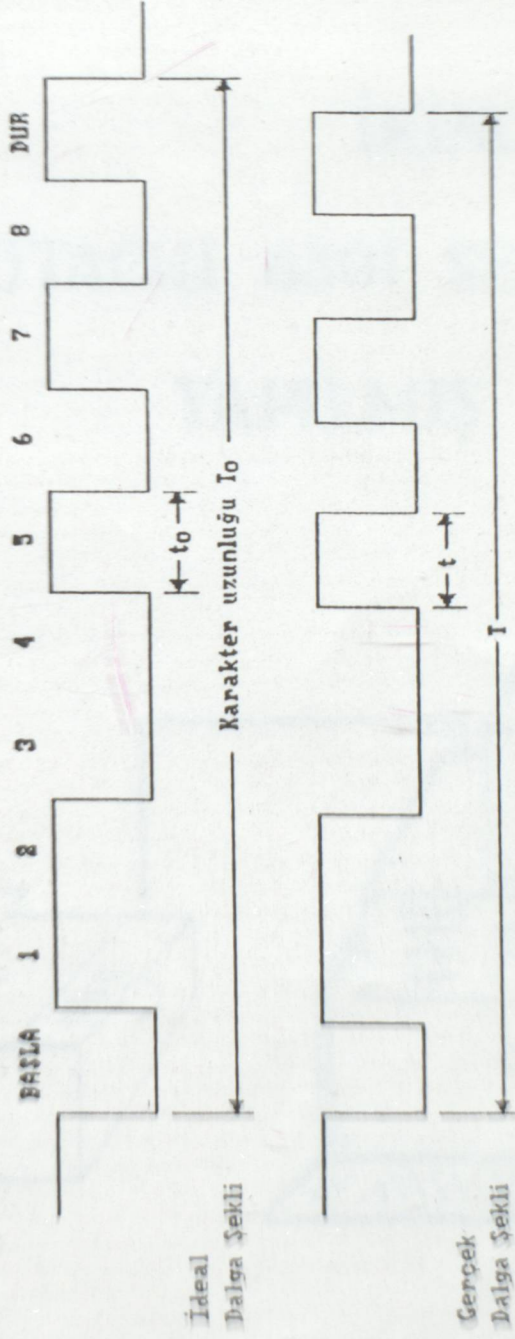
Sekil 1.20 Osilasyon devresinin baskılı devresinin tasarım örneği.

1.25 SCI'nin Alıcı Sınırları

HD63P01M1'in içinde bulunan seri iletişim arabiriminin (SCI) alıcı sınırları ve zamanlama diyagramı Çizelge 17'de gösterilmiştir.

Çizelge 17

Bit bozulma toleransı ($t-t_0$)/ t_0	Karakter bozulma toleransı ($I-I_0$)/ I_0
±25%	±3.75%



2.3 GİRİŞ

Bu çevirici IBM uyumlu kişisel bilgisayarlar ve sinibilgisayarlar üzerinde çalışır.

Temel çevirici programı Hitachi 6301 mikroçipli geliştirme sisteminin çevirici programıyla uyumlu.

Çevirici ana dizini, çevirici programı ve Hitachi 6301 mikroçipli geliştirme sistemi için gerekli programları içerir.

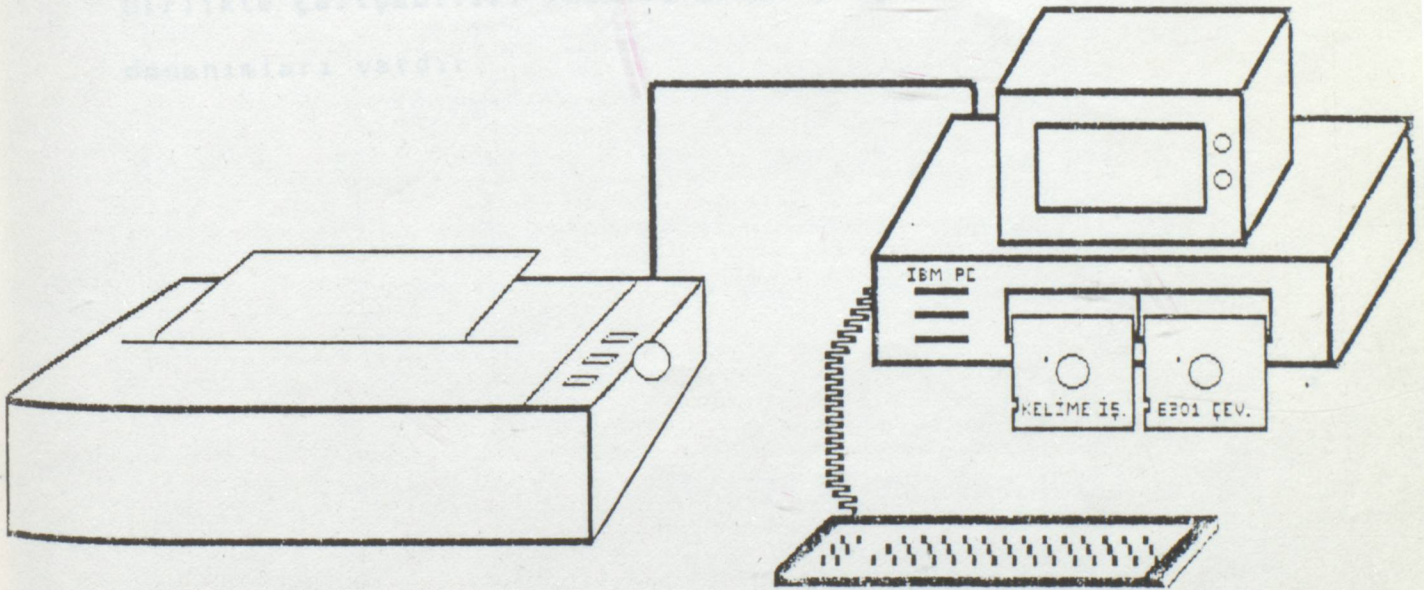
Hitachi 6301 mikroçipli geliştirme sistemi için gerekli programları içerir. Bu konunun detayları için Hitachi 6301 mikroçipli geliştirme sisteminin kullanım kılavuzuna bakınız.

Bu çevirici kendine özgü bir yazılım paketine sahiptir. Bu yazılım paketini kullanmak için gerekli yazılım paketini satın almanız gerekmektedir. Aynı zamanda Hitachi 6301 mikroçipli geliştirme sistemi için gerekli programları da satın almanız gerekmektedir.

Bu çevirici Hitachi 6301 mikroçipli geliştirme sistemi için gerekli programları içerir. Bu konunun detayları için Hitachi 6301 mikroçipli geliştirme sisteminin kullanım kılavuzuna bakınız.

Bu çevirici Hitachi 6301 mikroçipli geliştirme sistemi için gerekli programları içerir. Bu konunun detayları için Hitachi 6301 mikroçipli geliştirme sisteminin kullanım kılavuzuna bakınız.

İKİNCİ BÖLÜM HITACHI 6301 SERİSİ İÇİN YAPILMIŞ ÇEVİRİCİ



2.1 GİRİŞ

Bu çevirici IBM, IBM uyumlu kişisel bilgisayarlar ve minibilgisayarlar üzerinde çalışır.

Temel çevirici programı Hitachi 6301 mikroişleyici geliştirme sisteminin çevirici programıyla uygunluk gösterir. Çevirici söz dizimi, yerleşim düzeni ve makine kodu üretimi Hitachi 6301 çevirici ile bağdaştırılmıştır. Komut belirleyici singeler ve adresleme modlarının yazılımı bu konunun kapsamı içindedir.

Bu çevirici kendine özgü komutlarıyla yüksek seviyeli sayılabilecek bir dil kullanabilmekte ve bu dilden makine diline geçebilmektedir. Yine Hitachi firmasının üretimi olan 6301 mikroişleyici geliştirme seti ve donanımıyla birlikte çalışabilir. PROM ve EPROM programlama yazılım ve donanımları vardır.

2.2 HD63P01M1'in Çevirici Dili (Assembly Language)

Çevirici, kullanıcı ile işlem sahası arasında ilişki kurarak adresleme modunu belirler. Kullanışlı 7 ayrı adresleme modu vardır.

- (1) Birikeç adresleme
- (2) İçerik yoluyla adresleme
- (3) Hemen adresleme
- (4) Doğrudan adresleme
- (5) Genişletilmiş doğrudan adresleme
- (6) Dizinlenmiş adresleme
- (7) Bağlı adresleme

Adresleme modları hakkında ayrıntıya girilmeden önce iki ayrı işleme gerek duyan komutlar konusuna değinmek yararlı olacaktır. Bu 8 komut AIM, OIM, EIM, TIM, BCLR, BSET, BTGL ve BTST komutlarıdır.

Birinci işlem AIM, OIM ile TIM komutlarının verisini ve BTGL, BSET, BCLR ile BTST komutlarının ise bit işlemi için gerekli olan bit değerini içeri almaktır. İkinci işlem dizinlenmiş veya doğrudan adreslenmiş bellek adresini belirler.

2.2.1 Birikeç Adresleme

Birikeç A veya B üzerinde işlem yapan 13 komut vardır. Bunlar ASL, ASR, CLR, COM, DEC, INC, LSR, NEG, PSH, PUL, ROL, ROR ve TST komutlarıdır. Çevirici aracılığıyla birikeç adresleme modundaki komutlardan her biri 1 Bayt'lık makine koduna dönüştürülür.

Örnek :

İfade	Makine Kodu (Hexadecimal)		
	1.Bayt	2.Bayt	3.Bayt
ASL A veya ASLA	4E	--	--
ASR B veya ASRB	57	--	--

2.2.2 İçerik Yoluyla Adresleme

Bu modda işlem sahasındaki komut, işleme gerek duyulmadan çalışmanın clear-cut olması için adrese izin verir. İçerik yoluyla adreslenebilen 31 komut vardır. Bunlar ABA, ABX, ASLD, CBA, CLC, CLI, CLV, DAA, DES, DEX, INS, INX, LSRL, MUL, NOP, PSHX, PULX, RTI, RTS, SBA, SEC, SEI, SEV, SWI, TAB, TAP, TBA, TPA, TSX, TXS ve WAI komutlarıdır. Çevirici aracılığıyla her komut 1 Bayt'lık makine koduna dönüştürülür.

2.2.3 Hemen Adresleme

Hemen adreslenebilen 16 komut vardır. Bunlar ADC, ADD, AND, BIT, CPM, CPX, EOR, LDA, LDS, LIX, ORA, SBC, SUB, LDD, ADDD ve SUBD komutlarıdır.

İşlem sahası # işareti ile başlar. Bunu verinin tipini belirten özel işaretler izler. Herhangi bir durumda çevirici, özel olarak ayrılmış bellekte bu verileri ikili (binary) olarak saklar. Sonuçta CPX, LDS, LDX, LDD, ACDD ve SUBD komutlarında veri 0 ile 65535 arasında, diğer komutlarda ise veri 0 ile 255 arasında olabilir.

#'C (C:ASCII karakter) , gösterimi # işaretini izleyen karakterin 7 bit ASCII koduna çevrildiğini

gösterir. #'C genel şeklinin CPX, LDS ve LDX komutlarıyla kullanılması önerilmez. Eger kullanılırsa karakterin ASCII kodu üçüncü bayta yazılır. Çevirici hemen adresleme modundaki CPX, LDS, LDX, LDD, ADDD ve SUBD komutlarını 3 Bayt'lık, diğer komutları ise 2 Bayt'lık makine koduna dönüştürür. Şekil 2.1'de bu adresleme modunun bilgi akış diyagramı verilmiştir.

Örnek :

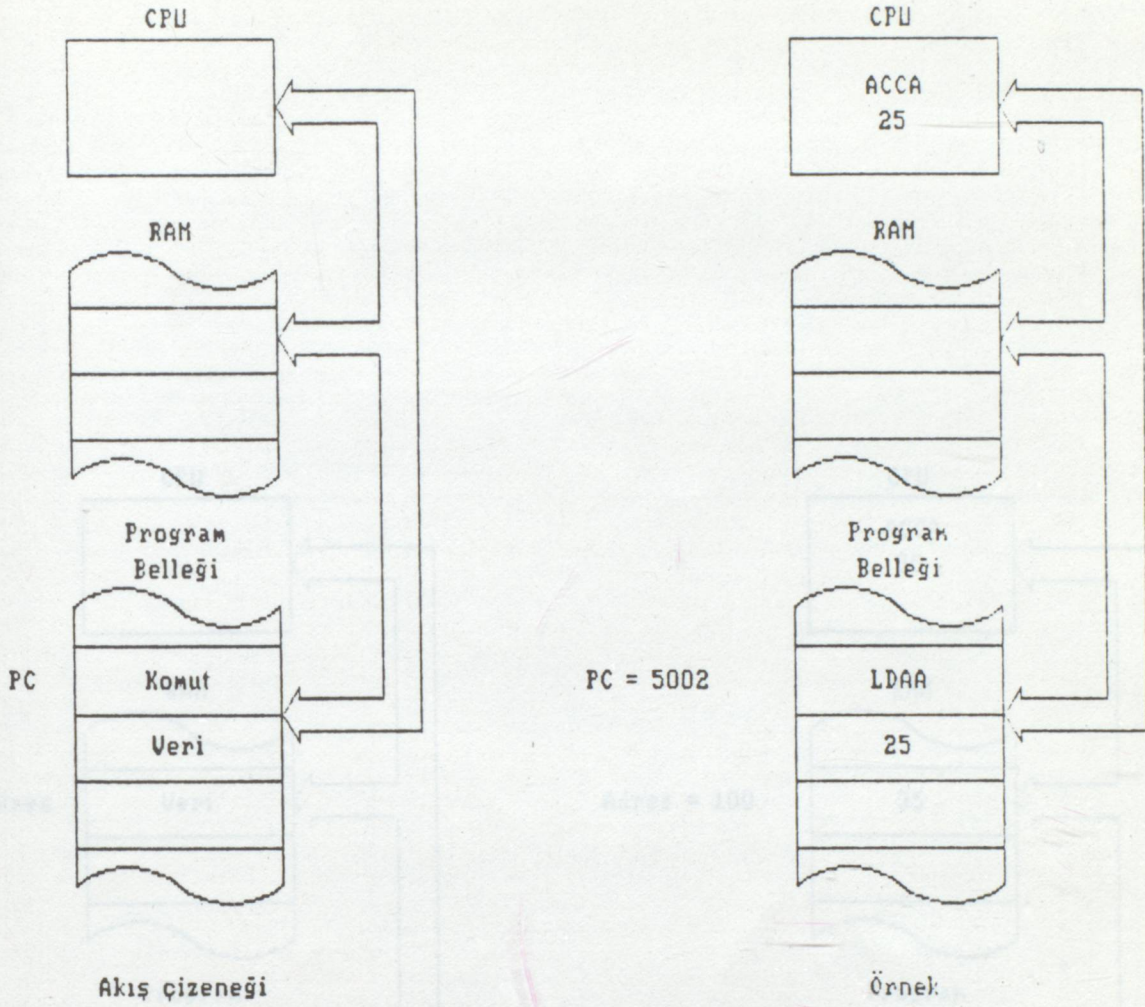
İfade	Makine kodu (Hexadecimal)		
	ETİKET= 100		
	1.Bayt	2.Bayt	3.Bayt
LDA #25	86	19	--
LDA #ETİKET	86	64	--
LDA #ETİKET+25	86	7D	--
LDA #'A	86	41	--
CPX #256	8C	01	00

2.2.4 Doğrudan Adresleme

Doğrudan adresleme modundaki komut, çevirici tarafından 2 Bayt'lık makine koduna dönüştürülür. İkinci bayt dönüşümden sonra 8 Bit'lik adresi gösterir.

Genişletilmiş doğrudan adreslemede komut, çevirici tarafından 3 Bayt'lık makine koduna çevrilir. İkinci bayt adresin üst baytını, üçüncü bayt ise adresin alt baytını gösterir.

Çevirici, 0 ile 255 arasındaki bellek adreslerinde doğrudan adreslemeyi, 255'den büyük bellek adreslerinde

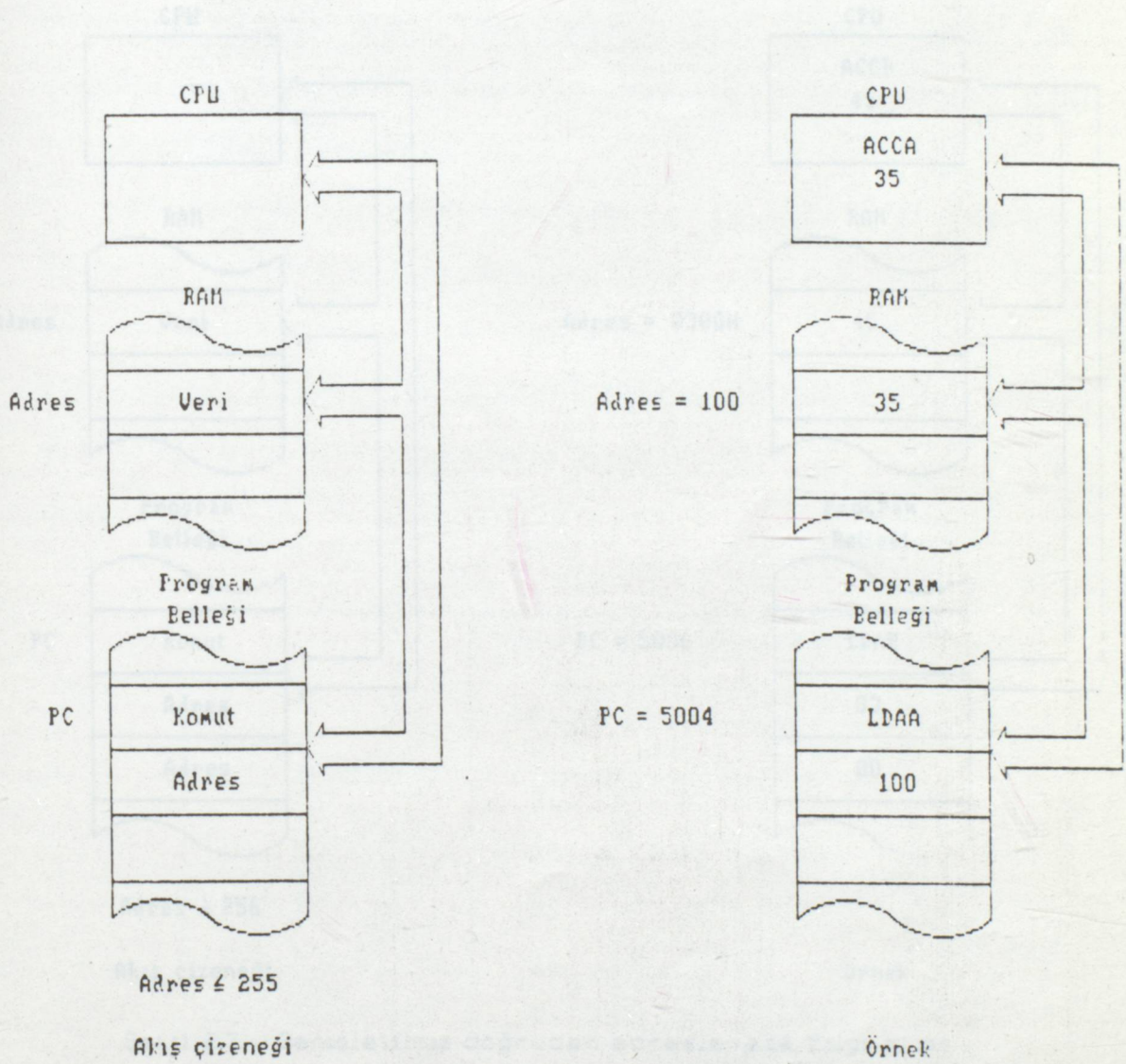


Şekil 2.1 Hemen adreslemede bilgi akışı

ise genişletilmiş doğrudan adreslemeyi kendiliginden seçerek kullanıcıya kolaylık sağlar. Şekil 2.2'de doğrudan adreslemenin, Şekil 2.3'de ise genişletilmiş doğrudan adreslemenin bilgi akış diyagramı verilmiştir.

