

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PIC MİKRODENETLEYİCİ TABANLI DENEY
MODÜLLERİ TASARIMI VE YAPIMI

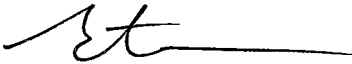
YÜKSEK LİSANS TEZİ

Hayrettin GÖKOZAN

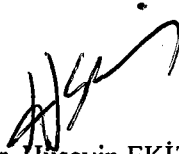
90917

Enstitü Anabilim Dalı: Elektronik ve Bilgisayar Eğitimi

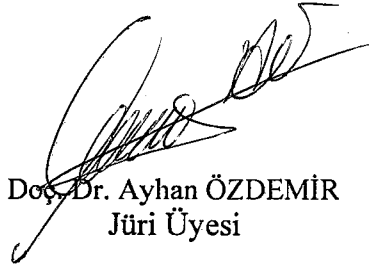
Bu tez .../.../1999 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



Doç. Dr. Ethem KÖKLÜKAYA
Jüri Başkanı



Doç. Dr. Hüseyin EKİZ
Jüri Üyesi



Yrd. Doç. Dr. Ayhan ÖZDEMİR
Jüri Üyesi

T.C. YÜKSEKÖĞRETİM BAKANLIĞI
DOKÜMANTASYON MERKEZİ

ÖNSÖZ

Çalışmamda bana desteklerini esirgemeyen Sayın Doç. Dr. Hüseyin EKİZ'e, mikroişlemcilerin programlanması ve devre dizaynında yardımcı olan Sakarya Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümü Araştırma Görevlisi Gürsel DÜZENLİ'ye, elektronik devrelerin ve deneylerin yapımında yardımcı olan Dumlupınar Üniversitesi Simav Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Bölümü Araştırma Görevlilerine ve çalışmalarım sırasında sonsuz sabır gösteren ailemin tüm fertlerine teşekkürü borç bilirim.

OCAK 1999

Hayrettin GÖKOZAN

İÇİNDEKİLER

ÖNSÖZ	II
İÇİNDEKİLER.....	III
SİMGELER ve KISALTMALAR.....	VI
ŞEKİLLER LİSTESİ.....	VIII
TABLolar LİSTESİ.....	X
ÖZET	XI
SUMMARY	XIII
BÖLÜM 1	
GİRİŞ.....	1
BÖLÜM 2	
PIC MİKRODENETLEYİCİ AİLESİ.....	3
2.1. PIC16C5X Microdenetleyicisi.....	4
2.1.1. PIC16C5X Microdenetleyicisinin genel özellikleri	4
2.1.2. PIC16C5x Microdenetleyicilerin kılıf yapıları	6
2.1.3. PIC'lerde Kullanılan frekans türleri	7
2.2. PIC'lerde Bulunan Bağlantı Uçları.....	9
2.2.1. PIC'in I/O uçları ve T0CKI girişi	10
2.2.2. Seçenek kayıtçısı (OPTION register).....	13
2.3. PIC'in Çalışma Şekli.....	14
2.3.1. Durum kayıtçısı (STATUS register)	16
2.4. PIC'in RAM Hafızası.....	18
2.5. PIC Programlama.....	21
2.5.1. Konfigurasyon kelimesi kayıtçısı.....	21
2.6. Saat Şeması/ Komut Çevrimi.....	22
2.7. Komut Akışı/Pipelining Hattı.....	23
BÖLÜM 3	
PIC PROGRAMLAMA KOMUTLARI.....	25

BÖLÜM 4

PROGRAMLAMA.....	40
4.1. Döngüler.....	40
4.2. Kayıtçı İçeriklerinin Karşılaştırılması.....	42
4.3. Programlamada Başlangıç Durumlarının Belirlenmesi.....	44
4.4. Dolaylı Yoldan Adresleme.....	45
4.5. Makro Programlama.....	46
4.6. Maskeleye.....	49

BÖLÜM 5

MODÜL TASARIMI ve YAPIMLARI.....	52
5.1. D.C. Motor Kontrol Devresi Tasarımı ve Yapımı.....	52
5.1.1. Doğru akım motorları.....	52
5.1.2. Doğru akım motorlarında zıt E.M.K. ve devir sayısı.....	54
5.1.3. D.C. motor kontrol devresi.....	55
5.2. Step Motor Modülü Tasarımı ve Yapımı.....	58
5.2.1. Step motorların yapısı ve çalışması.....	59
5.2.2. Step motor parametreleri ve uyarım yöntemleri.....	60
5.2.3. Step motor modülü.....	62
5.2.4. Devrenin çalışması.....	62
5.3. Trafik Işık Kontrolü Devresi Tasarımı ve Yapımı.....	65
5.3.1. Sinyalizasyon ve kavşaklar.....	65
5.3.2. Sinyalize yaya geçitleri ve tek yönlü yolların kesiştiği kavşaklar.....	66
5.3.3. Sola dönüşlerin az olduğu kavşaklar.....	67
5.3.4. Tek yönlü -T- kavşaklar.....	68
5.3.5. Ada çevresinde dönüş.....	69
5.3.6. Tasarlanan trafik ışık kontrolü devresi ve yol düzeni grafiği.....	69
5.4. Tuş Takımı Tasarımı ve Yapımı.....	73
5.4.1. Devre elemanları ve devrenin çalışması.....	73

BÖLÜM 6

SONUÇLAR ve ÖNERİLER.....	77
KAYNAKLAR.....	78
EKLER.....	79
Ek 1. D.C. Motor Programı.....	79

Ek 2. Step Motor Programı.....	81
Ek 3. Trafik Işıkları Programı	84
Ek 4. Keypad Programı.....	88
ÖZGEÇMİŞ	93



SİMGELER ve KISALTMALAR

ALU	Arithmetic Logic Unit
CERDIP	Ceramic Dual Inline Package
CLKIN	Clock In
CLKOUT	Clock Out
CMOS	Complementary Metal Oxide Semiconductor
DIL	Dual In Line
DIP	Dual Inline Package
DRT	Device Reset Timer
EEPROM	Electrical Erasable Read Only Memory
GND	Ground
HS	High Speed
I/O	In / Out
LP	Loop Power
MCLR	Main Clear
MCU	Microcontroller Unite
OSC	Oscillator
OTP	One Time Programmable
PC	Personal Computer
PIC	Programmable Integrated Circuit
POR	Power On Reset
RC	Resistor Capacitor
RISC	Reduce Instruction Set Computer
RTCC	Real Time Clock / Counter
ST	Schmitt Trigger
TOCS	Timer0 Clock Source Select Bit
TTL	Transistor Transistor Logic

VDD	Volt Digital Driver
WDT	Watch Dog Timer
XT	Ceramic Oscillator



ŞEKİLLER LİSTESİ

Şekil 2.1. PDIP, SSOP ve SOIC kılıflı PIC'ler.....	6
Şekil 2.2. PIC'lerdeki giriş/çıkış lojik seviyeleri.....	10
Şekil 2.3. MCLR ve TOCKI uçlarında konum değiştirme değerleri.....	11
Şekil 2.4. PIC'in TOCKI girişi.....	12
Şekil 2.5. Seçenek kayıtçısı (OPTION Register).....	14
Şekil 2.6. PIC'in çalışma şeması.....	15
Şekil 2.7. Durum kayıtçısı açık şeması.....	17
Şekil 2.8. PIC'in hafıza haritası.....	20
Şekil 2.9. PIC'in konfigürasyon kelime kayıtçısı.....	22
Şekil 2.10. PIC'in Saat şeması.....	23
Şekil 2.11. PIC'in komut çevrimi.....	23
Şekil 4.1. Sayma döngüsü.....	40
Şekil 4.2. Şartlı döngü program örneği.....	41
Şekil 4.3. Kayıtçı içeriğinin sabit bir sayı ile kontrol edilmesi.....	42
Şekil 4.4. Bir kayıtçının içeriğinin W ile karşılaştırılması.....	43
Şekil 4.5. PIC programlamada program başının belirlenmesi.....	44
Şekil 4.6. Dolaylı yoldan adresleme.....	46
Şekil 4.7. Makro programı genel yapısı.....	47
Şekil 4.8. Örnek makro programı.....	47
Şekil 4.9. Program içinde makro çağırma.....	48
Şekil 4.10. Makro parametresinin test kayıtçı ile toplanması.....	48
Şekil 5.1. İçinden akım geçen bir iletkenin manyetik alan içindeki durumu.....	53
Şekil 5.2. İçinden akım geçen bir bobinin alan içindeki ve kutuplar arasındaki endüvinin durumu.....	53
Şekil 5.3. D.C. motor hız kontrol devresi.....	56
Şekil 5.4. D.C. motor program akış şeması.....	57
Şekil 5.5. Step motorun çalışma prensibi.....	58

Şekil 5.6. Sabit mıknatıslı step motor.....	59
Şekil 5.7. Değişik yapıdaki step motor çeşitleri.....	61
Şekil 5.8. Step motor uyartım yöntemleri (İki, Üç ve Dört fazlı uyartım).	62
Şekil 5.9. Step motor kontrolü programı iş akış şeması.	63
Şekil 5.10. Step motor kontrol devre şeması.....	64
Şekil 5.11. Yaya geçidi fazları.	66
Şekil 5.12. Tek yönlü yaya kavşaklarındaki fazlar.....	66
Şekil 5.13. Sola dönüşün az olduğu kavşaklardaki fazlar.....	67
Şekil 5.14. Tek yönlü T kavşaklarda faz düzeni.	68
Şekil 5.15. Ada çevresinde dönüşe sahip kavşak.	69
Şekil 5.16. Trafik ışıkları kontrol devre şeması.	71
Şekil 5.17. Trafik ışık kontrolü iş akış şeması.	72
Şekil 5.18. Tuş takımı devre şeması.	74
Şekil 5.19. Tuş takımı program akış diyagramı.	75
Şekil 5.20. Tuş takımı program akış diyagramı(Devam).	76

TABLolar LİSTESİ

Tablo 2.1.a PIC16C5X Microdenetleyicileri ve özellikleri	5
Tablo 2.1.b PIC16C5X Microdenetleyicileri ve özellikleri	5
Tablo 2.2.a PIC'lerin osilatör türlerine göre sınıflandırılması	8
Tablo 2.2.b PIC'lerin osilatör türlerine göre sınıflandırılması	8
Tablo 2.3. PIC'lerde bağlantı uçları	9
Tablo 2.4. PIC16C5x'e ait özel fonksiyon kayıtçısı	19
Tablo 4.1. Farklı PIC'lerin EPROM hafızaları	45
Tablo 5.1. Step motor uyarım yöntemleri	61

ÖZET

Anahtar kelimeler: PIC, Mikrodenetleyiciler.

PIC mikrodenetleyiciler, dünyanın en çok tercih edilen 8 bit RISC (Reduce Instruction Set Computer) mikrodenetleyicileridir. Bu işlemciler, 1990'dan bu güne kadar yarım trilyon işlemci ve 100000 programlama sistemi satışı ile bir dünya standardı olmuştur. PIC'i üreten Arizona Microchip, sadece geçen iki yıl içinde 58 adet yeni PIC üretmiştir. Bunun yanında önümüzdeki günlerde 44 yeni tür PIC daha satışa sunulacaktır. Microchip PIC işlemciler diğer işlemcilere göre; daha yüksek performans, aynı yapıdaki aile gurupları, çoklu hafıza teknolojisi ve kolay kullanımlı programlama sistemine sahiptir. Bu özellik, aynı bilgi birikimi ile daha hızlı ve daha kolay tasarımlar yapma olanağı sağlar.

Microchip Technology'nin PIC16C5x mikrodenetleyicisi; fiyat bakımından en ucuz işlemcisi olmasının yanında 8 bitli, yüksek performanslı, tamamen statik, EPROM/ROM temelli CMOS teknolojisi özelliklerine sahiptir. PIC16C5x aynı fiyat sınıfındaki mikroişlemcilere göre çok daha büyük bir performans sağlar. 12 bit uzunluğundaki komutu, diğer 8 bitlik mikroişlemcilere göre çok daha büyük komut sıkıştırma (2:1) ile çalışır. Tüm komutlar tek adımda işlenirken(200 ns), sadece atlamalar iki adımda gerçekleşir.

Komutlar basit ve kolay hatırlandığı için tasarım aşaması da zaman kazandırmaktadır. PIC16C5x ürünleri, özel donanımı sayesinde sistem maliyetini düşürür ve güç harcamasını da azaltır. Sahip olduğu POR (Power On Reset-Gerilim verildiğinde Reset yap) ve DRT (Device Reset Timer-Element Reset süresi) sayesinde harici reset devrelerine gerek kalmaz. Düşük güçlü osilatörden ucuz osilatör yapılarına kadar dört farklı osilatör ile çalışabilirler. SLEEP modu,

Watchdog Timer ve yazılım koruma özellikleri sayesinde sistem maliyeti düşmekte, güç tüketimi azalmakta ve güvenilirlik artmaktadır.

Program tasarımcılar için geliştirilmiş olan silinebilir mikroişlemciler ile kolaylık sağlanırken, bir defa programlanabilen mikroişlemciler ile seri üretimde maliyet düşürülmesi üstünlüğü verilmektedir.

Bu çalışmada, birinci bölümde PIC mikrodenetleyici genel olarak tanıtılmakta, ikinci bölümde PIC16C5x'in teknik, kullanım ve programlama özellikleri anlatılmaktadır. Üçüncü bölümde komutlar örneklerle açıklanmakta, dördüncü bölümde programlamayı hızlı ve kolay öğrenebilmek için gerekli olan temel ve genel programlar önerilmektedir. Beşinci bölümde ise D.C. motor kontrol devresi tasarımı ve yapımı, step motor kontrol devresi tasarımı ve yapımı, trafik ışık kontrol devresi tasarımı ve yapımı ile tuş takımı tasarımı ve yapımı ayrıntılı olarak anlatılmıştır. Ekte ise örnek devrelerin programları verilmiştir.

DESIGN AND IMPLEMENTATION OF PIC MICROCONTROLLER BASED EXPERIMENT MODULES

SUMMARY

KEYWORDS: PIC, Microcontroller

PICmicro – the world’s most popular 8-bit RISC microcontrollers (MCU) – are available in OTP, EEPROM, Enhanced FLASH and ROM memory configurations to offer us the only complete 8-bit RISC MCU solution on the market today. The 8-bit RISC MCU architecture has become a worldwide standard, with half-a-billion PICmicro devices and 100,000 development systems shipped since 1990. The Company launched more than 58 new PICmicro OTP, EEPROM, FLASH and ROM products to market in the last two years alone. And Microchip’s product breadth continues to expand with 44 new RISC-based MCUs on the way. Only Microchip PICmicro’s provide superior performance, a seamless migration path between product families, multiple memory technologies within the same pinout and feature set, and comprehensive development tools that easily allow simple product enhancements or modifications. All this to help us build a solid design foundation, now and in the future.

The PIC16C5X from Microchip Technology is a family of low-cost, high performance, 8-bit, fully static, EPROM/ ROM-based CMOS microcontrollers. It employs a RISC architecture with only 33 single word/single cycle instructions. All instructions are single cycle (200 ns) except for program branches which take two cycles. The PIC16C5X delivers performance an order of magnitude higher than its

competitors in the same price category. The 12-bit wide instructions are highly symmetrical resulting in 2:1 code compression over other 8-bit microcontrollers in its class. The easy to use and easy to remember instruction set reduces development time significantly.

The PIC16C5X products are equipped with special features that reduce system cost and power requirements. The Power-On Reset (POR) and Device Reset Timer (DRT) eliminate the need for external reset circuitry. There are four oscillator configurations to choose from, including the power-saving LP (Low Power) oscillator and cost saving RC oscillator. Power saving SLEEP mode, Watchdog Timer and code protection features improve system cost, power and reliability.

The UV erasable CERDIP packaged versions are ideal for code development, while the cost-effective One Time Programmable (OTP) versions are suitable for production in any volume. The user can take full advantage of Microchip's price leadership in OTP microcontrollers while benefiting from the OTP's flexibility.

The PIC16C5X products are supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a 'C' compiler, fuzzy logic support tools, a low-cost development programmer, and a full featured programmer. All the tools are supported on IBM PC and compatible machines.

The PIC16C5X series fits perfectly in applications ranging from high-speed automotive and appliance motor control to low-power remote transmitters/receivers, pointing devices and telecom processors. The EPROM technology makes customizing application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages, for through hole or surface mounting, make this microcontroller series perfect for applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16C5X series very versatile even in areas where no microcontroller use has been considered before (e.g., timer functions, replacement of "glue" logic in larger systems, coprocessor applications).

BÖLÜM 1. GİRİŞ

Günümüzde çamaşır makinalarını, video cihazlarını ve kontrol gerektiren diğer elektronik cihazları mikroişlemcisiz düşünmek imkansız hale gelmiştir. Mikroişlemciler, çok az harici elemanlara ihtiyaç duymaları ve çok amaçlı kullanım için bir temel yapıtaşı olmaları nedeniyle günlük hayatta kullanılan elektronik cihazların kontrollerinde tercih edilmektedirler.

Mikro işlemciler 80'li yıllardan itibaren gösterdiği hızlı gelişmeler sonucu, önceden belirlenmiş bir mantığı kullanan lojik kapıların (TTL, CMOS gibi) yerini almış durumdadır [1, 2]. Bu özelliklere ek olarak mikro işlemcilerin diğer bir özelliği, çok az harici elemanla (Osilatör gibi) çalışmalarıdır.

Mikrodenetleyicilerde program (ROM) ve veri hafızaları (RAM) ile I/O (Giriş/Çıkış) birimine kadar hepsi tek bir CPU (Merkezi İşlem Ünitesi) üzerindedir. Mikrodenetleyici olarak kullanılan ilk seri, 8048 serisi olarak adlandırılan mikrodenetleyici ailesidir. Günümüzde ise, çok büyük miktarda 16 bit gibi farklı mikrodenetleyiciler piyasada mevcuttur (Intel, Thomson, Siemens, Mitsubichi, Texas Instrument gibi). Bunlar genelde otomotiv endüstrisinde yaygın olarak kullanılmaktadır.

Bu gelişmeler olurken Microchip şirketi, Intel ve Motorola'ya karşı 8 bitlik küçük yapı ve az pinli PIC mikrodenetleyicilerini piyasaya sürdü. Kısa bir süre öncesine kadar, bu mikrodenetleyicilerin varlığını çok az bir endüstriyel kesim bilmekte ve bunlar da mouse ve klavye içerisindeki fonksiyonları yerine getirmek gibi küçük amaçlı işler için bu mikrodenetleyicileri kullanmakta idi. Fakat günümüzde Avrupa'da yarı profesyonel amaçlı kullanımlar için gittikçe yaygınlaşmaktadır. Şu anda 8 bitlik mikrodenetleyici satışlarında dünya ikincisi durumundadır. PIC'lerin bu kadar çabuk yaygınlaşmasının en büyük nedeni, çok ucuz ve işlevleri açısından 16

bitlik mikroişlemcilerde yakın veya eşdeğer özellikte olmasıdır. Programlanmadan önce PC’de simüle edilebilmesi PIC mikrodenetleyicileri üstün duruma geçirmektedir.

Tez sırasında, mikroişlemci ve mikrodenetleyici terimleri ayrı anlamlarda kullanılacaktır. Entegre, çevre birimleri ile ilişki kurabilecek arabirim devrelerini içermiyorsa (6802, Z80, 8085 vb) 'mikroişlemci' olarak isimlendirilecektir. Aynı entegrede, mikroişlemci ve çevre birimleri ile haberleşmeyi sağlayacak arabirim elemanları birlikte bulunuyorsa (PIC, 8051 vb) 'mikrodenetleyici' olarak adlandırılacaktır.



BÖLÜM 2. PIC MİKRODENETLEYİCİ AİLESİ

80'li yılların sonunda, Microchip firmasının ilk PIC'i (Programmable Integrated Circuit) piyasaya sürmesi ile PIC ailesi dünyaya gelmiş oldu. Bu yıllarda sadece endüstrinin çok az bir kesiminde kullanılırken, günümüzde farklı uygulamalarda yaygın olarak kullanılmaktadır [3].

PIC mikrodenetleyicisi diğer standart mikroişlemcilerden görünüş itibarı ile de farklılık gösterir. PIC'ler sadece 40'lı DIP (Dual Inline Package) veya DIL (Dual In line) kılıfında değil, 18 veya 28'li DIP kılıfında da imal edilmektedirler[4]. Fakat 12 veya 28 Giriş/Çıkış uçları bakımından büyük kardeşleri ile eşdeğer özelliğe sahiptirler. Bu uçlar sadece programlanan amaç doğrultusunda Giriş/Çıkış işlemleri için kullanılabilir. Bu nedenle harici bir hafıza yongasının bağlanması normal şartlarda mümkün değildir. Bu durumda program hafızasının chip içinde olması gerekir. Bu işlemlerin yapılması sonucunda; yapı küçüklüğü, pin sayısı azlığı ve en önemlisi ucuzluğu sağlanmış olur. Fakat özel bir programlama cihazına gereksinim vardır.

İlk çıkan PIC serisi, PIC16C5X serisidir ve bu seri şu anda en çok kullanılan elemanlardır. Bu serinin program kapasitesi, hafıza alanı ve özellikleri diğer PIC'lere göre çok daha azdır. Fakat çok küçük ve ucuz olması yanında basit ve kolay programlanması, yaygın olarak kullanılmasının başlıca nedenidir. Şu anda çok sayıda PIC çeşidi mevcuttur. Bunlar; düşük, orta ve yüksek performanslı (Low-Middle-End) PIC'lerden oluşmaktadır. PIC serilerinden bazıları özellik bakımından 16 bitlik mikroişlemcileri kesinlikle aratmamaktadırlar.

2.1. PIC16C5X Microdenetleyicisi

PIC ailesinin en basit mikrodenetleyicisi 16C5X serisidir [5]. Bu seride Őu anda on farklı iŐlemci (CPU) mevcuttur. Bunların iç yapıları aynıdır fakat EPROM program hafızası büyüklüğü ve GiriŐ/ÇıkıŐ (I/O) uçları bakımından farklılıklar gösterirler.

2.1.1. PIC16C5X Microdenetleyicisinin genel özellikleri

Yüksek performanslı RISC-CPU (Reduce Instruction Set Computer) yapısında tüm komutlar (atlamalar hariç) tek adımda yapılır. Bu denetleyicinin genel özellikleri;

512-2048 Byte'lik EPROM.

25-72 Byte'lik RAM.

Doğrudan, dolaylı ve relative adresleme.

Osilatör türü ve osilasyon frekansı seçmeli (0-20 MHz.).

Watch-Dog-Timer.

Sayıcı/Bölücü.

12-20 I/O pinli.

2.5-6.25 V. Besleme gerilimli.

Düşük güç tüketimi.

SLEEP modlu.

Power-On-Reset şeklinde özetlenebilir.

Bu seride Őu anda on farklı chip mevcuttur. Bunlarda, program veri hafıza alanı ve I/O pin sayısı hariç tüm özellikler aynıdır. Tablo 2.1a ve 2.1b'de PIC16C5x serisindeki CPU'ların özellikleri görölmektedir.

Tablo 2.1.a PIC16C5X Microdenetleyicileri ve özellikleri

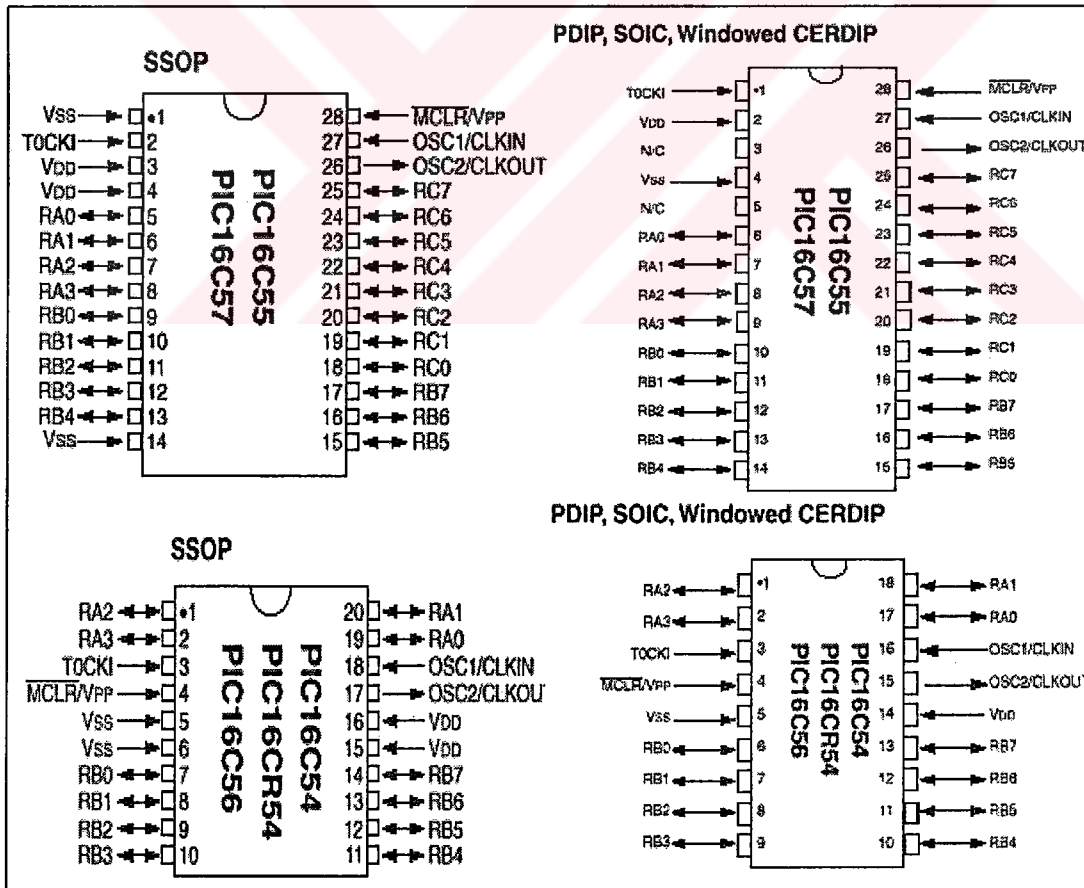
		PIC16C52	PIC16c54s	PIC16CR54s	PIC16C55s	PIC16C56s
Clock	Maksimum operasyon frekansı (MHz)	4	20	20	20	20
Hafıza	EPROM Program Hafızası (x12kelime)	385	512	-	512	1K
	ROM Program hafızası (x12 kelime)	-	-	512	-	-
	RAM Bilgi Hafızası (bytes)	25	25	25	24	25
Çevre birimleri	Zaman Modülü (s)	TMR0	TMR0	TMR0	TMR0	TMR0
Özellikler	I/O Pinleri	12	12	12	20	12
	İnstruction numarası	33	33	33	33	33
	Kılıf no	18-pin DIP, SOIC	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 28-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP

Tablo 2.1.b PIC16C5X Microdenetleyicileri ve özellikleri

		PIC16CR56s	PIC16c57s	PIC16CR57s	PIC16C58s	PIC16CR58s
Clock	Maksimum operasyon frekansı (MHz)	20	20	20	20	20
Hafıza	EPROM Program Hafızası (x 12 kelime)	-	2 K	-	2 K	-
	ROM Program hafızası (x 12 kelime)	1K	-	2 K	-	2K
	RAM Bilgi Hafızası (bytes)	25	72	72	73	73
Çevre Birimleri	Zaman Modülü (s)	TMR0	TMR0	TMR0	TMR0	TMR0
Özellikler	I/O Pinleri	12	20	20	12	12
	İnstruction numarası	33	33	33	33	33
	Kılıf no	18-pin DIP, SOIC; 20-pin SSOP	28-pin DIP, SOIC; 28-pin SSOP	28-pin DIP, SOIC; 28-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP

2.1.2. PIC16C5x Microdenetleyicilerin kılıf yapıları

Değişik kılıflarda PIC16C5x elemanı mevcut olmasına karşılık, burada sadece 2 tanesi incelenecektir. Bunlar PIC16C54 ve PIC16C56'nın 18'li ile PIC16C55 ve PIC16C57'nin 28'li DIL kılıflarıdır. 28'li DIL yapıda olanlar, diğerlerine göre daha fazla sayıda I/O ucuna sahip olduğundan daha büyük bir kılıfa ve tabii ki daha fazla pine ihtiyaç duyarlar. DIL kılıfının yanında diğer kılıf türleri; SSOP (Surface Small Outline Package) ve SOIC (Small Outline Integrated Circuit)'dir. Bunlardaki bacak bağlantıları diğer kılıf şekline göre farklılık gösterir. Ayrıca SSOP ve SOIC kılıf yapılar genellikle endüstriyel amaçlı olduklarından ve özel kullanımlarda pek kullanışlı olmadıklarından burada bu tür kılıflı PIC'lerden bahsedilmeyecektir. Şekil 2.1'de PDIP, SSOP ve SOIC kılıflı PIC'ler görülmektedir.