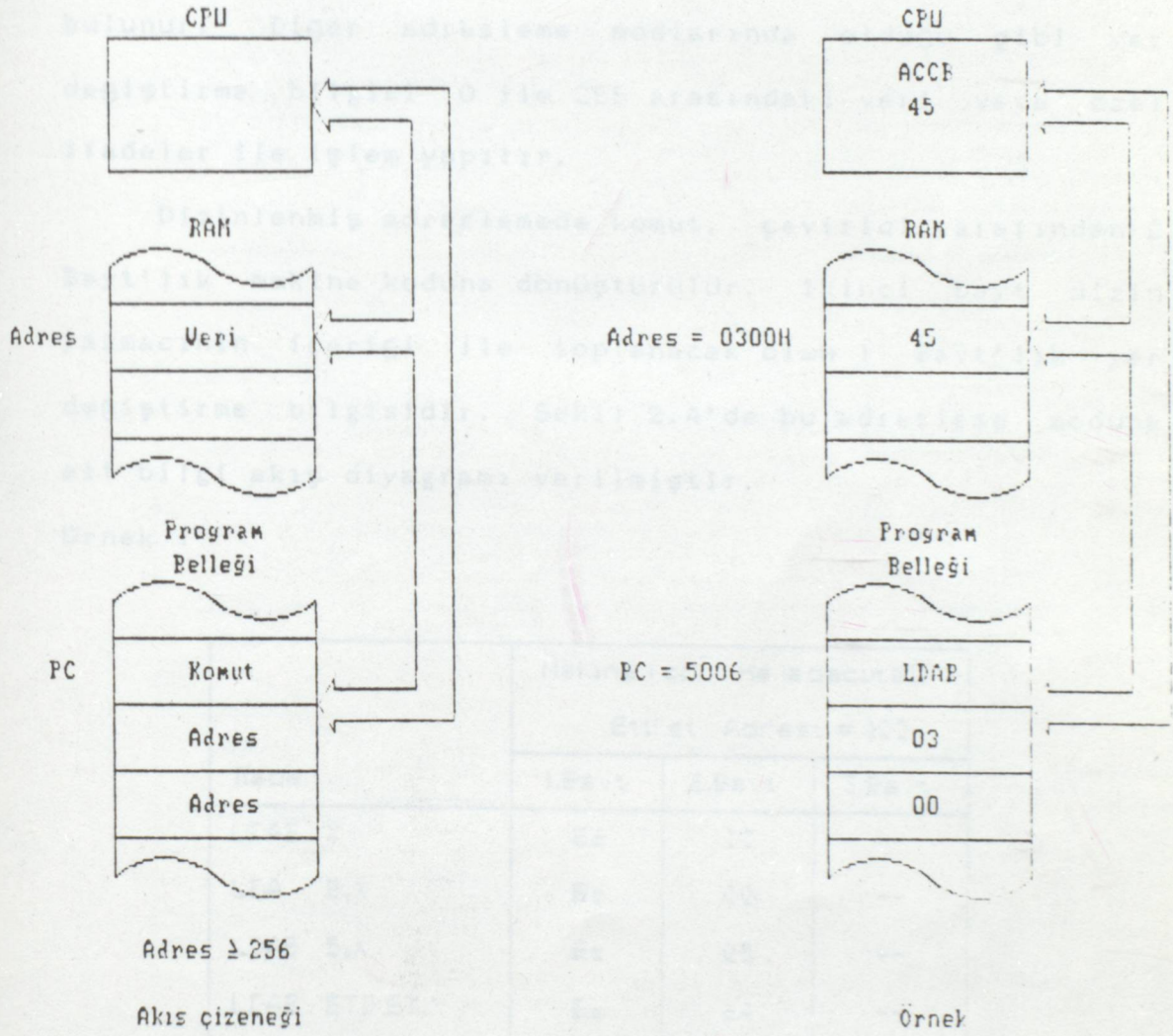
Örnek :

İfade	Makine Kodu (Hexadecimal)		
	1.Baıt	2.Baıt	3.Baıt
LDAA 100	9E	E4	--
LDAA ETİKET	9E	E4	--
LDAA ETİKET-200	E6	01	20



Şekil 2.2 Doğrudan adreslemede bilgi akışı

2.2.5 - Diziye Geniş Adresleme



Şekil 2.3 - Genişletilmiş doğrudan adreslemede bilgi akışı

2.2.5 Dizinlenmiş Adresleme

Bu adresleme modunda dizin yazmacının içeriği ile (16 Bit) yer değiştirme bilgisi aşağıdaki gibi işlem görür.

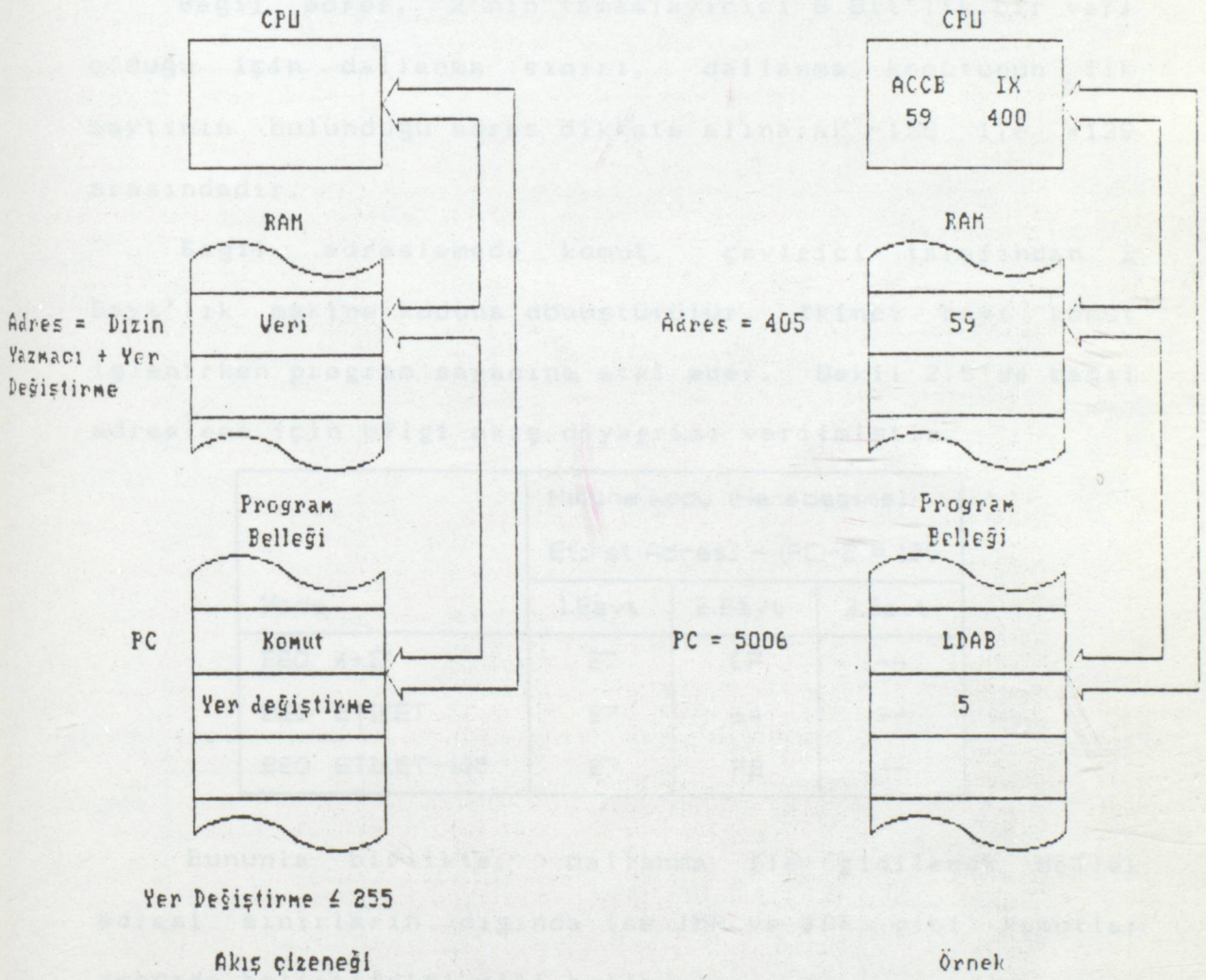
$$\text{Bellek adresi} = \text{Yer değiştirme bilgisi} + (\text{IX})$$

Böylece komutun veri işleyeceği bellek adresi bulunur. Diğer adresleme modlarında olduğu gibi yer değiştirme bilgisi 0 ile 255 arasındaki veri veya özel ifadeler ile işlem yapılır.

Dizinlenmiş adreslemede komut, çevirici tarafından 2 Bayt'lık makine koduna dönüştürülür. İkinci bayt dizin yazmacının içeriği ile toplanacak olan 1 Bayt'lık yer değiştirme bilgisidir. Şekil 2.4'de bu adresleme moduna ait bilgi akış diyagramı verilmiştir.

Örnek :

İfade	Makine Kodu (Hexadecimal)		
	Etiket Adresi = 100		
	1.Bayt	2.Bayt	3.Bayt
LIAB X	E6	10	--
LDA E,X	E6	00	--
LDAB S,X	E6	05	--
LDAB ETIKET,X	E6	64	--
LDAB ETIKET+5,X	E6	6F	--



Şekil 2.4 Dizinlenmiş adreslemede bilgi akışı

2.2.6 Bağıl Adresleme

Bu adresleme modu dallanma komutları ile sınırlıdır.

Adres ilişkisi aşağıdaki gibidir.

Bellek Adresi = (PC) + 2 + Bağıl Adres

(PC) : Dallanma komutunun bulunduğu ilk adres

Bellek Adresi : Dallanma ile gidilecek adres

Bağıl adres, 2'nin tamamlayıcısı 8 Bit'lik bir veri olduğu için dallanma sınırı, dallanma komutunun ilk baytının bulunduğu adres dikkate alınarak -126 ile +129 arasındadır.

Bağıl adreslemede komut, çevirici tarafından 2 Bayt'lık makine koduna dönüştürülür. İkinci bayt komut işlenirken program sayacına etki eder. Şekil 2.5'de bağıl adresleme için bilgi akış diyagramı verilmiştir.

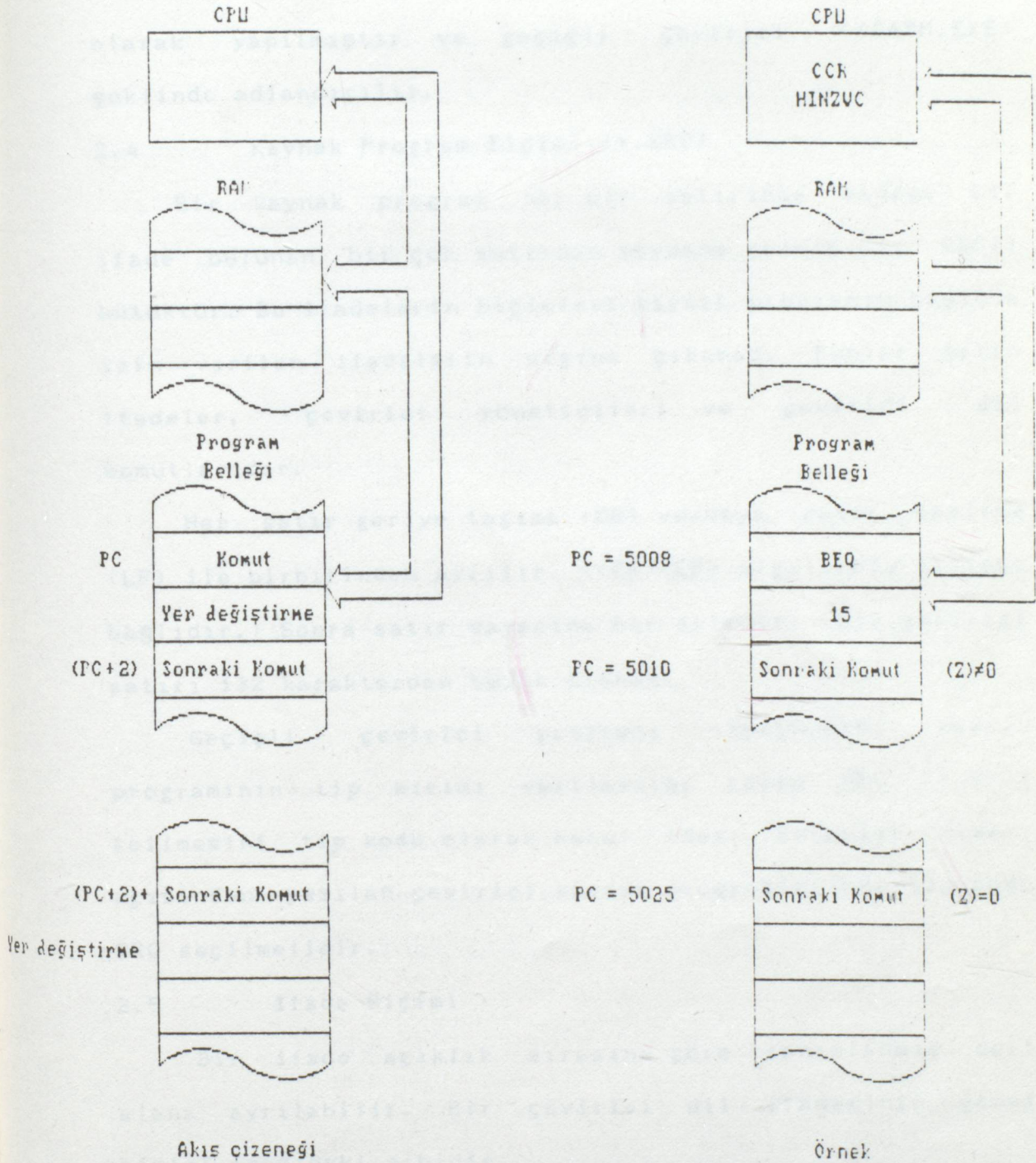
İfade	Makine Kodu (Hexadecimal)		
	Etiket Adresi - (PC)-2 = 100		
	1.Bayt	2.Bayt	3.Bayt
BEQ X+17	27	0F	--
BEQ ETİKET	27	64	--
BEQ ETİKET-105	27	FE	--

Bununla birlikte, dallanma ile gidilecek bellek adresi sınırların dışında ise JMP ve JSR gibi komutlar aşağıda belirtildiği gibi kullanılır.

İfade	Makine Kodu (Hexadecimal)		
	1.Bayt	2.Bayt	3.Bayt
JMP 300	7E	01	20
JSR 300	ED	01	20

Deviricinin Kullanışı

Bu devirici normal performansta çalışır.



Şekil 2.5 Eşil adreslemeye bilgi akışı

2.3 Çeviricinin Kullanımı

Bu çevirici normal çeviriciden (XASM.EXE) ayrı olarak yapılmıştır ve geçişli çevirici (XSASM.EXE) şeklinde adlandırılır.

2.4 Kaynak Program Biçimi (*.SRC)

Bir kaynak program her bir satırında sadece bir ifade bulunan bir çok satırdan meydana gelmiş bir ASCII kütüktür. Bu ifadelerin biçimleri kaynak programın başında izin verilen ifadelerin dışına çıkamaz. Bunlar makro ifadeler, çevirici yöneticileri ve çevirici dil komutlarıdır.

Her satır geriye taşıma (CR) ve/veya satır besleme (LF) ile birbirinden ayrılır. (<CR><LF> algılamacı sisteme bağlıdır.) Sonra satır sayacına bir eklenir. Bir çevirici satırı 132 karakterden büyük olamaz.

Geçişli çevirici programı (XSASM.EXE) kaynak programının tip kısmı verilmediği zaman SRC (*.SRC) kelimesini tip kodu olarak kabul eder. Kolaylık olması açısından yazılan çevirici kaynak programlarında tip kodu SRC seçilmelidir.

2.5 İfade Biçimi

Bir ifade açıklık sırasına göre tanımlanmış dört alana ayrılabilir. Bir çevirici dil ifadesinin genel biçimi aşağıdaki gibidir.

ETİKET İŞLEYİCİ İŞLEM AÇIKLAMALAR

Etiket ve açıklama alanları seçime bağlı olarak kullanılır veya kullanılmaz. İşlem alanı bir çok işlemden

oluşuyorsa işlemler virgül, tab veya boşluk özel karakterleri ile birbirinden ayrılır. Çevirici bu ifadelerin her birinin anlamlarını bulur ve işlem yapar. Bir ifade yukarıda anlatılan alanlardan bir veya birkaçını içermek zorundadır. Örneğin boş bırakılan bir satır kullanılamaz.

Çevirici dil ifadesinde hiç bir şekilde devam satırına izin verilmez. İfade bir satır içinde tamamlanmalıdır.

2.5.1 Etiket Alanı

Bulduğu yerin adresini veya değerini taşıyabilen kullanıcı tarafından belirlenen bir ifadedir. Sadece satırın solundaki ilk 6 karakterlik sembol etiket olarak kabul edilir. Etiket içindeki karakterlerin yeri ve içeriği bazı kurallara bağlıdır. Etiket ilk karakteri bir sayı (0-9), ".", "\$" ve "_" (alt çizgi) olamaz. Bununla birlikte sistem programı içinde özel olarak değerlendirilen ".", "\$" ve "_" karakterleri ilk karakter olmamak şartıyla etiket olarak kullanılabilir.

Örnek :

```
ABCD      LDAA #25
```

Bu durumda ABCD etiket olarak kabul edilip ve bulunulan yerin adres değerini alır.

Örnek :

```
ABC
```

```
KAL
```

TEK LDAA VERİİ

Yukarıdaki örnekte ise üç ayrı etiket aynı değeri alır ve her üçüde dikkate alınarak işlem yapılır.

2.5.2 İşleyici Alanı

İşleyici alanında makro komutları, komut belirleyici sembol veya çevirici yönetim ifadeleri bulunabilir. Etiket alanından sonra bırakılacak bir boşluk karakteri veya tab ile etiket alanından ayrılır.

2.5.3 İşlem Alanı

Bu alanda işleyici ile ilgili işlem ifadeleri bulunur. İşleyici alanından sonra bırakılacak boşluk veya tab ile işleyici alanından ayrılır.

2.5.4 Açıklama Alanı

Diğer ifadelerin bulunması durumunda satırın sonunda işlem ifadelerinden sonra yer alır. Belirlenmiş bir yapı şekli yoktur ve null, rubout, carriage return, line feed, form feed dışındaki ASCII karakterlerden meydana gelir.

Tek başına bir satır açıklama alanı olarak kullanılacaksa "*" karakteri ile başlamak zorundadır.

Bu alan üzerinde çevirici tarafından hiç bir işlem yapılmaz ve kaynak programına etkisi yoktur.

2.6 Biçim Kontrolü

Kaynak programının biçimi yatay olarak boşluk ve TAB'larla kontrol edilir. Bu iki karakterin çevirici programına etkisi yoktur. Çevirici anlamlı ASCII bilgilere rastlıyıncaya kadar işlem yapmaz. Bu kullanıcıya yazım rahatlığı sağlar.

ETIKET JSR ADRES DEGER BIRIKECTE

ETIKET JSR ADRES ;DEGER BIRIKECTE

Yukarıdaki iki satır örneği de çevirici tarafından aynı biçimde anlaşılır.

2.7 İfadeler

İfadeler sembol çeşitleri, sayılar, aritmetik işlem sembolleri ve özel işlem koduna sahip parantezli değerlerden oluşur.

Aritmetik İşlem Sembolleri :

- (a) + Toplama
- (b) - Çıkarma
- (c) * (veya !*) Çarpma
- (d) / (veya !/) Bölme
- (e) ! Üstel

Mantıksal İşlem Sembolleri :

- (a) !+ Mantıksal toplama
- (b) !. Mantıksal çarpma
- (c) !x Mantıksal XOR

Diğerleri :

(a) !< 2 Bayt'lık değeri sola kaydırma (shift)

(b) !> 2 Bayt'lık değeri sağa kaydırma (shift)

(c) !L 2 Bayt'lık değeri sola döndürme (rotate)

(d) !> 2 Bayt'lık değeri sağa döndürme (rotate)

İşlemlerle birlikte "(" ve ")" parantezleri kullanılabilir.

İşlem önceliği aşağıdaki gibidir.

Seviye	İşlem
Yüksek	()
Orta	* / ! !. !+ !x !< !> !L !R
Düşük	+ -

Eşit öncelikli işlemler soldan sağa doğru yapılır.

Sayılar :

On tabanlı <sayı> veya &<sayı>

Onaltı tabanlı 4<sayı> veya <sayı>H

Sekiz tabanlı @<sayı> veya <sayı>O veya <sayı>Q

İki tabanlı %<sayı> veya <sayı>B

ASCII ifadeler : '<karakter> tek bir karakter.

2.8 Çevirici Çıkış kütüğü (*.LST)

Çevirici çıkış düzeni genellikle aşağıda gösterilen biçimdedir.

<u>KOLON</u>	<u>İÇERİK</u>
1 - 5	5 hanelik satır sayıcısı
6	Bölge sayıcısının bölüm bayrağı
8 - 11	Aktif bölge sayıcısının değeri (hex)
13 - 14	Makine çalışma kodu (hex)
16 - 22	Çalışma koduna bağlı Dallanma olmayan çalışma kodu ise
16 - 17	İşlenenin ilk baytı (hex)
18 - 19	İşlenenin ikinci baytı
22	İşlenenin bölüm bayrağı Dallanma çalışma kodu ise
16 - 17	Bağıl dallanma sayacı (hex)
19 - 22	Dallanılan bölgenin adresi
24 - 29	Etiket alanı
31 - 36	İşlem alanı
37 - 44	İşlenen alanı (açıklama alanına kadar uzanır)
46 - son kolon	Açıklama alanı

Eğer seçilirse, normal çevirici çıkışının sonunda sembol geçiş-belirtme tablosu görülebilir.

2.9 Çevirici amaç çıkış kütüğü (*.OBJ)

Bu kütük sadece bağlayıcının çalışabilmesi için gerekli giriş verilerini ASCII kodlu olarak saklar. Bir bağdaştırıcı kütük olduğu için yerleşim biçiminin

anlatılmasına gerek yoktur.

2.10 Program Bölümleri

Program bölümleri temel olarak, HD6301 için yerdeğişirlik ve bağlayıcıda kullanılır. Beş değişik program bölümü vardır. Bu bölümlerin adları ve işlevleri aşağıda anlatılmıştır.

ASCT Gerçek Bölüm (yerideğiştirilemez)

Bir kullanıcı programında sınırlı sayıda gerçek bölüm vardır. Programcı tarafından ayrılmış değişmeyen program bölgelerinde kullanılır. Örneğin ACIA (Asenkron Seri Veri İletişim Arabirimi) ve FIA (Paralel Veri Arabirimi) için ayrılmış adreslerin belirtildiği yerlerde kullanılabilir.

BSCT Temel Bölüm

Kullanıcı programında sadece bir tane temel bölüm vardır. Bağlayıcı programının gerek duyduğu bazı verileri üzerinde bulundurur. BSCT adresleme alanı 0-255 arasında sınırlıdır.

CSCT Ortak Karanlık Bölüm (ilk koşulsuz)

Sadece bir tane CSCT vardır. Bu bölüm ortak karanlık oluşturmak amacıyla kullanılır. İlk koşullandırılmaz.

COMM Ortak Adlandırılmış Bölüm

BSCT, DSCT ve PSCT alanlarının içindeki ortak adlandırılanlar ayrılmıştır. İlk şartlandırma sırasında belleğin bazı bölümleri işaretli alan olarak ayrılarak COMM bölgesini oluşturur.

DSCT Veri Bölümü

Bu bölümden sadece bir tane bulunabilir. Bağlayıcı programının gerek duyduğu bazı verileri içerir. DSCT genellikle genişletilmiş adresleme modundaki değişkenlere erişmek için kullanılır.

PSCT Program Bölümü

DSCT ile benzer çalışır fakat PSCT komutlarla ilgili işlem yapar.

2.11 Çevirici Yönetim Komutları

YÖNETİCİ İŞLEVİ

ÇEVİRİCİ KONTROL YÖNETİCİLERİ

NAM	Programın adı
OPT (REL)	Yerideğiştirilebilir amaç çıkışı seç
(LOAD)	Gerçek amaç çıkışı seç
(OBJ)	Amaç çıkışı seç
(NEM)	Amaç kod doğrudan belleğe
ORG	Origin
ASCT	Gerçek bölüm
BSCT	Temel bölüm
CSCT	Ortak boş bölüm
DSCT	Veri bölümü
PSCT	Program bölümü
COMM	Ortak adlandırılmış bölüm
FAIL	Kasıtlı hata
END	Program bitti

SEMBOL BELİRLEME YÖNETİCİLERİ

EQU	Değişmez değeri belirlemek
SET	Gecici değeri belirlemek
MACR	Makro belirtecini başlatma
ENDM	Makro belirtecini bitirme

VERİ BELİRLEME/SAKLAMA AYIRMA YÖNETİCİLERİ

FCC	Karakter dizisi veri
FCB	Bir bayt veri
FDB	İki bayt veri
BSZ	Sıfır koyulmuş blok
RMB	Ayrılmış bellek bölgesi

ŞARTLI ÇEVİRİCİ YÖNETİCİLERİ

Bu ifade sağlanmış ise çevir

IFGT	O'dan büyük
IFGE	O'dan büyük veya O'a eşit
IFLT	O'dan küçük
IFLE	O'dan küçük veya O'a eşit
IFEQ	O'a eşit
IFNE	O'a eşit değil

Dizi sağlanmış ise çevir

IFC	Karşılaştı
IFNC	Karşılaştırma
ENDC	Şartlı Çevirmeyi bitir

ÇEVİRİCİ ÇIKIŞ KÜTÜK BİÇİMİNİ KONTROL EDEN YÖNETİCİLER

PAGE	Sayfa başı
TTL <dizi>	Sayfa başına konacak mesaj
SPC <n>	n tane boş satır bırak
OPT CREF	Sembol Geçiş-Belirleme tablosunu seç
CLIST	Şartlı ve şartsız yöneticileri yaz
MD	Makro belirteç listesi
MC	Makro çağırma listesi
MEX	Makro genişlemeyi yaz
UNA	Şartlı çevirme sağlanmadıysa çevirme
PAGE	Sayfa uzunluğunu değiştir
NOPAGE	Sayfasız çıkış listesi
LIST	Yazıcı kütük çıkışına izin ver
NOLIST	Yazıcı kütük çıkışına izin verme

Bazı OPT çıkış seçeneklerinin yanına aşağıda belirtilen şekilde komutlar eklenebilir.

OPT <seçenek>[,<seçenek>][...] k1,k2,k3

k1,k2,k3 : R Birinci geçişin sonunda reset
 N "NO" gibi önekleri ayır
 D Kabul edilen biçim (Default)

Çevircinin Makro Yetenekleri

Çevirici dille yapılan programlanada, tekrarlanan ifadeler olduğunda, bu ifadelerin karışmaması ve anlaşılır şekilde görünebilmesini sağlamak amacıyla çevircinin makro yetenekleri kullanılabilir.

Örneğin PIA (Paralel giriş/çıkış arabirimi) yongasının ilk koşullarını yerine getiren program

çeviriciye aşağıdaki şekilde yazılabilir.

```

SETPIA   MACR   DDR,DATA,CONTROL-REG,CTLDTA
        LDAA   #CTLDTA
        STAA  CONTROL-REG
        LDAA  #DATA
        STAA  DDR
SETPIA   23,24,25,#26
        ENDM
        END

```

DDR=23, DATA=24, CONTROL-REG=25, CTLDTA=#26 degerini alır.

Eger MACR belirteci kullanılırsa program aşağıdaki gibi olabilir.

```

SETPIA   MACR
        LDAA   #\3
        STAA  \2
        LDAA  #\1
        STAA  \0
SETPIA   27,28,29,30
        ENDM
        END

```

Programcı LDA A,42E şeklinde bir satır yazarsa, çevirici bunu LDAA #42E şeklinde anlar ve işleme koyar.

2.12 6301 Geçişli Çevirici

Çevirici dille program tasarımının zorlukları göz önüne alınır, 6301 tek yonga mikrobilgisayarının yazılımını çok iyi bilmeyen bir kişi kolaylıkla program yazamaz. Bu nedenle tasarlanması zor olan programlar bile özel geçiş komutları yardımıyla yapılabilir hale gelir. Ayrıca program düzenli olması nedeni ile kolaylıkla geliştirilebilir veya değiştirilebilir. Tasarım sırasında mantık ve yazılım hataları en aza indirilir.

Özel geçiş komutlarının kullanılabilmesi için geçişli çeviricinin seçildiği (OPT STRUCT) programın başında belirtilir.

Eğer çıkış listesinde etiket ve kodların üretilmesi isteniyorsa

`__LST = 1` (istenmiyorsa =0) şeklinde bir satır yazılır.

`__SAVA = 1` şeklindeki bir satır ile PSHA kodunun üretilmesine izin verilebilir.

`__SAVB = 0` şeklindeki bir satır ile PSHB kodunun üretilmesine izin verilmeyebilir.

2.13 İşlenen Sözdizimi

İşlenenler bazı standart çevirici dillerden alınmış veya tamamen özel sözdizimine sahiptir.

2.14 LET Atama Deyimi

LET <atanacak değişkenin adı> := işlenen

LET sadece 8 bit'lik işlemlerde kullanılabilen bir özel atama deyimidir.

Örnekler :

LET M := VER11 ; bellekten belleğe transfer

LET M := -VER11+VER12

2.14.1 Anahtar Kelimeler ve İşleçler

LET A := Anahtar kelime

Anahtar kelime OFF, RESET, CLEARED, FALSE, ON, SET, TRUE kelimelerinden biri olabilir.

Öf : LET A := RESET

Bayrakların sözdizimi aşağıdaki gibidir.

CARRY C bit (elde biti)

C.BIT C bit (elde biti)

OVERFLOW V bit (taşma biti)

V.BIT V bit (taşma biti)

INTERRUPT I bit (kesilme biti)

I.BIT I bit (kesilme biti)

İşlenen alanında NOT (2'ye tamamlayıcı) deyimi her anahtar kelimenin veya bayrağın önüne gelebilir.

2.14.2 Karışık Atama İşlemleri

LET <Atanacak ifade> := işleç1 işlenen1 işleç2

işlenen2

İşleç1 NOT veya -(eksi) olabilir.

İşlenen1 Herhangi bir geçerli işlenen olabilir.

İşleç2 Aşağıda belirtilen bir ikili bir işleç olabilir.

İşlenen2 Herhangi bir geçerli işlenen olabilir.

Çizelge 1 Geçerli İkili İşleçler

<u>İkili İşleç</u>	<u>Açık İfadesi</u>	<u>Örnek</u>	<u>Komut</u>
+	toplama	D+H	ADD
ADD	toplama	D ADD H	ADD
ADC	eldeli toplama	D ADC H	ADC
-	çıkarma	D - H	SUB
SUB	çıkarma	D SUB H	SUB
SBC	eldeli çıkarma	D SBC H	SBC
RES.BY	örtülen bitleri "0"	D RES.BY H	
RES.BIT	belirtilen bit "0"	D RES.BIT N	
LSR	saga mantıksal kaydır	D LSR N	LSR
ASL	sola aritmetik kaydır	D ASL N	ASL
ASR	saga aritmetik kaydır	D ASR N	ASR
ROR	saga döndür	D ROR N	ROR
ROL	sola döndür	D ROL N	ROL
AND	mantıksal ve	D AND H	AND
ORA	mantıksal veya	D ORA H	ORA
EOR	mantıksal özel veya	D EOR H	EOR
SET.BY	örtülen bitleri "1"	D SET.BY H	
SET.BIT	belirtilen bit "1"	D SET.BIT N	
CLEARED.BY	örtülen bitler "0"	D CLEARED.BY H	
CLEAR.BIT	belirtilen bit "0"	D CLEAR.BIT N	
NAND	mantıksal ve-değil	D NAND H	
NOR	mantıksal veya-değil	D NOR H	

H ve D bir bayt'lık veri, N ise kaydırma/döndürme sayacı (0-7 arası geçerli).

2.15 IF Deyimi

Bir şarta bağlı olarak iki seçenekten birinin yürütülmesini sağlayan deyimdir. Sözdizimi aşağıdaki gibidir.

```
IF işlenen <ilişki işleci> THEN <ifade>
veya IF işlenen <ilişki işleci> <ifade>
veya IF işlenen <ilişki işleci> işlenen
    <ifade>
ELSE
    <ifade>
    __ENDS
```

2.16 __ENDS Deyimi

__ENDS deyimi blok yapılı işlemlerde bitirme deyimi olarak kullanılır.

2.17 ON.ERROR ve ON.NOERROR Deyimleri

Bu iki deyim elde bitinin durumunu test eden IF deyiminin özel durumudur. ON.NOERROR deyimi elde bitinin "0" konumunu, ON.ERROR deyimi ise "1" konumunu test eder.

```
Or: ON.ERROR THEN LET A := B
ON.NOERROR
    _CALLR ALT1
ON.NOERROR
LET M := B
    __ENDS
    __ENDS
```

2.18 LEAVE ve RESTART Deyimleri

Bu iki deyim IF deyimi ile birlikte kullanılabilir ve program akışını denetler. LEAVE deyimi bir program öbeğinin veya çevriminin bazı durumlarda terkedilmesini sağlamak amacıyla kullanılır.

örnek :

```

BEGIN KISIM1
.
.
IF RESULT IS NE THEN LEAVE KISIM1
.
.
__ENDS KISIM1

```

RESTART deyimi bir program öbeğinin veya çevriminin bazı durumlarda yeniden yürütülmesini sağlamak amacıyla kullanılır.

örnek :

```

WHILE A NE B
.
.
IF A EQ 0 THEN RESTART LOOP
.
.
__ENDS (WHILE çevrimi)

```

2.19 FOR Deyimi

FOR deyimi ile belirtilen alanın içindeki ifadeler bir çarpışma bağı olmaksızın belirtilen sayıda yürütülmesi sağlanır.

Örnekler :

```
FOR A := 1 TO N BY 2
```

```
  .
  .
  .
__ENDS
```

A'nın değeri N'den büyük olana kadar A, 1'den başlayarak 2 eklenerek artırılır.

```
FOR G := N DOWNTO 1
```

```
  .
  .
  .
__ENDS
```

G'nin değeri 1'den küçük olana kadar N'nin değerinden 1 eksiltilerek küçültülür.

2.20 REPEAT ve UNTIL Deyimleri

REPEAT deyimi, bir program öbeği içindeki ifadelerin belirli bir şarta bağlı olarak tekrarlanarak yürütülmesini sağlar.

Örnekler :

```
REPEAT UNTIL BIT 3 SET. IN DURUM
```

```
REPEAT
```

```
  __CALLR ALT2 B,A
```

```
UNTIL A NE B
```

```
REPEAT
```

```
  .
  .
  .
__ENDS
```

2.21 THRU Deyimi

THRU deyimi, 6301'in DEC komutu ile ana işlem birimine bağlıdır.

Örnek :

```
THRU D
```

```
LET E := D ASL 1
```

```
IF RESULT IS C THEN LEAVE LOOP
```

```
__ENDS
```

2.22 WHILE Deyimi

REPEAT deyimi ile aynı anlamdadır fakat WHILE deyimi kullanıldığında, şartın gerçekleşip gerçekleşmediği, ifadeler yürütülmeden önce test edilir.

Örnek :

```
WHILE A NE B
```

```
.
```

```
.
```

```
.
```

```
__ENDS
```

2.23 BEGIN Deyimi

BEGIN deyimi, program öbekleri oluşturmak amacıyla kullanılır. Böylece LEAVE/RESTART gibi deyimlerin işlerliği sağlanır. BEGIN deyiminin yanına yazılacak olan program öbeğinin adı 6 karakteri geçmemelidir.

Örnek :

```
BEGIN KISIM1
```

```
.
```

```
.
```

```
.
```

```
__ENDS KISIM1
```

2.24 CALLR Deyimi

CALLR deyimiyle ana programdan iki veri alınarak bir alt program gidilerek yürütülmesi sağlanır ve geri döndürülür.

Örnekler :

`_CALLR ALT2 D,M`

ALT2 alt programı çağrılır ve D'nin değeri A yazmacına, M'nin değeri ise E yazmacına aktarılır.

2.25 `_CALLS` Deyimi

`_CALLS` deyimiyle ana programdan, en çok 16 veri yığına atılarak bir alt programa gidilir. Alt program yürütülür ve geri dönlür.

Yığına her zaman 17 Byte veri atılır. İlk veri programda `_CALLS` deyiminin yanında verilen veri sayını belirler.

Örnekler :

`_CALLS ALT2`

`_CALLS ALT3 M,VER11,D`

2.26 `_RETURN` Deyimi

`_RETURN` deyimiyle alt programdan çağrılan programa şartlı veya şartsız olarak geri dönlür.

Örnekler :

`_RETURN`

`_RETURN ON C`

`_RETURN ERROR`

`_RETURN NOERROR`

2.27 `SUBROUTINE` Deyimi

Alt programların çevirici tarafından belirlenebilmesi için alt programın ilk satırı olarak kullanılır. `SUBROUTINE` deyiminin yanına yazılacak ad en çok 6 geçerli karakterden oluşabilir.

Örnekler :

SUBROUTINE ALT1,LOCAL

SUBROUTINE ALT2,SCOPE

SUBROUTINE ALT3,GLOBAL

2.28 ENTRYPOINT Deyimi

ENTRYPOINT deyimi programcının bir etiketin değerinin dışarıdan girilebilmesi için kullanılır.

Örnek :

ENTRYPOINT VER11,SCOPE

2.29 PROBE Deyimi

PROBE deyimi bir işlenen ifadesinin durumunu test etmek amacıyla kullanılır.

Örnek :

PROBE BILGI

2.30 BEGINMODULE Deyimi

Bu deyim programcının programın ana kısım ve gövdesini biçimlendirmesini sağlar.

BEGINMODULE ISIM,KİMLİK,TARİH

Örnek :

BEGINMODULE DENEME,MART

2.31 ENDMODULE Deyimi

Bu deyim BEGINMODULE ile başlayan programın bittiğini belirtmek için kullanılır. Program adı aynı şekilde kullanılmalıdır.