Şekil 2.1. PDIP, SSOP ve SOIC kılıflı PIC'ler.

2.1.3. PIC'lerde Kullanılan frekans türleri

PIC'ler kendi aralarında farklı türlere ayrılabilirler. Fakat bu ayırmda referans, çalışma veya görünüş değil, maksimum çalışma frekansdır. PIC'ler çalışma frekansına göre 4 gruba incelenebilirler. Farklı gruplar RC, XT, HS ve LP uzantıları ile ayırt edilebilirler.

RC ve XT uzantılı olan PIC'ler 0-4 MHz arasındaki frekansla çalışırlar. Bu iki grup arasındaki fark; kullanılan osilatör çeşididir. XT uzantılı PIC'ler, osilasyon için Quartz veya seramik rezonanslı elemanlar kullanırlar. Quartz veya seramik rezonanslı elemanların dış etkenlere çok az bağımlı olması yani osilasyon frekansının dış etkenlerden çok az etkilenmesi nedeniyle çoğu uygulamalar için XT uzantılı PIC'ler tercih edilir. RC türü PIC'lerde ise 'RC' kısaltmasından da anlaşılacağı gibi osilasyonun oluşmasında direnç ve kondansatörden faydalanılır. RC osilatör yapısı, XT'deki Quartz veya seramik elemanlara göre çok daha ucuz olmasına karşılık, sıcaklık gibi dış etkenlerden çok etkilenir. Bu nedenle; RC ile oluşturulan osilatörlerde, osilasyon frekansı \pm %20 toleranslı çalışır.

İşlem hızının çok önemli olduğu sistemlerde yüksek hızlı (High-Speed) PIC çeşitleri kullanılır. Bunlar HS türü PIC'lerdir ve 20 MHz'e kadar kullanılabilirler. Bu işlem hızında, belirli bir süre içinde diğerlerine göre çok daha fazla işlem yapma olanağı oluşur. Bu tür PIC'lerde osilatör olarak Quartz kullanılır.

Düşük güçlü sistemlerde kullanılmak üzere düşük güç tüketimli (Low-Power) PIC'ler üretilmiştir. LP uzantısı ile gösterilen bu PIC'ler, diğer PIC çeşitlerine göre çok daha az güç tüketimine sahiptirler. Fakat en fazla 32 KHz'lik bir hızda çalışabilirler. Bu tür PIC'lerde osilatör olarak Quartz kristali temelli osilatör kullanılır. Tablo 2.2.a ve 2.2.b'de farklı osilatörlere sahip PIC'lerin güç harcama ve performans değerleri toplu olarak görülmektedir.

Tablo 2.2.a PIC'lerin osilatör türlerine göre sınıflandırılması

OSC	PIC16C5X-RC	PIC16C5X-XT	PIC16C5X-10
RC	VDD: 3.0 V.ile 6.25 V. IDD :5 V.ta max.3.3mA IPD :3 V.ta max.9µA. Freq :4 MHz max.	Kullanılmıyor	Kullanılmıyor
XT	VDD: 3.0 V.ile 6.25 V. IDD :5.5 V.ta max.1.8mA IPD :3 V.ta max.0.6µA. Freq :4 MHz max.	VDD: 3.0 V.ile 6.25 V. IDD :5 V.ta max.3.3mA IPD :3 V.ta max.9µA. Freq :4 MHz max.	Kullanılmıyor
HS	Kullanılmıyor	Kullanılmıyor	VDD: 4.5 V.ile 5.5 V. IDD :5.5 V.ta max.10mA IPD :3 V.ta max.9µA. Freq :10 MHz max.
LP	VDD: 2.5 V.ile 6.25 V. IDD :3 V.ta max.15µA IPD :3 V.ta max.9µA. Freq :40 KHz max.	VDD: 2.5 V.ile 6.25 V. IDD :3 V.ta max.15µA IPD :3 V.ta max.9µA. Freq :40 KHz max.	VDD: 2.5 V.ile 6.25 V. IDD :3 V.ta max.15µA IPD :3 V.ta max.9µA. Freq :40 KHz max.

Tablo 2.2.b PIC'lerin osilatör türlerine göre sınıflandırılması

OSC	PIC16C5X-HS	PIC16C5X-LP	PIC16C5X/JW
RC	Kullanılmıyor	Kullanılmıyor	VDD: 3.0 V.ile 6.25 V. IDD :5.5 V.ta max.3.3mA IPD :3 V.ta max.9µA. Freq :4 MHz max.
XT	Kullanılmıyor	Kullanılmıyor	VDD: 3.0 V.ile 6.25 V. IDD :5.5 V.ta max.3.3mA IPD :3 V.ta max.9µA. Freq :4 MHz max.
HS	VDD: 4.5 V.ile 5.5 V. IDD :5.5 V.ta max.20mA IPD :3 V.ta max.9µA. Freq :20 MHz max.	Kullanılmıyor	VDD: 4.5 V.ile 5.5 V. IDD :5.5 V.ta max.20mA IPD :3 V.ta max.9µA. Freq :20 MHz max.
LP	VDD: 2.5 V.ile 6.25 V. IDD :3 V.ta max.15µA IPD :3 V.ta max.0.6µA. Freq :40 KHz max.	VDD: 2.5 V.ile 6.25 V. IDD :3 V.32kHz.de max.15µA IPD :3 V.ta max.9µA. Freq :40 KHz max.	VDD: 2.5 V.ile 6.25 V. IDD :3 V.32kHz.de max.15µA IPD :3 V.ta max.9µA. Freq :40 KHz max.

Tabloda verilen akım ve gerilim değerleri standart kullanım için ortalama değerlerdir. Aynı zamanda, uygun çalışma frekansında ve yüklenmemiş Giriş-Çıkış (Boş uçlar) durumları için bulunmuş olan değerlerdir.

2.2. PIC'lerde Bulunan Bağlantı Uçları

Osilatör türü değişse de, PIC'de bulunan bağlantı uçları değişmez Şekil 2.1. Bu uçların açıklaması Tablo 2.3'de toplu olarak görülmektedir.

Tablo 2.3. PIC'lerde bağlantı uçları

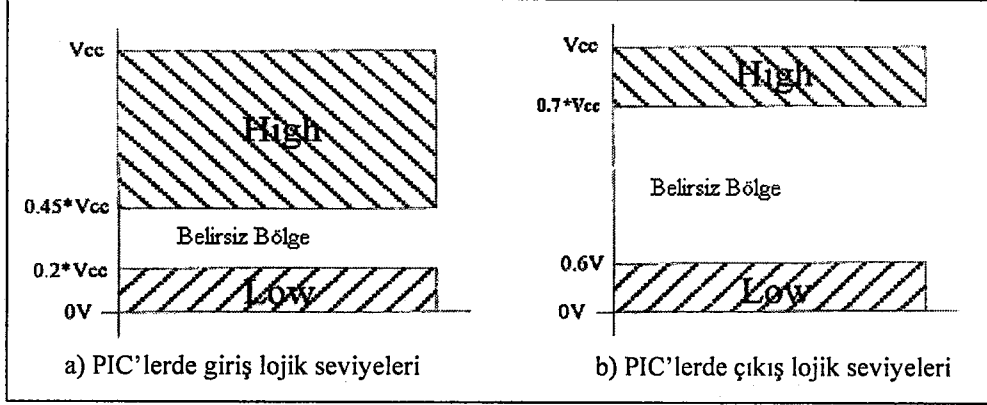
İsim	DIP, SOIC No	SSOP No	I/O Türü	Giriş Seviyesi	Açıklama
RA0	17	19	I/O	TTL	Çift yönlü I/O Port.
RA1	18	20	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RB0	6	7	I/O	TTL	Çift yönlü I/O Port.
RB1	7	8	I/O	TTL	
RB2	8	9	I/O	TTL	
RB3	9	10	I/O	TTL	
RB4	10	11	I/O	TTL	
RB5	11	12	I/O	TTL	
RB6	12	13	I/O	TTL	
RB7	13	14	I/O	TTL	
TOCKI	3	3	I	ST	Saat giriş timer0.
MCLR/ Vpp	4	4	I	ST	Reset.Program girişi için gerekli voltaj (Vss veya V _{DD} bağlanmalı).
OSCI/ CLKIN	16	18	I	ST	Osilatör kristali için gerekli saat kaynak girişi.
OSC2/ CLKOUT	15	17	O	—	Osilatör kristali çıkışı. Osilatör kristali için gerekli saat kaynak girişi.
V _{DD}	14	15,16	P	—	Lojik ve I/O pinleri için pozitif kaynak.
V _{SS}	5	5,6	P	—	Lojik ve I/O pinleri için toprak değeri.

I = giriş, O = Çıkış, I/O = Giriş/Çıkış, P = Güç, — = Kullanılmıyor,
TTL = TTL Giriş, ST = Schmitt Trigger Giriş.

Tablo 2.3'de görüleceği gibi, çoğu PIC uçları I/O için kullanılır. PIC16C54 ve 56'nın 12, PIC16C55 ve 57'nin ise 20 tane I/O ucu vardır. Bu I/O uçları, 2 veya 3 gruba ayrılmıştır ve her grupta 4 veya 8 I/O ucu bulunur.

PIC'lerin terminallerindeki (portlar) giriş geriliminin lojik seviyesi, besleme gerilimine göre göre değişir. $0 V - 0.2 * V_{cc}$ arası gerilim değerleri 'LOW-0' seviye, $0.45 * V_{cc} - V_{cc}$ arası ise 'HIGH-1' seviye olarak tanımlanır. Bu iki seviye arasında kalan alan ise tanımsızdır (belirsizdir).

PIC'lerdeki çıkış lojik seviyeler, girişteki lojik seviyelere benzerler. 0.6 V'a kadar çıkış gerilimi '0' seviyeyi, $0.7 \cdot V_{cc}$ gerilimden büyük değerler ise '1' seviyeyi temsil ederler. PIC'lerde kullanılan '0' ve '1' gerilim seviyeleri Şekil 2.2'de şematik olarak görülmektedir.



Şekil 2.2. PIC'lerdeki giriş/çıkış lojik seviyeleri

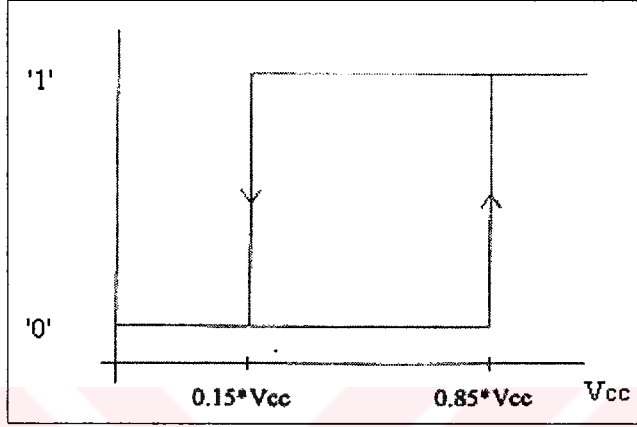
PIC'in çıkışında akan akımlar, PIC'e özel bir soğutma uygulanmaksızın, '1' seviye için 20 mA ve '0' seviye için 25 mA'dir. İki veya üç terminalli (Portlu) çıkışlarda tek bir terminalin kullanılması durumunda, bu değerlerin iki katı çıkış akımı akıtılabilir. PIC'lerde giriş akımları mikroamper seviyesindedir.

2.2.1. PIC'in I/O uçları ve T0CKI girişi

PIC'deki tüm I/O uçları çift yönlüdür (Bidirectional). Bu özellik nedeni ile her I/O ucu giriş veya çıkış olarak tanımlanabilir. Giriş ve çıkış uçları arasında kısa devre oluşmaması için, her ucun veri akış yönü önceden belirlenmelidir. PIC'deki giriş ve çıkış durumları arasındaki değişimi TRIS-kaydedici birimi ile yapılır. Bu kayıtçı silindiğinde yani '0' ile yüklendiğinde, '0' olan uç çıkış olarak seçilmiş olur. Bir ucu giriş olarak seçmek için ilgili bit'in '1' yapılması gerekir.

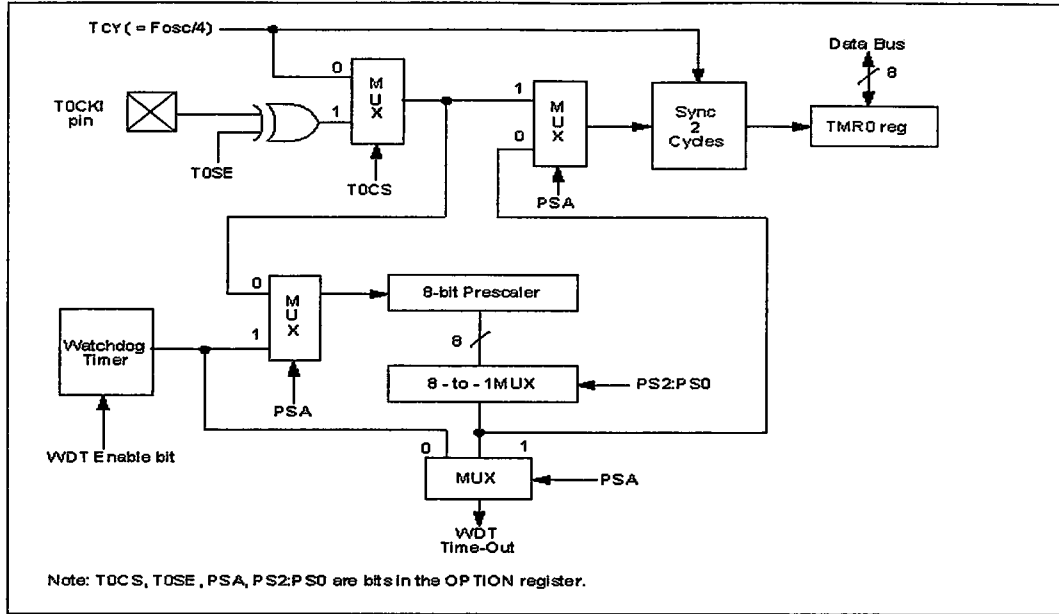
I/O uçları 8'li veya 4'lü gruplar şeklinde toplanmalarına karşı, her bir uç giriş veya çıkış olarak seçilebilir. Bu işlem; PORT-kayıtçısı ile her uçun ayrı ayrı kontrol edilebilmesi ve TRIS-kayıtçısı ile veri akışı yönünün (Giriş veya Çıkış)

ayarlanabilmesi özellikleri yardımıyla gerçekleştirilir. Diğer uçlar \overline{MCLR} ve TOCKI (Eskiden RTCC) I/O uçlarına göre farklı yapıdadırlar. Bunlar normal TTL girişi yerine Schmitt-Trigger girişe sahiptirler ve sabit/belirli bir giriş değerinde konumlarını değiştirirler. Bu sabit değer '0'dan '1' için $0.15 \cdot V_{cc}$ ve '1'den '0' için $0.85 \cdot V_{cc}$ 'dir (Şekil 2.3).



Şekil 2.3. MCLR ve TOCKI uçlarında konum değiştirme değerleri

Schmitt-Trigger yapıya sahip bir uç $0.85 \cdot V_{cc}$ değerinde '0'dan '1'e konum değiştirirken, diğer terminal girişleri $0.45 \cdot V_{cc}$ ile V_{cc} arasındaki bir değerde '1' konumuna durum değiştirir. Bu özellik reset girişi olarak kullanılan \overline{MCLR} girişi için çok önemlidir. Çünkü Schmitt-Trigger yapıya sahip giriş çok yavaş artan bir RC yapıyı takip edebilir. PIC'in diğer bir özelliği; TOCKI (Timer0 Clock Input) girişidir. TOCKI girişi '0' nolu zamanlayıcı tetikleme girişidir ve PIC içerisinde bulunan gerçek zamanlı sayıcıya (RTCC – Real Time Clock / Counter) gerekli tetikleme sinyalini uygulamak için kullanılır. Şekil 2.4.a ve Şekil 2.4.b'de bu sayıcının yapısı görülmektedir .



Şekil 2.4. PIC'in TOCKI girişi

Dışarıdan bağlanan TOCKI işaret TOSE biti yardımı ile tersi yada tersi alınmadan MUX (Multiplexer)'un 1 girişine iletilir. Bu işlem EXOR yapısı ile sağlanır. EXOR'u takip eden Multiplexer yardımı ile PIC'in kendi işareti ile dışarıdan bağlanan işaret arasında tercih yapılabilir. Bu tercih T0CS biti ile yapılır. T0CS bitindeki değer '1' ise dışarıdan bağlanan işaret seçilmiş olur.

Multiplexer'den gelen tetikleme sinyali doğrudan TMR0'a uygulanabileceği gibi, bir ön bölücüden geçirildikten sonra TMR0'a uygulanabilir. Kullanılan ön bölücünün değeri 2^n ile ifade edilen ve PS0'dan PS2'ye kadar olan 3 bite karşılık gelir. Kullanılan ikinci Multiplexer, uygulanan işaretin bir sonraki TMR0'a iletilip ileilmeyeceğini belirler. Bu işlemde elde edilen işaret, iki komut işlem süresi (8 pals) geciktirilerek TMR0'a iletilir. Bu işlem, dışarıdan bağlanan tetikleme sinyali ile dahili tetikleme palsinin senkronize edilebilmesi için yapılır.

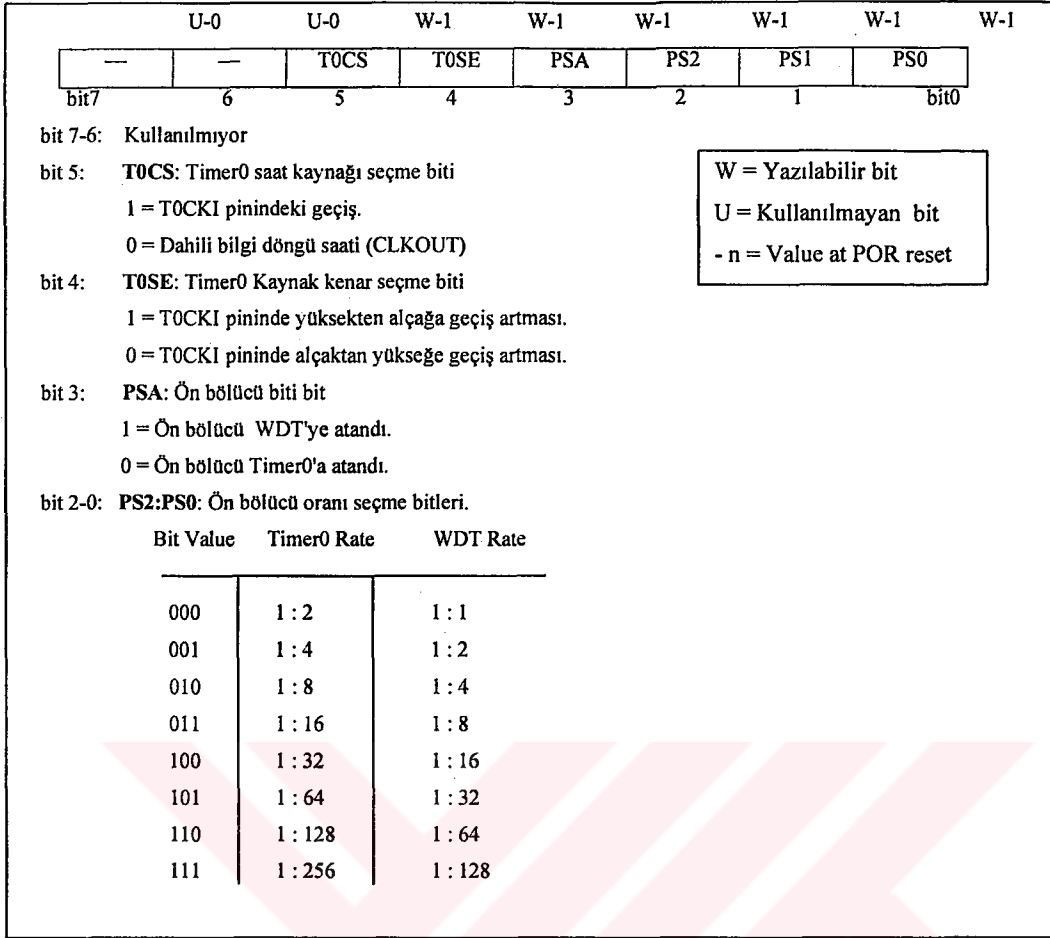
TMR0'a uygulanan her tetikleme palsi ile kayıtçının değeri bir artırılır. TMR0'nun değeri 0FFh olduğunda taşma oluşur ve kayıtçı 00h değerini alır. TMR0, takip eden kısımlarda anlatılacak kayıtçılar gibi kullanılır.

PIC'in dięer bir özellięi, TMR0 ile çok yakın ilgisi olan gözetleme sayacı WDT (Watch Dog Timer) giriřine sahip olmasıdır. Adından anlaşılacağı üzere gözetleme Sayacı giriři PIC'i uyandıran bir araçtır. WDT bir tetikleme sinyali üreticidir ve belirli aralıklarla uyarılmaması durumunda PIC'i sıfırlar. Bu özellik sayesinde hatasız sistemler oluşturulabilir. Bu durumda yazılan program içine WDT'yi belirli aralıklarla sıfırlayacak kısımlar eklenmelidir. Böylece program hatalı çalıştığında veya işlem bozukluğu oluştuğunda program doğru çalışmayacağından WDT'yi sıfırlayan devre gecikmeli çalışır. Bu durumda sıfırlama oluşur ve program yeniden başlar. WDT çok önemli bir işleve sahip olmasından dolayı, PIC iç yapısında dięer birimlerden bağımsız olarak görev yapmaktadır. WDT'nin çalışma frekansı PIC içerisinde ayrı olarak üretilir.

Normal şartlar altında WDT'nin sıfırlanmaması durumunda, her 18.2 ms'de bir WDT tarafından sıfırlama sinyali üretilir. Bu süre TMR0 yardımıyla 8 kata kadar artırılabilir. Bu artırma özellięi, T0CKI kullanılmıyorsa geçerlidir.

2.2.2. Seçenek kayıtçısı (OPTION register)

Yukarıda anlatılanlar řekil 2.5'de görülen seçenek kayıtçısında özetlenmektedir.

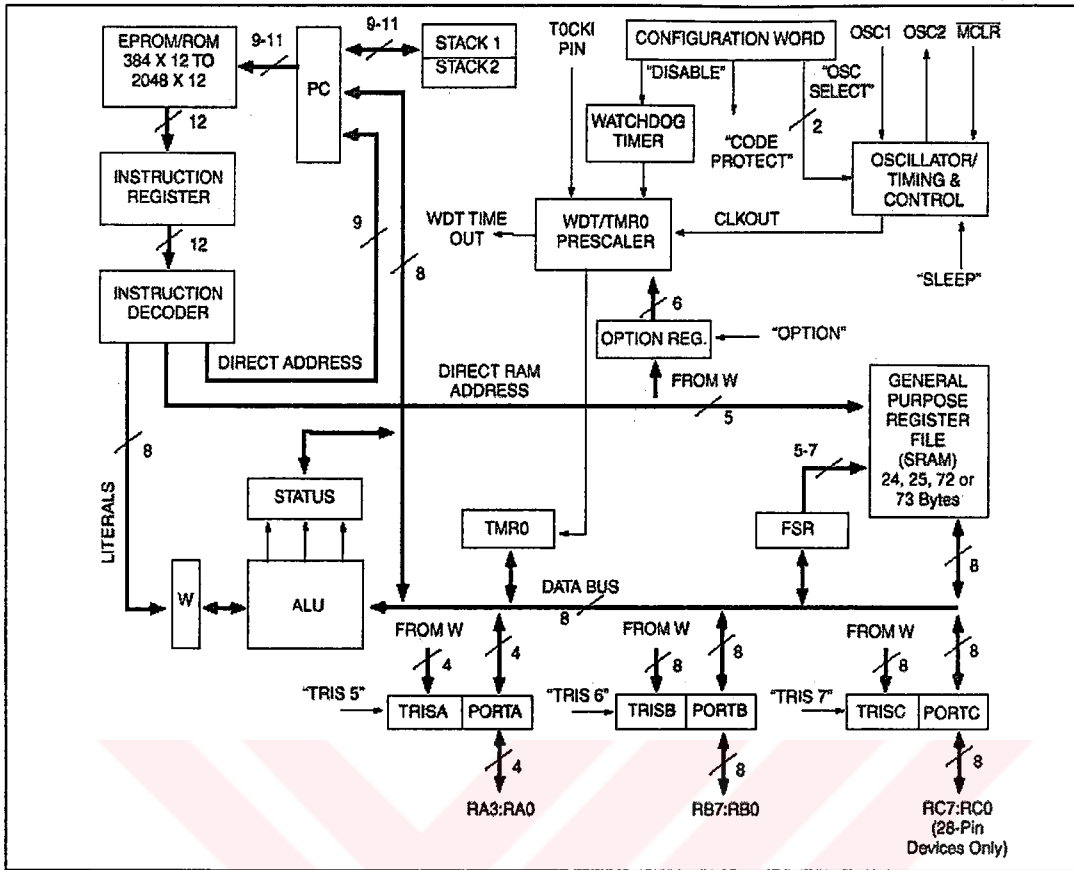


Şekil 2.5. Seçenek kayıtçısı (OPTION Register).

Yukarıda anlatılan durumlar, seçenek kayıtçısına ait bit'in '1' yapılması durumunda geçerlidir. Şekil 2.5 ön bölmeyi göstermektedir. Burada unutulmaması gereken nokta, ön bölme işleminin T0CKI veya WDT için kullanılabileceği, her ikisi için aynı anda kullanılmayacağıdır.

2.3. PIC'in Çalışma Şekli

Önceki kısımlarda anlatıldığı gibi tüm PIC'ler RISC-CPU yapısına sahiptirler ve komutlarını içindeki EPROM'dan alırlar. Bu EPROM, diğer mikro işlemcilerinkine göre farklı kelime uzunluğuna sahiptir (12 Bit). Bunun sebebi, tüm PIC'lerin sadece Bir-Byte uzunluğundaki komut ile programlanmalarıdır. Komut iki kısımdan oluşmaktadır: Komut ve veri. İşlenecek veri veya yaptırılacak işlem hafızada yani EPROM'da saklanmaktadır (Şekil 2.6).



Şekil 2.6. PIC'in çalışma şeması

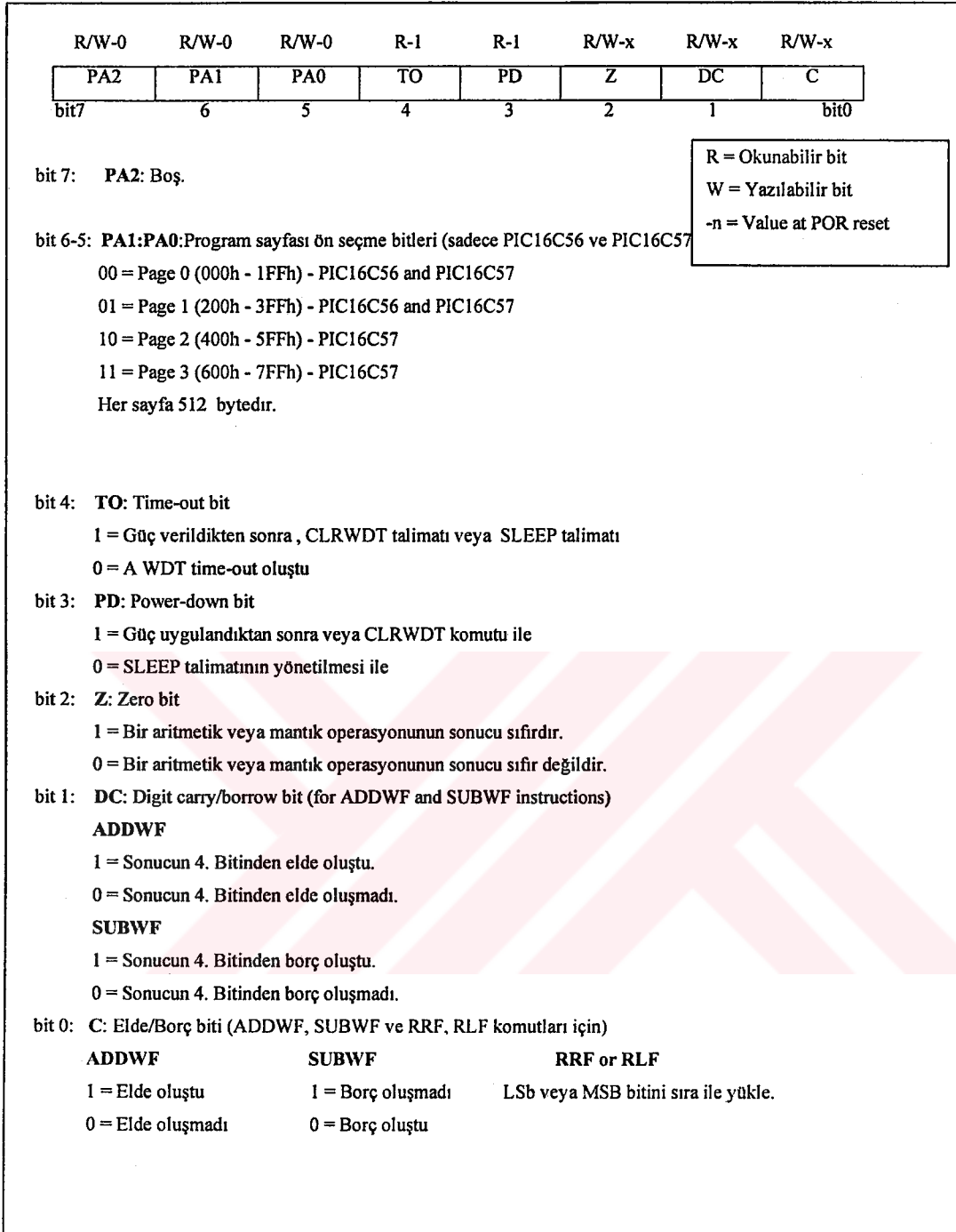
PIC'de herhangi bir programın çalıştırılması sırasında program sayıcısı (PC-Program Counter) ile bir sonraki işlenecek komut adreslenerek belirlenir. Buna bağlı olarak iki kat derinlikli Hardware Stack (yığın bölgesi) mevcuttur. Yığın bölgesi içine en fazla iki adres geçici olarak saklanabilir. Ana programdan bir alt programa atlandığında, ana programa ait adres bu bölgeye yüklenir. Alt programdan geri döndüğünde ise; bu bölgeye konulan adres geri alınarak, bu adresten itibaren işleme devam edilir. Yığın bölgesinde en fazla iki adet adres saklanabildiğinden, içiçe en fazla iki alt program çalıştırılabilir. Bu program yani içiçe iki alt program çalıştığı anda bir üçüncü alt program çalıştırılırsa, ilk programa ait adres atılarak bu üçüncü programa ait değerler yığın bölgesinde saklanır. Sonuç olarak; yığın bölgesinden atılan adres ana programa ait ise hiçbir zaman ana programa geri dönülemez ve program hatalı çalışır.

Bir programın çalıştırılması sırasında program sayıcısı tarafından adreslenen komut, önce komut kayıtcısına yüklenir. Buradan asıl işlenecek olan değer ise yeni değerlendirme lojiğine gönderilir. Değerlendirme lojiği iki kısımdan oluşur: Komut kodçözücüsü yani çalışma kayıtcısı (WORK-Register—W-Kayıtçı) ve Aritmetik-Lojik-Birimi (ALU). Komut kod çözücüsü kendisine iletilen komutu inceleyerek, ne tür bir komut olduğuna ve nerede işleneceğine karar verir. Komut kod çözücüsünün, komut sayacına, RAM hafızasına ve ALU'ya doğrudan müdahale etme yetkisi vardır. Bu nedenle atlamalar, hafızaya müdahaleler, aritmetik işlemler ve mantıksal işlemler burada yapılabilmektedir. Komut kod çözücüden gelen veriler W-kayıtçısına gönderilir. W-kayıtçısı diğer mikroişlemcilerin akümülatörlerine benzer. Bu kayıtçı, tüm ara hafızalama için gerekli olan verilerin veya sonuçların işlenebilmesi için kullanılır.

2.3.1. Durum kayıtcısı (STATUS register)

ALU'da işlenen işlemin durumu veya işlemin sonucu hakkında bilgi edinmek için, durum kayıtcısına (STATUS Registere) bakılır. Bu kayıtçıda her bit, bir özel durumu temsil etmektedir. Şekil 2.7'de durum kayıtcısının tüm bitleri açıklanmaktadır.

Taşma biti (CARRY bit), tüm hesaplamalar ve kaydırma işlemlerinde artan bit olarak kullanılır. Toplama işlemlerinde taşma (elde) oluştuğunda bu bit '1' olur. Çıkarma işlemleri yapılmadan önce bu bit '1' yapılmalıdır. Çünkü çıkarma işlemi sonucu elde edilen değer eğer negatif ise bu bit '0' değerini alır. Fakat negatif değil ise bu bit'e dokunulmaz.



Şekil 2.7. Durum kayıtcısı açık şeması.

Bu nedenle herhangi bir hata oluşmaması için bu bit işlem yapmadan önce '1' yapılmalıdır. Kaydırma işlemlerinde ise kaydırma bu bitten faydalanılarak gerçekleştirilir.

DIGIT CARRY (DC) bit, aritmetik işlemlerde elde (carry) biti ile aynı işi yapar. Aralarındaki fark, elde biti için tüm Bayt'in (8 bit) değerlendirilmesine karşılık DC bitinde sadece ilk dört bitin değerlendirilmesidir. Bu ilk dört bit'e 'NIBBLE bit' denir. Yani aritmetik toplama işlemi 15'den büyük olduğunda bu bit '1' değerini alır.

ZERO bit, herhangi bir işlemin sonucu sıfır olduğunda '1' yapılır. Burada işlemin türü önemli değildir. Yani herhangi bir işlemin sonunda çalışma kayıtcısı=0 değerini alıyorsa bu bit '1' olur.

Durum kayıtcısındaki 3. Bitin '0' olması, PIC'in güç kesim (POWER DOWN) durumunda olduğunu gösterir. Bu durumdan kurtulabilmek için PIC'e ya sıfırlama palsi uygulanmalı veya WDT uyandırma sinyali göndermelidir.

Durum kayıtcısındaki 4. Bitin '1' olması, WDT'nin zaman içinde sıfırlanmadığını ve zamanın dolduğunu (TIME OUT olduğunu) gösterir. Durum kayıtcısındaki 5. ve 6. Bitlerin sadece PIC 16C56 ve PIC 16C57'de bir etkisi söz konusudur. Bu bitler, kullanılmakta olan EPROM'un sayfasını değiştirmeye yardım ederler. Her EPROM sayfası 512 Byte büyüklüğündedir. PIC16C54 ve PIC16C55'in EPROM'u 512 Byte büyüklüğünde olduğundan bu bitlere ihtiyaç duymazlar. Bununla beraber PIC16C56, 1024 Byte ve PIC16C57, 2048 Byte büyüklüğünde EPROM'a sahip olduklarından PIC16C56 için durum kayıtcısındaki 5. bite, PIC16C57'de ise her iki Bit'e ihtiyaç duyulur. PIC, çalıştığı sürece (besleme olduğu sürece) veri saklamak veya geçici olarak yedeklemek için RAM hafızasına sahiptir. Programlama hafızasına karşı buradaki veriler 8 bitlik bölümlerle kullanılmaktadır. PIC'in yapısına göre bu hafızanın büyüklüğü 32'den 80 Byte'a kadar uzanabilir.

2.4. PIC'in RAM Hafızası

PIC'in RAM hafızasının ilk 8 Byte'i, veri saklamak veya geçici yedeklemek için değil özel fonksiyonlar için ayrılmıştır. Bunlara özel fonksiyon kayıtcısı (SPECIAL FUNCTION Register-SFR) denir. PIC16C5x'e ait SFR, Tablo 2.4'de görülmektedir.

Tablo 2.4. PIC16C5x'e ait özel fonksiyon kayıtcısı

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on MCLR and WDT Reset
N/A	TRIS	I/O Kontrol kayıtcıları (TRISA, TRISB, TRISC)								1111 1111	1111 1111
N/A	OPTION	Seçenek kayıtcısı								--11 1111	--11 1111
00h	INDF	Data hafızasını adreslemek için FSR'nin içeriğini kullanır. (Fiziksel bir kayıtcı değildir) Dolaylı adreslerde kullanılır.								Xxxx xxxx	Uuuu uuuu
01h	TMRO	8-bit gerçek zaman saat sayıcı								Xxxx xxxx	Uuuu uuuu
02h ⁽¹⁾	PCL	Low order 8 bits of PC								1111 1111	1111 1111
03h	STATUS	PA2	PA1	PA0	TO	PD	Z	DC	C	0001.1xxx	000q quuu
04h	FSR	Indirect veri hafıza adres gösterici(Dolaylı adreslerde kullanılır).								1xxx xxxx	1uuu uuuu
05h	PORTA					RA3	RA2	RA1	RA0	--- xxxx	--- uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	Uuuu uuuu
07h ⁽²⁾	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	Uuuu uuuu

Legend: Gölge kutular = kullanılmamış, -- = unimplemented, read as '0' (if applicable)
x = bilinmeyen, u = değişmeyen, q = muhtemel değerler için

Not 1: Program sayıcının üst baytlarına direk ulaşabilmek için

2: Dosya adresi 07h, PIC16C54/CR54A/C56 üzerinde genel bir kayıtcıdır.

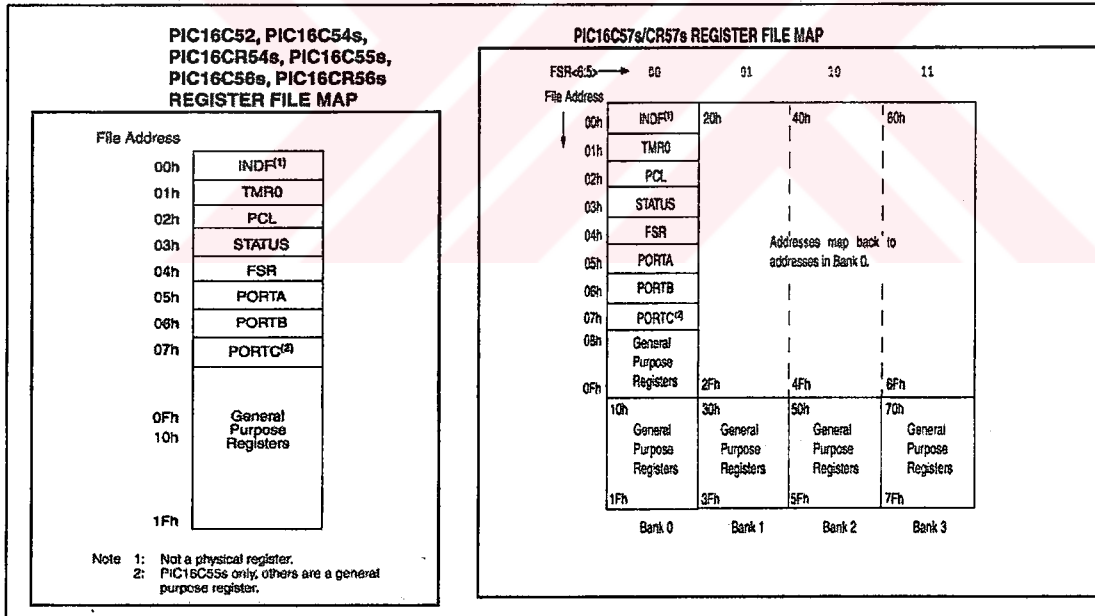
RAM hafızanın ilk 8 Byte haricindeki yerler, veri saklamak veya geçici yedekleme işlemleri yani kısacası hafıza olarak kullanılmaktadır. Kayıtcının sıfır adresi fiziksel olarak mevcut değildir. Bu hafıza, dolaylı adresleme (INDIRECT Adressing) işlemleri için kullanılır.

Dolaylı yoldan adresleme işlemi kullanışlı olmasına karşılık kullanımı zordur. Dosya seçme kayıtcısının (File Select Register-FSR) 0. ile 4. bitleri kullanılarak dolaylı adresleme işlemi yapılabilir. Bu bitler ile 004'den 1F4'e kadarki değerler oluşturulabilir. Bu sayılar aslında, başka bir adrese ait kayıtcının değeridir. Bu değer 'gösterge' olarak isimlendirilir ve bunun nedeni dolaylı yoldan bir kayıtcının adresini göstermesidir. FSR'de dolaylı yoldan belirlenmiş adresin değeri veya içeriği IND'de aynen oluşur ve IND'nin yeri de '0' nolu adrestir. Bir örnek verirsek; FSR'ye 9 yazılırsa, 9 nolu kayıtcının içeriği IND kayıtcısında aynen oluşur. Böylece, IND kayıtcısını okumakla 9 nolu kayıtcının içeriği okunmuş olur. Bir sonraki kayıtcı TMR0'ın sayı değerini içermektedir. Bundan sonraki kayıtcılar daha önce açıklamış

olduğumuz program sayıcısı (PC), durum (STATUS) ve dosya seçici (FSR) kayıtçılarını içermektedir.

Bundan sonraki 3 kayıtçı ise dış dünya ile bağlantımızı sağlayan kayıtçılardır. Bunların içeriği I/O terminalinin durumuna bağlıdır. Bu kayıtçıları okumak veya yazmak demek, terminalden giriş veya çıkış işlemi yapmak demektir. Şekil 2.8'de hafızanın kayıtçılara ayrılmış grafiksel şekli görülmektedir.

Terminalleri tam kullanabilmek için 3 kayıtçıya ihtiyaç vardır. Bunlar daha önce açıklanan TRIS kayıtçılardır. Bunlar, çıkış terminalleri tristate durumuna yani yüksek empedanslı duruma sokmak için kullanılırlar. Fakat bu işlem sadece terminalin giriş olarak kullanılması durumunda oluşur. TRIS kayıtçısının durumunu değiştirmek, sadece W kayıtçısı ile mümkündür. W kayıtçısına yazılan değer, özel bir komut ile TRIS kayıtçısına kopyalanabilir.



Şekil 2.8. PIC'in hafıza haritası

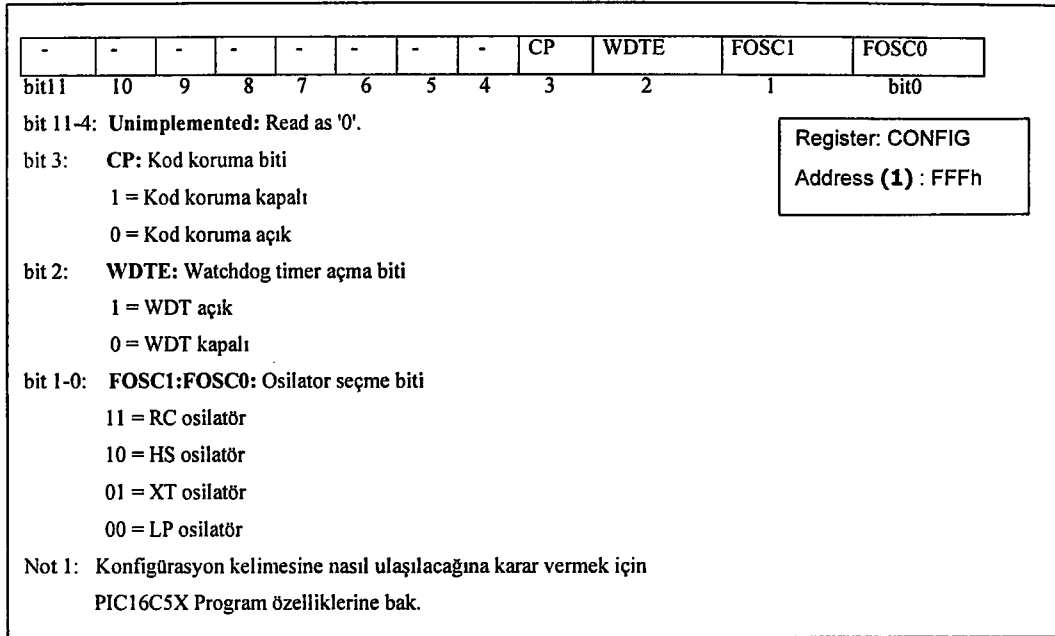
2.5. PIC Programlama

PIC'in EPROM'unu programlamak için özel bir programlama cihazına ihtiyaç vardır. Bu cihazlar ile programlanacak PIC'ler genelde iki gruba ayrılır: Silinebilen ve silinemeyen PIC'ler.

Silinebilen PIC'lerin üstünde küçük bir penceresi vardır ve ultraviyole (UV) ışık ile PIC'in EPROM'u silinebilir. Silinebilir PIC'ler yaklaşık 100 defa silinip yazılabilir. Daha fazla silme işleminde tam silme sağlanamaz veya tam programlama yapılamaz. Ayrıca üzerinde pencere bulunan PIC'ler sadece UV ışığı geçirme özelliğine sahip olduklarından pencereci chiplerin üretimi çok pahalıdır. İkinci grup PIC'ler silinemeyen veya bir defa programlanabilen (One Time Programmable-OTP) PIC'lerdir. OTP PIC'lerin, pencereci PIC'ler arasında herhangi bir üretim farkı yoktur. Sadece kılıflama farkı vardır. Aralarındaki diğer bir fark ise fiyatlarıdır. PIC'lerin türüne göre pencereci PIC'ler OTP'lerden 10 ilâ 30 kat daha pahalıdır. Bu nedenle pencereci PIC'ler genelde deneme aşamalarında kullanılırlar.

2.5.1. Konfigurasyon kelimesi kayıtçısı

PIC EPROM'u programlanmadan önce, konfigurasyon kelimesi isimli kayıtçısı vardır. Bu kayıtçı ile PIC'in donanım (Hardware) ayarları yapılır ve bu kayıtçı sadece programlama esnasında değiştirilebilir. PIC 16C5x'e ait bu kayıtçı Şekil 2.9'da görülmektedir.



Şekil 2.9. PIC'in konfigürasyon kelime kayıtçısı.

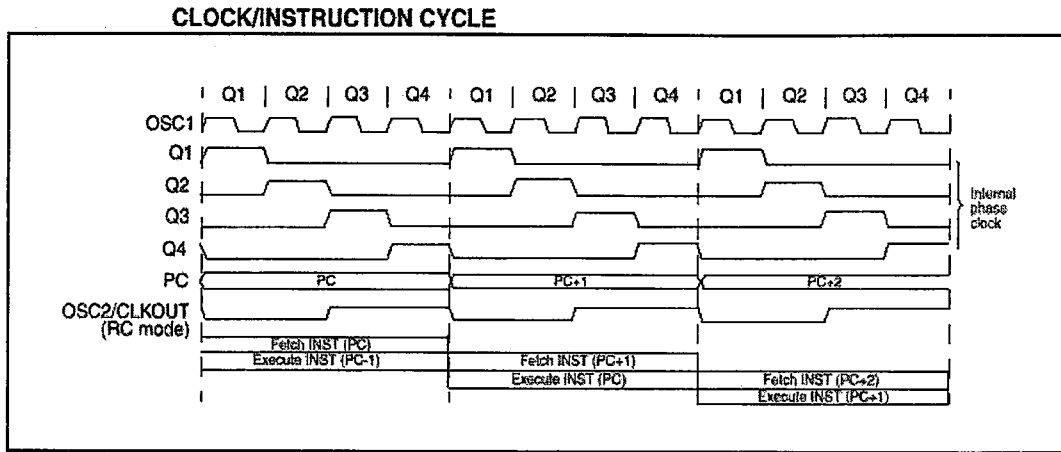
Kayıtçıda ilk iki bit, osilatörün türünün ayarlanması için kullanılır. Burada dikkat edilmesi gereken husus, programlanacak PIC'in hangi tür osilatörle çalıştığına dikkat edilmesidir. Buna bir örnek verirsek; PIC16C54/LP-4 düşük güçte çalışır (Low Power) ve maksimum osilasyon frekansı 4 MHz'dir. Bunun anlamı, işlemci LP modda programlanmalı ve çalışırken en fazla 4MHz'de çalıştırılmalıdır.

WDTE'nin çalıştırılıp çalıştırılmayacağını bit 2 belirler. WDTE'nin çalışması için bu bit '1' yapılmalıdır. Bu kayıtçının en önemli biti, okuma koruma bit'i olan dördüncü bit'dir. Bu bit programlama sırasında '0' ise, bu PIC'in EPROM'u bir daha asla okunamaz. Çünkü okuma esnasında 12 bit'lik kelimenin sadece ilk 5 bit'i okunmaktadır. Bundan dolayı program %100 korunmuş olur.

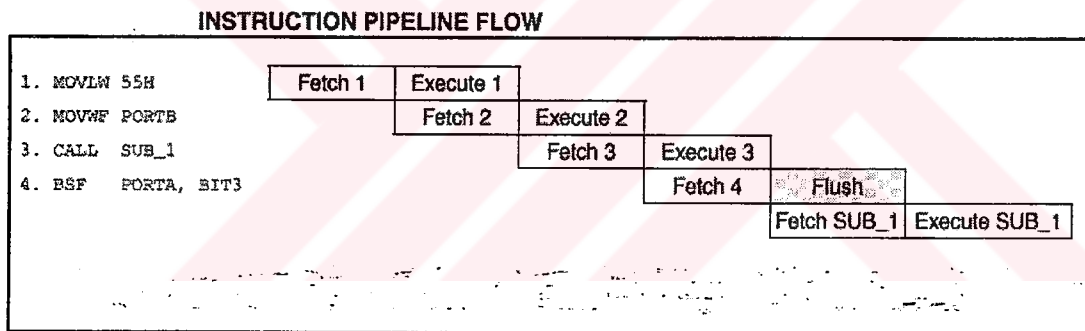
2.6. Saat Şeması/ Komut Çevrimi

Saat girişi (OSC1/CLKIN pini) PIC içinde dörder blok halinde kullanılır. Böylece üst üste çakışmayan Q₁, Q₂, Q₃ ve Q₄ isimli sinyaller oluşturulur. Bunun sebebi, komut ve işlemlerin çakışmasını önlemektir. Program sayıcısı (PC) her Q₁ palsinde artırılır ve

program hafızasındaki komut her Q_4 sinyalinde komut kayıtcısına iletilir. Komut kayıtcısındaki işlem Q_1 - Q_4 arasında sonuçlandırılır (Şekil 2.10 ve Şekil 2.11).



Şekil 2.10. PIC'in Saat şeması



Şekil 2.11. PIC'in komut çevrimi

2.7. Komut Akışı/Pipelining Hattı

Bir komut çevrimi dört Q çevriminden ($Q_1..Q_4$) oluşur. Komutun iletilmesi (Fetch) ve işlenmesi (Execute) birer çevrimde oluşacak şekilde hattın iletilir. Herhangi bir komutun program sayıcısının (PC) değişmesini gerektiriyorsa, bu işlem iki çevrimde işlenebilir.

İletim çevrimi (fetch cycle) program sayacının (PC) Q_n 'de artmasıyla başlamaktadır. İşlem çevriminde (execution cycle) iletilmiş komut, komut kayıtcısına Q_n çevriminde

gönderilir. Komutun incelenmesi ve sonuçlanması Q_2 , Q_3 ve Q_4 çevrimlerinde yapılır. Veri hafızasından okuma Q_2 'de ve yazma ise Q_4 'de yapılır.



BÖLÜM 3. PIC PROGRAMLAMA KOMUTLARI

Aşağıda PIC16C5X'e ait komutlar ve kısa uygulamaları görülmektedir [7].

ADDWF	Add W and f	ANDLW	And literal with W
Yazılım şekli:	[label] ADDWF f,d	Yazılım şekli:	[label] ANDLW k
Komut:	$0 \leq f \leq 31$ $d \in [0,1]$	Komut:	$0 \leq k \leq 255$
İşlem:	$(W) + (f) \rightarrow (\text{dest})$	İşlem:	$(W).AND.(k) \rightarrow (W)$
Etkilenen durumlar:	C, DC, Z	Etkilenen durumlar:	Z
Kodlama:	0001 11df ffff	Kodlama:	1110 kkkk kkkk
Açıklama:	W kayıtçısı ile 'f' 'in içeriklerini topla. Eğer 'd' değeri '0' ise sonucu W'de sakla. Eğer 'd' = '1' ise, sonucu 'f' 'de geri depola.	Açıklama:	W kayıtçısının içeriği, 8 bitlik literal 'k' ile VE işlemine tabi tutulur. Sonuç W kayıtçısına yerleştirilir.
Kelime:	1	Kelime:	1
Çevrim:	1	Çevrim:	1
Örnek:	ADDWF FSR,0	Örnek:	ANDLW 0x5F
Komuttan önce:	W = 0x17 FSR = 0xC2	Komuttan önce:	W = 0xA3
Komuttan sonra:	W = 0xD9 FSR = 0xC2	Komuttan sonra:	W = 0x03

ANDWF	AND W with f
Yazılım şekli:	[label] ANDWF . f,d
Komut:	$0 \leq f \leq 31$ $d \in [0,1]$
İşlem:	(W) .AND. (f) \rightarrow (dest)
Etkilenen durumlar:	Z
Kodlama:	0001 01df ffff
Açıklama:	W kayıtçısının içeriğini 'f' kayıtçısının içeriği ile VE işlemine tabi tut. Eğer sonuç $d=0$ ise bunu W kayıtçısında sakla. Eğer $d=1$ ise, 'f' kayıtçısında sakla.
Kelimeler:	1
Çevrim:	1
Örnek:	ANDWF FSR, 1
Komuttan önce:	W = 0x17 FSR = 0xC2
Komuttan sonra:	W = 0x17 FSR = 0x02

BCF	Bit Clear f
Yazılım şekli:	[label] BCF f,b
Komut:	$0 \leq f \leq 31$ $0 \leq b \leq 7$
İşlem:	$0 \rightarrow (f)$
Etkilenen durum:	Yok
Kodlama:	0100 bbbf ffff
Açıklama:	'f' kayıtçısındaki 'b' biti silindi.
Kelime:	1
Çevrim:	1
Örnek:	BCF FLAG_REG,7
Komuttan önce:	FLAG_REG = 0xC7
Komuttan sonra:	FLAG_REG = 0x47

BSF **Bit Set f**

Yazılım şekli: [label] BSF f,b

Komut: $0 \leq f \leq 31$
 $0 \leq b \leq 7$

İşlem: $1 \rightarrow (f \langle b \rangle)$

Etkilenen durum: Yok

Kodlama: 0100 bbbf ffff

Açıklama: 'f' kayıtçısındaki 'b' biti kuruldu.