Örnek :

ENDMODULE DENEME

2.32 Program örnekleri

Geçişli çeviriciyle bir programın nasıl yazılacağına ve tasarımı nasıl yapılacağına daha iyi anlaşılabilmesi için bu bölümde bir kaç program örneği verilmiştir.

```

*****
* BU PROGRAM 100-65535 MIKROSANIYE *
* UZUNLUGUNDAKI DARBELERIN GENISLIGINI *
* OLCER VE BCD OLARAK SAKLAR. *
*****
* PWCT2 ICTEKI RAM *
*****
      ORG     480
ICRUP   RMB     2
PWDATA  RMB     2
ENDF    RMB     1
HEXD    RMB     2
DECD    RMB     3
DECDT   RMB     3
DEA     RMB     2
*****
*      ICTEKI YAZMACLAR *
*****
PWDDR   EQU     401
TCSR1   EQU     408
ICR     EQU     40D
*****
*      ANA PROGRAM *
*****
      ORG     4E00
PWHN    LDS     #4FF
        BCLR   0, ENDF
        LEAA  #412
        STAA  TCSR1
        CLI
PWHN1   BTST   0, ENDF
        BEQ   PWHN1
        BCLR   0, ENDF
        LDD   PWDATA
        STD   HEXD
        JSR   HEX
FEND    BRA    FEND

```



```

*****
*          PWCNT : DARBE GENISLIGI OLCME          *
*****
*  CIKIS : PWDATA (DARBE GENISLIGI)              *
*          ENDF   (BITIS BAYRAGI)                *
*****
PWCNT  LDD      ICR
        BTST    1,TCSR1
        BEQ     PWCT1
        LDD     ICR
        STD     ICRUP
        BCLR    1,TCSR1
        BRA     PWCT2
PWCT1  LDD      ICR
        SUBD    ICRUP
        STD     PWDATA
        BSET    0,ENDF
        BSET    1,TCSR1
PWCT2  RTI
*****
*          HEX TO BCD DONUSTURUCU                *
*****
*  GIRIS : ACCD  (2 BYTE HEX SAYI)               *
*  CIKIS : DECD  (5 HANE BCD SAYI)              *
*****
HEX    CLRA
        CLRB
        STD     DECD
        STAA    DECD+2
        LDAB    #16
HEX2   ASL     HEXD+1
        ROL     HEXD
        LDX     #3
HEX1   LDAA    DECD-1,X
        ADCA    DECD-1,X
        DAA
        STAA    DECD-1,X
        DEX
        BNE     HEX1
        DECB
        BNE     HEX2
        RTS
        END

```

.MAIN.

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *

Page: 1

```

1                                     ASCT
2                                     *****
3                                     * BU PROGRAM 100-65535 MIKROSANIYE *
4                                     * UZUNLUGUNDAKI DARBELERIN GENISLIGINI *
5                                     * OLCER VE BCD OLARAK SAKLAR. *
6                                     *****
7                                     * ICTEKI RAM *
8                                     *****
9                                     ORG 480
10A 0060 0002 ICRUP RMB 2
11A 0062 0002 PWDATA RMB 2
12A 0064 0001 ENDF RMB 1
13A 0065 0002 HEXD RMB 2
14A 0067 0003 DECD RMB 3
15A 006A 0003 DECDT RMB 3
16A 006D 0002 DEA RMB 2
17                                     *****
18                                     * ICTEKI YAZMACLAR *
19                                     *****
20 0001 P2DDR EQU 401
21 0008 TCSR1 EQU 408
22 000D ICR EQU 40D
23                                     *****
24                                     * ANA PROGRAM *
25                                     *****
26 E000 ORG 4E000
27A E000 8E 00FF PWMN LDS 44FF
28A E003 71 FE 84 BCLR 0,ENDF
29A E006 86 12 LDAA 4412
30A E008 97 08 STAA TCSR1
31A E00A 0E CLI
32A E00B 7B 01 84 PWMN1 BTST 0,ENDF
33A E00E 27 FB BEQ PWMN1
34A E010 71 FE 84 BCLR 0,ENDF
35A E013 DC 82 LDD PWDATA
36A E015 DD 85 STD HEXD
37A E017 BD E039 JSR HEX
38A E01A 20 FE PEND BRA PEND
39                                     *****
40                                     * PWCNT : DARBE GENISLIGI OLCME *
41                                     *****
42                                     * CIKIS : PWDATA (DARBE GENISLIGI) *
43                                     * ENDF (BITIS BAYRAGI) *
44                                     *****
45A E01C DC 0D PWCNT LDD ICR
46A E01E 7B 02 08 BTST 1,TCSR1
47A E021 27 09 BEQ PWCT1
48A E023 DC 0D LDD ICR
49A E025 DD 80 STD ICRUP
50A E027 71 FD 08 BCLR 1,TCSR1

```

.MAIN.

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
 YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *
 Page: 2

```

51A E02A 20 0C          BRA    PWCT2
52A E02C DC 0D          PWCT1 LDD    ICR
53A E02E 93 80          SUBD   ICRUP
54A E030 DD 82          STD    PWDATA
55A E032 72 01 84      BSET   0,ENDF
56A E035 72 02 08      BSET   1,TCSR1
57A E038 3B            PWCT2 RTI
58                      *****
59                      *      HEX TO BCD DONUSTURUCU      *
60                      *****
61                      * GIRIS : ACCD (2 BYTE HEX SAYI) *
62                      * CIKIS : DECD (5 HANE BCD SAYI) *
63                      *****
64A E039 4F            HEX    CLRA
65A E03A 5F            CLRB
66A E03B DD 87          STD    DECD
67A E03D 97 89          STAA  DECD+2
68A E03F C6 10          LDAB  #16
69A E041 78 0086      HEX2  ASL    HEXD+1
70A E044 79 0085      ROL   HEXD
71A E047 CE 0003      LDX   #3
72A E04A A6 86          HEX1  LDAA  DECD-1,X
73A E04C A9 86          ADCA  DECD-1,X
74A E04E 19            DAA
75A E04F A7 86          STAA  DECD-1,X
76A E051 09            DEX
77A E052 26 F6          BNE   HEX1
78A E054 5A            DECB
79A E055 26 EA          BNE   HEX2
80A E057 39            RTS
81                      END

```

Symbol Table:

```

DEA    008D DECD    0087 DECDT    008A ENDF    0064 HEX    E039 HEX1    E04A
HEX2   E041 HEXD    0085 ICR    000D ICRUP    0060 NARG    0000 F2DDR    0001
PEND   E01A PWCT   E01C PWCT1   E02C PWCT2   E038 PWDATA  0062 PWN    E000
PWN1   E00B TCSR1  0008
RUN COMPLETE -      0 ERRORS DETECTED

```

```

OPT      NOLIST,CREF
NAM      DARBEGEN
ASCT
*****
* BU PROGRAM 100-65535 MIKROSANIYE *
* UZUNLUGUNDAKI DARBELERIN GENISLIGINI *
* OLCER VE BCD OLARAK SAKLAR. *
*****
OPT      LIST
*****
*      ICTEKI RAM *
*****
      CIKIS  ORG      $80
ICRUP     RMB      2
PWDATA   RMB      2
ENDF     RMB      1
HEXD     RMB      2
DECD     RMB      3
DECDT    RMB      3
DEA      RMB      2
*****
*      ICTEKI YAZMACLAR *
*****
P2DDR    EQU      $01
TCSR1    EQU      $08
ICR      EQU      $0D
*****
*      ANA PROGRAM *
*****
      ORG      $E000
PWMN     LDS      #$FF
          BCLR    0,ENDF
          LDAA   #$12
          STAA   TCSR1
          CLI
PWMN1    BTST    0,ENDF
          BEQ    PWMN1
          BCLR    0,ENDF
          LDD    PWDATA
          STD    HEXD
          JSR    HEX
PEND     BRA     PEND
*****
*      PWCNT : DARBE GENISLIGI OLCME *
*****
*      CIKIS : PWDATA (DARBE GENISLIGI) *
*      ENDF   (BITIS BAYRAGI) *
*****
PWCNT    LDD     ICR
          BTST   1,TCSR1
          BEQ    PWCT1
          LDD    ICR
          STD    ICRUP

```

```

      BCLR      1,TCSR1
      BRA      PWCT2
PWCT1  LDD      ICR
      SUBD     ICRUP
      STD      PWDATA
      BSET     0,ENDF
      BSET     1,TCSR1
PWCT2  RTI

```

```

*****
*          HEX TO BCD DONUSTURUCU          *
*****
* GIRIS : ACCD (2 BYTE HEX SAYI)          *
* CIKIS : DECD (5 HANE BCD SAYI)          *
*****

```

```

HEX   CLRA
      CLRB
      STD      DECD
      STAA     DECD+2
      LDAB     #16
HEX2  ASL      HEXD+1
      ROL      HEXD
      LDX      #3
HEX1  LDAA     DECD-1,X
      ADCA     DECD-1,X
      DAA
      STAA     DECD-1,X
      DEX
      BNE      HEX1
      DECE
      BNE      HEX2
      RTS
      END

```

DAREEG

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *

Page: 1

```

          9A 0000
10
11
12
13          0060
14A 0080 0002
15A 0082 0002
16A 0084 0001
17A 0085 0002
18A 0087 0003
19A 008A 0003
20A 008D 0002
21
22
23
24          0001
25          0008
26          000D
27
28
29
30          E000
31A E000 6E 00FF
32A E003 71 FE 64
33A E006 66 12
34A E008 97 08
35A E00A 0E
36A E00B 7B 01 64
37A E00E 27 FB
38A E010 71 FE 64
39A E013 1C 62
40A E015 DD 85
41A E017 BD E039
42A E01A 20 FE
43
44
45
46
47
48
49A E01C 1C 0D
50A E01E 7B 02 08
51A E021 27 09
52A E023 1C 0D
53A E025 1D 60
54A E027 71 FD 08
55A E02A 20 0C
56A E02C 1C 0D
57A E02E 93 60
58A E030 1D 62

```

```

          OPT   LIST
*****
*           ICTEKI RAM           *
*****
          ORG   $80
ICRUP   RMB   2
PWRDATA RMB   2
ENDF    RMB   1
HEXD    RMB   2
DECD    RMB   3
DECDT   RMB   3
DEA     RMB   2
*****
*           ICTEKI YAZMACLAR     *
*****
P2LDR   EQU   $01
TCSR1   EQU   $08
ICR      EQU   $0D
*****
*           ANA PROGRAM          *
*****
          ORG   $E000
PWNM    LDS   #$FF
        BCLR  0,ENDF
        LDAA  #$12
        STAA TCSR1
        CLI
PWNM1   BTST  0,ENDF
        BEQ  PWNM1
        BCLR  0,ENDF
        LDD  PWRDATA
        STD  HEXD
        JSR  HEX
FEND    BRA   FEND
*****
*           PWCNT : DARBE GENISLIGI OLCME *
*****
*           CIKIS : PWRDATA (DAREE GENISLIGI) *
*           ENDF (BITIS BAYRAGI) *
*****
PWCNT   LDD   ICR
        BTST  1,TCSR1
        BEQ  PWCNT1
        LDD  ICR
        STD  ICRUP
        BCLR  1,TCSR1
        BRA  PWCNT2
PWCNT1  LDD   ICR
        SUBD ICRUP
        STD  PWRDATA

```

DAREEG

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
 YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *
 Page: 2

```

59A E032 72 01 84          ESET  0,ENDF
60A E035 72 02 08          BSET  1,TCSR1
61A E038 3B                PWCT2  RTI
62                          *****
63                          *      HEX TO BCD DONUSTURUCU      *
64                          *****
65                          * GIRIS : ACCD (2 BYTE HEX SAYI)      *
66                          * CIKIS : DECD (5 HANE BCD SAYI)      *
67                          *****
68A E039 4F                HEX    CLRA
69A E03A 5F                CLRB
70A E03B DD 87             STD    DECD
71A E03D 97 89             STAA  DECD+2
72A E03F C6 10             LDAB  #16
73A E041 78 0086          HEX2   ASL  HEXD+1
74A E044 79 0085          ROL   HEXD
75A E047 CE 0003          LDX   #3
76A E04A A6 86            HEX1   LDAA DECD-1,X
77A E04C A9 86            ADCA  DECD-1,X
78A E04E 19              DAA
79A E04F A7 86            STAA  DECD-1,X
80A E051 09              DEX
81A E052 26 F6            BNE   HEX1
82A E054 5A              DECB
83A E055 26 EA            BNE   HEX2
84A E057 39              RTS
85                          END

```

Symbol Table:

```

008D DEA  00020*
0087 DECD 00018*00070 00071 00076 00077 00079
008A DECDT 00019*
0084 ENDF 00016*00032 00036 00038 00059
E039 HEX  00041 00068*
E04A HEX1 00076*00061
E041 HEX2 00073*00063
0085 HEXD 00017*00040 00073 00074
003D ICR  00026*00049 00052 00056
0060 ICRUF 00014*00053 00057
0001 P2DDR 00024*
E01A FEND 00042*00042*
E01C PWCNT 00049*
E02C PWCT1 00051 00056*
E038 PWCT2 00055 00061*
0082 PWDATA 00015*00039 00056
E000 PWMN  00031*
E00E PWMN1 00036*00037
000B TCSR1 00025*00034 00050 00054 00060

```

```

OPT      NOLIST,NOPAGE
TTL      DARBE GENISLIGININ OLCULMESI
NAM      DARBEGEN
ASCT

```

```

*****
* BU PROGRAM 100-65535 MIKROSANIYE *
* UZUNLUGUNDAKI DARBELERIN GENISLIGINI *
* OLCER VE BCD OLARAK SAKLAR. *
*****

```

```

OPT      LIST

```

```

*****
*      ICTEKI RAM *
*****

```

```

          ORG      $80
ICLUP    RMB      2
PWDATA   RMB      2
ENDF     RMB      1
HEXD     RMB      2
DECD     RMB      3
DECDDT   RMB      3
DEA      RMB      2

```

```

*****
*      ICTEKI YAZMACLAR *
*****

```

```

P2DDR    EQU      $01
TCR1     EQU      $08
ICR      EQU      $0D

```

```

*****
*      ANA PROGRAM *
*****

```

```

          ORG      $E000
PWMN     LDS      #$FF
          BCLR    0,ENDF
          LDAA   #$12
          STAA   TCSR1
          CLI
PWMN1    BTST    0,ENDF
          BEQ    PWMN1
          BCLR    0,ENDF
          LDD    PWDATA
          STD    HEXD
          JSR    HEX
PEND     BRA     PEND
          PAGE

```

```

*****
*      PWCNT : DARBE GENISLIGI OLCME *
*****
*      CIKIS : PWDATA (DARBE GENISLIGI) *
*      ENDF   (BITIS BAYRAGI) *
*****

```

```

PWCNT    LDD      ICR
          BTST    1,TCSR1
          BEQ    PWCT1

```



```

LDD      ICR
STD      ICRUP
BCLR     1,TCSR1
BRA      PWCT2
PWCT1    LDD      ICR
          SUBD     ICRUP
          STD      PWDATA
          BSET     0,ENDF
          BSET     1,TCSR1
PWCT2    RTI
          SPC      2

```

```

*****
*      HEX TO BCD DONUSTURUCU      *
*****
* GIRIS : ACCD (2 BYTE HEX SAYI)   *
* CIKIB : DECD (5 HANE BCD SAYI)  *
*****
HEX      CLRA
          CLRB
          STD      DECD
          STAA     DECD+2
          LDAB     #16
HEX2     ASL      HEXD+1
          ROL      HEXD
          LDX      #3
HEX1     LDAA     DECD-1,X
          ADCA     DECD-1,X
          DAA
          STAA     DECD-1,X
          DEX
          BNE      HEX1
          DECB
          BNE      HEX2
          RTS
          END

```

DAREEG

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1966
YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *

Page: 1

DAREE GENISLIGININ OLCULMESI

	OPT	LIST
10A 0000		
11	*****	
12	* ICTEKI RAM	*
13	*****	
14 0080	ORG	\$80
15A 0080 0002	ICRUP	RMB 2
16A 0082 0002	PWDATA	RMB 2
17A 0084 0001	ENDF	RMB 1
18A 0085 0002	HEXD	RMB 2
19A 0087 0003	DECD	RMB 3
20A 008A 0003	DECDT	RMB 3
21A 008D 0002	DEA	RMB 2
22	*****	
23	* ICTEKI YAZMAKLAR	*
24	*****	
25 0001	P2DDR	EQU \$01
26 0008	TCSR1	EQU \$08
27 000D	ICR	EQU \$0D
28	*****	
29	* ANA PROGRAM	*
30	*****	
31 E000	ORG	\$E000
32A E000 8E 00FF	PWMN	LDS #\$FF
33A E003 71 FE 84		ECLR 0, ENDF
34A E006 86 12		LDAA #\$12
35A E008 97 06		STAA TCSR1
36A E00A 0E		CLI
37A E00B 7B 01 84	PWMN1	BTST 0, ENDF
38A E00E 27 FB		BEQ PWMN1
39A E010 71 FE 84		ECLR 0, ENDF
40A E013 DC 82		LDD PWDATA
41A E015 DD 85		STD HEXD
42A E017 ED E039		JSR HEX
43A E01A 20 FE	PEND	BRA PEND


```
*****
*      BU PROGRAMIN AMACI GECISLI YAPILMIS CEVIRICININ KOMUTLARINDAN      *
*      BAZILARININ KULLANILIS ORNEKLERINI VERMEKTIR.                      *
*****
```

```
OPT    STRUCT
__LST  = 1
BEGINMODULE  DENE1,MAY87
__LNG  = 0
__SAVA = 0
__SAVB = 0
ASCT
ORG    $0000
VER1   RMB    1
VER11  FCB    2
VER12  FCB    4
NUM    RMB    1
DURUM  FCB    0
CEVRE  FCB    1
CALIS  RMB    $20
PSCT
BEGIN  DENE1
LET A := $23
LET B := VER11
LET VER1 := VER11 + VER12
LET DURUM := VER1 SET.BIT 3
LET CALIS := VER1 ROL 4
LET CALIS := CALIS CLEARED.BY A
IF CALIS GT VER1 THEN LET DURUM := RESET
    IF CALIS EQ VER1
        _CALLR DENE2 VER11,VER12
        LET DURUM := SET
    _ENDS
IF BIT 7 SET.IN DURUM
    LDAB $55
    CLR  NUM
    LET CARRY := SET
ELSE
    TAB
    LET NUM := #1
    LET CARRY := CLEARED
    _ENDS
WHILE CALIS GT VER1
    LET CALIS := VER11 SUB VER12
    ON.ERROR THEN RESTART DENE1
    _ENDS
REPEAT
    _CALLS DENE3 DURUM
    LDAA 1,X
    STAA DURUM
    PROBE DURUM
    IF RESULT IS 2 LEAVE LOOP
    _ENDS
REPEAT UNTIL BIT 3 CLEAR.IN DURUM
LET NUM := CLEARED
REPEAT
    LET NUM := NUM + 1
    LDAB NUM
```

```

DENEME FOR A := #1 TO B
        LET CALIS := -VER11 AND VER12
        THRU B
        LET VER1 := NOT B
        IF RESULT IS C THEN LET CARRY := RESET
        _ENDS
    _ENDS
1 UNTIL NUM EQ CALIS
2 SUBROUTINE DENE2,LOCAL
3   LET A := B
4   CLR B
5   _RETURN ON C
6 SUBROUTINE DENE3,LOCAL
7   LDAA 1,X
8   ANDA #64
9   SUBA #1
10  STAA 1,X
11  _RETURN NOERROR
12  _ENDS DENE1
13  ENDMODULE DENEME
14
15  ORG +0000
16  VER1 RRG 1
17  VER11 FCB 2
18  VER12 FCB 4
19  NUM RRG 1
20  DENEM FCB 0
21  CENEM FCB 1
22  CALIS TRD 400
23  _PRT
24  _PRTIN DENE1
25  _END
26  LET A := VER11
27  ANDA VER11
28  LET VER1 := VER11 AND VER12
29  LDAA VER11
30  ANDA VER12
31  STAA VER1
32  _UNTIL NUM EQ CALIS
33  CLR NUM
34  ORG #0
35  STAA DENEM
36  LET CALIS := VER1 AND 4
37  LDAA VER1
38  _END
39  ORG #0
40  STAA CALIS
41  IF CALIS GT 0 THEN LET DENEM := RESET
42  LDAA CALIS
43  _PRT VER1