Kelime: 1

Çevrim: 1

Örnek: BSF FLAG_REG,7

Komuttan önce: FLAG_REG = 0x0A

Komuttan sonra: FLAG_REG = 0x8A

BTFSK **Bit Test f, Skip if Clear**

Yazılım şekli: [label] BTFSK f,b

Komut: $0 \leq f \leq 31$
 $0 \leq b \leq 7$

İşlem: Sıçra şayet $(f \langle b \rangle) = 0$

Etkilenen durum: Yok

Kodlama: 0110 bbbf ffff

Açıklama: Eğer 'f' kayıtçısındaki 'b' biti 0 ise bir sonraki komutu atla. Eğer 'b' biti 0 ise halihazırdaki komutun işlenmesi sırasında tutulan bir sonraki komutu dışarı at ve yerine 2 sayıklık bir komut olan NOP'u icra et.

Kelime: 1

Çevrim: 1(2)

Örnek: BURADA BTFSK FLAG,1
HATA GOTO
:
DOĞRU
:

Komuttan önce: PC= address (BURADA)

Komuttan sonra: if FLAG<1> = 0,
PC=address (DOĞRU);
if FLAG<1> = 1,
PC=address (HATA)

BTfSS	Bit Test f, Skip if Set
Yazılım şekli:	[label] BTfSS f,b
Komut:	$0 \leq f \leq 31$ $0 \leq b \leq 7$
İşlem:	Sıçra şayet ($f \langle b \rangle = 1$)
Etkilenen durum:	Yok
Kodlama:	0111 bbbf ffff
Açıklama:	Eğer 'f' kayıtçısındaki 'b' biti 1 ise bir sonraki komutu atla. Eğer 'f' kayıtçısındaki 'b' biti 1 ise halihazırdaki komutun işlenmesi sırasında tutulan bir sonraki komutu dışarı at ve yerine 2 sayıklık bir komut olan NOP'u icra et.
Kelime:	1
Çevrim:	1(2)
Örnek:	BURADA BTfSS FLAG,1 HATA GOTO : DOĞRU :
Komuttan önce:	PC= address (BURADA)
Komuttan sonra:	if FLAG<1> = 0, PC=address (HATA); if FLAG<1> = 1, PC=address (DOĞRU)

CALL	Subroutine Call
Yazılım şekli:	[label] CALL k
Komut:	$0 \leq k \leq 255$
İşlem:	(PC) +1 → Top of Stack; k → PC<7:0>; (STATUS<6:5>) → PC<10:9>; 0 → PC<8>
Etkilenen durum:	Yok
Kodlama:	1001 kkkk kkkk
Açıklama:	Alt program çağırma, önce (PC+1) dönüş adresini yığıt bölgesine koy. 8 bit acil adres PC bitlerine yüklenir. (7:0). PC'nin üst bitleri (9:10), STATUS'dan yüklenir(6:5), PC(8) silinir. CALL 2 sayıklık bir işlemdir.
Kelime:	1
Çevrim:	2
Örnek:	BURADA CALL ORADA
Komuttan önce:	PC= address (BURADA)
Komuttan sonra:	PC=address (ORADA); TOS=address (BURADA + 1)

CLRF Clear f

Yazılım şekli: [label] CLRF f
Komut: $0 \leq f \leq 31$
İşlem: $00h \rightarrow (f);$
Etkilenen durum: Z
Kodlama: 0000 011f ffff
Açıklama: 'f' kayıtcısının içeriği silinir ve Z biti kurulur.
Kelime: 1
Çevrim: 1
Örnek: CLRF FLAG_REG
Komuttan önce: FLAG_REG = 0x5A
Komuttan sonra: FLAG_REG = 0x00
 Z = 1

CLRWDT Clear Watchdog Timer

Yazılım şekli: [label] CLRWDT
Komut: Yok
İşlem: $00h \rightarrow$ WDT;
 $0 \rightarrow$ WDT prescaler
 $1 \rightarrow$ TO;
 $1 \rightarrow$ PD
Etkilenen durum: TO,PD
Kodlama: 0000 0000 0100
Açıklama: CLRWDT komutu WDT'yi siler ve prescaler'i sıfırlar. Prescaler WDT'ye yönlendirilmemişse ve Timer0 değilse TO ve PD durum bitleri kurulur.
Kelime: 1
Çevrim: 1
Örnek: CLRWDT
Komuttan önce: WDT counter = ?
Komuttan sonra: WDT counter = 0x00
 TO = 1
 PD = 1

CLRW Clear W

Yazılım şekli: [label] CLRW
Komut: Yok
İşlem: $00h \rightarrow (W);$
 $1 \rightarrow Z$
Etkilenen durum: Z
Kodlama: 0000 0100 0000
Açıklama: W kayıtcısı temizlenir ve sıfır biti (Z) kurulur.
Kelime: 1
Çevrim: 1
Örnek: CLRW
Komuttan önce: W = 0x5A
Komuttan sonra: W = 0x00
 Z = 1

COMF Complement f

Yazılım şekli: [label] COMF f,d

Komut: $0 \leq f \leq 31$ $d \in [0,1]$ İşlem: $(f) \rightarrow (\text{dest})$

Etkilenen durum: Z

Kodlama: 0010 01df ffff

Açıklama: 'f' kayıtcısının içeriğinin tümleyeni alınır. Eğer $d=0$ ise sonuç W kayıtcısında depolanır. Eğer $d=1$ ise sonuç 'f' kayıtcısına geri depolanır.

Kelime: 1

Çevrim: 1

Örnek: COMF REG1,0

Komuttan önce: REG1 = 0x13

Komuttan sonra: REG1 = 0x13
W = 0xEC

DECF Decrement f**Decrement f**

Yazılım şekli: [label] DECF f,d

Komut: $0 \leq f \leq 31$ $d \in [0,1]$ İşlem: $(f) - 1 \rightarrow (\text{dest})$

Etkilenen durum: Z

Kodlama: 0000 11df ffff

Açıklama: 'f' kayıtcısını azalt. Eğer $d=0$ ise sonuç W kayıtcısına kaydedilir. Eğer $d=1$ ise sonuç 'f' kayıtcısına geri depolanır.

Kelime: 1

Çevrim: 1

Örnek: DECF CNT,1

Komuttan önce: CNT = 0x00

Z = 0

Komuttan sonra: CNT = 0x00

Z = 1

DECFSZ **Decrement f, Skip if 0**

Yazılım şekli: [label] DECFSZ f,d
Komut: $0 \leq f \leq 31$
 $d \in [0,1]$
İşlem: (f) - 1 → d; skif if result = 0
Etkilenen durum: Yok
Kodlama: 0010 11df ffff
Açıklama: 'f' kayıtçısının içeriği azaltılır. Eğer d=0 ise sonuç W kayıtçısına kaydedilir. Eğer d=1 ise sonuç 'f' kayıtçısına geri depolanır. Eğer sonuç 0 ise, bir sonraki komut atılır ve 2 sayıklık NOP işlemi icra edilir.

Kelime: 1
Çevrim: 1(2)
Örnek: BURADA DECFSZ CNT,1
GOTO DÖNGÜ
DEVAM

Komuttan önce: PC = address (BURADA)

Komuttan sonra: CNT = CNT - 1;
if CNT = 0,
PC = address (DEVAM)
if CNT ≠ 0,
PC = address (BURADA+1)

GOTO **Unconditional Branch**

Yazılım şekli: [label] GOTO k
Komut: $0 \leq k \leq 511$
İşlem: $k \rightarrow PC\langle 8:0 \rangle$;
STATUS<6:5> → PC<10:9>
Etkilenen durum: Yok
Kodlama: 101k kkkk kkkk
Açıklama: GOTO şartsız dallanmadır. 9 bitlik immediate değer, PC'nin <8:0> bitlerine yüklenir. PC'nin yüksek iki biti <6:5> STATUS'dan yüklenir. GOTO iki bitlik bir komuttur.

Kelime: 1
Çevrim: 2
Örnek: GOTO ORAYA

Komuttan sonra: PC = address (ORAYA)

INCF	Increment f	INCFSZ	Increment f, Skip if 0
Yazılım şekli:	[label] INCF f,d	Yazılım şekli:	[label] INCFSZ f,d
Komut:	$0 \leq f \leq 31$ $d \in [0,1]$	Komut:	$0 \leq f \leq 31$ $d \in [0,1]$
İşlem:	$(f) + 1 \rightarrow (dest)$	İşlem:	$(f) + 1 \rightarrow (dest)$, skip if result=0
Etkilenen durum:	Z	Etkilenen durum:	Yok
Kodlama:	0010 10df ffff	Kodlama:	0011 11df ffff
Açıklama:	'f' kayıtçısının içeriği artırılır. Eğer d=0 ise sonuç W kayıtçısına kaydedilir. Eğer d=1 ise sonuç 'f' kayıtçısına geri depolanır.	Açıklama:	'f' kayıtçısının içeriği artırılır. Eğer d=0 ise sonuç W kayıtçısına kaydedilir. Eğer d=1 ise sonuç 'f' kayıtçısına geri depolanır. Sonuç 0 ise halihazırda elaltında tutulan sonraki işlem komutu atılır ve yerine iki sayıklık işlem olan NOP komutu icra edilir.
Kelime:	1	Kelime:	1
Çevrim:	1	Çevrim:	1(2)
Örnek:	INCF CNT,1	Örnek:	BURADA DECFSZ CNT,1 GOTO DÖNGÜ DEVAM
Komuttan önce:	CNT = 0xFF Z = 0	Komuttan önce:	PC = address (BURADA)
Komuttan sonra:	CNT = 0x00 Z = 1	Komuttan sonra:	CNT = CNT + 1; if CNT = 0, PC = address (DEVAM) if CNT ≠ 0, PC = address (BURADA+1)

IORLW Inclusive OR literal with W

Yazılım şekli: [label] IORLW k

Komut: $0 \leq k \leq 255$ İşlem: (W).OR. (k) \rightarrow (W)

Etkilenen durum: Z

Kodlama: 1101 kkkk kkkk

Açıklama: W kayıtcısının içeriği ile 8 bitlik literal 'k' OR işlemine tabi tutulur. Sonuç W kayıtcısına yerleştirilir.

Kelime: 1

Çevrim: 1

Örnek: IORLW 0x35

Komuttan önce: W = 0x9A

Komuttan sonra: W = 0xBF

Z = 0

IORWF Inclusive OR W with f

Yazılım şekli: [label] IORWF f,d

Komut: $0 \leq f \leq 31$ İşlem: (W).OR. (f) \rightarrow (dest)

Etkilenen durum: Z

Kodlama: 0001 00df ffff

Açıklama: W kayıtcısı ile 'f' kayıtcısı OR işlemine tabi tutulur. d=0 ise, sonuç W kayıtcısına yerleştirilir. d= 1 ise sonuç tekrar 'f' kayıtcısına yeniden konur.

Kelime: 1

Çevrim: 1

Örnek: IORWF RESULT,0

Komuttan önce: RESULT = 0x13

W = 0x91

Komuttan sonra: RESULT = 0x13

W = 0x93

Z = 0

MOVF	Move f
Yazılım şekli:	[label] MOVF f,d
Komut:	$0 \leq f \leq 31$ $d \in [0,1]$
İşlem:	(f) → (dest)
Etkilenen durum:	Z
Kodlama:	0010 00df ffff
Açıklama:	'f' kayıtçısının içeriği 'd'ye taşınır. d=0 ise, hedef W kayıtçısıdır. d= 1 ise hedef dosyalayıcısı 'f' dir. d=1 durumu durum bayrağı Z'yi etkilediğinden bir dosya kayıtçısını test etmek için kullanışlıdır.
Kelime:	1
Çevrim:	1
Örnek:	MOVF FSR,0
Komuttan sonra:	W = FSR kayıtçısındaki değer

MOVWF	Move W to f
Yazılım şekli:	[label] MOVWF f
Komut:	$0 \leq f \leq 31$
İşlem:	(W) → (f)
Etkilenen durum:	Yok
Kodlama:	0000 001f ffff
Açıklama:	Veriyi W kayıtçısından 'f' kayıtçısına taşır.
Kelime:	1
Çevrim:	1
Örnek:	MOVWF TEMP_REG
Komuttan önce :	TEMP_REG = 0xFF W = 0x4F
Komuttan sonra:	TEMP_REG = 0x4F W = 0x4F

MOVLW	Move Literal to W
Yazılım şekli:	[label] MOVLW k
Komut:	$0 \leq k \leq 255$
İşlem:	k → (W)
Etkilenen durum:	Yok
Kodlama:	1100 kkkk kkkk
Açıklama:	Sekiz bitlik literal 'k' W kayıtçısına yüklenir. Farketmez durumlar 0 olarak assamble edilir.
Kelime:	1
Çevrim:	1
Örnek:	MOVLW 0x5A
Komuttan sonra:	W = 0x5A

NOP	No Operation
Yazılım şekli:	[label] NOP
Komut:	Yok
İşlem:	Yok
Etkilenen durum:	Yok
Kodlama:	0000 0000 0000
Açıklama:	İşlem yok işlemi
Kelime:	1
Çevrim:	1
Örnek:	NOP

OPTION Load OPTION Register

Yazılım şekli: [label] OPTION
 Komut: Yok
 İşlem: (W) → OPTION
 Etkilenen durum: Yok
 Kodlama: 0000 0000 0010
 Açıklama: W kayıtçısının içeriği seçenek kayıtçısına yüklenir.
 Kelime: 1
 Çevrim: 1
 Örnek: OPTION
 Komuttan önce : W = 0x07
 Komuttan sonra: OPTION = 0x07

RETLW Return with Literal in W

Yazılım şekli: [label] RETLW k
 Komut: $0 \leq k \leq 255$
 İşlem: $k \rightarrow (W)$
 TOS → PC
 Etkilenen durum: Yok
 Kodlama: 1000 kkkk kkkk
 Açıklama: W kayıtçısı 8 bit 'k' değeri ile yüklenir. Program sayıcı yığıt bölgesinin en üstündeki dönüş adresi ile yüklenir. Bu iki sayıklık bir komuttur.
 Kelime: 1
 Çevrim: 2
 Örnek: CALL TABLE ;Wcontains
 ;table offset
 ;value
 ;W now has table
 ;value.
 TABLE ADDWF PC ;W= offset
 RETLW k1 ;Begintable
 RETLW k2 ;
 RETLW kn ;End of
 table

Komuttan önce : W = 0x07

Komuttan sonra: W = value of k8

RLF Rotate Left f through Carry

Yazılım şekli: [label] RLF f,d
Komut: $0 \leq f \leq 31$
 $d \in [0,1]$
İşlem: Aşağıdaki açıklamaya bakın.
Etkilenen durumlar: C
Kodlama: 0011 01df ffff
Açıklama: f kayıtlarının içeriği taşma Flagtan sola doğru 1 bit kaydırılır. Eğer d= 0 ise sonuç W kayıtçısına yerleştirilir. d=1 ise sonuç tekrar f kayıtçısına konur.

Kelime: 1
Çevrim: 1
Örnek: RLF REG1,0
Komuttan önce: REG1 = 1110 0110
C = 0
Komuttan sonra: REG1 = 1110 0110
W = 1100 1100
C = 0

RRF Rotate Right f through Carry

Yazılım şekli: [label] RRF f,d
Komut: $0 \leq f \leq 31$
 $d \in [0,1]$
İşlem: Aşağıdaki açıklamaya bakın.
Etkilenen durumlar: C
Kodlama: 0011 00df ffff
Açıklama: f kayıtlarının içeriği taşma Flagından sağa doğru 1 bit kaydırılır. Eğer d= 0 ise sonuç W kayıta yerleştirilir. d=1 ise sonuç tekrar f kayıtına konur.

Kelime: 1
Çevrim: 1
Örnek: RRF REG1,0
Komuttan önce: REG1 = 1110 0110
C = 0
Komuttan sonra: REG1 = 1110 0110
W = 0111 0011
C = 0

SLEEP Enter SLEEP Mode

Yazılım şekli:	[label] SLEEP
Komut:	Yok.
İşlem:	00h→ WDT; 0→WDT Prescaler; 1→TO; 2→PD
Etkilenen durumlar:	TO, PD
Kodlama:	0000 0000 0011
Açıklama:	Time-out durum bir (TO) kurulur. Güç kesme durum biti (PO) silinir. WDT ve Prescaler temizlenir.
Kelime:	1
Çevrim:	1
Örnek:	SLEEP

SUBWF subtract w from f

Yazılım şekli:	[label] SUBWFLF f,d
Komut:	$0 \leq f \leq 31$ $d \in [0,1]$
İşlem:	$(f) - (W) \rightarrow (\text{dest})$
Etkilenen durumlar:	C, DC, Z
Kodlama:	0000 10df ffff
Açıklama:	W kayıtçısını f kayıtçısından çıkar.(2 tümleyen yöntemi ile) $d=0$ ise sonuç w kayıtçısına depolanır. $d=1$ ise sonuç f'ye geri yüklenir.
Kelime:	1
Çevrim:	1
Örnek 1:	SUBWF REG1, 1
Komuttan önce:	REG1 = 3 W = 2 C = ?
Komuttan sonra:	REG1 = 1 W = 2 C = 1 ; Sonuç pozitiftir.
Örnek 2:	
Komuttan önce:	REG1 = 2 W = 2 C = ?
Komuttan sonra:	REG1 = 0 W = 2 C = 1 ; Sonuç 0' dir.
Örnek 3:	
Komuttan önce:	REG1 = 1 W = 2 C = ?
Komuttan sonra:	REG1 = FF W = 2 C = 0 ; Sonuç negatiftir.

SWAPF	Swapf Nibbles in f	TRIS	Load TRIS register
Yazılım şekli:	[label] SWAPF f,d	Yazılım şekli:	[label] TRIS f
Komut:	$0 \leq f \leq 31$ $d \in [0,1]$	Komut:	$f = 5, 6 \text{ OR } 7$
İşlem:	$(f<3:0>) \rightarrow (dest<7:4>);$ $(f<3:0>) \rightarrow (dest<7:4>);$	İşlem:	$(W) \rightarrow \text{TRIS REGISTER } f$ $(f<3:0>) \rightarrow (dest<7:4>);$
Etkilenen durumlar:	Yok	Etkilenen durumlar:	Yok
Kodlama:	0011 10df ffff	Kodlama:	0000 0000 0fff
Açıklama:	f kayıtlarının düşük ve yüksek nibble'leri değiştirilir. d=0 ise, sonuç w kayıtcısına yerleştirilir. d=1 ise f kayıtcısına konur.	Açıklama:	f TRIS kayıtcısı (f=5,6,7), W w kayıtcısının içeriği ile yüklenir.
Kelime:	1	Kelime:	1
Çevrim:	1	Çevrim:	1
Örnek:	SWAPH REG1, 0	Örnek:	TRIS PORTA
Komuttan önce:	REG1 = 0XA5	Komuttan önce:	W = 0XA5
Komuttan sonra:	REG1 = 0XA5 W = 0X5A	Komuttan sonra:	TRIS = 0X5A

XORLW Exclusive OR literal with W

Yazılım şekli: [label] XORLW k
Komut: $0 \leq k \leq 255$
İşlem: (W) .XOR. k \rightarrow (W)
Etkilenen durumlar: Z
Kodlama: 1111 kkkk kkkk
Açıklama: W kayıtçısının içeriğini 8 bit literal k ile XOR işlemine tabii tut. Sonucu w kayıtçısına yerleştir.
Kelime: 1
Çevrim: 1
Örnek: XORLW 0XAF
Komuttan önce: W = 0XB5
Komuttan sonra: W = 0X1A

XORWF Exclusive OR W with f

Yazılım şekli: [label] XORWF f,d
Komut: $0 \leq f \leq 31$
 $d \in [0,1]$
İşlem: (W) .XOR. (f) \rightarrow (dest)
Etkilenen durumlar: Z
Kodlama: 0001 10df ffff
Açıklama: W kayıtçısının içeriğini f kayıtçısınınki ile ExOR işlemine tabii tut. d=0 ise sonucu W kayıtçısına kaydet. d= 1 ise sonucu f kayıtçısına yerleştir.
Kelime: 1
Çevrim: 1
Örnek: XORW REG,1
Komuttan önce: REG = 0xAF
W = 0XB5
Komuttan sonra: REG = 0x1A
W = 0xB5

BÖLÜM 4. PROGRAMLAMA

Basic, Pascal ve C ileri programlama dillerindeki gibi anlaşılır bir yazılıma ve görünümüne sahip olmayan assembler’de program yazmak kolay değildir. Bu zorluk iyi bir programlama tekniği ile giderilebilir. Programlamaya yeni başlayanların, yazma teknikleri eksiklikleri ve alt program yazılımlarının olmamasından dolayı, başlangıçta iyi program yazma ihtimali düşüktür. Bu nedenle, önce, ilk programlama anından itibaren iyi program yazabilmek için gerekli olan program yazma teknikleri ve alt programlar tanıtılacaktır [8, 9].

4.1. Döngüler

Alt program, belirli bir durum oluşana kadar tekrarlanan işlemlerdir. En çok kullanılan alt programlar, çeşitli türden döngülerdir. Tüm döngüler hemen hemen aynı yapıya sahiptirler. En yaygın anlamı ile döngü, belirli bir sayıda işlem yaptıktan sonra fonksiyonunu bitiren programlardır. Bu tür döngü örneği, Şekil 4.1’de görülmektedir.

```
*****
;
;sayma döngüsü, sayma COUNT registere yazılır.
;*****
;
DÖNGÜ          :          ;döngü içeriği buraya yazılır.
                :
                :          ;
                DECFSZ    SAYISI,1    ;döngü sayısını bir azalt ve döngü sayısı>0 ise
                                ;döngü devam edilir.
                GOTO      DÖNGÜ
                :          ;döngü sayısı<=0 ise buradan devam edilir.
```

Şekil 4.1. Sayma döngüsü.

Bir alt programın tekrarlanma sayısını gösteren döngü adedi, bir kayıtçıda yazılı olmalıdır. Bu kayıtçı 'COUNT' olarak adlandırılır. Döngü programının her çalışması sonunda, bu kayıtçı bir azaltılır. Döngü programının tekrarlanmasına 'COUNT' kayıtçısındaki değer sıfır olana kadar devam edilir. 'COUNT' kayıtçısındaki değer sıfır olunca döngüden çıkılır.

Diğer bir döngü çeşidi; sabit bir sayıda değil, belirlenen bir koşul oluşana kadar devam eden döngülerdir. PIC'de yazılan Assembler programlama, ileri programlama dillerindeki gibi sınırsız koşul belirleme imkanına sahip değildir. PIC programlamada, sadece bitlerin durumu kontrol edilebilir. Bu nedenle koşul olarak da sadece bitlere dayalı koşullar kullanılabilir. Şekil 4.2'deki koşullu döngü örneğinde, bir kayıtçadaki BIT'in sıfır olup olmadığı kontrol edilmektedir.

```

;*****
;şartlı döngü, Döngüden çıkmak için 'BYTE' kayıtçısındaki 'BIT' biti '0' olmalıdır.
;*****

DÖNGÜ      :                               ;döngü içeriği buraya yazılır.
           :                               ;
           :                               ;
           :                               ;
           BTFSC      BYTE,BIT             ;bit kontrol ediliyor, eğer BIT=1 ise
                                           döngü tekrar edilir

           GOTO      DÖNGÜ

           :                               ;döngüden sonra buradan devam
                                           edilir.

```

Şekil 4.2. Şartlı döngü program örneği

Şekil 4.2'deki programda kayıtçı 'BYTE', kontrol edilen bit ise 'BIT'dir. Döngü, BYTE'deki BIT'in '0' olması durumunda ana programa dönmektedir.

4.2. Kayıtçı İçeriklerinin Karşılaştırılması

PIC'de yazılan programda birkaç bit'in durumunu aynı anda kontrol etmek için herhangi bir komut olmadığından, bu işlem bir program yardımı ile gerçekleştirilir (Şekil 4.3).

```

*****
;
;register içeriğinin sabit bir sayı ile kontrol edilmesi
*****
;
MOVWF    TEMP        ;W kayıtçısı yedeklenir.
MOVF     REG,0       ;kontrol edilecek kayıtçı yedeklenir.
BCF      STATUS,ZERO ;ZERO flag silinir
SUBLW    KONST       ;sabit (KONST) ve W birbirinden
                    ;çıkarılır
BTFS    STATUS,ZERO  ;ZERO flag =1 sorgulaması yapılır.
GOTO     EŞİT        ;ZERO =1 ise EŞİT döngüye dallanılır
EŞİTDE   MOVF        TEMP,0   ;W nin yedeklenen içeriği, W'ye
                    ;yeniden yazılır
;
;
;buraya eşit olmadığı durumda
;                    ;yapılması istenen işlem yazılır
;
;
EŞİT     MOVF        TEMP,0   ;W'nin yedeklenen içeriği, W'ye
                    ;yeniden yazılır
;
;
;buraya eşit olduğu durumdaki
;                    ;işlemler yazılır
;
;

```

Şekil 4.3. Kayıtçı içeriğinin sabit bir sayı ile kontrol edilmesi.

Şekil 4.3’de görülen programda, iki sayı birbiri ile karşılaştırılır. Bu karşılaştırmada iki sayı birbirinden çıkarılıp, sonucun sıfır olup olmadığına bakılır ve elde edilen sonuca göre yapılacak işleme karar verilir. Yazılan alt programın özelliği, karşılaştırma yapılmadan önce W kayıtçısının yedeklenmesidir. Bunun yapılmasının nedeni; bazı durumlarda karşılaştırmaların sonucuna göre, W’nin değerinin değiştirilmesidir. Şekil 4.3’deki programda, karşılaştırma yapılmadan önce 'W' kayıtçısı 'TEMP'’te yedeklenir. Kontrol edilecek kayıtçının içeriği 'W' kayıtçısına yüklenir. 'W' kayıtçısındaki değerden, sabit bir sayı olan 'KONST' değeri çıkarılır.