```

DENEME

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *

Page: 1

```

1
2
3
4
5P 0000
6      0001
7P 0000
      P 0000
8      0000
9      0000
10     0000
11
12     0000
13A 0000 0001
14A 0001 02
15A 0002 04
16A 0003 0001
17A 0004 00
18A 0005 01
19A 0006 0020
20
21P 0000
      P 0000
22P 0000
      P 0000 86 23
23P 0002
      P 0002 D6 01
24P 0004
      P 0004 96 01
      P 0005 9B 02
      P 0008 97 00
25P 000A
      P 000A 96 00
      P 000C 8A 08
      P 000E 97 04
26P 0010
      P 0010 96 00
      P 0012 C6 04
      P 0014 49
      P 0015 5A
      P 0016 26 FC
      P 0018 97 06
27P 001A
      P 001A 43
      P 001B 94 06
      P 001D 97 06
28P 001F
      P 001F 96 06
      P 0021 91 00

```

```

*****
*      BU PROGRAMIN AMACI GECISLI YAPILMIS CEVIRICININ
*      BAZILARININ KULLANILIS ORNEKLERINI VERMEKTIR.
*****
      OPT   STRUCT
      _LST  = 1
      BEGINMODULE   DENEME,MAY87
      NAM DENEME
      IDNT MAY87
      _LNG  = 0
      _SAVA = 0
      _SAVB = 0
      ASCT
      ORG   $0000
VERI   RMB   1
VERI1  FCB   2
VERI2  FCB   4
NUM    RMB   1
DURUM  FCB   0
CEVRE  FCB   1
CALIS  RMB   $20
      PSCT
      BEGIN  DENE1
      _B000
      LET A := #423
      LDAA #423
      LET B := VERI1
      LDAB VERI1
      LET VERI := VERI1 + VERI2
      LDAA VERI1
      ADDA VERI2
      STAA VERI
      LET DURUM := VERI SET.BIT 3
      LDAA VERI
      ORAA #8
      STAA DURUM
      LET CALIS := VERI ROL 4
      LDAA VERI
      LDAB #4
      ROLA
      DECB
      BNE *-2
      STAA CALIS
      LET CALIS := CALIS CLEARED.BY A
      COMA
      ANDA CALIS
      STAA CALIS
      IF CALIS GT VERI THEN LET DURUM := RESET
      LDAA CALIS
      CMP A VERI

```

DENEME

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
 YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *
 Page: 2

```

P 0023 2F 03          BLE _L000
P 0025 7F 0004       CLR DURUM
P 0028                _L000
29P 0028             IF CALIS EQ VER1
P 0028 96 06         LDAA CALIS
P 002A 91 00         CMP A VER1
P 002C 26 0B         BNE _L001
30P 002E             _CALLR DENE2 VER11,VER12
P 002E 96 01         LDAA VER11
P 0030 D6 02         LDAB VER12
P 0032 BD 00D1       JSR DENE2
31P 0035             LET DURUM := SET
P 0035 86 FF         LDAA #FF
P 0037 97 04         STAA DURUM
32P 0039             _ENDS
P 0039                _L001
33P 0039             IF BIT 7 SET.IN DURUM
P 0039 96 04         LDAA DURUM
P 003B 85 80         BITA #128
P 003D 27 0B         BEQ _L002
34P 003F C6 55         LDAB #55
35P 0041 7F 0003       CLR NUM
36P 0041             LET CARRY := SET
P 0044 0D            SEC
37P 0045             ELSE
P 0045 20 06         BRA _L003
P 0047                _L002
38P 0047 16          TAB
39P 0047             LET NUM := #1
P 0048 86 01         LDAA #1
P 004A 97 03         STAA NUM
40P 004C             LET CARRY := CLEARED
P 004C 0C            CLC
41P 004D             _ENDS
P 004D                _L003
42P 004D             WHILE CALIS GT VER1
P 004D                _B001
P 004D 96 06         LDAA CALIS
P 004F 91 00         CMP A VER1
P 0051 2F 0A         BLE _L004
43P 0053             LET CALIS := VER11 SUB VER12
P 0053 96 01         LDAA VER11
P 0055 90 02         SUBA VER12
P 0057 97 06         STAA CALIS
44P 0059             ON.ERROR THEN RESTART DENE1
P 0059 25 A5         BCS _B000
45P 005B             _ENDS
P 005B 20 F0         BRA _B001
P 005D                _L004
P 005D                _E001

```

LEKEME

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1966
YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *

Page: 3

```

46P 005D                                REPEAT
P 005D                                _E002
47P 005D                                _CALLS DENE3 DURUM
P 005D 30                                TSX
P 005E 09                                DEX
P 005F 86 01                             LDAA #001
P 0061 36                                PSHA
P 0062 96 04                             LDAA DURUM
P 0064 36                                PSHA
P 0065 86 FF                             LDAA #FF
P 0067 36                                PSHA
P 0068 86 FF                             LDAA #FF
P 006A 36                                PSHA
P 006B 86 FF                             LDAA #FF
P 006D 36                                PSHA
P 006E 86 FF                             LDAA #FF
P 0070 36                                PSHA
P 0071 86 FF                             LDAA #FF
P 0073 36                                PSHA
P 0074 86 FF                             LDAA #FF
P 0076 36                                PSHA
P 0077 86 FF                             LDAA #FF
P 0079 36                                PSHA
P 007A 86 FF                             LDAA #FF
P 007C 36                                PSHA
P 007D 86 FF                             LDAA #FF
P 007F 36                                PSHA
P 0080 86 FF                             LDAA #FF
P 0082 36                                PSHA
P 0083 86 FF                             LDAA #FF
P 0085 36                                PSHA
P 0086 86 FF                             LDAA #FF
P 0088 36                                PSHA
P 0089 86 FF                             LDAA #FF
P 008B 36                                PSHA
P 008C 86 FF                             LDAA #FF
P 008E 36                                PSHA
P 008F 86 FF                             LDAA #FF
P 0091 36                                PSHA
P 0092 ED 00D6                          JSR DENE3
48P 0095 A6 01                             LDAA 1,X
49P 0097 97 04                             STAA DURUM
50P 0097                                PROBE DURUM
P 0099 96 04                             LDAA DURUM
P 009B 84 FF                             ANDA #FF
51P 009D                                IF RESULT IS 2 LEAVE LOOP
P 009D 27 02                             BEQ _E002
52P 009F                                _ENDS
P 009F 20 EC                             BRA _E002
P 00A1                                _E002

```


DENEME

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *

Page: 4

```

53P 00A1                                REPEAT UNTIL BIT 3 CLEAR.IN DURUM
  P 00A1                                _B003
  P 00A1 96 04                          LDAA DURUM
  P 00A3 85 08                          BITA #B
  P 00A5 26 FA                          BNE _B003
54P 00A7                                LET NUM := CLEARED
  P 00A7 7F 0003                        CLR NUM
55P 00AA                                REPEAT
  P 00AA                                _B004
56P 00AA                                LET NUM := NUM + 1
  P 00AA 96 03                          LDAA NUM
  P 00AC 9B 01                          ADDA 1
  P 00AE 97 03                          STAA NUM
57P 00B0  D6 03                          LDAB NUM
58P 00B0                                FOR A := #1 TO B
  P 00B2                                _B005
  P 00B2 86 01                          LDAA #1
  P 00B4 4A                              DECA
  P 00B5                                _BX005
59P 00B5                                LET CALIS := -VER11 AND VER12
  P 00B5 B6 FFFF                        LDAA -VER11
  P 00B8 94 02                          ANDA VER12
  P 00BA 97 06                          STAA CALIS
60P 00BC                                THRU B
  P 00BC                                _B006
61P 00BC                                LET VER1 := NOT B
  P 00BC 53                              COMB
  P 00BD D7 00                          STAB VER1
62P 00BF                                IF RESULT IS C THEN LET CARRY := RESET
  P 00BF 24 01                          BCC _L006
  P 00C1 0C                              CLC
  P 00C2                                _L006
63P 00C2                                __ENDS
  P 00C2 5A                              DECB
  P 00C3 26 F7                          BNE _B006
  P 00C5                                _E006
64P 00C5                                __ENDS
  P 00C5 4C                              INCA
  P 00C6 11                              CBA
  P 00C7 2C 02                          BGE _L007
  P 00C9 20 EA                          BRA _BX005
  P 00CB                                _L007
  P 00CB                                _E005
65P 00CB                                UNTIL NUM EQ CALIS
  P 00CB 96 03                          LDAA NUM
  P 00CD 91 06                          CMP A CALIS
  P 00CF 26 D9                          BNE _B004
66P 00D1                                SUBROUTINE DENE2,LOCAL
  P 00D1                                DENE2
67P 00D1                                LET A := B

```


3.1-Çıktı

Hesaplama hızı, tek yonga mikrobilgisayarlar ile gerçekleştirilen bu uygulamanın amacı, tıbbi alanlarda ve donanımların özelliklerinin daha iyi anlaşılmasını sağlamaktır.

Tıbbi alanlarda kullanılabilir ve verimli olarak kullanılabilir.

Donanımların desteğiyle ilgili olarak, donanımların

İçerisindeki bilgilerin kullanılması çok kolaydır. Donanımların

birbirine bağlanması, donanımların bir

birbirine bağlanması, donanımların bir

birbirine bağlanması, donanımların bir

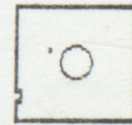
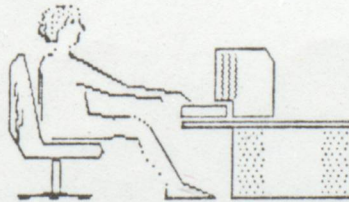
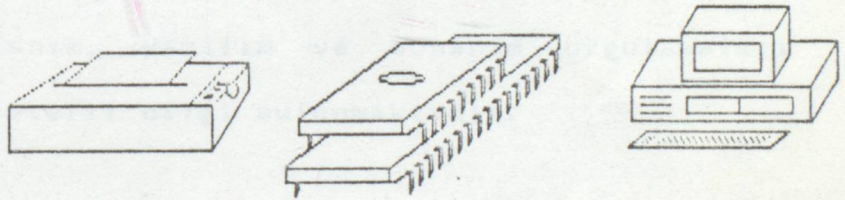
birbirine bağlanması, donanımların bir

birbirine bağlanması, donanımların bir

birbirine bağlanması, donanımların bir

birbirine bağlanması, donanımların bir

birbirine bağlanması, donanımların bir



ÜÇÜNCÜ BÖLÜM

TEK YONGA MİKROBİLGİSAYAR

UYGULAMASI

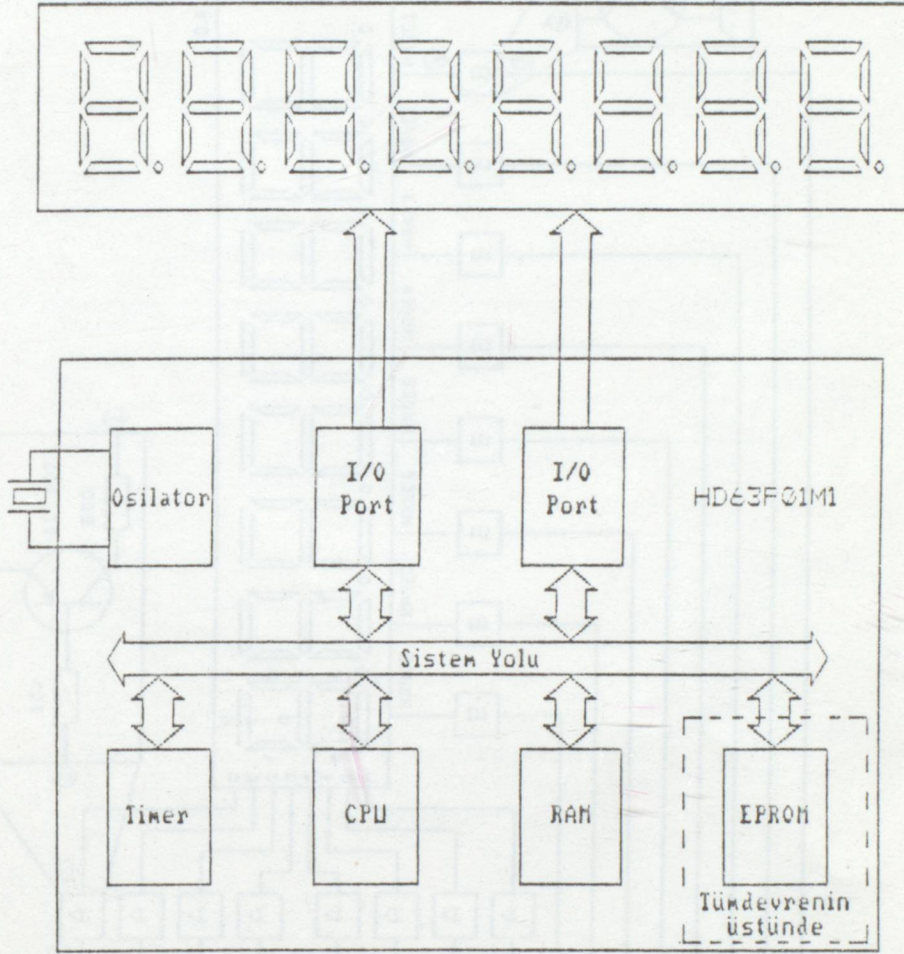
3.1 GİRİŞ

HD63P01M1 CMOS tek yonga mikrobilgisayarı ile gerçekleştirilen bu uygulamanın amacı tümdevrenin yazılım ve donanım özelliklerinin daha iyi anlaşılmasını sağlamaktır.

Tümdevrenin kullanılabilmesi, yeterli yazılım ve donanım desteğine bağlı olmakla birlikte bu destek sağlandığı takdirde kullanımı çok kolaydır. Donanımının bir kaç tümdevreden oluşması nedeniyle baskılı devresinin tasarımı kolaydır. 6301 serisi için yapılmış bir çeviriciyle yazılımı kolaylıkla yapılabilir. Komut kümesinin Motorola 6800 serisinin komut kümesine benzemesi nedeniyle makine dilini anlamak ve bu konu hakkında kaynak bulmak kolaydır.

Ayrıca yapımcı Hitachi firmasının bu tümdevreler için hazırladığı kullanım, yazılım ve donanım uygulamaları kitaplarında da yeterli bilgi bulunmaktadır.

3.2 HD63F01M1 MCU ile Gerçekleştirilen 8 Hane X 8 Parçalı LED Göstergeyi Dinamik Sürme Uygulaması

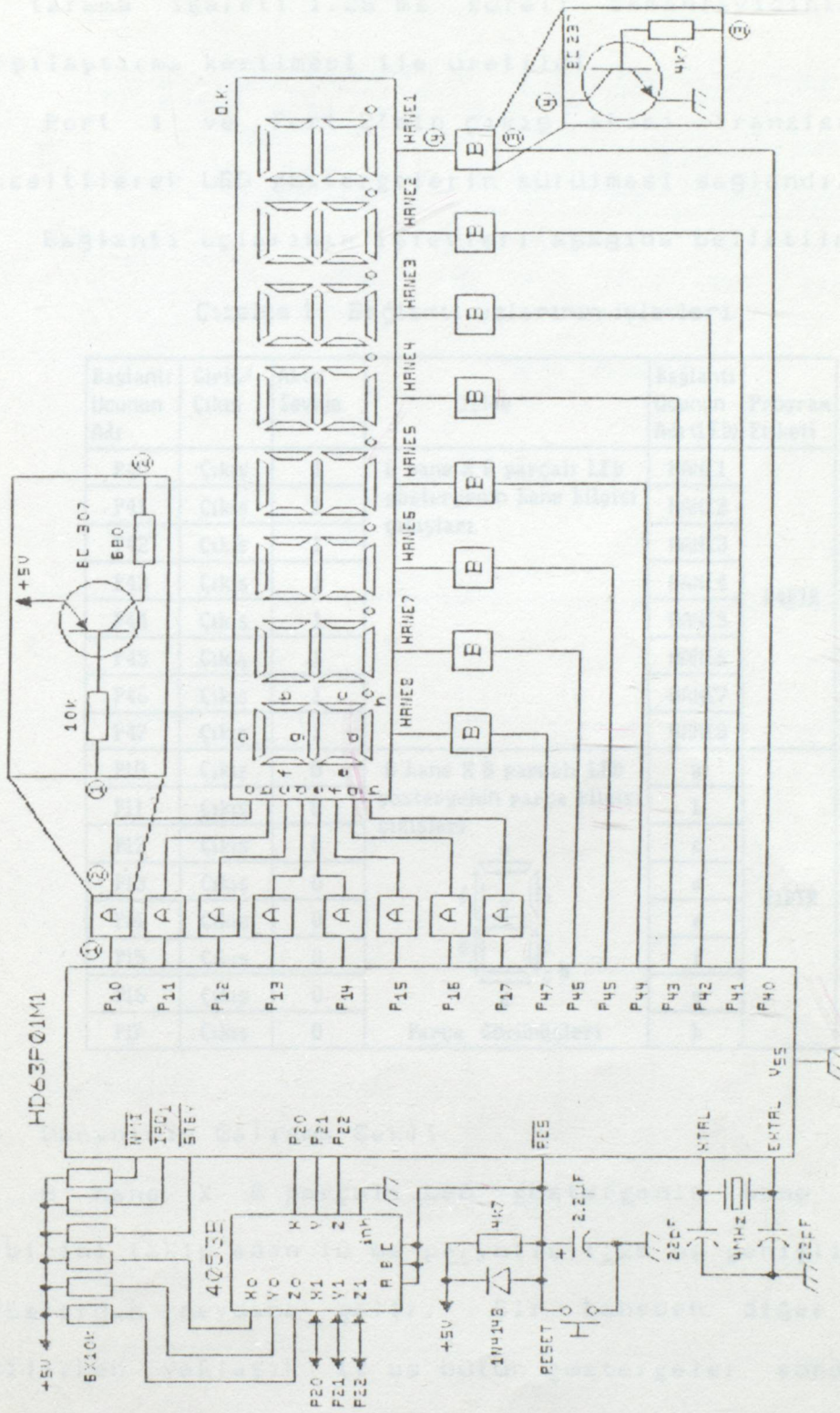


Şekil 3.1 Gerçekleştirilen sistemin blok diyagramı

3.3 Uygulamanın Donanım Özellikleri

HD63F01M1 ile 8 hane X 8 parçalı LED gösterge dinamik olarak sürüldü.

MCU saat frekansı 1 MHz seçildi ve yonga üzerindeki osilatöre 4 MHz'lik bir kristal bağlanarak üretildi.



Şekil 3.2 8 hane X 8 parçalı LED göstergesi dinamik süren devre.

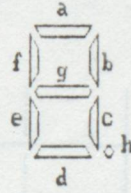
Tüm göstergenin tarama süresi 10 ms (100 Hz) seçildi ve tarama işareti 1.25 ms süreli zamanlayıcının çıkış karşılaştırma kesilmesi ile üretildi.

Port 1 ve Port 2'nin çıkış akımı tranzistor ile yükseltilerek LED göstergelerin sürülmesi sağlandı.

Bağlantı uçlarının işlevleri aşağıda belirtilmiştir.

Çizelge 1 Bağlantı uçlarının işlevleri

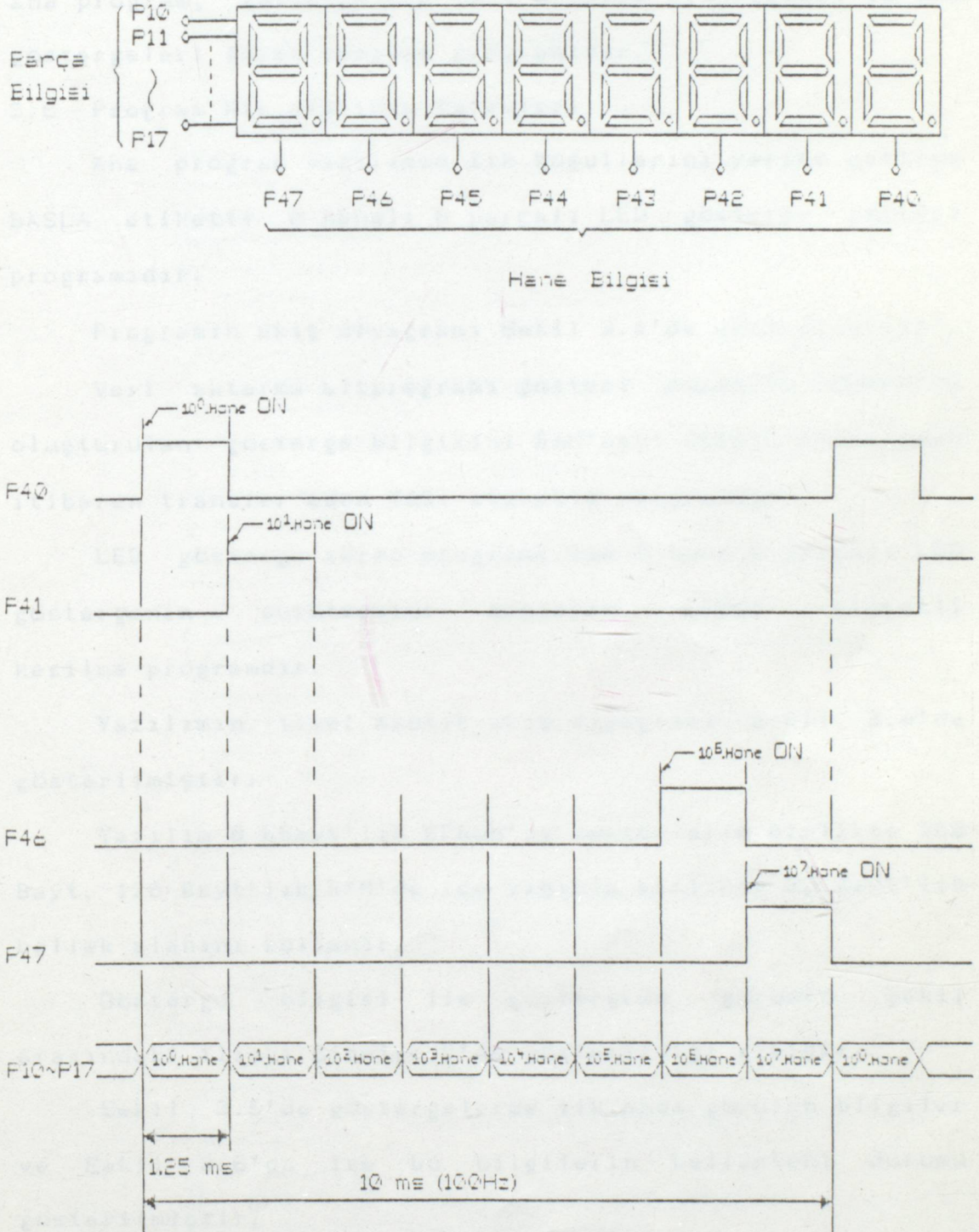
Bağlantı Ucunun Adı	Giriş/Çıkış	Aktif Seviye	İşlev	Bağlantı Ucunun Adı (LED)	Program Etiketi
P40	Çıkış	1	8 hane X 8 parçalı LED göstergenin hane bilgisi çıkışları	HANE1	P4DTR
P41	Çıkış	1		HANE2	
P42	Çıkış	1		HANE3	
P43	Çıkış	1		HANE4	
P44	Çıkış	1		HANE5	
P45	Çıkış	1		HANE6	
P46	Çıkış	1		HANE7	
P47	Çıkış	1		HANE8	
P10	Çıkış	0	8 hane X 8 parçalı LED göstergenin parça bilgisi çıkışları	a	P1DTR
P11	Çıkış	0		b	
P12	Çıkış	0		c	
P13	Çıkış	0		d	
P14	Çıkış	0		e	
P15	Çıkış	0		f	
P16	Çıkış	0		g	
P17	Çıkış	0		h	



Parça Görünüşleri

3.4 Donanımın Çalışma Şekli

8 hane X 8 parçalı LED göstergenin hane bilgisi birbirini takip eden 10 ms periyotlu 1.25 ms genişliğindeki darbelerden meydana gelir. Bir haneden diğer haneye geçilirken yaklaşık 17 us bütün göstergeler söndürülür. Şekil 3.3'de çalışma prensibi ayrıntılı olarak gösterilmiştir.



Şekil 3.3 8 hane X 8 parçalı LED göstergesi dinamik sürme düzeni.

3.5 Yazılım Özellikleri

Yazılım başlıca üç kısımdan meydana gelmiştir. Bunlar ana program, göstergelere veri aktarma altprogramı ve LED göstergeleri süren kesilme programıdır.

3.6 Program Kısımlarının İşlevleri

Ana program yazılımın ilk koşullarını yerine getiren BASLA etiketli 8 haneli 8 parçalı LED gösterge gösteri programıdır.

Programın akış diyagramı Şekil 3.4'de gösterilmiştir.

Veri aktarma altprogramı gösteri amacıyla EPROM'da oluşturulan gösterge bilgisini RAM'deki GOSBIL adresinden itibaren transfer eden TASI etiketli programdır.

LED gösterge sürme programı ise 8 hane 8 parçalı LED göstergenin sürülmesini sağlayan LEDGS etiketli kesilme programıdır.

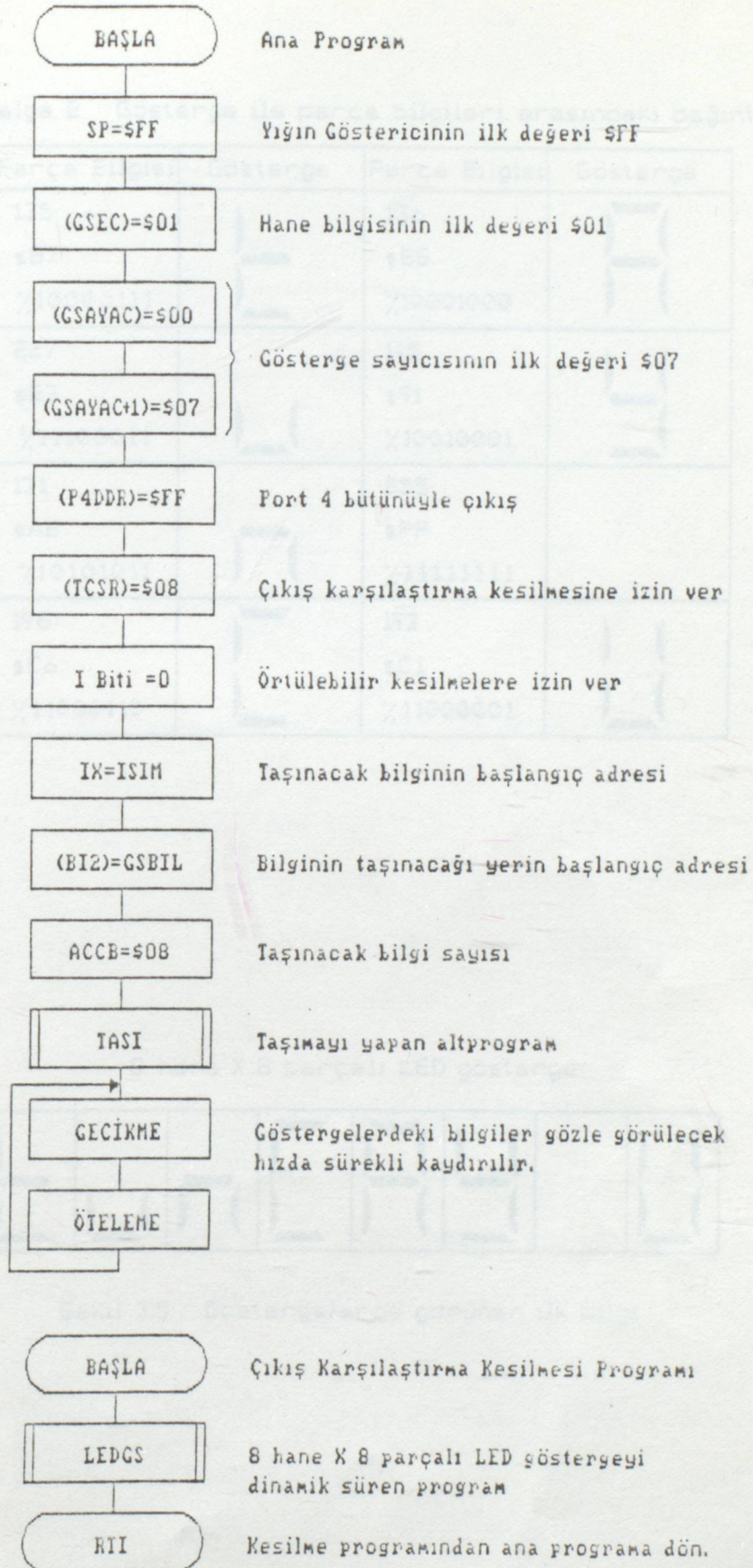
Yazılımın temel mantık akış diyagramı Şekil 3.4'de gösterilmiştir.

Yazılım 8 KBayt'lık EPROM'da vektörlerle birlikte 158 Bayt, 128 Bayt'lık RAM'de ise yığınla birlikte 22 Bayt'lık bellek alanını kullanır.

Gösterge bilgisi ile göstergede görünen şekil arasındaki ilişki Çizelge 2'de gösterildiği gibidir.





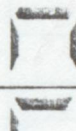


Şekil 3.5'de göstergelerde ilk anda görülen bilgiler ve Şekil 3.6'da ise bu bilgilerin bellekteki durumu gösterilmiştir.

Şekil 3.7'de gerçekleştirilen sistemin bellek haritası gösterilmiştir.

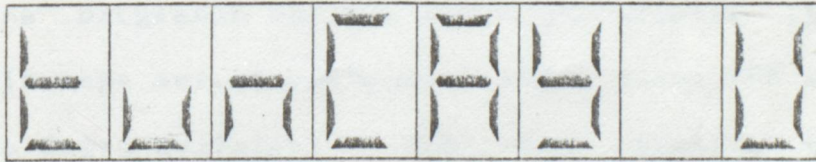


Şekil 3.4 Programın Akış diyagramı

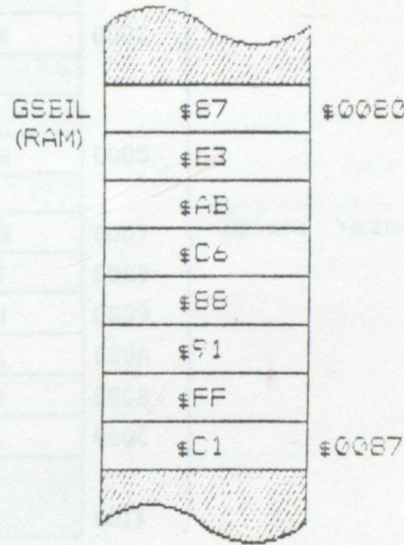
Çizelge 2 Gösterge ile parça bilgileri arasındaki bağlantı

Parça Eilgisi	Gösterge	Parça Bilgisi	Gösterge
135 \$B7 %10000111		136 \$B8 %10001000	
227 \$E3 %11100011		145 \$91 %10010001	
171 \$AB %10101011		255 \$FF %11111111	
198 \$C6 %11000110		193 \$C1 %11000001	

8 hane X 8 parçalı LED gösterge



Şekil 3.5 Göstergelerde görünen ilk bilgi



Şekil 3.6 Gösterge bilgilerinin bellek haritası

3.7 Programın çalışma şekli

Verilen kaynak programda her satırın yanına açıklama konulmuş olmakla birlikte programın çalışmasının bir metin halinde anlatılmasında fayda vardır.

3.7.1 Ana Program

Ana programın başında yığın göstericiye içteki RAM belleğin tepe adresi yüklenerek yığın alanı \$FF adresinden aşağıya doğru belirlenir. GSEC (\$8A) adresine (00000001) yüklenerek ilk göstergenin seçildiği belirlenir. GSAYAC (\$88) adresine (\$00) yüklenerek gösterge sayıcının üst baytı sıfırlanır. Alt baytına ise, (\$89) adresine 7 yüklenir. Bu sayıcı geriye sayarak taranan göstergelyi belirler. Sonra Port 1 ve Port 2 veri yönü yazmaçlarına \$FF bilgisi yüklenerek bu portlar bütünüyle çıkış yapılır.

P1DDR	0000
P1DTR	0002
P4DDR	0005
P4DTR	0007
TCSR	0008
FRCH	0009
FRCL	000A
OCRH	000B
OCRL	000C
	001F

İçteki Yazmaçlar

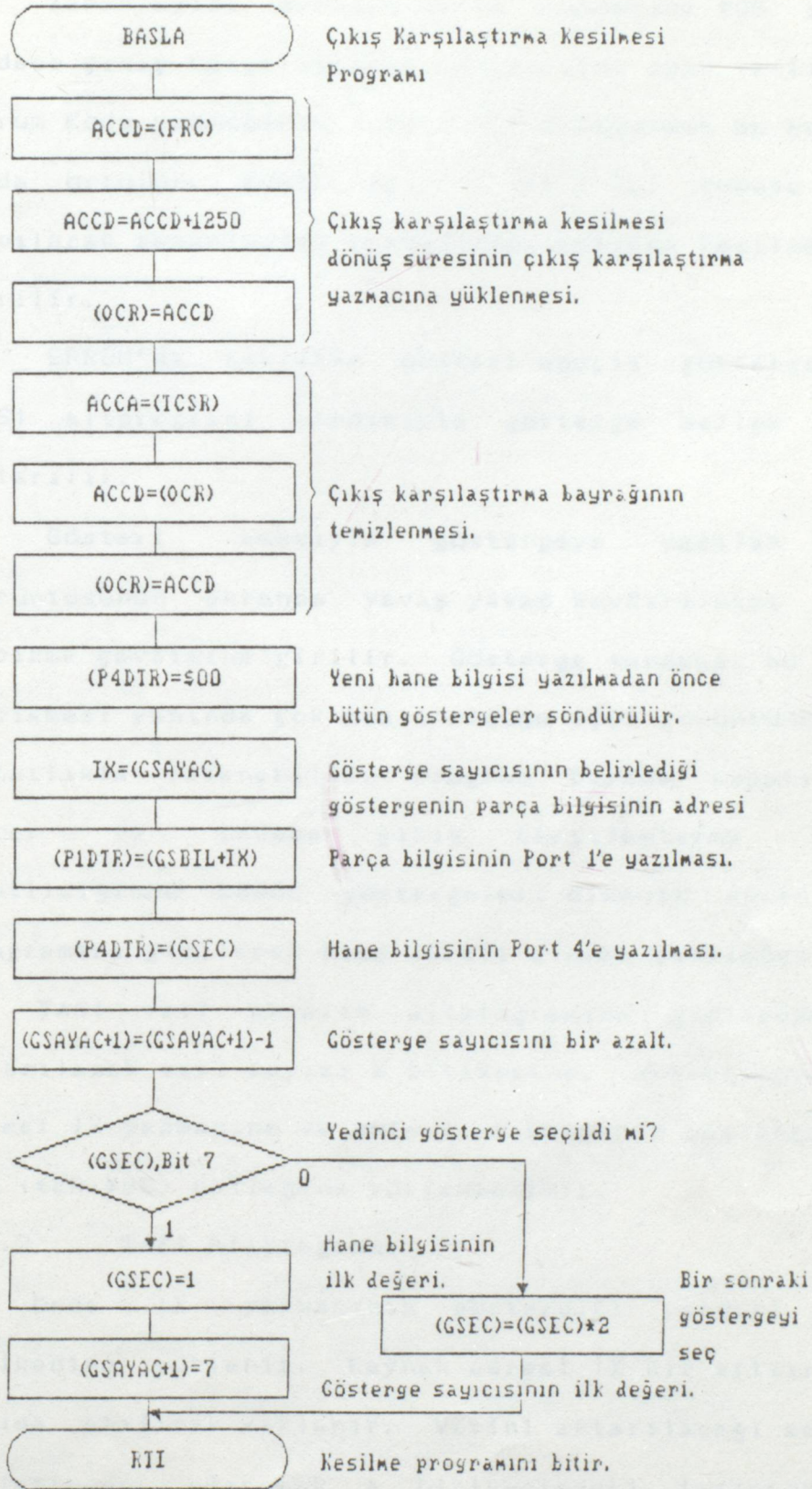
GSBIL	0080
	0087
GSAYAC	0088
GSAYAC+1	0089
GSEC	008A
BI2H	008B
BI2L	008C
↑ YIĞIN ↑	00FF

İçteki RAM

BASLA	E000
	E043
TASI	E100
	E10F
LEDGS	E400
	E42F
ISIM	E810
	E817
VEKTÖRLER	FFEE
	FFFF

Tümdevrenin üstündeki EPROM

Şekil 3.7 Sistemin bellek haritası



Şekil 3.E Kesilme programının akış diyagramı

Zamanlayıcı denetim/durum yazmacına #08 yüklenerek sadece çıkış karşılaştırma kesilmesine izin verilir. Fakat Durum Kodu yazmacının 1 biti "1" olduğundan bu kesilme ilk anda örtülür. Bunun için 1 biti CLI komutu ile "0" yapılarak zamanlayıcı çıkış karşılaştırma kesilmesine izin verilir.

EPR0M'da saklanan gösteri amaçlı gösterge bilgisi TASI altprogramı yardımıyla gösterge bellek bölgesine aktarılır.

Gösteri amacıyla göstergeye yazılan bilginin görüntüsünün ekranda yavaş yavaş kaydırılması için bir gecikme çevrimine girilir. Gösterge taraması bu kaydırma gecikmesi yanında çok hızlı olduğu için görüntünün kaydığı rahatlıkla izlenebilir. Program burada kapalı çevrime girer ve sadece çıkış karşılaştırma kesilmesi üretildiğinde LEDGS göstergeleri dinamik süren kesilme programına gidilerek kısa süreli olarak çevrimden çıkarılır.

TASI veri aktarım altprogramına gidilmeden önce aktarılacak veri sayısı B birikecine, aktarılacak verinin adresi IX yazmacına ve aktarılan bölgenin başlangıç adresi B12 (#8B:#8C) belleğine yüklenmelidir.

3.7.2 TASI Altprogramı

Önce IX yazmacının gösterdiği yerdeki veri A birikecine yüklenir. Kaynak adresi IX bir arttırılır ve yığına atılarak saklanır. Verini aktarılacağı adres B12 belleğinden alınarak A birikecindeki taşınacak veri aktarılır. B12 belleğindeki adres bilgisi bir eklenir ve

tekrar BI2 belleğinde saklanır. Kaynak adresi yığından geri alınarak IX yazmacına yüklenir. Aktarılacak veri sayısı yani B birikeci bir azaltılır ve sifıra eşit olana kadar yapılanlar tekrarlanır. B birikeci sıfır olduğunda bütün veriler aktarılmıştır ve RTS komutu işlenerek altprogramdan geri dönülür.