Çıkarma işleminin sonucunu kontrol etmek için, STATUS kayıtçısındaki ZERO Flag kontrol edilir. Eğer, ZERO Flag=0 ise iki değer aynı olduğu anlaşılır ve buna ait programa atlanır (dallanılır). Gidilen alt programda TEMP’deki değer W’ye yüklenir ve istenen işlemlerin yapılmasına devam edilir. Karşılaştırma sonucunda ZERO Flag=1 ise, iki sayının aynı olmadığına karar verilir. Bu durumda, sayıların eşit olmaması durumunda yapılması istenen işlemler yapılır. Burada dikkat edilmesi gereken husus; BIT’lerin önceden yapılan başka işlemlerin sonucu olarak farklı bir değere sahip olma ihtimaline karşı kontrol işlemi yapılmadan önce silinmesi gerektiğidir.

İki sayının veya kayıtçının karşılaştırılması yanında, W kayıtçısının değerinin bir kayıtçının içeriği ile karşılaştırılması yapılabilir. Bu işlemi yapan program Şekil 4.4’de görülmektedir.

```

*****
;
;bir kayıtçının içeriğinin 'W' ile karşılaştırılması.
*****
                BCF          STATUS,ZERO      ;Zero flag silinir
                SUBWF       REG,0            ;REG ile W çıkarılır
                BTFSC       STATUS,ZERO      ;Zero flag=1 mi diye kontrol edilir
                GOTO        EŞİT             ;ZERO =1 ise EŞİT döngüye git
EŞİTDE          :
EŞİT            :
                :
                :

```

Şekil 4.4. Bir kayıtçının içeriğinin W ile karşılaştırılması.

Şekil 4.4'deki programda 'W' değeri karşılaştırılacak kayıtçının içeriğini temsil eden REG değerinden çıkarılır. REG değerinin değişmemesi için sonuç W'ye yazılır. Sonucun belirlenmesinde, Şekil 4.3'deki verilen programdaki yol takip edilmektedir.

4.3. Programlamada Başlangıç Durumlarının Belirlenmesi

PIC'leri programlamada dikkat edilmesi gereken diğer önemli bir özellik, RESET işlemi sonucunda programın yeniden başlaması durumudur. PIC'ler, RESET işleminden sonra baştan başlama yerine programın en sonundan çalışmaya başlarlar. Bu nedenle, en son adresten ana programa gidecek bir atlama komutu gerekir. Şekil 4.5'deki program bu söylenen işlemi yapar. PIC'deki Ana programın mutlaka 0000h adresinde olması gerekmez. Ana program başka bir adreste olabilir. Önemli olan, RESET anında ana programın olduğu yere gidilecek bir atlama komutunun olmasıdır.

```

;*****
;
;program başı (PIC16C56)
;*****
SON          ORG          03ffh          ;EPROM'un son adresi
             GOTO         ANA           ;ANA programa git
             ORG          0000h         ;ANA programın başı
ANA          :
             :                       ;ANA program buradan itibaren
             :                       başlıyor
             :
             :

```

Şekil 4.5. PIC programlamada program başının belirlenmesi.

Her PIC farklı EPROM hafızasına sahip olduğundan, bitiş adresleri de farklı olmaktadır. Tablo 4.1'de çeşitli PIC'lere ait RESET adresleri görülmektedir.

Tablo 4.1. Farklı PIC'lerin EPROM hafızaları

PIC	SON adres
16c54	01ffh
16c55	01ffh
16c56	03ffh
16c57	07ffh
16c71	03ffh
16c84	03ffh

Ayrıca Reset adresi mutlaka en son adres olacak diye bir şart yoktur. Bu nedenle, programlanacak PIC'e ait üreticinin el kitabına bakılması gerekir.

4.4. Dolaylı Yoldan Adresleme

Programlamada kullanışlı olan bir başka teknik (metot), dolaylı yoldan adreslemedir. Bu metoda alışmak biraz vakit olsa da öğrenildikten sonra sık kullanılan bir metottur. Dolaylı yoldan programlama için iki kayıtçıya ihtiyaç vardır. Bunlardan birincisi 'FSR' kayıtçısıdır. FSR'ye adreslenecek kayıtçı konulur.

Örneğin 13h adresli kayıtçı ile bir işlem yapacaksak FSR'ye 13h yazmamız gerekir. Bu adresten sonra 14h işleyeceksek, sadece FSR'yi bir arttırmamız yeterlidir. Buraya kadar herhangi bir ayrıcalık görülmemektedir.

FSR'ye 13h yazıldığını ve bu adrese yapılacak işlemin, ilgili adresin içeriğini silmek olduğunu varsayalım. Bu işlemleri yapmak için 'clrf 13h' dememiz gerekmektedir. Aynı işlem dolaylı adresleme yönteminde 00h adresli kayıtçı ile de yapılabilir. Sonuç olarak 00h adresli kayıtçı ile yapılacak tüm işlemler, FSR kayıtçısı ile belirlenen adresle dolaylı yoldan yapılmış olur. Bunu bir örnek ile daha iyi açıklayabiliriz (Şekil 4.6).

```

;*****
;dolaylı yoldan adresleme
;*****

START      MOVLW      D'17'      ;W değerini D'17' ile yükle
           MOVWF     SAYICI     ;ve sayıcı kayıtcıya yükle
           MOVLW     10h
           MOVWF     FSR        ;10h'ı FSR'ye yükle
DÖN        CLRF     IND         ;FSR'deki adresin içeriğini sil
           INC      FSR,1      ;FSR'yi bir artır
           DECFSZ   SAYICI     ;SAYICI'yı bir azalt, sıfır olunca bir
                                   sonrakinini atla
           GOTO     DÖN
                                   ;SAYICI sıfır olduğunda buradan
                                   devam edilir

```

Şekil 4.6. Dolaylı yoldan adresleme.

Şekil 4.6'daki örnekte, dolaylı adresleme ile bir hafıza bölümü silinmektedir. Dolaylı adresleme olmadan, 17 adresin teker teker silinmesi gerekirdi ve toplam 17 satırlık bir işlem oluşurdu. Fakat dolaylı adresleme ile silinecek adres FSR'ye yazıldı (burada 10h) ve 00h yani IND adresi silinerek 10h'ın içeriği silinmiş oldu. 11h'nolu adresin içeriğini silmek için FSR bir arttırılır. FSR'nin gösterdiği adresi silerseniz, 11h'ın içeriği silinmiş olur. Böylece; bir döngü işlemi ile, bir hafıza bölümü direkt adresleme yöntemine göre çok daha kısa bir programla silinmiş olur.

4.5. Makro Programlama

Makro programlama alt programlar gibidir. Yani programın bir yerinde yazıldıktan sonra, gerektiğinde yada çağırıldığında oraya gidilir ve işlem orada yapılır. Sonra eski yerine (gidiş adresine) dönlür. Bu işlem alt programlar için de geçerlidir. Aralarındaki fark, assemble işleminde makro, program içerisinde kaç defa kullanılıyorsa o kadar programa eklenir. Sonuç olarak simülasyon programında makrolar aynı alt programlar gibi işlem görmektedir. Fakat PIC'e aktarılma işlemi

yapılmak için assemble edildiğinde makronun kullanıldığı her yere makronun aynen yazılması sağlanmaktadır.

Makronun kullanım amacı, ana program içinde bir program parçasının devamlı veya çok defa kullanılması gerektiğinde devamlı olarak yazılmamasını sağlamaktır. Fakat assemble edildiğinde makro olan yerler sanki her defa yazılmış gibi algılanır. Makronun tanımını Şekil 4.7'de görülmektedir.

```
<makroname> macro [<parametre 1> <parametre 2> <parametre n>]
    <makrocodu>
endm
```

Şekil 4.7. Makro programı genel yapısı.

Makro programı 'makro' kelimesi ile başlar ve 'endm' ile biter. Makro programlamaya ait bir örnek Şekil 4.8'de görülmektedir.

```
sil        makro
;bu makro W ve DENEME registerlerini siliyor
        clrw
        clrf        DENEME
endm
```

Şekil 4.8. Örnek makro programı.

Bir makro, program içinde Şekil 4.9'daki gibi çağrılır.

movlw	10	;W=10
movwf	DENEME	;DENEME=10
sil		;W=0, DENEME=0

Şekil 4.9. Program içinde makro çağırma.

Şekil 4.9'da görüldüğü gibi, makro tanımlamada kullanılan özel isim 'sil', program içinde makronun çağrılması için yeterlidir. Makrolar, tanımından görüldüğü gibi, parametreler içerebilir. Aşağıdaki makroda, makro parametresi TEST kayıtcısı ile toplanmaktadır (Şekil 4.10).

topla	macro	sayı	
;sayı içeriği ile DENEME'nin birbiri ile toplanması			
	movlw	sayı	;W=sayı
	addwf	DENEME,1	;DENEME=DENEME+W
endm			

Şekil 4.10. Makro parametresinin test kayıtcısı ile toplanması.

Parametre kullanımında çok dikkat edilmesi gereken husus, makroda kullanılan parametrenin türüdür. Yani makro parametresinin kayıtcı adresini mi yoksa kayıtcının içeriğini mi kullanacağı belirlenmelidir. Bu örnekte, MOVLW Komutu kullanıldığında kayıtcının adresi kullanılmakta, yani kayıtcının içeriği kullanılmamaktadır.

Bu işlemi program içinde açıklarsak;

topla 07h

ile TEST +07h toplamı yapılır. Fakat:

topla TEST 2

ile TEST+'TEST 2' nin adres değeri toplanır içeriği toplanmaz.

Makrolarda parametre olarak kayıtçı adresleri ve sabit değer kullanıldığında komut olarak MOVLW ve ADDLW komutları, parametrelerde kayıtçının içeriğinin kullanılması durumunda ise MOVF ve ADDLW komutları kullanılmalıdır.

Makro programında dikkat edilmesi gereken diğer bir önemli husus, makronun içinde kullanılan atlama komutlarıdır. Çünkü, makro programları assembler programında birden fazla bulunabileceğinden ve atlama işlemi bir defa kullanılabildiğinden hata oluşur.

4.6. Maskeleye

Programlama sırasında bir kayıtçıda bulunan bir veya birden fazla bitin değerinin kontrol edilmesi gerekebilir. Bu işlemin diğer bitlerin değerleri değiştirmeden yapılabilmesi için BCF ve BSF komutları kullanılır. Fakat bu komutlarla birden çok bitin kontrolü zor ve uzundur. Ayrıca programın çalışması sırasında 0 veya 1 yapılması gereken bitin değeri bilinemeyebilir. Bu bitlerin değeri programın çalışmasından sonra belirlenmektedir. Bu durumda kayıtçının 0 veya 1 yapılması gereken bitlerini belirlemek için, maskeleye biti oluşturulur. Bu bit maskesi ile beraber lojik komutlar kullanılarak 0 veya 1 yapılacak kayıtçılarda işlenir.

Bir kayıtçıda istenilen bitleri '1' yapmak için kullanılan lojik komut OR'dur. Örneğin, TEST kayıtçının 2. ve 4. bitlerini '1' yapmak istiyorsak, bit maskeleye byte'i 1010b olmalıdır. Böylece, TEST kayıtçının diğer bitlerine dokunulmadan 2. ve 4. bitler '1' yapılmış olur.

Sonuç olarak:

$$\begin{array}{rcl}
 \text{TEST} & = & 00010001 \\
 \underline{\text{OR MASKE}} & = & 00001010 \\
 & & 00011010
 \end{array}$$

bulunur.

Aynı şekilde bir kayıtçının belirli bitlerini '0' yapmak için, AND lojik komutu kullanılır. Bu işlemde '0' yapılacak bitlerin dışındaki tüm bitler '1' yapılır. Örneğin; TEST kayıtçısının 2. ve 4. bitlerini '0' yapmak için bit maskeleme byte'i 11110101b olmalıdır.

Sonuç olarak:

$$\begin{array}{rcl}
 \text{TEST} & = & 00111011 \\
 \underline{\text{AND MASKE}} & = & 11110001 \\
 & & 00110001
 \end{array}$$

bulunur.

Böylece TEST registerlerdeki 2.ve 4. bitlerin haricindeki bitlere dokunulmadan, 2. ve 4. bitler '0' yapılır.

BÖLÜM 5. MODÜL TASARIMI ve YAPIMLARI

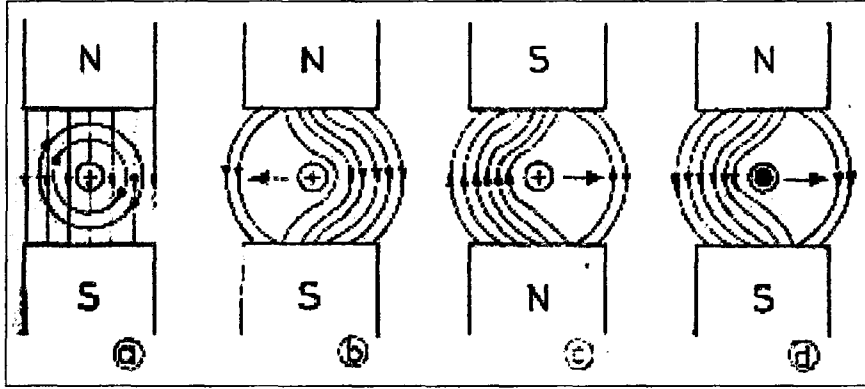
5.1. D.C. Motor Kontrol Devresi Tasarımı ve Yapımı

5.1.1. Doğru akım motorları

Doğru akım motoru, doğru akım elektrik enerjisini mekanik enerjiye dönüştüren elektrik makinasıdır. Manyetik alan içerisinde bulunan bir iletkenin akım geçerse iletken, içinden geçen akımın ve manyetik alanın yönüne bağlı olarak hareket eder. İletkenin hareket yönü sol el kuralı ile bulunur.

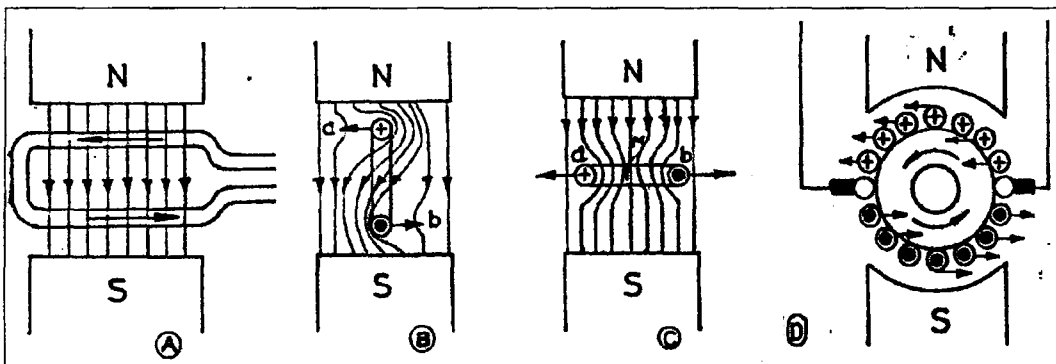
Sol el kuralı: sol el dört parmak birbirine birleştirilerek açılır. Avuç içi N kutbuna bakacak şekilde tutulur. Dört parmağın gösterdiği yön akım yönü ise iletken baş parmak yönünde hareket eder.

İçinden akım geçen bir iletkenin manyetik alan içindeki durumu şu şekilde açıklanabilir: iletken içinden geçen akım, iletken etrafında bir alan oluşturur (Şekil 5.1.a). Kutuplar arasında ve iletken etrafında oluşan bu alan, Şekil 5.1.a'da görüldüğü gibi, sol tarafta ana alana ters, sağda ise ana alanı kuvvetlendirecek yöndedir. Bunun sonucu olarak kutup alanı Şekil 5.1.b'deki gibi durum alarak, iletken üzerinde itici bir kuvvet etkisi yapar ve iletkeni alanın dışına doğru iter. İletken içinden geçen akımın yönü sabit kalmak şartıyla manyetik alanın yönü değiştirilecek olursa iletkenin hareket yönü değişir (Şekil 5.1.c). Manyetik alanın yönü sabit tutulup iletkenin geçen akımın yönü değiştirilirse hareket yönü de değişir.



Şekil 5.1. İçinden akım geçen bir iletkenin manyetik alan içindeki durumu.

İçinden akım geçen bir bobinin manyetik alan içindeki durumu da şu şekilde açıklanabilir: N-S kutupları arasına, bir eksen etrafında dönebilen bir bobin yerleştirilirse, bobinden akım geçtiği anda bobinin a-b kenarları üzerinde itici kuvvetler doğar. Şekil 5.2'de görüldüğü gibi bobin a kenarından sola, b kenarından sağa doğru itilir. Bu durumda a ve b kenarları birbirlerinin ters yönünde kuvvetler tarafından itileceğinden bobinin hareketi durur. Bobinin durmasına engel olmak için, a kenarı N kutbunun etkisi altına girerken, içinden geçen akım yönünün değiştirilmesi gerekir. Bu durum bobin uçlarını iki dilimli bir kollektöre bağlayıp, fırçalardan gerilim verilerek sağlanır.



Şekil 5.2. İçinden akım geçen bir bobinin alan içindeki ve kutuplar arasındaki endüvinin durumu.

Kutuplar arasına tek bir bobin yerine birden fazla bobin yerleştirilir ve bobin uçları kollektöre bağlanırsa sistemin düzgün bir şekilde dönmesi sağlanır. Böylece motora elektrik enerjisi uygulanması durumunda motor döner (Şekil 5.2.d).

5.1.2. Doğru akım motorlarında zıt E.M.K. ve devir sayısı

Bir doğru akım motorunun endüvisinden akım geçtiğinde, meydana gelen döndürücü kuvvetten dolayı, endüvinin döneceği yukarıda açıklandı. Bilindiği gibi, manyetik alan içerisinde dönen ve iletkenleri kuvvet çizgileri tarafından kesilen endüvi üzerinde e.m.k. endüklenir. Endüviye uygulanan gerilim (U) ile, endüvide endüklenen e.m.k.'nın yönleri birbirine terstir. Endüviye uygulanan gerilime ters yönde olan bu e.m.k.'ya 'zıt e.m.k.' denir. Endüviye uygulanan U gerilimi ile, zıt e.m.k.(Ez) birbirine ters yönde olduklarından, endüviden akan akım, iki gerilim farkından dolayı akan akımdır. Endüvi devresinin direncini Ra ile gösterirsek akan akım:

$$I_a = \frac{U - E_z}{R_a} \text{ Amper,} \quad (5.1)$$

zıt e.m.k.'nın değeri ise:

$$E_z = U - I_a.R_a \text{ Volt} \quad (5.2)$$

olarak bulunur.

Doğru akım motorlarında, makinenin gücüne göre geçen akımı ayarlama işini zıt e.m.k. yapar. Yükten dolayı oluşan devir sayısının azalması durumunda Ia artarak motorun gücünü karşılarken, motorun yükünün azalması durumunda oluşan devir sayısının artması sonucunda Ia azalarak motorun devrini düşürür.

Doğru akım motorlarında devir sayısını bulmak için zıt e.m.k. formülünden faydalanılır. Devir sayısı (n) :

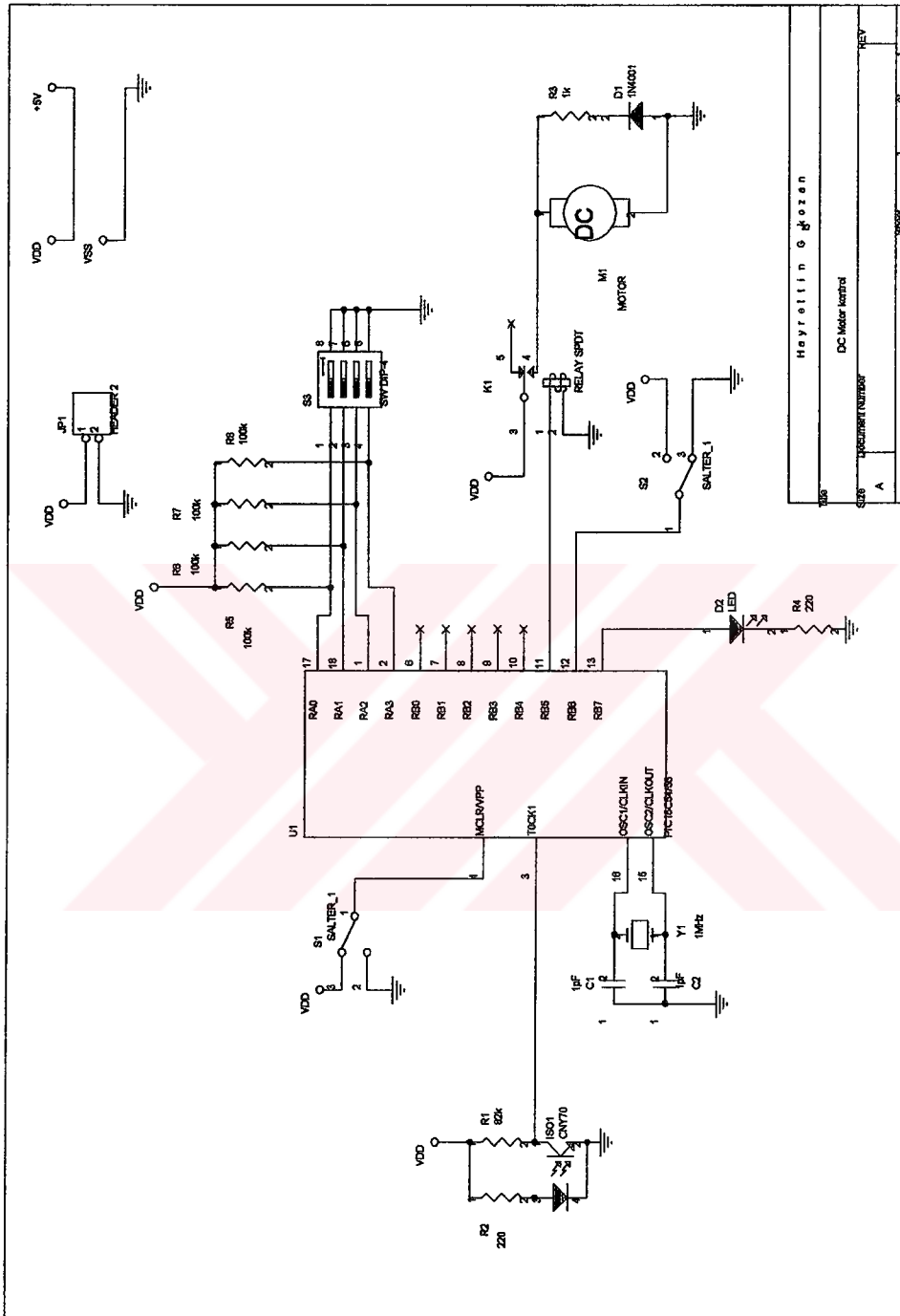
$$n = \frac{U - I_a.R_a}{k.\Phi} \quad (5.3)$$

formülüyle bulunur. Formülde $I_a.R_a$ çok küçük ve k sabit sayı olduğundan, devir sayısı uygulanan gerilim (U) veya Φ manyetik akısına (maxvell) bağlı olarak değişir. Uygulamalarda, U gerilimini değiştirmek sureti ile devir sayısı değiştirilir.

5.1.3. D.C. motor kontrol devresi

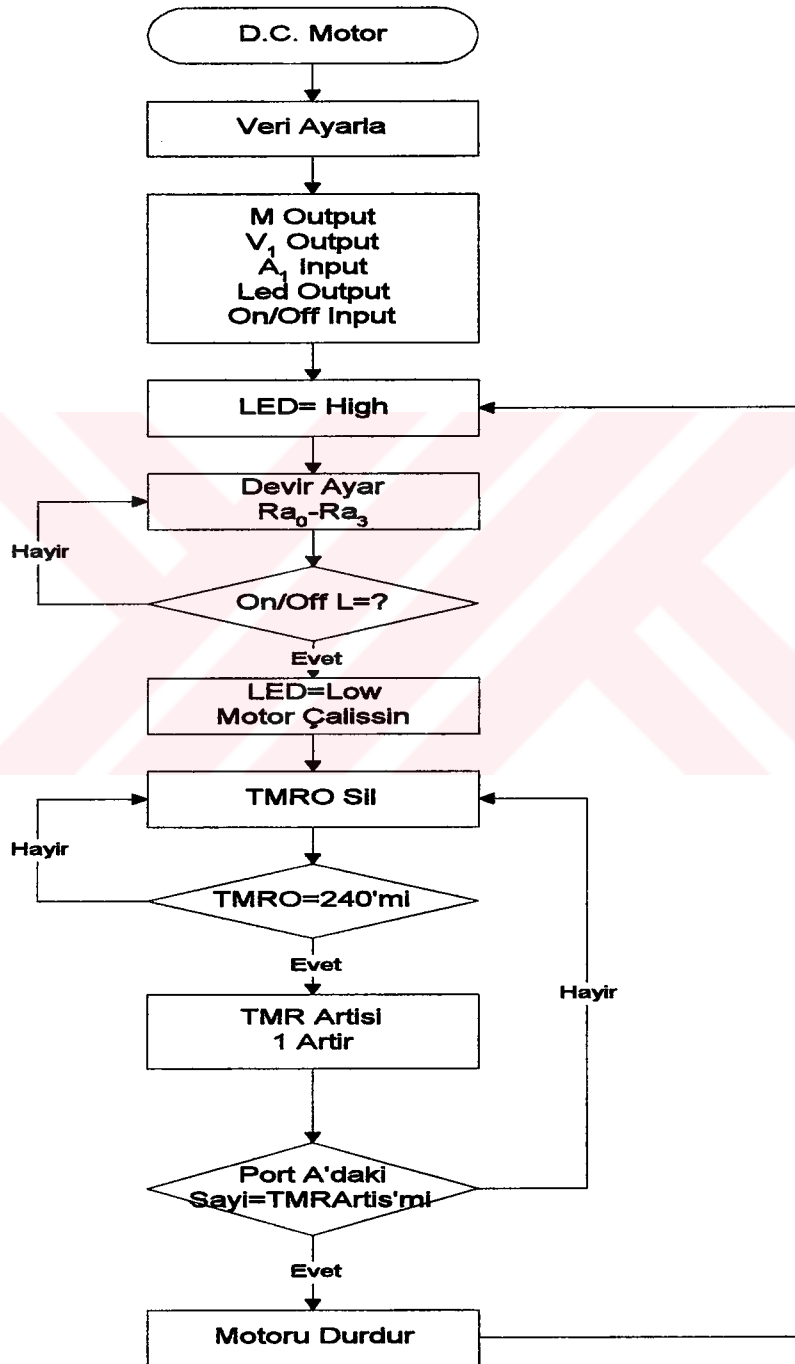
Kullanılan bir motorun dakikadaki veya toplam devir sayısının kontrol edilmesi önemlidir. Bu çalışmada PIC16C54 kullanarak belirli bir sayıda dönme sağlayan motorun kontrolü amaçlanmıştır.

Tasarlanan D.C. motor kontrol devresi, devir ayarlamasının yapılacağı anahtarlama kısmı ile motor ve devir sayısını kontrol edecek kısımdan oluşmaktadır. Anahtarlama kısmı ile gerekli devir sayısı ayarlandıktan sonra motorun çalıştırma anahtarı kapatılır. Bu durumda daha önce yanık olan 'hazır' ledi söner. Motorun miline bağlı olan delikli bir levha yardımı ile opto kuplör tarafından elde edilen sinyal mikrokontrolöre gönderilir. Devir sayısı bu şekilde sayılan motorun toplam yapmış olduğu devir, anahtarlama kısmı ile ayarlanan devir sayısına ulaştığında motorun enerjisini sağlayan röle devreden çıkartılarak motor durdurulur.