3.7.3 Çıkış Karşılaştırma Kesilmesi Programı

Serbest sayıcının içeriği alınarak buna 1250 eklenir. Özellikle bu sayının eklenmesinin nedeni sistem saatiyle sürülen sayıcının her adımında 1 mikrosaniyelik süre geçtiği için 1.25 milisaniyelik süreyi elde etmektir. Gelecekteki karşılaştırma işlemi için önce çıkış karşılaştırma bayrağı sıfırlanır ve yeni karşılaştırma değeri çıkış karşılaştırma yazmacına yüklenir.

Gösterge seçme işlemini yapan Port 4 sıfırlanır ve seçilen gösterge ve gösterge bilgisi belirlenir. Önce gösterge bilgisi Port 1'e yüklenir. Gösterge sayıcısı bir azaltılır. Son göstergenin seçilip seçilmediği test edilir ve seçildiyse ilk şartlanma yapılarak kesilme programından RTI komutu işlenerek ana programa dönülür. Eğer son gösterge seçilmediyse ASL kaydırma komutuyla bir sonraki gösterge seçilerek ana programa dönülür.

Bir sonraki çıkış karşılaştırma kesilmesinde bu işlemler tekrarlanarak devam eder.

Kaynak programı ve çıkış programı aşağıda verilmiştir.

OPT NOLIST,CREF
 NAM LEDGOS
 ASCT

```
*****
* BU PROGRAM 8 HANE X 8 PARCALI DINAMIK GOSTERGE DENETIM PROGRAMIDIR *
* TEK YONGA CALISMA SEKLI (MODE 7) *
* ZAMANLAYICI, EPROM, RAM, YIGIN, PORT 1, PORT 4 *
* CIKIS KARSILASTIRMA KESILMESI *
*****
```

OPT LIST

```
*****
* ICTEKI RAM *
*****
ORG $0080 128 BYTE RAM BASLANGIC ADRESI
GSBIL RMB 8 GOSTERGE BILGISI
GSAYAC RMB 2 GOSTERGE SAYICISI
GSEC RMB 1 GOSTERGE SECICISI
B12 RMB 2 2 BYTE'LIK BILGI SAKLAYICI
*****
```

```
*****
* ICTEKI YAZMAKLAR *
*****
P1DDR EQU $00 PORT 1 VERI YONU YAZMACI
P1DTR EQU $02 PORT 1 VERI YAZMACI
TCSR EQU $08 ZAMANLAYICI DENETIM DURUM YAZMACI
FRC EQU $09 SERBEST SAYICI
OCR EQU $0B CIKIS KARSILASTIRMA YAZMACI
P4DDR EQU $05 PORT 4 VERI YONU YAZMACI
F4DTR EQU $07 PORT 4 VERI YAZMACI
*****
```

```
*****
* ANA PROGRAM *
*****
ORG $E000
EASLA LDS #$00FF YIGIN GOSTERICI SP=$FF
LDAA #$01
STAA GSEC BIRINCI GOSTERGE SECILDI
CLR GSAYAC GOSTERGE SAYICISINI SIFIRLA
LDAA #7
STAA GSAYAC+1 GOSTERGE SAYICISI = 7
LDAA #$FF
STAA P1DDR PORT 1 BUTUNUYLA CIKIS
STAA P4DDR PORT 4 BUTUNUYLA CIKIS
LDAA #$08 ZAMANLAYICI CIKIS KARSILASTIRMA
STAA TCSR KESILMESINE IZIN VER
CLI ORTULEBILIR KESILMELERE IZIN VER
LDX #ISIM ISIM KARAKTERLERINI
LDD #GSBIL GOSTERGE BOLGESINE TASI
STD B12 GIDILECEK ADRES
LDAB #6 TASINACAK BILGI SAYISI
JSR TASI TASIMAYI YAPAN ALTPROGRAM
IEVAM LDAA #$03 GECIKME SAGLAYAN PROGRAM
LDX #40000
GECIK DEX
BNE GECIK
```

DECA		
BNE	GECIK	
LDAA	GSEIL	BIRINCI GOSTERGEDEKI BILGI
LDX	##0000	
KAYDIR	LDAB	GSBIL+1,X GOSTERGELEERDEKI BILGILILERI KAYDIR
	STAB	GSBIL,X
	INX	
	CPX	##0007
	BNE	KAYDIR
	STAA	GSBIL,X SEKIZINCI GOSTERGENIN BILGISI
	BRA	DEVAM

```
*****
* TASI : BELLEK BOLGELERINI TRANSFER EDEN ALTPROGRAM *
*****
* GIRIS : IX (TASINACAK BILGININ BASLANGIC ADRESI) *
* B12 (BILGININ TASINACAGI YERIN BASLANGIC ADRESI) *
* ACCB (TASINACAK BILGI SAYISI) *
*****
```

	ORG	##E100	
TASI	LDAA	0,X	(IX) BILGIYI A YAZMACINA YUKLE
	INX		KAYNAK ADRESINI BIR ARTIR
	PSHX		KAYNAK ADRESINI YIGINDA SAKLA
	LDX	B12	GIDILECEK YERIN ADRESINI IX'E YUKLE
	STAA	0,X	A BIRIKECINDEKI BILGIYI TASI
	INX		GIDILECEK YERIN ADRESINI BIR ARTIR
	STX	B12	VE BELLEKTE SAKLA
	PULX		KAYNAK ADRESINI YIGINDAN AL
	DECB		TASINACAK BILGI SAYISINI BIR AZALT
	BNE	TASI	SIFIR ISE ALTPROGRAMI BITIR.
	RTS		

```
*****
* LEDGS : 8 HANE X 8 PARCALI LED GOSTERGEYI DINAMIK SUREN *
* KESILME PROGRAMI *
*****
```

```
*****
* GIRIS : GSBIL (8 PARCALI LED GOSTERGE BILGI ADRESI) *
*****
```

LEDGS	ORG	##E400	
	LDD	FRC	SAYICIYI D YAZMACINA YUKLE
	ADDD	##1250	SAYICIYA 1250 EKLE
	STD	OCR	CIKIS KARSILASTIRMA YAZMACINA YUKLE
	LDAA	TCSR	CIKIS KARSILASTIRMA BAYRAGINI TEMIZLE
	LDD	OCR	
	STD	OCR	
	CLR	F4DTR	GOSTERGE SECICI PORTU TEMIZLE
	LDX	GSAYAC	SECILEN GOSTERGENIN YERINI BELIRLE
	LDAA	GSBIL,X	GOSTERGE BILGISINI AL
	STAA	F1DTR	VE BILGI PORTUNA YUKLE
	LDAA	GSEC	SECILEN GOSTERGEYI BELIRLE
	STAA	F4DTR	GOSTERGE SECME PORTUNA YUKLE
	DEC	GSAYAC+1	GOSTERGE SAYICISINI BIR AZALT
	BTST	7,GSEC	YEDINCI GOSTERGE SECILDIMI?
	BNE	LEDGS1	SECILMEDIYSE DEVAM
	ASL	GSEC	BIR SONRAKI GOSTERGEYI SEC
	BRA	LEDGS2	VE KESILME PROGRAMINI BITIR

```

LEDGS1 LDAA #401
        STAA GSEC      BIRINCI GOSTERGEYI SEC
        LDAA #7
        STAA GSAYAC+1  GOSTERGE SAYICISINA 7 YUKLE
LEDGS2 RTI      KESILME PROGRAMINI BITIR

```

```

*****
*   BILGI TABLOSU   *
*****

```

```

        ORG   $E610      TUNCAY U GOSTERGE BILGISI
ISIM   FCB   $87,$E3,$AB,$C6  TUNC
        FCB   $88,$91,$FF,$C1  AY U

```

```

*****
*   VEKTOR ADRESLERI   *
*****

```

```

        ORG   $FFEE
        FDB   BASLA  TRAP  KOMUT VE ADRES HATASI
        FDB   BASLA  SCI   SERI ILETISIM ARABIRIMI
        FDB   BASLA  TOF   ZAMANLAYICI TASTI BAYRAGI
        FDB   LEDGS  OCF   CIKIS KARSILASTIRMA BAYRAGI
        FDB   BASLA  ICF   GIRIS YAKALANDI BAYRAGI
        FDB   BASLA  IRQ   ORTULEBILIR KESILME
        FDB   BASLA  SWI   YAZILIM ILE KESILME
        FDB   BASLA  NMI   ORTULEMEZ KESILME
        FDB   BASLA  RES   RESET

```

```

*
END

```

LEDCOS

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
 YILDIZ UNIVERSITESI ELEKTRONİK ve HAB. BÖL. * Ar.Gör.Tuncay UZUN *
 Page: 1

		OPT	LIST
10A 0000			
11		*****	
12		* ICTEKI RAM	
13		*****	
14	0080	ORG	\$0080 12B BYTE RAM BASLANGIC
15A 0080	0008	GSEIL RMB	6 GOSTERGE BILGISI
16A 0088	0002	GSAYAC RMB	2 GOSTERGE SAYICISI
17A 008A	0001	GSEC RMB	1 GOSTERGE SECICISI
18A 008B	0002	B12 RMB	2 2 BYTE'LIK BILGI SAKLAY
19		*****	
20		* ICTEKI YAZMACLAR	
21		*****	
22	0000	P1DDR EQU	\$00 PORT 1 VERI YONU YAZMAC
23	0002	P1DTR EQU	\$02 PORT 1 VERI YAZMACI
24	0008	TCSR EQU	\$08 ZAMANLAYICI DENETIM DUR
25	0009	FRC EQU	\$09 SERBEST SAYICI
26	000B	OCR EQU	\$0B CIKIS KARSILASTIRMA YAZ
27	0005	P4DDR EQU	\$05 PORT 4 VERI YONU YAZMAC
28	0007	P4DTR EQU	\$07 PORT 4 VERI YAZMACI
29		*****	
30		* ANA PROGRAM	
31		*****	
32	E000	ORG	\$E000
33A E000	8E 00FF	BASLA LDS	\$\$00FF YIGIN GOSTERICI SP=\$FF
34A E003	85 01	LDAA	\$\$01
35A E005	97 8A	STAA	GSEC BIRINCI GOSTERGE SECILD
36A E007	7F 0088	CLR	GSAYAC GOSTERGE SAYICISINI SIF
37A E00A	86 07	LDAA	\$7
38A E00C	97 89	STAA	GSAYAC+1 GOSTERGE SAYICISI = 7
39A E00E	86 FF	LDAA	\$\$FF
40A E010	97 00	STAA	F1DDR PORT 1 BUTUNUYLE CIKIS
41A E012	97 05	STAA	F4DDR PORT 4 BUTUNUYLE CIKIS
42A E014	86 08	LDAA	\$\$08 ZAMANLAYICI CIKIS KARSI
43A E016	97 08	STAA	TCSR KESILMESINE IZIN VER
44A E018	0E	CLI	ORTULEBILIR KESILMELERE
45A E019	CE E810	LDX	\$\$1SIM ISIM KARAKTERLERINI
46A E01C	CC 0080	LDD	\$\$GSEIL GOSTERGE BOLGESINE TASI
47A E01F	DD 8B	STD	E12 GIDILECEK ADRES
48A E021	C6 08	LDAB	\$8 TASINACAK BILGI SAYISI
49A E023	ED E100	JSR	TASI TASIMAYI YAPAN ALTIROGR
50A E026	86 03	DEVAM LDAA	\$\$03 GECIKME SAGLAYAN PROGRA
51A E028	CE 0000	LDX	\$\$0000
52A E02B	09	GECIK DEX	
53A E02C	26 FD	BNE	GECIK
54A E02E	4A	DECA	
55A E02F	26 FA	BNE	GECIK
56A E031	96 80	LDAA	GSEIL BIRINCI GOSTERGELEKI EI
57A E033	CE 0000	LDX	\$\$0000
58A E036	E6 81	KAYDIR LDAB	GSEIL+1,X GOSTERGELEMEKI BILGILI
59A E038	E7 80	STAB	GSEIL,X

LEDGOS

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986

YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *

Page: 2

```

60A E03A 08                INX
61A E03B 8C 0007          CPX    #40007
62A E03E 26 F6           BNE    KAYDIR
63A E040 A7 80           STAA  GSEIL,X      SEKIZINCI GOSTERGENIN E
64A E042 20 E2           BRA    DEVAM
65
66 *      TASI    : BELLEK BOLGELERINI TRANSFER EDEN ALTP
67 *****
68 *      GIRIS   : IX    (TASINACAK BILGININ BASLANGIC A
69 *                B12  (BILGININ TASINACAGI YERIN BASL
70 *                ACCB (TASINACAK BILGI SAYISI)
71 *****
72                ORG    #E100
73A E100 A6 00          TASI  LDAA  0,X      (IX) BILGIYI A YAZMACIN
74A E102 08            INX
75A E103 3C            PSHX
76A E104 DE 88          LDX   B12          KAYNAK ADRESINI YIGINDA
77A E106 A7 00          STAA  0,X          KAYNAK ADRESINI YIGINDA
78A E108 08            INX                GIDILECEK YERIN ADRESIN
79A E109 DF 88          STX   B12          A BIRIKECINDEKI BILGIYI
80A E10B 38            PULX                GIDILECEK YERIN ADRESIN
81A E10C 5A            DECB                VE BELLEKTE SAKLA
82A E10D 2E F1          BNE   TASI          KAYNAK ADRESINI YIGINDA
83A E10F 39            RTS                TASINACAK BILGI SAYISIN
84 *****                SIFIR ISE ALTPROGRAMI B
85 *      LEDGS   : 8 HANE X 8 PARCALI LED GOSTER
86 *                KESILME PROGRAMI
87 *****
88 *      GIRIS   : GSBIL (8 PARCALI LED GOSTER
89 *****
90                ORG    #E400
91A E400 DC 09          LEDGS LDD   FRC      SAYICIYI D YAZMACINA YU
92A E402 C3 04E2        ADDD  #1250        SAYICIYA 1250 EKLE
93A E405 DD 0B          STD   OCR          CIKIS KARSILASTIRMA YAZ
94A E407 96 0B          LDAA  TCSR         CIKIS KARSILASTIRMA BAY
95A E409 DC 0B          LDD   OCR
96A E40B DD 0B          STD   OCR
97A E40D 7F 0007        CLR   F4DTR        GOSTERGE SECICI PORTU T
98A E410 DE 88          LDX   GSAYAC       SECILEN GOSTERGENIN YER
99A E412 A6 80          LDAA  GSBIL,X      GOSTERGE BILGISINI AL
100A E414 97 02         STAA  F1DTR        VE BILGI PORTUNA YUKLE
101A E416 96 8A          LDAA  GSEC         SECILEN GOSTERGEYI BELI
102A E418 97 07         STAA  F4DTR        GOSTERGE SECME PORTUNA
103A E41A 7A 0089        DEC   GSAYAC+1     GOSTERGE SAYICISINI BIR
104A E41D 7B 80 8A      BTST  7,GSEC       YEDINCI GOSTERGE SECILD
105A E420 26 05         BNE   LEDGS1       SECILMEDYSE DEVAM
106A E422 78 008A      ASL   GSEC         BIR SONRAKI GOSTERGEYI
107A E425 20 08         BRA   LEDGS2       VE KESILME PROGRAMINI B
108A E427 86 01         LEDGS1 LDAA  #401
109A E429 97 8A         STAA  GSEC         BIRINCI GOSTERGEYI SEC

```

LEDGOS

Hitachi 6301 Structured Cross Assembler Version 1.10 August 1986
 YILDIZ UNIVERSITESI ELEKTRONIK ve HAB. BOL. * Ar.Gor.Tuncay UZUN *
 Page: 3

```

110A E42B 66 07          LDAA    #7
111A E42D 97 89          STAA   GSAYAC+1      GOSTERGE SAYICISINA 7 Y
112A E42F 3B          LEDGS2 RTI          KESILME PROGRAMINI BITI
113                      *****
114                      *      BILGI TABLOSU
115                      *****
116          E810          ORG    $E810          TUNCAY U GOSTER
117A E810 87 E3 AB C6   ISIM   FCB    $87,$E3,$AB,$C6      TUNC
118A E814 88 91 FF C1   FCB    $88,$91,$FF,$C1          AY U
119                      *****
120                      *      VEKTOR ADRESLERI
121                      *****
122          FFEE          ORG    $FFEE
123A FFEE E000          FDB    BASLA  TRAP      KOMUT VE ADRES HATASI
124A FFF0 E000          FDB    BASLA  SCI       SERI ILETISIM ARABIRIMI
125A FFF2 E000          FDB    BASLA  TOF       ZAMANLAYICI TASTI BAYRA
126A FFF4 E400          FDB    LEDGS  OCF       CIKIS KARSILASTIRMA BAY
127A FFF6 E000          FDB    BASLA  ICF       GIRIS YAKALANDI BAYRAGI
128A FFF8 E000          FDB    BASLA  IRQ       ORTULEBILIR KESILME
129A FFFA E000          FDB    BASLA  SWI       YAZILIM ILE KESILME
130A FFFC E000          FDB    BASLA  NMI       ORTULEMEZ KESILME
131A FFFE E000          FDB    BASLA  RES       RESET
132                      *
133                      END

```

Symbol Table:

```

E000 BASLA 00033*00125 00124 00125 00127 00128 00129 00130 00131
006B E12 00016*00047 00076 00079
E026 DEVAM 00050*00064
0009 FRC 00025*00091
E02B GECIK 00052*00053 00055
008B GSAYAC 00016*00036 00038 00098 00103 00111
0080 GSBIL 00015*00046 00056 00058 00059 00063 00099
008A GSEC 00017*00035 00101 00104 00106 00109
E810 ISIM 00045 00117*
E036 KAYDIR 00058*00062
E400 LEDGS 00091*00126
E427 LEDGS1 00105 00108*
E42F LEDGS2 00107 00112*
000B DCR 00026*00093 00095 00096
0000 P1DDR 00022*00040
0002 P1DTR 00023*00100
0005 F4DDR 00027*00041
0007 F4DTR 00028*00097 00102
E100 TASI 00049 00073*00082
000E TCSR 00024*00043 00094

```

RUN COMPLETE -

0 ERRORS DETECTED

HD63P01M1 HD63P01M1 HD63PB01M1 CMOS MCU

The HD63P01M1 is a high performance CMOS MCU. It is the smallest MCU with an 8K ROM. The HD63P01M1 is available in a variety of packages to meet your needs.

- 128 Bytes
- 8K ROM
- 40 Pins
- 128 Bytes of Data
- 25 KHz to 500 KHz
- 10 Bit Programmable Timer
- Serial Communication Interface
- Low Power Consumption Mode
- Sleep Mode, Standby Mode
- Monitor Mode, Lock Mode
- Watchdog Timer
- 2.5V to 5.5V
- 100 nA Standby Current
- 100 nA Sleep Current
- 100 nA Standby Current
- 100 nA Sleep Current

EK A

HD63P01M1 CMOS MCU KATALOG BİLGİLERİ

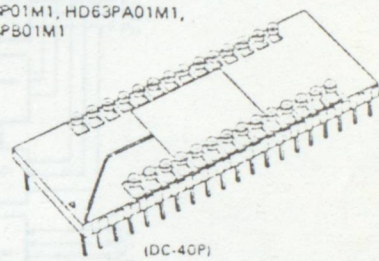
HD63P01M1	100
HD63P01M1	100
HD63P01M1	100
HD63P01M1	100

HD63P01M1, HD63PA01M1, HD63PB01M1 CMOS MCU (Microcomputer Unit)

The HD63P01M1 is an 8-bit single chip Microcomputer Unit (MCU) which has 4096 bytes or 8192 bytes of EPROM on the package. It is pin and function (except ROM) compatible with the HD6301V1. The HD63P01M1 can be used to emulate the HD6301V1 for software development or it can be used in production to allow for easy firmware changes with minimum delay.