Şekil 5.3. D.C. motor hız kontrol devresi.

Şekil 5.3'deki D.C. motorun devir sayısı kontrolünü yapacak programın akış şeması Şekil 5.4'de görülmektedir. Buraya kadar yapılan işlemlerde D.C. motorun devir sayısı kontrolü gerçekleştirilmiştir. Bu devreye, motorun kaç devir döndüğünü gösteren bir devre ilave edilebilir.

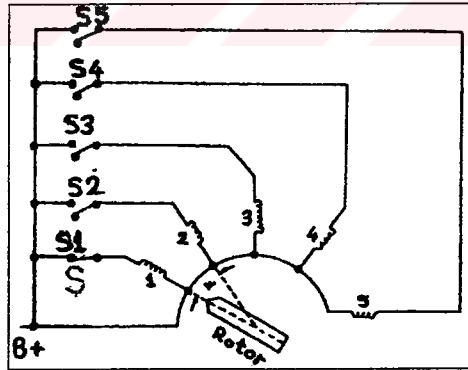


Şekil 5.4. D.C. motor program akış şeması.

5.2. Step Motor Modülü Tasarımı ve Yapımı

Aldığı palsa bağlı olarak hareket eden (belirli açılarla dönme hareketi yapan), dijital kontrol sistemlerinde mil, iş tablası, yazıcı çizici ve benzeri elemanların konum kontrolü amacıyla kullanılan motora, 'step (adım) motor' denir. Bu tür motorda dönme hareketi uygulanan palsa bağlı olarak değişir. Birkaç kW güce kadar imal edilebilen bu motorlar, programlama ile çalışan makinelerin tabla kontrolünde, yazıcı cihazların şerit ve kalem kontrolünde, x-y grafik çizimlerinde yaygın olarak kullanılırlar.

Normal D.C. motorlarda, gerilim uygulanması ile dönme hareketi başlayıp devam ederken, step motorlarda bir palsın uygulanması ile palsın sebep olacağı miktarda (örneğin 7 derece) dönme olduktan sonra motor yeni bir palsın gelmesini bekler. Her bir pals için dönme miktarı aynıdır. Rotoru çok kutuplu, statoru ise 3 veya 4 faz sargılı olarak yapılan step motorda rotor kutup sayısı, her bir pals ile oluşması istenen dönme miktarı elde edilecek şekilde seçilir.



Şekil 5.5. Step motorun çalışma prensibi.

Step motorun çalışmasının açıklandığı temel işlem Şekil 5.5'de görülmektedir. Adım işlemi bir daire içerisinde yerleştirilmiş elektromıknatıslar serisine benzetilebilir. 1 numaralı anahtarın kapatılması durumunda sabit mıknatıs 1 numaralı elektromıknatıs tarafından ve 2 numaralı anahtarın kapatılması ile de sabit mıknatıs 2 numaralı

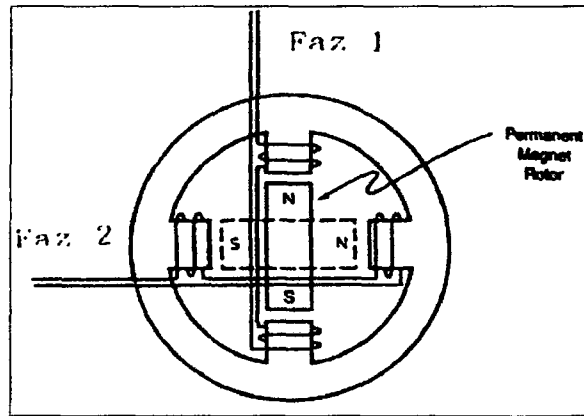
elektromıknatıs tarafından çekilir. Anahtarların düzgün bir sıra ile kapatılması durumunda, sabit mıknatıs tam bir daireyi tamamlar.

5.2.1. Step motorların yapısı ve çalışması

Step motorlar Sabit mıknatıslı (Permanent Magnet-PM) ve Değişken relüktanslı (Variable Reluctance-VR) olmak üzere iki değişik yapıda imal edilirler. Bu çalışmada PM türü step motor kullanılacağından bu tip step motor incelenmiştir.

5.2.1.a Sabit mıknatıslı (PM) step motorlar

Sabit mıknatıslı motorlar, elektromanyetik alan ile sabit mıknatıslı rotor arasındaki reaksiyona göre çalışırlar. En basit PM motor, dört kutuplu yarıkli stator içerisinde hareket eden iki kutuplu sabit mıknatıstan oluşur. Bu şekildeki step motorun yapısı Şekil 5.6'da görülmektedir. Şekil 5.6'da görülen step motorda faz 1'e gerilim uygulanırsa, bobinlerin alt tarafında N kutbu, üst tarafında S kutbu olacak şekilde bir manyetik alan oluşur. Farklı kutuplar birbirini çekeceğinden rotor kendisini manyetik alanın içerisinde bulur.



Şekil 5.6. Sabit mıknatıslı step motor.

Faz 1'den enerji kesilerek, faz 2'ye uygulanırsa; bu durumda sađ taraf N kutbu sol taraf S kutbu olacak şekilde bir manyetik alan oluşur. Rotor bu hareketi takip ederek kendisini kesik çizgilerle gösterilen duruma getirir. Faz 1 ters yönde enerjilendirilir ve pals uygulanmaya devam edilirse motor, D.C. motorlardaki hareketi yapar.

5.2.2. Step motor parametreleri ve uyartım yöntemleri

Yapı bakımından D.C. ve A.C. motorlardan farklı olan step motorlarda, farklı parametreler bulunur. Bu parametrelerden birisi palsın verildiđi süre ile motorun tepkisi arasındaki süreyi ifade eden ve adım tepkisi (step response) olarak isimlendirilen (ms cinsinden) parametredir. Adımlama oranı (stepping rate) olarak anılan diđer parametre, motorun bir saniyede yapabileceđi maksimum adım sayısını vermektedir. Diđer önemli parametre ise, uygulanan her bir pals ile motorun yaptığı açığı ifade eden adım açısıdır (step angle).

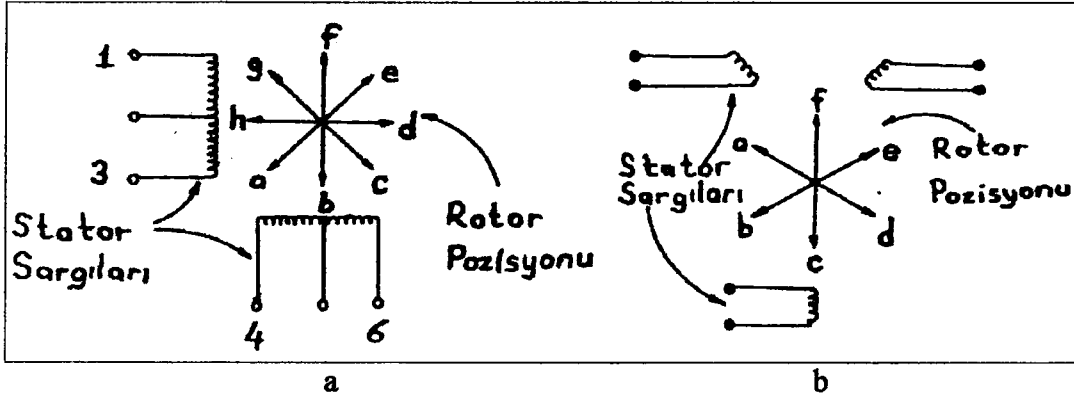
Step motorlar deđişik uyartım sistemlerine sahiptirler. Çıkış gücü, adım tepkisi, minimum güç girişı, verim gibi parametreler farklı uyartım yöntemlerine göre deđişiklik gösterirler. Stator sargılarına ve istenilen performansa bađlı olarak seçimi yapılan uyartım yöntemleri aşıđıdaki guruplara ayrılabilir.

İki faz ve iki faz modified

Üç faz ve üç faz modified

Dört faz ve dört faz modified

Faz ile stator sargıları ifade edilirken, modified ile aynı anda sürülen iki sargı açıklanır.



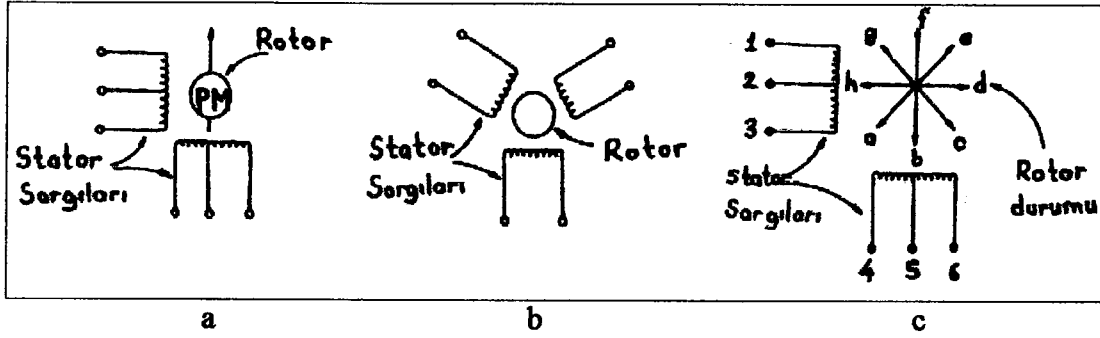
Şekil 5.7. Değişik yapıdaki step motor çeşitleri.

Şekil 5.7.a'da görülen orta uçlu stator sargıya sahip step motorun orta ucunun kullanılması durumunda 4 fazlı motor oluşur. Şekil 5.7.b'de görülen yıldız şeklinde yapıya sahip motorlarda ise, 3 faz veya 3 faz modified yöntemleri kullanılır. Kullanacağımız yöntemle ilgisi olması bakımından iki faz modified yöntemini inceleyelim.

İki faz modified yönteminde iki sargı (orta uçlar ihmal edilerek yalnızca kenarlardaki uçlar devrede düşünülür) aynı anda enerjilendirilir. İlk anda akım stator içerisinden 3-1 yönünde akar, bir sonraki adımda ters yönde (1-3) akar. Aşağıdaki çizelgede 2 faz ve 2 faz modified yöntemlerinin uygulanması durumunda sargıların ve rotorun durumu incelenmektedir.

Tablo 5.1. Step motor uyarım yöntemleri.

UYARTIM YÖNTEMİ	ENERJİLENDİRİLEN SARGILAR	ROTOR POZİSYONU
İki faz B+ Komitasyon	3-1	F
	6-4	H
İki faz B- Komitasyon	1-3	B
	4-6	D
2 faz modified yöntemi	3-1 ve 6-4	G
	1-3 ve 6-4	A
	1-3 ve 4-6	C
	3-1 ve 4-6	E



Şekil 5.8. Step motor uyartım yöntemleri (İki, Üç ve Dört fazlı uyartım).

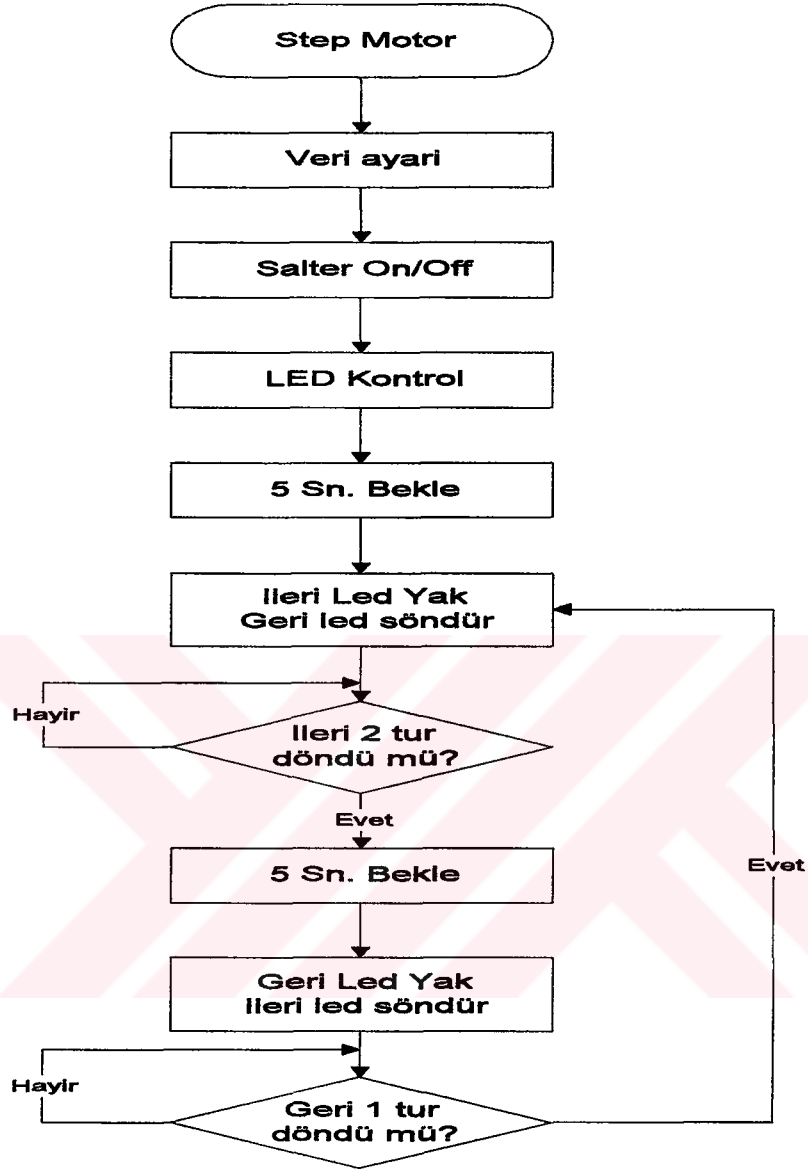
Şekil 5.8.b'de üç faz uyartım yöntemi, şekil c'de ise dört faz uyartım yöntemi şekilleri görülmektedir. Üç fazlı yöntemde bobinler ayrı ayrı enerjilenirken, dört fazlı uyartım yönteminde bobinlerin orta uçları kullanılmaktadır.

5.2.3. Step motor modülü

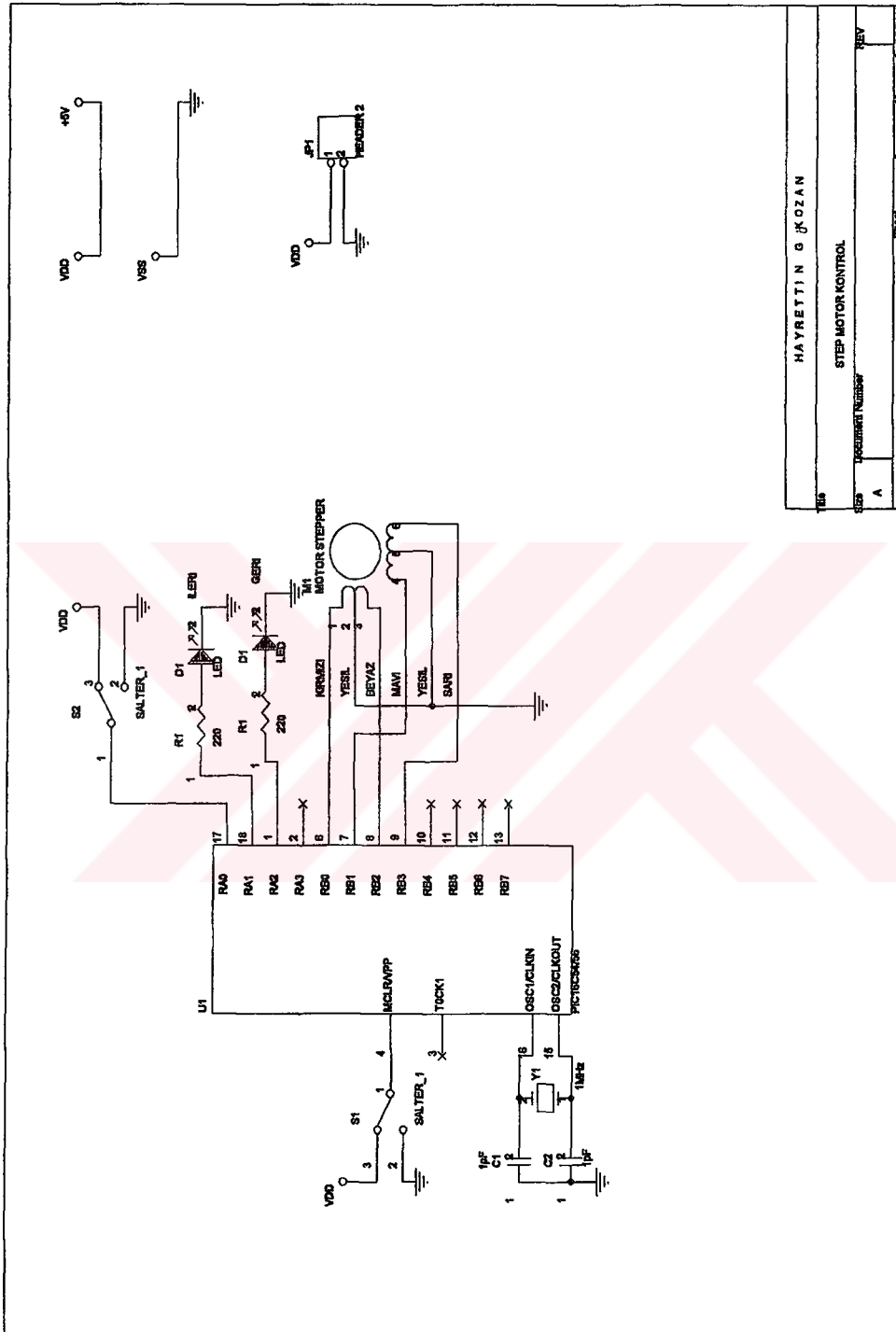
Nümerik kontrollü (NC) cihazların bir kısmında ve bilgisayar nümerik kontrollü (CNC) cihazların tamamında servo ve step motorlar kullanılmaktadır. Step motorlarda, girişlere verilen sinyale göre dönme elde edileceği için tezgahların daha rahat ve kolay çalışması sağlanmaktadır. Bu modülde step motorun sürülme sisteminin anlaşılması ve eğitim için kullanımı amaçlanmıştır.

5.2.4. Devrenin çalışması

Devrede step motorun çalıştırılıp durdurulması için bir on/off anahtarı ile motorun ileri ve geri dönüşlerini belirlemek için iki adet led kullanılmıştır. Ayrıca devrenin çalışma frekansını sağlayacak kristal bulunmaktadır. Motor mili iki tam devir ileri yönde döndükten sonra bir tam devir de geri dönmektedir.



Şekil 5.9. Step motor kontrolü programı iş akış şeması.



Şekil 5.10. Step motor kontrol devre şeması.

HAYRETTİN ÖZÖZAN		
STEP MOTOR KONTROL		
ÖLÇÜ	ÖLÇÜMÜ	REVİZYON
A		1

5.3. Trafik Işık Kontrolü Devresi Tasarımı ve Yapımı

Sinyal olarak adlandırılan ışıklı işaretler; yollarda ve özellikle kavşaklarda düzenli ve güvenli bir trafik akışı sağlamak için kullanılan kontrol gereçleridir. Gerçek hayatta sürekli karşılaştığımız bu gereçler, yollardaki gecikme ve tıkanıklıkları önlemenin yanında güvenliği sağlar, kaza olasılığını azaltır ve yoğun trafiğin bulunduğu yollarda geçiş üstünlüğü vererek tıkanıklıkları önlerler.

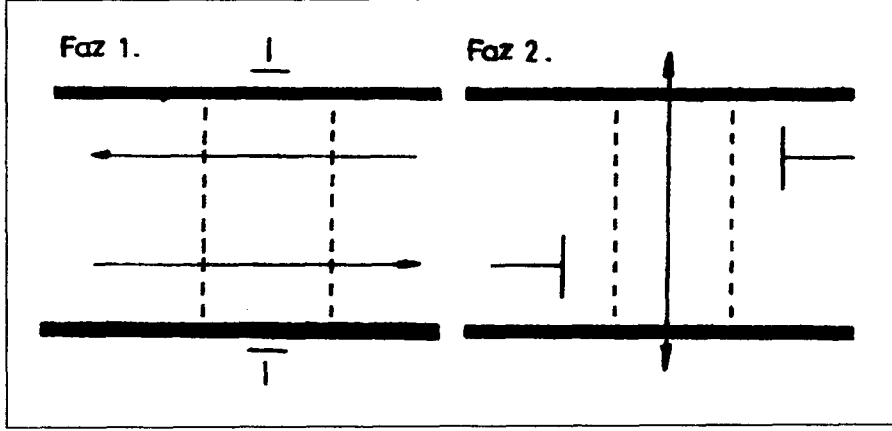
Bahsedilen bu işlemleri yapan sistemin; sürücü ve yayaların dikkatini çekmesi, basit ve kesin anlamları olması, sürücülere intikal ve reaksiyon için yeterli zamanı tanınması gibi niteliklere sahip olması gerekir. Tasarlanan bu modül ile belirli özelliklere sahip kavşakların trafik akışı sağlanarak, ışık sisteminin mikrodenetleyici ile kontrolü açıklanıp, belirli bir kavşak uygulaması gösterilecektir. Önce sinyalizasyon ile ilgili temel kavramları ve kavşak düzenlerini inceleyelim.

5.3.1. Sinyalizasyon ve kavşaklar

Sinyalize bir sistemle birbirini izleyen değişik renkteki ışıklı sinyallerin bir devrine 'Sinyal Devresi', ışıklı sinyallerin bu devreyi tamamlaması sırasında geçen toplam zamana da 'Devre Süresi' veya 'Peryod' denir. Devre süresi; taşıt akımları için ayrılan yeşil ışık süresi ve yeşil ışıklar arasında geçen sürelerin toplamı (kayıp zaman) olmak üzere iki kısımdan meydana gelir.

Sistemin bulunduğu kavşağın özelliklerine göre değişiklik gösteren bu kısımlar 8 sn. ile 120 sn. arasında bir değere sahip olabilirler. Bir sinyal devresi içerisinde bir veya birden fazla trafik akımını aynı anda öngören kumanda şekline 'Faz yöntemi' adı verilir. Sinyalizasyon sisteminde seçilecek faz yöntemi, kavşağa girişi olan yol sayısına ve kesişen trafik yoğunluğuna bağlıdır. Bu değerlere bağlı olarak 2, 3, 4

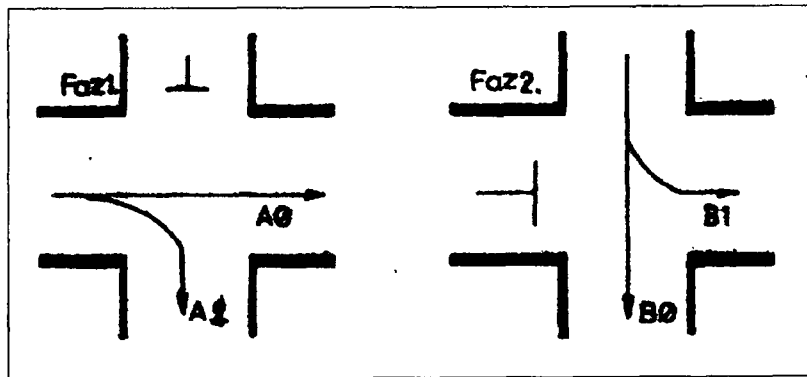
fazlı sistemler kullanılır. Faz sayısının az olması kayıp zamanı azaltacağından mümkün olduğu kadar az faz kullanılmalıdır.



Şekil 5.11. Yaya geçidi fazları.

5.3.2. Sinyalize yaya geçitleri ve tek yönlü yolların kesiştiği kavşaklar

Kavşak olmayan yerlerdeki yaya geçitlerinin sinyalize edilmesinde taşıt ve yayalar olmak üzere 2 faz kullanılır (Şekil 5.12). Tek yönlü yolların birleştiği kavşaklarda ise sinyalizasyon sistemi kavşaklardaki trafik akış yoğunluğunun yönüne göre düzenlenir.

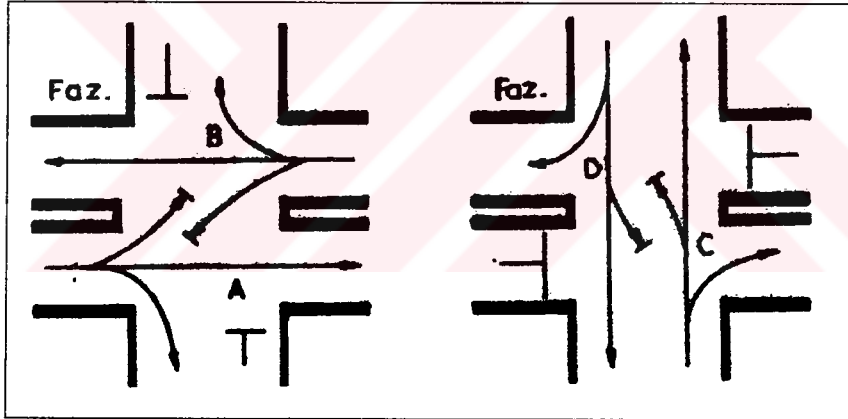


Şekil 5.12. Tek yönlü yaya kavşaklarındaki fazlar.

Şekil 5.12'deki gibi sola ve sağa dönüş yapan taşıtların bulunduğu bir kavşakta B_1 sola dönüşü ile A_1 sağa dönüşünü yapan taşıtlar, aynı anda geçiş hakkı olan yayalara yol vereceklerdir.

5.3.3. Sola dönüşlerin az olduğu kavşaklar

Çift yönlü taşıt akımı bulunan yollarda sola dönüş yapan taşıt hacmi 60 taşıt/saat veya daha az ise, sola dönüşler için özel donanım yapmak gerekmez. Sola dönüş yapan araçlar fazların değişimi sırasında yeşiller arasındaki süreden faydalanarak manevralarını yaparlar. Ayrıca kavşak alanının geniş ve yeterli olması halinde sola dönüş yapacak taşıtların kavşak içinde depo edilerek, yeşiller arasındaki sürenin uzatılması suretiyle ek bir faz etkisinin uygulanması sağlanabilir.

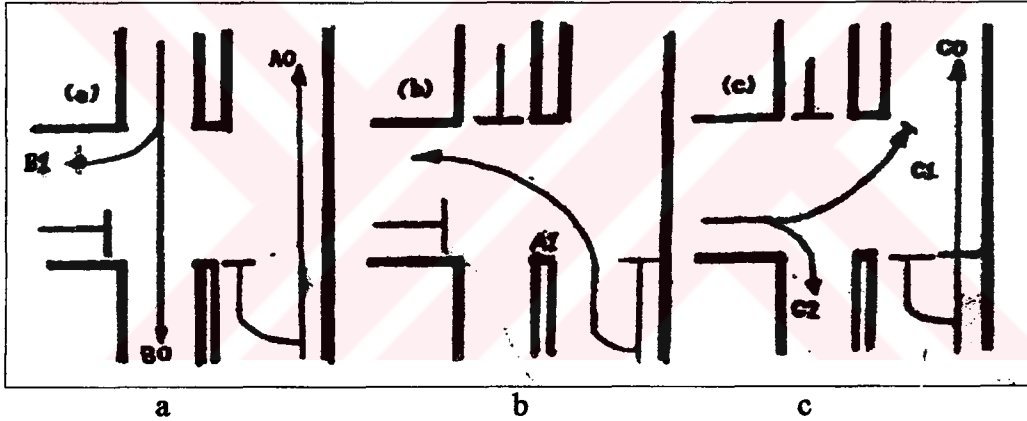


Şekil 5.13. Sola dönüşün az olduğu kavşaklardaki fazlar.