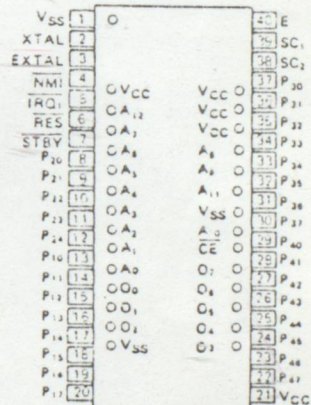
— The specifications for HD63PA01M1 and HD63PB01M1 are preliminary. —

HD63P01M1, HD63PA01M1,
HD63PB01M1



■ PIN ARRANGEMENT

HD63P01M1, HD63PA01M1, HD63PB01M1



(Top View)

(NOTE) EPROM is not included.

■ FEATURES

- Pin Compatible with HD6301V1
- On Chip Function Compatible with HD6301V1
 - 128 Bytes of RAM
 - 29 Parallel I/O
 - 16 Bit Programmable Timer
 - Serial Communication Interface
- Low Power Consumption Mode
 - Sleep Mode, Standby Mode
- Minimum Instruction Cycle Time
 - 1 μ s (f = 1MHz), 0.67 μ s (f = 1.5MHz), 0.5 μ s (f = 2MHz)
- Bit Manipulation, Bit Test Instruction
- Protection from System Upset
 - Address Trap, Op-Code Trap
- Up to 65k Words Address Space
- Applicable to 4k or 8k Bytes of EPROM
 - 4096 Bytes: HN482732A
 - 8192 Bytes: HN482764, HN27C64

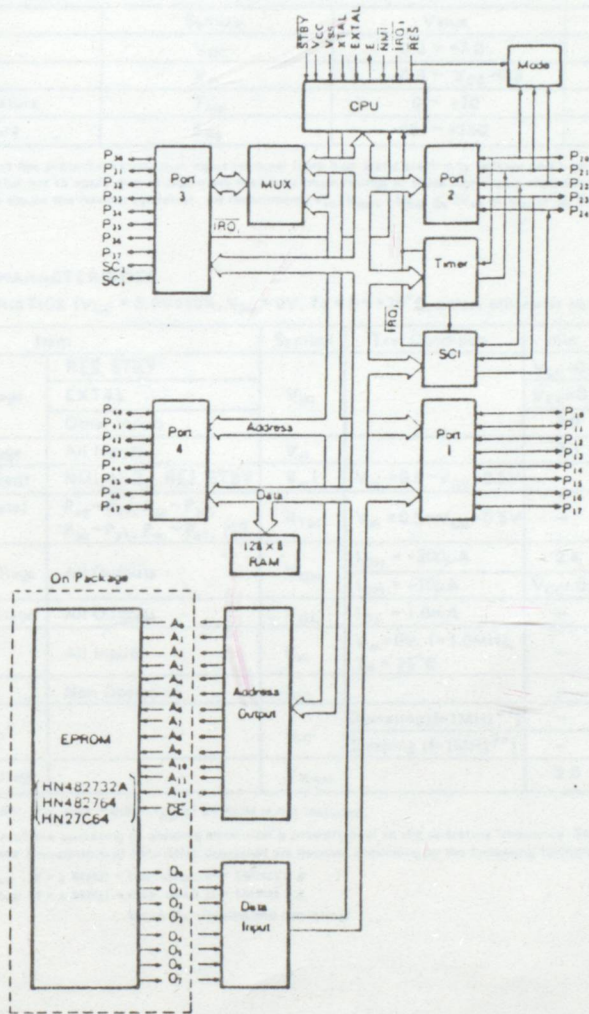
■ TYPE OF PRODUCTS

Type No.	Bus Timing	EPROM Type No.
HD63P01M1	1MHz	HN482732A-30, HN482764-3, HN27C64-30
HD63PA01M1*	1.5MHz	HN482732A-30, HN482764-3, HN27C64-30
HD63PB01M1*	2MHz	HN482732A-25, HN482764, HN27C64-25

* Preliminary

HD63P01M1, HD63PA01M1, HD63PB01M1

■ BLOCK DIAGRAM



HD63P01M1, HD63PA01M1, HD63PB01M1

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage	V_{in}	-0.3 ~ $V_{CC}+0.3$	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend $V_{in}, V_{out} - V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

● DC CHARACTERISTICS ($V_{CC} = 5.0V \pm 10\%$, $V_{SS} = 0V$, $T_a = 0 \sim +70^\circ C$, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit	
Input "High" Voltage	RES, STBY		$V_{CC}-0.5$	-	$V_{CC} \pm 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	-			
	Other Inputs		2.0	-			
Input "Low" Voltage	All Inputs	V_{IL}	-0.3	-	0.8	V	
Input Leakage Current	NMI, IRQ ₁ , RES, STBY	I_{in}	$V_{in} = 0.5 \sim V_{CC} - 0.5V$	-	-	1.0	μA
Three State (off-state) Leakage Current	P ₁₀ ~P ₁₇ , P ₂₀ ~P ₂₄ , P ₃₀ ~P ₃₇ , P ₄₀ ~P ₄₇ , IS3	I_{TSI}	$V_{in} = 0.5 \sim V_{CC} - 0.5V$	-	-	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	$I_{OH} = -200\mu A$	2.4	-	-	V
			$I_{OH} = -10\mu A$	$V_{CC}-0.7$	-	-	V
Output "Low" Voltage	All Outputs	V_{OL}	$I_{OL} = 1.6mA$	-	-	0.55	V
Input Capacitance	All Inputs	C_{in}	$V_{in} = 0V, f = 1.0MHz, T_a = 25^\circ C$	-	-	12.5	pF
Standby Current	Non Operation	I_{CC}		-	2.0	15.0	μA
Current Dissipation*		I_{CC}	Operating (f=1MHz)**	-	6.0	10.0	mA
			Sleeping (f=1MHz)**	-	1.0	2.0	
RAM Stand-By Voltage		V_{RAM}		2.0	-	-	V

* $V_{IH} \text{ min} = V_{CC} - 1.0V, V_{IL} \text{ max} = 0.8V, I_{CC}$ of EPROM is not included.

** Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at f = x MHz operation are decided according to the following formula;

$$\begin{aligned} \text{typ. value (f = x MHz)} &= \text{typ. value (f = 1MHz)} \times x \\ \text{max. value (f = x MHz)} &= \text{max. value (f = 1MHz)} \times x \\ &\text{(both the sleeping and operating)} \end{aligned}$$

HD63P01M1, HD63PA01M1, HD63PB01M1

• AC CHARACTERISTICS ($V_{CC} = 5.0V \pm 10\%$, $V_{SS} = 0V$, $T_a = 0 \sim +70^\circ C$, unless otherwise noted.)
BUS TIMING

Item	Symbol	Test Condition	HD63P01M1			HD63PA01M1			HD63PB01M1			Unit
			min	typ	max	min	typ	max	min	typ	max	
Cycle Time	t_{CYC}		1	—	10	0.666	—	10	0.5	—	10	μs
Address Strobe Pulse Width "High"*	PW_{ASH}		220	—	—	150	—	—	110	—	—	ns
Address Strobe Rise Time	t_{ASr}		—	—	20	—	—	20	—	—	20	ns
Address Strobe Fall Time	t_{ASf}		—	—	20	—	—	20	—	—	20	ns
Address Strobe Delay Time*	t_{ASD}		60	—	—	40	—	—	20	—	—	ns
Enable Rise Time	t_{Er}		—	—	20	—	—	20	—	—	20	ns
Enable Fall Time	t_{Ef}		—	—	20	—	—	20	—	—	20	ns
Enable Pulse Width "High" Level*	PW_{EH}		450	—	—	300	—	—	220	—	—	ns
Enable Pulse Width "Low" Level*	PW_{EL}		450	—	—	300	—	—	220	—	—	ns
Address Strobe to Enable Delay* Time	t_{ASED}		60	—	—	40	—	—	20	—	—	ns
Address Delay Time	t_{AD1}	Fig. 1	—	—	250	—	—	190	—	—	160	ns
	t_{AD2}		—	—	250	—	—	190	—	—	160	ns
Address Delay Time for Latch*	t_{ADL}	Fig. 2	—	—	250	—	—	190	—	—	160	ns
Data Set-up Time	Write	t_{DSW}	230	—	—	150	—	—	100	—	—	ns
	Read	t_{DSR}	80	—	—	60	—	—	50	—	—	ns
Data Hold Time	Read	t_{HR}	0	—	—	0	—	—	0	—	—	ns
	Write	t_{HW}	20	—	—	20	—	—	20	—	—	ns
Address Set-up Time for Latch*	t_{ASL}		60	—	—	40	—	—	20	—	—	ns
Address Hold Time for Latch	t_{AHL}		30	—	—	20	—	—	20	—	—	ns
Address Hold Time	t_{AH}		20	—	—	20	—	—	20	—	—	ns
$A_0 \sim A_7$ Set-up Time Before E*	t_{ASM}		200	—	—	110	—	—	60	—	—	ns
Peripheral Read Access Time	Non-Multiplexed Bus*	t_{ACCN}	—	—	650	—	—	395	—	—	270	ns
	Multiplexed Bus*	t_{ACCM}	—	—	650	—	—	395	—	—	270	ns
Oscillator stabilization Time	t_{AC}	Fig. 10	20	—	—	20	—	—	20	—	—	ms
Processor Control Set-up Time	t_{PCS}	Fig. 11	200	—	—	200	—	—	200	—	—	ns

* These timings change in approximate proportion to t_{CYC} . The figures in this characteristics represent those when t_{CYC} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item	Symbol	Test Condition	HD63P01M1			HD63PA01M1			HD63PB01M1			Unit	
			min	typ	max	min	typ	max	min	typ	max		
Peripheral Data Set-up Time	Port 1, 2, 3, 4	t_{DSU}	Fig. 3	200	—	—	200	—	—	200	—	—	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4	t_{PDH}	Fig. 3	200	—	—	200	—	—	200	—	—	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition		t_{OSD1}	Fig. 5	—	—	300	—	—	300	—	—	300	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition		t_{OSD2}	Fig. 5	—	—	300	—	—	300	—	—	300	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2, 3, 4	t_{PWD}	Fig. 4	—	—	300	—	—	300	—	—	300	ns
Input Strobe Pulse Width		t_{PWS}	Fig. 6	200	—	—	200	—	—	200	—	—	ns
Input Data Hold Time	Port 3	t_{IH}	Fig. 6	150	—	—	150	—	—	150	—	—	ns
Input Data Setup Time	Port 3	t_{IS}	Fig. 6	0	—	—	0	—	—	0	—	—	ns

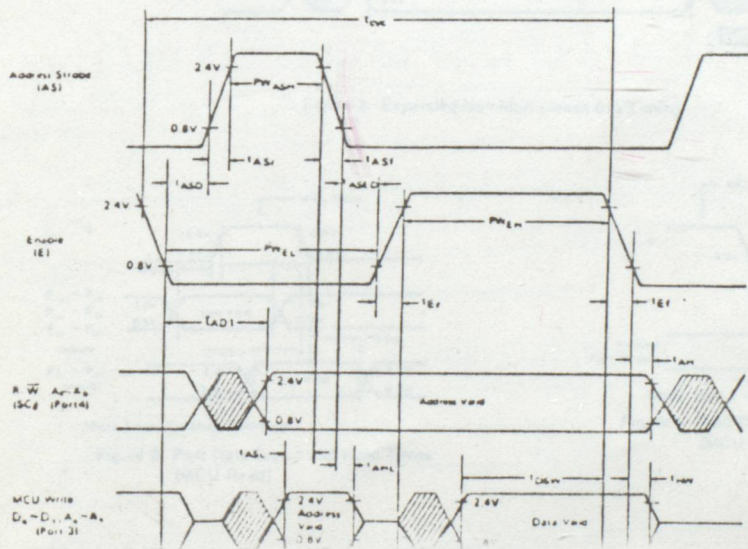
* Except P21

TIMER, SCI TIMING

Item	Symbol	Test Condition	HD63P01M1			HD63PA01M1			HD63PB01M1			Unit
			min	typ	max	min	typ	max	min	typ	max	
Timer Input Pulse Width	t_{PW_T}		2.0	-	-	2.0	-	-	2.0	-	-	t_{cyc}
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 7	-	-	400	-	-	400	-	-	400	ns
SCI Input Clock Cycle	t_{SCYC}		2.0	-	-	2.0	-	-	2.0	-	-	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	-	0.6	0.4	-	0.6	0.4	-	0.6	t_{SCYC}

MODE PROGRAMMING

Item	Symbol	Test Condition	HD63P01M1			HD63PA01M1			HD63PB01M1			Unit
			min	typ	max	min	typ	max	min	typ	max	
RES "Low" Pulse Width	PW_{RSTL}		3	-	-	3	-	-	3	-	-	t_{cyc}
Mode Programming Set-up Time	t_{MPS}	Fig. 8	2	-	-	2	-	-	2	-	-	t_{cyc}
Mode Programming Hold Time	t_{MPH}		150	-	-	150	-	-	150	-	-	ns



HD63P01M1, HD63PA01M1, HD63PB01M1

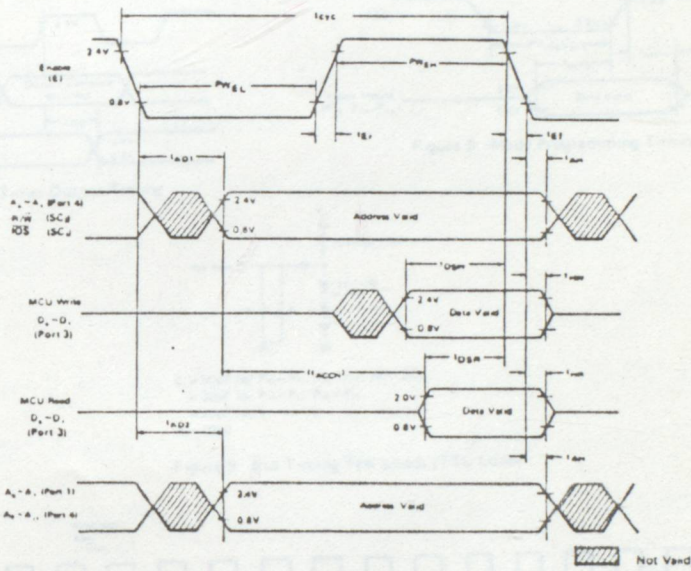
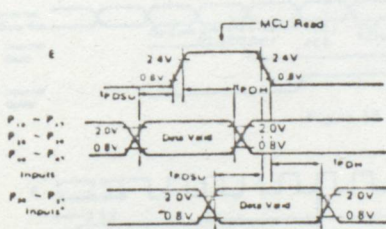
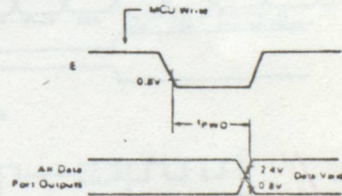


Figure 2 Expanded Non-Multiplexed Bus Timing



*Port 3 Non-Latched Operation
Figure 3 Port Data Set-up and Hold Times (MCU Read)



Note) Port 2 Except P_{11}
Figure 4 Port Data Delay Times (MCU Write)

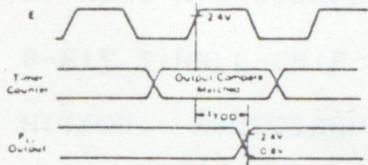


Figure 7 Timer Output Timing

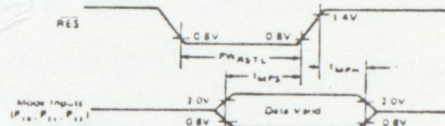
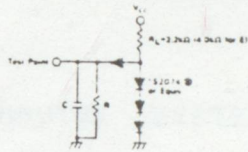


Figure 8 Mode Programming Timing



$C = 90\text{pF}$ for $P_{10}-P_{11}$; $P_{12}-P_{13}$; SC_1 , SC_2
 $= 30\text{pF}$ for $P_{14}-P_{15}$; $P_{16}-P_{17}$
 $= 40\text{pF}$ for E
 $R = 12\text{k}\Omega$

Figure 9 Bus Timing Test Loads (TTL Load)

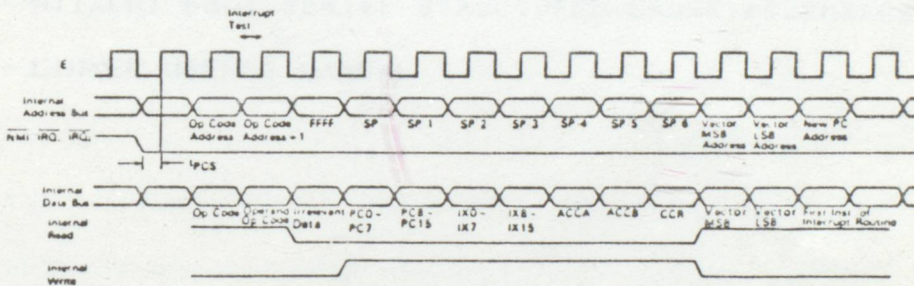


Figure 10 Interrupt Sequence

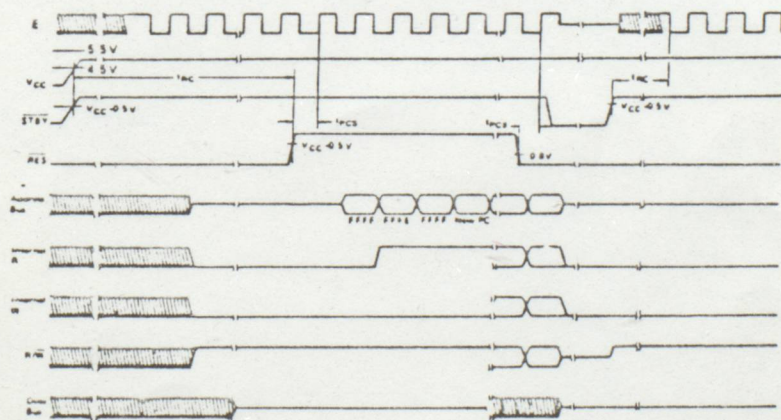


Figure 11 Reset Timing

KAYNAKÇA

1. HITACHI MICROCOMPUTER DATA BOOK
8-BIT SINGLE-CHIP
2. HITACHI MICROCOMPUTER SYSTEM
HD6301X0, HD6303X, HD63701X0 APPLICATION NOTES
-HARDWARE-
3. HITACHI MICROCOMPUTER SYSTEM
CMOS 8-BIT SINGLE CHIP MICROCOMPUTER UNIT (CMOS MCU)
HD6301X0, HD6303X. USER'S MANUAL
4. HITACHI 6301 SERIES STRUCTURED CROSS ASSEMBLER and
LINKER USER'S MANUAL

GEÇMİŞ

1963 yılında Şile'de doğmuşum. İlk öğrenimimi 1974 yılında Bartın Cumhuriyet İlkokulunda, orta öğrenimimi ise 1977 yılında İstanbul Halide Edip Adıvar Ortaokulunda tamamladım. Aynı yıl meslek lisesi giriş sınavını kazandım. İstanbul Kartal Endüstri Meslek Lisesinden 1980 yılında mezun oldum. 1981 yılında girdiğim Yıldız Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Bölümünden 1985 yılında mezun oldum. Aynı yıl, yüksek lisans eğitimi için Yıldız Üniversitesi Fen Bilimleri Enstitüsünün açmış olduğu giriş sınavını kazandım. Şu anda Elektronik ve Haberleşme Bölümünde yüksek lisans eğitimime devam etmekteyim ve 6 Mayıs 1986 tarihinden beri Yıldız Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü Devreler ve Sistemler Anabilim dalında araştırma görevlisi olarak çalışmaktayım.