Şekil 5.13'de çift yönlü yolların kesiştiği dörtlü bir kavşakta sola dönüşlerin az olmasından yararlanılarak uygulanan iki fazlı bir düzen görülmektedir. Bu şekildeki faz düzenine göre sağa dönüşler düz gidişler ile birlikte kesilmektedir. Bu durum kavşağın geometrik yapısına göre değişebilir.

5.3.4. Tek yönlü -T- kavşaklar

Düz gidiş yapan taşıt yoğunluğunun yanı sıra sağa dönüş yapan taşıt sayısının artması halinde tek fazlı T kavşak düzeni uygulanması gerekir. Bu yöntemin seçiminde önemli faktör kavşağın geometrik yapısı ve akış yönüne göre taşıt yoğunluğudur. Tek fazlı bu sistemde Şekil 5.14'deki sistem uygulanarak düz gidenler ile birlikte sağa dönüşü izin verilebilir. A_2 yönünde akan taşıtlara ise yolun genişliğine bağlı olarak A_0 ile aynı anda yol verilebileceği gibi A_0 yönüne akan trafiğe dur denilerek A_1 ve A_2 'ye izin verilip daha sonraki fazda B_0 olarak izin verilebilir. Şekil a'daki sistemde tek fazlı yöntem kullanılırken, Şekil 5.14.b ve Şekil 5.14.c'deki sistemde iki fazlı yöntem kullanılır.

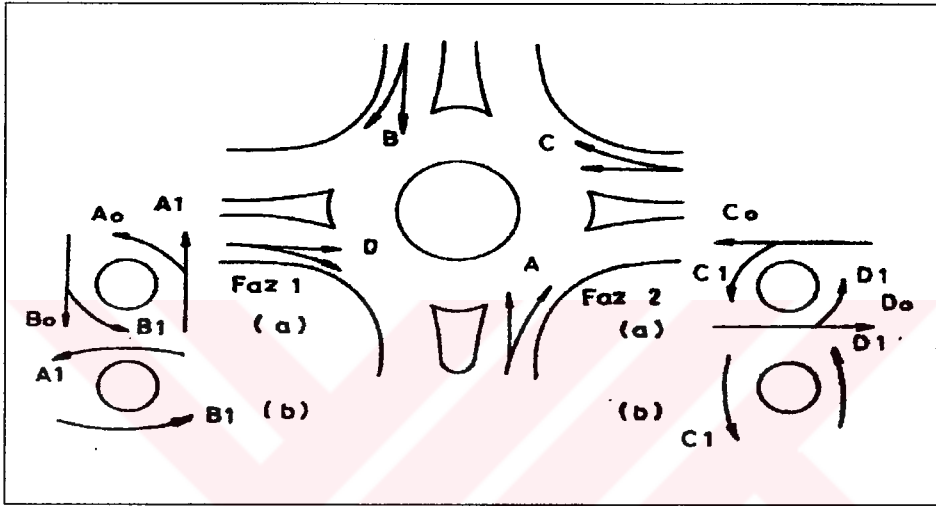


Şekil 5.14. Tek yönlü T kavşaklarda faz düzeni.

Tek fazlı bu sistemde şekil 5.14.a'daki sistem uygulanarak düz gidenler ile birlikte sağa dönüşü izin verilebilir. A_2 yönünde akan taşıtlara ise yolun genişliğine bağlı olarak A_0 ile aynı anda yol verilebileceği gibi A_0 yönüne akan trafiğe dur denilerek A_1 ve A_2 'ye izin verilip (Şekil 5.14.b), daha sonraki fazda B_0 olarak (Şekil 5.14.c) izin verilebilir. Şekil 5.14.a'daki sistemde tek fazlı yöntem kullanılırken, Şekil 5.14.b ve Şekil 5.14.c'deki sistemde iki fazlı yöntem kullanılır.

5.3.5. Ada çevresinde dönüş

Şekil 5.15'de iki fazlı düzenle yönetilen sinyalizasyon edilmiş bir döner kavşak örneği görülmektedir. Her iki fazın (a) hareketinde düz gidiş ile birlikte kavşağa giren sola dönecek taşıtlar ortadaki adanın çevresinde depolanmakta, (b) hareketinde ise kavşağı boşaltmaktadırlar.



Şekil 5.15. Ada çevresinde dönüşe sahip kavşak.

Sola dönüş yapacak taşıtlar kavşak içine girmiş olduklarından fazların (b) hareketleri için verilen süre çok kısa olabilmekte, bir sonraki fazla kavşağa girecek taşıtlar kavşağı boşaltmakta olan taşıtlarla aynı yönde seyrettiklerinden zaman kaybı önlenmediği gibi, trafik güvenliği de artırılmaktadır. Yayalara yeşil yanmadığı durumlarda, taşıtların sağa dönüşü serbest bırakılmıştır.

5.3.6. Tasarlanan trafik ışık kontrolü devresi ve yol düzeni grafiği

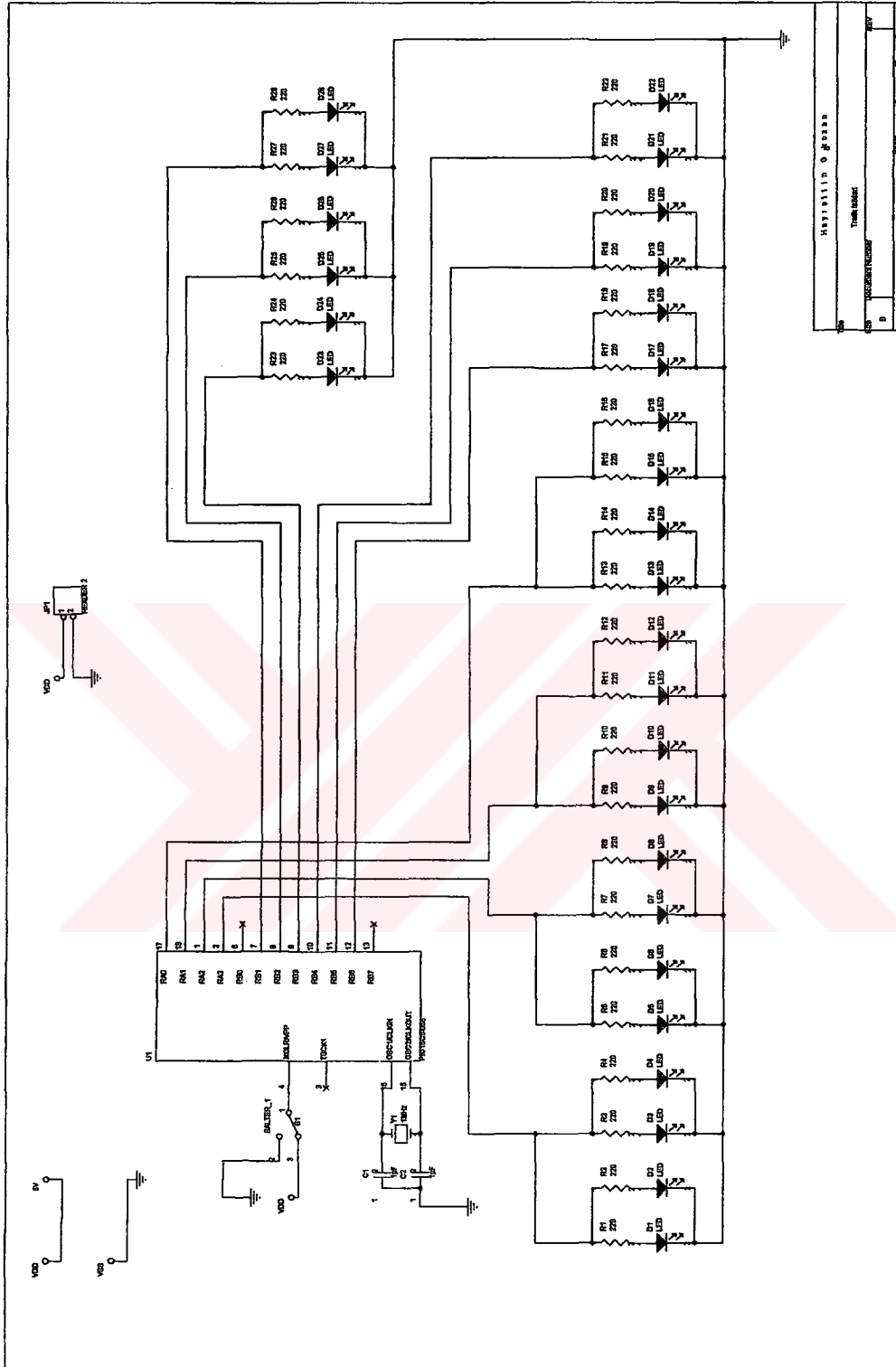
Tasarlanan sistemdeki yol düzeni bir döner kavşağı içermektedir (Şekil 5.15). Bu döner kavşaktaki trafik akışı ve yaya geçitleri kontrol edilmektedir. Kavşağın trafik akışı daha önceki kısımda açıklandığı şekildedir.

Bu çalışmada bir döner kavşağın bağımsız trafik akışı sağlanmıştır. Bu işlem PIC16C54 mikrokontrolörü ile gerçekleştirilmiştir. Mikrokontrolörden alınan enerji ledlere verilerek ışıklar kontrol edilmiştir. Bu işlem Şekil 5.16'da görülmektedir. Bu devrede PIC16C54 mikrokontrolörü, yapılan program dahilinde çıkış portlarına bağlı olan ledlere gerekli enerjiyi vermektedir. Kavşaktaki ışıkların yanma düzeni ve süresi yazılan programla istenilen şekilde değiştirilebilir.

Tasarlanan devrede ada çevresinde dönüşe sahip kavşakta A ve B yönünden gelen taşıtlar aynı anda geçiş hakkına sahiptirler. Bu taşıtların normal trafik akışı devam ederken C ve D yönünden gelen taşıtlara kırmızı ışık yandığından bu yollardan geçmek isteyen yayalara yeşil ışık yanmaktadır. 1. Fazda A ve B yollarındaki taşıtlara yeşil ışık yanarken, bu yolları geçmek isteyen yayalara kırmızı ışık yanmaktadır. Bu anda ada çevresindeki a_1 ve b_1 bölgesindeki ışıkların rengi yeşil olduğundan bu bölgedeki taşıtlar buldukları bölgeleri boşaltmaktadırlar. Bu anda c_1 ve d_1 bölgelerindeki taşıtlar adanın ilgili kısımlarında beklemekteyiz.

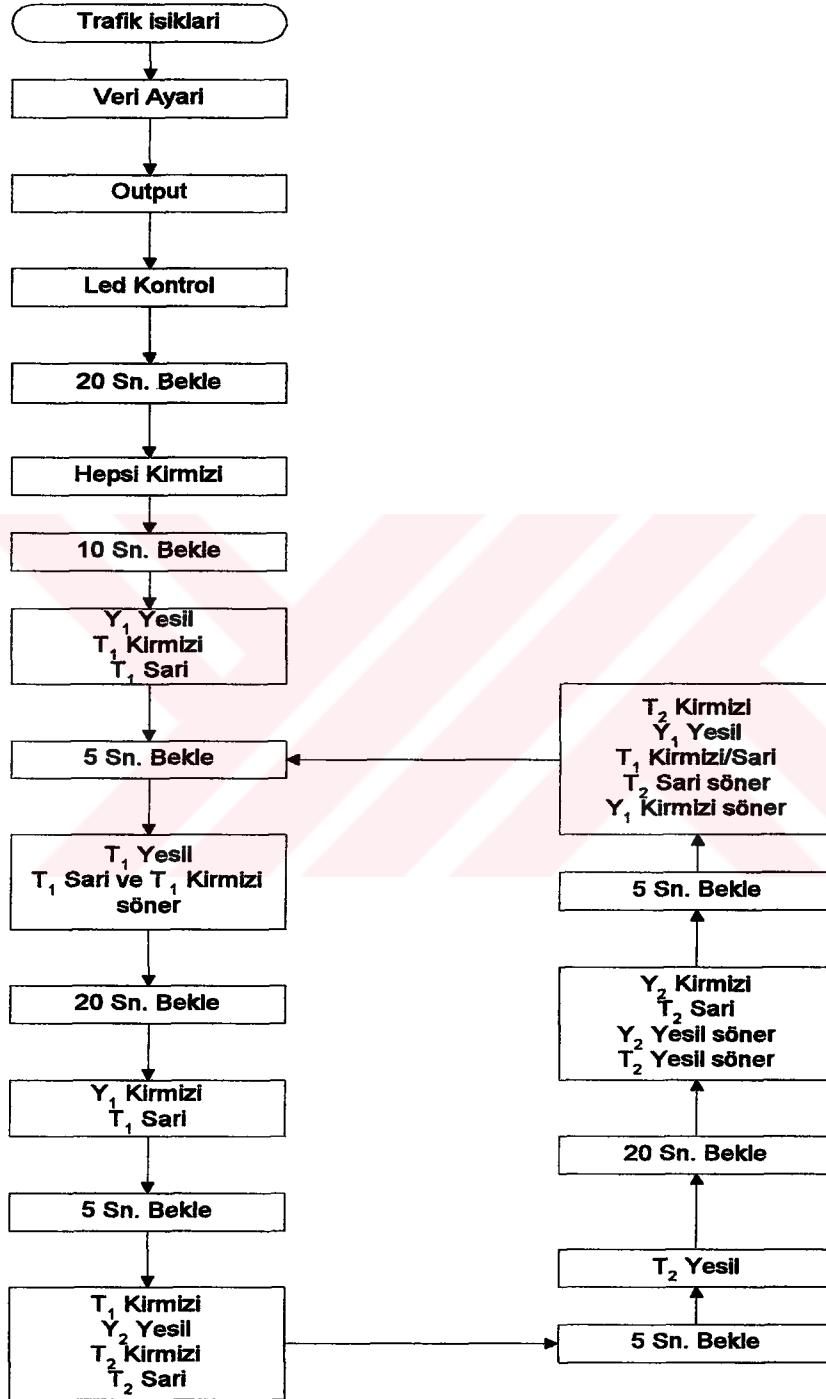
1. fazda lambaların yanma süreleri istenilen değere ayarlanabilir. Bu fazdan sonra yayalar ve taşıtların buldukları durumun değişeceğini belirtmek üzere 2. Faz olarak adlandırılan sarı ışıklar yanar. 2. faz, diğer fazlara göre daha az süreli olup bu fazda diğer kavşakta bulunan bütün direklerde sarı ışıklar yanar.

3. fazda ise her bir direkte ilk faz sırasında yanan sinyallerin tersi yanmaktadır. Bu durumda C ve D yönünden gelen taşıtlarla birlikte c_1 ve d_1 bölgesinde bekleyen taşıtlar da geçme hakkına sahiptir. Yine bu fazda A ve B yönlerinden gelen taşıtlara kırmızı ışık yandığından bu yollarda seyreden taşıtlar beklemekte, bu kavşaktaki yayalar ise karşıdan karşıya geçmektedir.



Şekil 5.16. Trafik ışıkları kontrol devre şeması.

Şekil 5.16'daki Trafik ışıklarının kontrolünü yapacak programın akış şeması Şekil 5.17'de görülmektedir. Buraya kadar yapılan işlemlerde bir kavşaktaki trafik akışı ile ilgili kontrol gerçekleştirilmiştir.



Şekil 5.17. Trafik ışık kontrolü iş akış şeması.

5.4. Tuş Takımı Tasarımı ve Yapımı

Bu çalışma analog bir ifadenin digital şekle dönüştürülmesi yani basılan bir butonun karşılığı olan sayı değerinin 7 segmentli göstergede ifade edilmesini amaçlamaktadır.

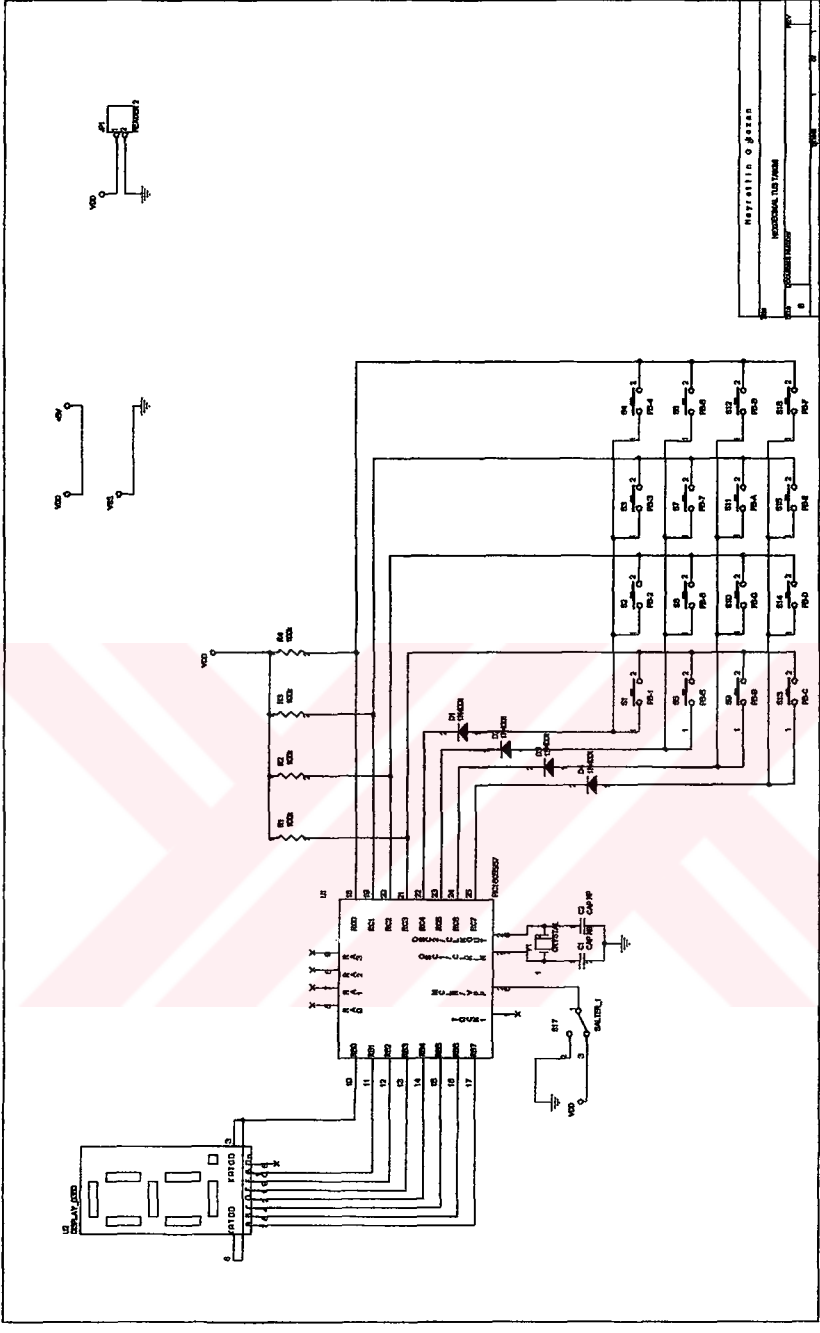
4 sütun ve 4 satırdan ibaret olan bir tuş konfigürasyonunda 0 ile 9 arası sayılar ile A, B, C, D, E ve F harflerinin görüntülenmesi gerçekleştirilmektedir.

5.4.1. Devre elemanları ve devrenin çalışması

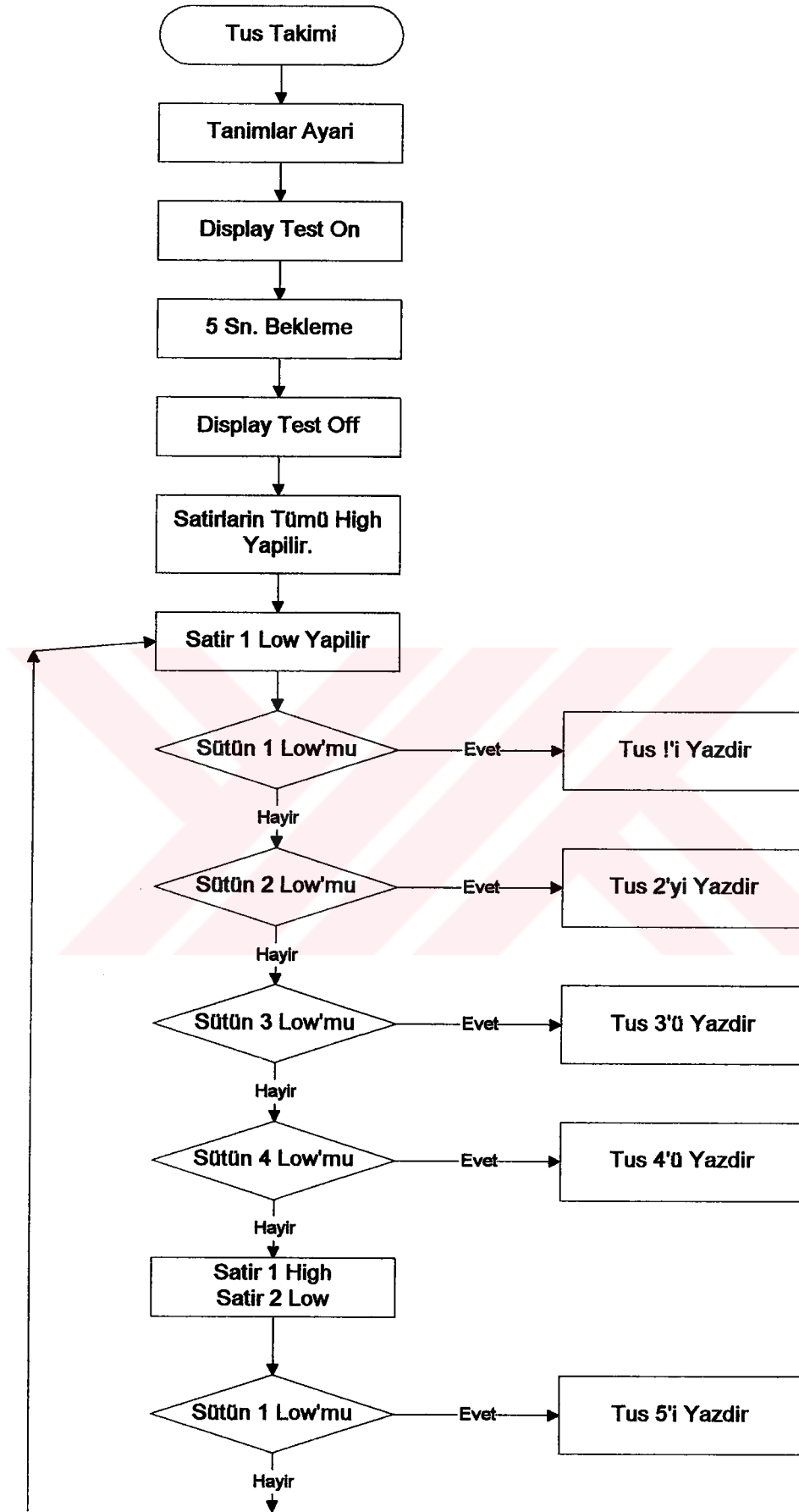
Yapılan devrede tuş fonksiyonlarını yerine getirmek üzere dizayn edilmiş 16 adet buton Şekil 5.18'deki gibi yerleştirilmiştir. Ayrıca 7 segmentli bir display kullanılmıştır. Kullanılan display ortak katotludur.

Enerji anahtarı kapatıldığında PIC16C55 mikrodenetleyici, tuş takımındaki satırları sıra ile önce low yapmaktadır. Basılan bir tuş olduğunda hangi satır ve hangi sütundan sinyal geldiğini karşılaştırarak göstergede bunu görüntülemektedir. Herhangi bir kısa devre durumuna karşı satırlara akım sınırlayıcı dirençler bağlanmıştır.

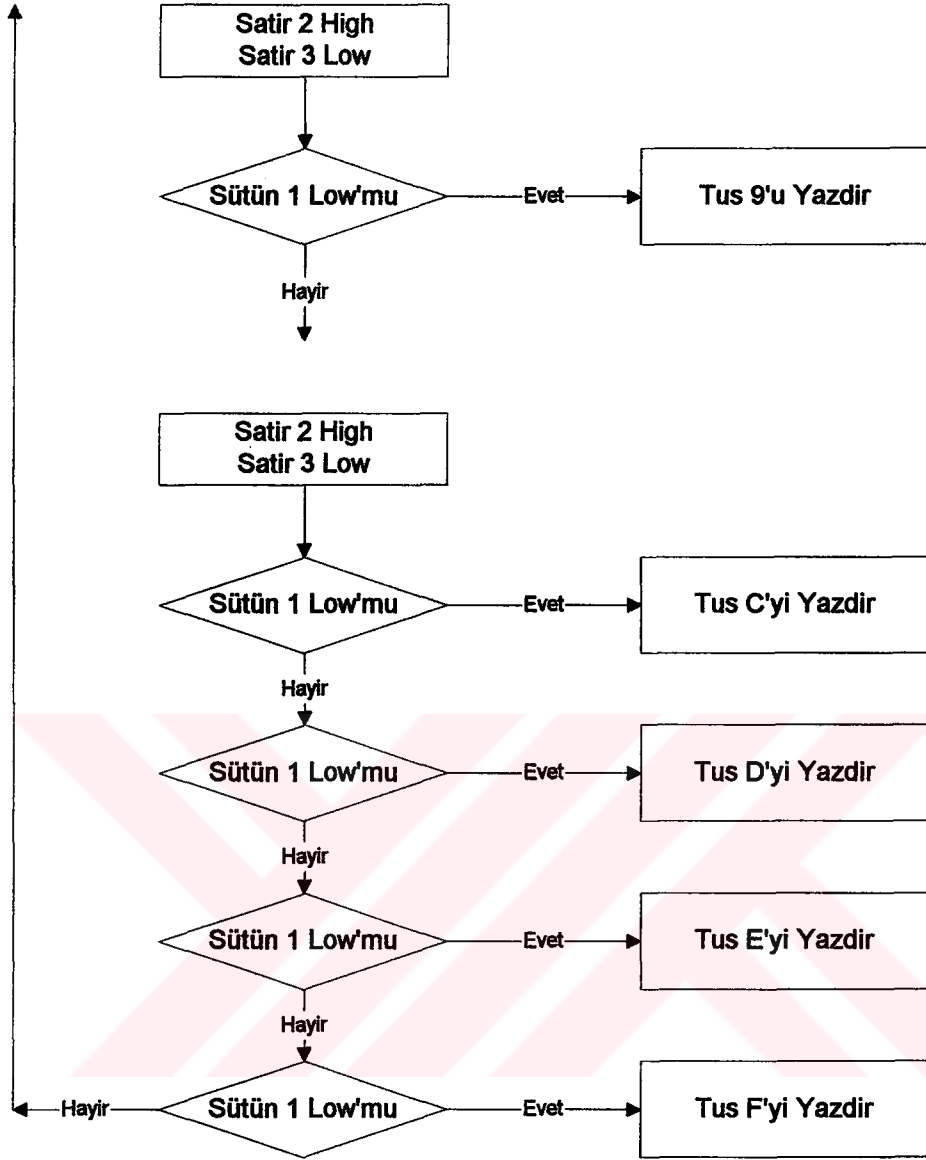
Kullanılan PIC16C55 mikrodenetleyicide R_{b0} - R_{b7} uçları 7 segment display için Çıkış (Output) ucu olarak, R_{c0} - R_{c7} uçları da Tuş takımı için giriş (Input) olarak kullanılmaktadır. Çalışma frekansını sağlamak için devreye 1 MHz. kristal bağlanmıştır.



Şekil 5.18. Tuş takımı devre şeması.



Şekil 5.19. Tuş takımı program akış diyagramı.



Şekil 5.20. Tuş takımı program akış diyagramı(Devam).

BÖLÜM 6. SONUÇLAR ve ÖNERİLER

Önceki bölümlerde ortaya konulduğu gibi PIC mikrodeneleyiciler çok geniş bir kullanım alanına sahiptir. PIC mikrodeneleyiciler, tüm mühendislik dallarında kolaylıkla uygulanabildiklerinden, dünya genelinde en çok tercih edilen 8 bit mikrodeneleyicidir. Bunun sebepleri; kullanım kolaylığı, simülasyon özelliği, ebat küçüklüğü, çok çeşitliliği ve her şeyden önemlisi yok denecek kadar çevre elemanlara olan gereksinimi ve sayısı azaltılmış, çok kullanışlı komutlarıdır. Bu özelliklerden dolayı öğrenimi, diğer mikroşlemcilerle göre daha kolay ve uygulanabilirliği az bir elektronik devre bilgisi ile mümkündür. Ayrıca ucuz oluşu, PIC'leri en çok tercih edilen mikrodeneleyici durumuna getiren sebeplerden birisidir.

Bu çalışmada tasarlanan devreler ve sunulan uygulamalar, günlük hayatta sıkça karşılaşılan cihazların bire bir uygulamasıdır. Uygulaması yapılan cihazlar ile pratikteki cihazların teknik ve kullanım bakımından herhangi bir farkı yoktur. PIC ile gerçekleştirilen uygulamaların maliyeti, diğer elemanlarla gerçekleştirilen cihazların fiyatına göre daha düşüktür. Bununla beraber, PIC'lerin varlığı ve daha önemlisi avantajları ülkemizde tam olarak anlaşılmamıştır. Bu nedenle Türkiye'de gerçekleştirilen mikroşlemcili elektronik devrelerin PIC'lerle daha hızlı ve daha ucuz gerçekleştirilme olanağı varken, yurtdışı ve yurtiçi kaynaklar araştırılmadığından bunlardan çok az haberdar olunabilmektedir.

Yapılan D.C. motor kontrol devresine ek olarak motorun kaç devirle döndüğünü gösteren bir gösterge konulabilir. Trafik ışıkları devresi başka bir kavşağa uygulanabilir. Tuş takımı devresi aracılığı ile, D.C. motorun devir sayısı girilebilir.

KAYNAKLAR

- [1] Thieser, M.,ve Finker, A., "PIC Controller", Franzis Verlag GmbH, 1998
- [2] König A., ve König M., "Mit dem PIC Controller erfolgreich arbeiten", Markt & Technik Verlag, 1996
- [3] Future's Technologies, Sayı 53, Sayfa 28-29, Eylül 1998,.
- [4] Microchip PIC16C5X User Maunal.
- [5] Urban C., "Mikrocontroller Mit RISC Struktur", Elektor Verlag, 1995
- [6] Microchip Development Systems Ordering Guide 1998
- [7] Microchip MPASM, Assembler User's Guide
- [8] Microchip MPLAB, IDE, Simulator, Editor User's Guide
- [9] Elektor, "PIC Anwendungssammlung", Elektro Verlag, 1996

EKLER

Ek 1. D.C. Motor Programı

```
TITLE "DC-MOTOR"
INCLUDE "P16C5X.INC"
LIST P=16C54
;*****
;SABİTLER
;*****
DESAY EQU 07
TMRAD EQU 08
;*****
;DEĞİŞKENLER
;*****
#DEFINE LED PORTB,7
#DEFINE SALTER PORTB,6
#DEFINE MOTON PORTB,5
;*****
;PORTLAR
;*****
;PORTA TRİS DEĞER
;RA0 1 X
;RA1 1 X
;RA2 1 X
;RA3 1 X
TR_A EQU B'00001111'
DEĞ_A EQU B'11111111'
;PORTB TRİS DEĞER
;RB0 X X
;RB1 X X
;RB2 X X
;RB3 X X
;RB4 X X
;RB5 0 0 ;DC MOT. ROLE ÇALIŞ. GERİLİMİ, İLK LOW (ÇALIŞMIYOR)
;RB6 1 X ;DC MOTOR ON/OFF DÜĞMESİ
;RB7 0 0 ;DC MOT. HAZIR GÖSTERGE LED, İLK LOW (LED SÖNÜK)
TR_B EQU B'01011111'
DEĞ_B EQU B'00000000'
;*****
;RESET/BAŞLANCIÇ DURUMU
;*****
ORG 1FF
```

```

        GOTO  ANA
        ORG   00
;*****
;ALT PROGRAMLAR
;*****
;*****
;ANA PROGRAMLAR
;*****
ANA    MOVLW B'00100000'
        OPTION
        MOVLW TR_A
        TRIS  PORTA
        MOVLW DEĞ_B           ;LED SÖNÜK, MOTOR ÇALIŞMIYOR
        MOVWF PORTB
        MOVLW TR_B
        TRIS  PORTB
A0     CLRF  TMRAD
        CLRF  DESAY
        BSF   LED             ;LED YANIYOR
A1     BTFSK  SALTER
        GOTO  MOTOR          ;MOTORU ÇALIŞTIR
        MOVF  PORTA,W        ;PORTA'DAKİ DURUMU W YÜKLE
        MOVWF DESAY         ;W'DEKİ PORTA DURUMU DEVİRSAYISI OLARAK AL
        GOTO  A1
MOTOR  BCF   LED             ;LEDİ SÖNDÜR
        BSF   MOTON          ;MOTOR ÇALIŞTIR
        CLRF  TMR0
A2     MOVLW D'240'
        SUBWF TMR0,W        ;TMR0-W SONUÇU POZİTİF İSE STATUS DAKİ C=1
        BTFSK STATUS,C     ;OLUR VE TMRAD BİR ARTIR
        GOTO  A2
        INCF  TMRAD,F
        CLRF  TMR0

        MOVF  DESAY,W
        SUBWF TMRAD,W        ;TMRAD-W SONUÇU POZİTİF İSE STATUS DAKİ C=1 OLUR
        BTFSK STATUS,C     ; VE İSTENEN DEVİRE ULAŞILIR VE MOTOR DURDURULUR
        GOTO  A2
        BCF   MOTON          ;MOTORU DURDUR
        GOTO  A0
END

```

Ek 2. Step Motor Programı

```

        TITLE "STEP MOTOR"
        INCLUDE "P16C5X.INC"
        LIST P=16C54
;*****
;SABİTLER
;*****
DÖN EQU 09
İLERİ EQU 07
GERİ EQU 08
YEDEK1 EQU 0A
YEDEK2 EQU 0B
YEDEK3 EQU 0C

;*****
;DEĞİŞKENLER
;*****

#DEFINE LEDİ PORTA,1
#DEFINE LEDG PORTA,2
#DEFINE SALTER PORTA,0

;*****
;PORTLAR
;*****

;PORTA TRİS DEĞER
;RA0 1 X ;$ALTER
;RA1 0 1 ;LED YANIYOR İLK AŞAMADA
;RA2 0 1 ;LED YANIYOR İLK AŞAMADA
;RA3 1 X
TR_A EQU B'00001001'
DEĞ_A EQU B'11111111'

;PORTB TRİS DEĞER
;RB0 0 0 ;KIRMIZI
;RB1 0 0 ;MAVİ
;RB2 0 0 ;BEYAZ
;RB3 0 0 ;SARI
;RB4 1 X
;RB5 1 X
;RB6 1 X
;RB7 1 X

TR_B EQU B'11110000'
DEĞ_B EQU B'00000000'

```

```

;*****
;RESET/BAŞLANCIÇ DURUMU
;*****

```

```

ORG 1FF
GOTO ANA
ORG 00

```

```

;*****
;ALT PROGRAMLAR
;*****

```

```

BEKLEME MOVLWD'4' ;W=4
MOVWF YEDEK3 ;YEDEK=4

BEK21 MOVLW D'255' ;W=255
MOVWF YEDEK2 ;YEDEK=255

BEK11 MOVLW D'255' ;W=255
MOVWF YEDEK1 ;YEDEK=255

BEK12 DECFSZ YEDEK1,F
GOTO BEK12
DECFSZ YEDEK2,F
GOTO BEK11
DECFSZ YEDEK3,F
GOTO BEK21
RETLW 0

```

```

;*****
;ANA PROGRAMLAR
;*****

```

```

ANA MOVLW B'00100000'
OPTION
MOVLW DEĞ_A ;LED YANIYOR
MOVWF PORTA
MOVLW TR_A
TRIS PORTA
MOVLW DEĞ_B ;MOTOR ÇALIŞMIYOR
MOVWF PORTB
MOVLW TR_B
TRIS PORTB
A1 BSF LEDG ;LED GERİ YAK
BSF LEDİ ;LED İLERİ YAK
A2 BTFSC SALTER
GOTO A2
;İLERİ DÖNDÜRME

İ2 BCF LEDG ;LED GERİ SÖNDÜR
BSF LEDİ ;LED İLERİ YAK
BCF STATUS,C ;KAYDIRMA STATUS'DAKİ C'DEN OLUYOR
MOVLW B'00010001'

```

```

MOVWF DÖN
MOVLW D'100'
MOVWF İLERİ
i1  BTFSK  SALTER
    GOTO  A1

MOVF  DÖN,W
MOVWF PORTB
CALL  BEKLEME
RLF  DÖN,F
BTFSK STATUS,C
GOTO  i3
MOVLW B'00010001'
MOVWF DÖN
BCF  STATUS,C
i3  DECFSZ İLERİ,F
    GOTO  i1

```

;GERİ DÖNDÜRME

```

BCF  LEDİ          ;LED İLERİ SÖNDÜR
BSF  LEDG          ;LED GERİ YAK
BSF  STATUS,C     ;KAYDIRMA STATUS'DAKİ C'DEN OLUYOR
MOVLW B'00001000'
MOVWF DÖN

MOVLW D'50'
MOVWF GERİ
G1  BTFSK  SALTER
    GOTO  A1
MOVF  DÖN,W
MOVWF PORTB

CALL  BEKLEME

RRF  DÖN,F
BTFSK STATUS,C
GOTO  G2
MOVLW B'00001000'
MOVWF DÖN
BSF  STATUS,C
G2  DECFSZ GERİ,F
    GOTO  G1
    GOTO  İ2
END

```


Ek 3. Trafik Işıkları Programı

```

        TITLE    "TRAFİK IŞIĞI"
;
        INCLUDE"P16C5X.INC"
        LIST     P=16C54
;
;*****
;SABİTLER
;*****
BEK_5  EQU     01           ;
BEK_10 EQU     02          ;
BEK_20 EQU     04          ;

YEDEK1 EQU     07           ;HERHANGİ BİR DEĞER YEDEKLEMELİK İÇİN
YEDEK2 EQU     08           ;HERHANGİ BİR DEĞER YEDEKLEMELİK İÇİN
YEDEK3 EQU     09           ;HERHANGİ BİR DEĞER YEDEKLEMELİK İÇİN
YEDEK4 EQU     0A          ;HERHANGİ BİR DEĞER YEDEKLEMELİK İÇİN

;*****
;DEĞİŞKENLER
;*****
#define Y_YA1  PORTA,2
#define K_YA1  PORTA,3

#define Y_YA2  PORTA,1
#define K_YA2  PORTA,0

#define Y_TA1  PORTB,1
#define S_TA1  PORTB,2
#define K_TA1  PORTB,3
#define Y_TA2  PORTB,6
#define S_TA2  PORTB,5
#define K_TA2  PORTB,4

;*****
;PORTLAR
;*****

;PORTA

TR_A  EQU     B'00000000'   ;RA0-3 OUTPUT
DEĞ_A EQU     B'00001111'   ;RA0-3 OUTPUT ÇIKIŞI HIGH (LED YANIYOR)

;PORTB

TR_B  EQU     B'10000001'   ;RB1-6 OUTPUT, DİĞERLERİ INPUT
DEĞ_B EQU     B'01111110'   ;RB1-6 OUTPUT ÇIKIŞI HIGH (LED YANIYOR), DİĞERLERİ LOW

```

```

;*****
;ALT DÖNGÜ BEKLEME
;*****
BEKLEME MOVLWD'19'           ;W=19
        MOVWF YEDEK3         ;YEDEK=19

BEK21  MOVLW D'255'          ;W=255
        MOVWF YEDEK2         ;YEDEK=255

BEK11  MOVLW D'255'          ;W=255
        MOVWF YEDEK1         ;YEDEK=255

BEK12  DECFSZ YEDEK1,F
        GOTO  BEK12
        DECFSZ YEDEK2,F
        GOTO  BEK11
        DECFSZ YEDEK3,F
        GOTO  BEK21
        RETLW 0

;*****
;ANA PROGRAM
;*****
ANA    MOVLW B'00111111'     ;T0CS=1, DİĞERLERİ ÖNEMSİZ
        OPTION
        MOVLW TR_A           ;W=TR_A DEĞERİ
        TRIS  PORTA

        MOVLW DEĞ_A          ;PORTA YANI YAYA 1 VE 2 NİN TÛM İŞIKLARINI YAK
        MOVWF PORTA

        MOVLW DEĞ_B          ;PORTB YANI TAŞIT 1 VE 2 NİN TÛM İŞIKLARINI YAK
        MOVWF PORTB
        MOVLW TR_B           ;W=TR_B DEĞERİ
        TRIS  PORTB
        MOVLW D'04'          ;W=4
        MOVWF YEDEK4
A1    CALL  BEKLEME          ;20S BEKLEME LEDLERİN TÛMÛ YANIYOR
        DECFSZ YEDEK4,F
        GOTO  A1

        BCF  Y_YA1           ;YAYA 1'İN YEŞİL İŞİĞİNİ SÖNDÜR
        BCF  Y_YA2           ;YAYA 2'İN YEŞİL İŞİĞİNİ SÖNDÜR
        BCF  Y_TA1           ;TAŞIT 1'İN YEŞİL İŞİĞİNİ SÖNDÜR
        BCF  S_TA1           ;TAŞIT 1'İN SARI İŞİĞİNİ SÖNDÜR

        BCF  Y_TA2           ;TAŞIT 2'İN YEŞİL İŞİĞİNİ SÖNDÜR
        BCF  S_TA2           ;TAŞIT 2'İN SARI İŞİĞİNİ SÖNDÜR

        MOVLW D'02'          ;W=2
        MOVWF YEDEK4

```

```

A2    CALL    BEKLEME                ;10S BEKLEME
      DECFSZ  YEDEK4,F
      GOTO    A2

      BCF     K_YA1                  ;YAYA 1'İN KIRMIZI IŞIĞINI SÖNDÜR
      BSF     Y_YA1                  ;YAYA 1'İN YEŞİL IŞIĞINI YAK

      BSF     S_TA1                  ;TAŞIT 1'İN SARI IŞIĞINI YAK

AK    MOVLW  D'01'                  ;W=1
      MOVWF  YEDEK4

A3    CALL    BEKLEME                ;5S BEKLEME
      DECFSZ  YEDEK4,F
      GOTO    A3

      BCF     K_TA1                  ;TAŞIT 1'İN KIRMIZI IŞIĞINI SÖNDÜR
      BCF     S_TA1                  ;TAŞIT 1'İN SARI IŞIĞINI SÖNDÜR
      BSF     Y_TA1                  ;TAŞIT 1'İN YEŞİL IŞIĞINI YAK

      MOVLW  D'04'                  ;W=4
      MOVWF  YEDEK4

A4    CALL    BEKLEME                ;20S BEKLEME
      DECFSZ  YEDEK4,F
      GOTO    A4

      BCF     Y_YA1                  ;YAYA 1'İN YEŞİL IŞIĞINI SÖNDÜR
      BSF     K_YA1                  ;YAYA 1'İN KIRMIZI IŞIĞINI YAK

      BSF     S_TA1                  ;TAŞIT 1'İN SARI IŞIĞINI YAK
      BCF     Y_TA1                  ;TAŞIT 1'İN YEŞİL IŞIĞINI SÖNDÜR
      MOVLW  D'01'                  ;W=1
      MOVWF  YEDEK4

A5    CALL    BEKLEME                ;5S BEKLEME
      DECFSZ  YEDEK4,F
      GOTO    A5

      BSF     K_TA1                  ;TAŞIT 1'İN KIRMIZI IŞIĞINI YAK
      BCF     S_TA1                  ;TAŞIT 1'İN SARI IŞIĞINI SÖNDÜR

      BSF     S_TA2                  ;TAŞIT 2'İN SARI IŞIĞINI YAK

      BSF     Y_YA2                  ;YAYA 2'İN YEŞİL IŞIĞINI YAK
      BCF     K_YA2                  ;YAYA 2'İN KIRMIZI IŞIĞINI SÖNDÜR
      MOVLW  D'01'                  ;W=1
      MOVWF  YEDEK4

A6    CALL    BEKLEME                ;5S BEKLEME
      DECFSZ  YEDEK4,F
      GOTO    A6

      BCF     K_TA2                  ;TAŞIT 2'İN KIRMIZI IŞIĞINI SÖNDÜR

```

```

BCF    S_TA2                ;TAŞIT 2'İN SARI IŞIĞINI SÖNDÜR
BSF    Y_TA2                ;TAŞIT 2'İN YEŞİL IŞIĞINI YAK
MOVLW D'04'                ;W=4
MOVWF  YEDEK4

A7     CALL  BEKLEME        ;10S BEKLEME
      DECFSZ YEDEK4,F
      GOTO  A7
      BCF   Y_YA2          ;YAYA 2'İN YEŞİL IŞIĞINI SÖNDÜR
      BSF   K_YA2          ;YAYA 2'İN KIRMIZI IŞIĞINI YAK

      BSF   S_TA2          ;TAŞIT 2'İN SARI IŞIĞINI YAK
      BCF   Y_TA2          ;TAŞIT 2'İN YEŞİL IŞIĞINI SÖNDÜR
      MOVLW D'01'         ;W=1
      MOVWF YEDEK4

A8     CALL  BEKLEME        ;5S BEKLEME
      DECFSZ YEDEK4,F
      GOTO  A8
      BSF   K_TA2          ;TAŞIT 2'İN KIRMIZI IŞIĞINI YAK
      BCF   S_TA2          ;TAŞIT 2'İN SARI IŞIĞINI SÖNDÜR

      BSF   S_TA1          ;TAŞIT 1'İN SARI IŞIĞINI YAK
      BSF   Y_YA1          ;YAYA 1'İN YEŞİL IŞIĞINI YAK
      BCF   K_YA1          ;YAYA 1'İN KIRMIZI IŞIĞINI SÖNDÜR
      GOTO  AK
      ORG  1FF
      GOTO  ANA
      END

```

Ek 4. Keypad Programı

```

TITLE "KEY-PAD"

INCLUDE "P16C5X.INC"

LIST P=16C55
;*****
;DEĞİŞKENLER
;*****

YEDEK1 EQU 0A ;HERHANGİ BİR DEĞER YEDEKLEMEK İÇİN
YEDEK2 EQU 08 ;HERHANGİ BİR DEĞER YEDEKLEMEK İÇİN
YEDEK3 EQU 09 ;HERHANGİ BİR DEĞER YEDEKLEMEK İÇİN

;*****
;TANIMLAR
;*****

#DEFINE SUT4 PORTC,0
#DEFINE SUT3 PORTC,1
#DEFINE SUT2 PORTC,2
#DEFINE SUT1 PORTC,3
#DEFINE SAT1 PORTC,4
#DEFINE SAT2 PORTC,5
#DEFINE SAT3 PORTC,6
#DEFINE SAT4 PORTC,7

;*****
;PORTLAR
;*****

;PORTC KEY-PAD TARAMA İÇİN
;
; TRİS DEĞER
;
;RC0 1 X ;X ÖNEMSİZ, BURADA X=1 ALINDI
;RC1 1 X
;RC2 1 X
;RC3 1 X
;RC4 0 1
;RC5 0 1
;RC6 0 1
;RC7 0 1
TR_C EQU B'00001111'

```

```

DEĞ_C EQU    B'11111111'

;PORTB      DISPLAY SÜRMEK İÇİN
;
;           TRİS          DEĞER
;
;RB0        0            1      ;İLK ÇALIŞMADA TÜM DİSPLAY YANACAK
;RB1        0            1
;RB2        0            1
;RB3        0            1
;RB4        0            1
;RB5        0            1
;RB6        0            1
;RB7        0            1
TR_B EQU    B'00000000'
DEĞ_B EQU    B'11111111'

```

```

;PORTA      KULLANILMIYOR
;
;           TRİS          DEĞER
;
;RB0        X            X      ;X ÖNEMSİZ, BURADA X=1 ALINDI
;RB1        X            X
;RB2        X            X
;RB3        X            X
TR_A EQU    B'11111111'
DEĞ_A EQU    B'11111111'

```

```

;*****
;RESET/BAŞLANGIC DURUMU
;*****

```

```

ORG    1FF          ;RESET ADRESİ
GOTO   ANA
ORG    00

```

```

;*****
;ALT PROGRAM
;*****

```

```

;TUŞU DISPLAYDE GÖSTERME

```

```

DISPLAY MOVWF PORTB
GOTO   A1

```

```

;DISPLAY İÇİN ABCDEFGK

```

```

TUŞ1  MOVLW B'01100000'
      GOTO  DISPLAY
TUŞ2  MOVLW B'11011010'
      GOTO  DISPLAY
TUŞ3  MOVLW B'11110010'
      GOTO  DISPLAY
TUŞ4  MOVLW B'01100110'
      GOTO  DISPLAY
TUŞ5  MOVLW B'10110110'
      GOTO  DISPLAY
TUŞ6  MOVLW B'10111110'
      GOTO  DISPLAY
TUŞ7  MOVLW B'11100000'
      GOTO  DISPLAY
TUŞ8  MOVLW B'11111110'
      GOTO  DISPLAY
TUŞ9  MOVLW B'11110110'
      GOTO  DISPLAY
TUŞ0  MOVLW B'11111100'
      GOTO  DISPLAY
TUŞA  MOVLW B'11101110'
      GOTO  DISPLAY
TUŞB  MOVLW B'00111110'
      GOTO  DISPLAY
TUŞC  MOVLW B'10011100'
      GOTO  DISPLAY
TUŞD  MOVLW B'01111010'
      GOTO  DISPLAY
TUŞE  MOVLW B'10011110'
      GOTO  DISPLAY
TUŞF  MOVLW B'10001110'
      GOTO  DISPLAY

```

```

;*****
;ANA PROGRAM
;*****

```

```

ANA  MOVLW B'0010000'      ;T0CS=1, DİĞERLERİ ÖNEMSİZ
      OPTION

      MOVLW TR_B
      TRIS  PORTB          ;INPUT/OUTPUT TANIMI
      MOVLW DEĞ_B         ;DISPLAY YANİYOR
      MOVWF PORTB         ;INPUT/OUTPUTTUN DEĞERLERİ

      MOVLW TR_C
      TRIS  PORTC          ;INPUT/OUTPUT TANIMI
      MOVLW DEĞ_C

```

```

MOVWF PORTC          ;INPUT/OUTPUTTUN DEĞERLERİ

MOVLW D'19'          ;W=19,          5S BEKLEME
MOVWF YEDEK3          ;YEDEK=19

BEK21  MOVLW D'255'   ;W=255
        MOVWF YEDEK2   ;YEDEK=255

BEK11  MOVLW D'255'   ;W=255
        MOVWF YEDEK1   ;YEDEK=255

BEK12  DECFSZ YEDEK1,F
        GOTO  BEK12

        DECFSZ YEDEK2,F
        GOTO  BEK11

        DECFSZ YEDEK3,F
        GOTO  BEK21

CLRF   PORTB          ;DISPLAYLER SÖNDÜRÜLÜR

MOVLW B'11111111'    ;RC4-7 HIGH YAPILDI, DİĞERLERİ ÖNEMSİZ ÇÜNKÜ INPUT
MOVWF PORTC          ;TUŞ TAKIMINA HIGH GÖNERİLİYOR (RC4-7)

```

;SATIR 1 TARAMA İLE TUŞ 1, 2, 3, 4 DÜN KONTROLU

```

A1     BCF   SAT1          ;SATIR 1 LOW YAPILDI DİĞER SATIRLAR HIGH

        BTFSS SUT1
        GOTO  TUŞ1
        BTFSS SUT2
        GOTO  TUŞ2
        BTFSS SUT3
        GOTO  TUŞ3
        BTFSS SUT4
        GOTO  TUŞ4

        BSF   SAT1          ;SATIR 1 HIGH YAPILDI DİĞER SATIRLAR HIGH

```

;SATIR 2 TARAMA İLE TUŞ 5, 6, 7, 8 İN KONTROLU

```

BCF   SAT2          ;SATIR 2 LOW YAPILDI DİĞER SATIRLAR HIGH

        BTFSS SUT1
        GOTO  TUŞ5
        BTFSS SUT2
        GOTO  TUŞ6
        BTFSS SUT3
        GOTO  TUŞ7

```


BTFFS SUT4
GOTO TUŞ8

BSF SAT2 ;SATIR 2 HIGH YAPILDI DİĞER SATIRLAR HIGH

;SATIR 3 TARAMA İLE TUŞ 9, 0, A, B NİN KONTROLU

BCF SAT3 ;SATIR 3 LOW YAPILDI DİĞER SATIRLAR HIGH

BTFFS SUT1
GOTO TUŞ9
BTFFS SUT2
GOTO TUŞ0
BTFFS SUT3
GOTO TUŞA
BTFFS SUT4
GOTO TUŞB

BSF SAT3 ;SATIR 3 HIGH YAPILDI DİRER SATIRLAR HIGH

;SATIR 4 TARAMA İLE TUŞ C, D, E, F NİN KONTROLU

BCF SAT4 ;SATIR 4 LOW YAPILDI DİĞER SATIRLAR HIGH

BTFFS SUT1
GOTO TUŞC
BTFFS SUT2
GOTO TUŞD
BTFFS SUT3
GOTO TUŞE
BTFFS SUT4
GOTO TUŞF
BSF SAT4 ;SATIR 4 HIGH YAPILDI DİĞER SATIRLAR HIGH
GOTO A1
END

ÖZGEÇMİŞ

Hayrettin GÖKOZAN, 1961 Yılında Denizli'de doğdu. İlk ve orta öğrenimini Denizli'de tamamladı. 1983 yılında Gazi Üniversitesi Teknik Eğitim Fakültesi Elektrik ve Elektronik Bölümü'nden mezun oldu. 1985-1993 yılları arasında Endüstri Meslek Liselerinde Atelye ve Meslek Dersleri öğretmenliği ile Bölüm Şefliği görevlerini yaptı. 1993 yılında Dumlupınar Üniversitesi Simav Teknik Eğitim Fakültesi'ne Öğretim Görevlisi olarak girdi. 1998 yılında Sakarya Üniversitesi Fen Bilimleri Enstitüsünde Yüksek Lisans öğrenimine başladı. Halen Dumlupınar Üniversitesi Simav Teknik Eğitim Fakültesi'nde Öğretim Görevlisi kadrosunda Elektronik ve Bilgisayar Eğitimi Bölüm Başkanı olarak görevini yürütmektedir. Hayrettin Gökozan evli ve iki çocuk babasıdır.