

CNC-TEZGAH EGITIMININ
BILGISAYAR PROGRAMLARIYLA YAPILMASI

Erol Seke /

Anadolu Universitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Elektronik Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman : Prof. Dr. Atalay Barkana

Temmuz-1988

Erol Seke' nin YUKSEK LISANS tezi olarak hazırladığı "CNC-Tezgah Eğitiminin Bilgisayar Programlarıyla Yapılması" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

.12.1.7.1988

Başkan : Prof.Dr. Atalay Barkana

Atalay Barkana

Üye : Yrd.Doç.Dr. İlker Gürkan

İlker Gürkan

Üye : Yrd.Doç.Dr. Nuri Yücel

Nuri Yücel

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 29 TEMMUZ 1988
gün ve ...182/3..... sayılı kararıyla onaylanmıştır.

Rüstem Kaya

Enstitü Müdürü

Prof. Dr. Rüstem KAYA

ÖNSÖZ

Bu yüksek lisans tezi CNC-tezgah eğitimine yardımcı kaynak olması amacıyla hazırlanmıştır. CNC-tezgahı ve G-komutlarını yeni öğrenen bir öğrencinin başvurabileceği, G-komutlarının ekranda deneyerek öğrenilmesini sağlayabilecek pratik bir kaynaktır. Bölüm-3'te G-komutları, bölüm-2.2'de de eğitici programın komutları açıklanmıştır. Öğrencinin öncelikle tezgahın koordinat sistemini öğrenmesi gerekmektedir. Bu amaçla bölüm-1'de CNC-tezgahında kullanılan koordinat sistemleri hakkında bilgi verilmiştir. İkinci olarak G-komutlarının incelenmesi daha sonra da eğitici programın komutlarının öğrenilerek uygulama yapılması tavsiye edilir. Bölüm-4'teki örnek programlar da G-komutlarının öğrenilmesinde faydalı olabilir.

Çalışmanın hazırlanmasında emeği geçenlere ve hiçbir konuda yardımlarını esirgemeyen tez danışmanım Prof.Dr. Atalay BARKANA'ya teşekkürü borç bilirim.

ÖZET

Çalışmada, makine dili ve BASIC kullanılarak eğitim amaçlı bir CNC-torna tezgahı ile bir kişisel bilgisayar arasında iletişim kurmak ve CNC yazılımında eğitim amacı güdülmüştür. Öncelikle CNC-torna tezgahının mikrobilgisayarı ile ilişkiyi sağlamak üzere bir devre ve makine dilinde bir program yapılmış daha sonra BASIC dilinin avantajlarını kullanarak tezgaha aktarılacak programların kişisel bilgisayarda hazırlanması ve denenmesi amacıyla bir program geliştirilmiştir. BASIC'in grafik olanakları ile tezgah programlarının ekran üzerinde denenmesi yoluna gidilmiştir. Ayrıca yine bu program sayesinde tezgah, kişisel bilgisayarın veri saklama olanaklarından yararlandırılmıştır.

Böylece öğrenme aşamasındaki bir kişi tezgah programlarını önce kişisel bilgisayarda deneyebilecek ve hatalı programların yol açabileceği hasarlardan korunmuş olacaktır.

PROGRAM ON CNC EDUCATION

In this work, it is aimed to have a communication between a CNC-lathe and a simple personal microcomputer using machine code and BASIC programming languages, the education on CNC-lathe programming can be realized on the personal microcomputer. First, a circuit and a program in machine code are prepared for communication between CNC-lathe and the personal microcomputer. Then by using advantages of BASIC-language, a program is developed in order to prepare CNC-programs on the personal microcomputer. Using graphic advantages of BASIC, the CNC-programs are executed on the screen. In addition, the data saving facilities are advantageous in the personal microcomputers compared with CNC's stand alone memory.

Students, learning CNC-programming can try their CNC-programs on the personal microcomputer first, so they can protect themselves from injury and the CNC-lathe from damages which may be caused by possible faulty programs.

İÇİNDEKİLER

	sayfa
ÖNSÖZ	iii
1. GİRİŞ	1
2. İLETİŞİM KARTI VE PROGRAMI	6
2.1. Devre Şeması	7
2.2. İletişim Programı	14
2.3. BASIC-Dilinde Eğitici Program	19
3. G-KOMUTLARI	25
4. CNC-DE PARÇA İŞLEME ÖRNEKLERİ	40
4.1 Konik Tapa	40
4.2 Dairesel Tornalama Örneği	43
4.3 King	46
5. SONUÇ VE TARTIŞMA	49
KAYNAKLAR DİZİNİ	51
KISALTMALAR	52

EKLER

1. BASIC Program Listesi.
2. Z80 komut listesi.
3. Devre Elemanlarının katalog Bilgileri.
4. Standard CNC G ve M-kodları.

SEKİLLER DİZİNİ

Sekil	Sayfa
1.1 Hareket yolu örneği	3
1.2 Artımlı sistemde koordinat sistemi merkezleri	4
2.1 Aradevrenin bağlantıları	6
2.2 PIO için adres seçici devre	7
2.3 "Bilgi alındı bağlantısı"	9
2.4 İlave edilen konektör bağlantıları	9
2.5 Tuş takımının matris düzenlemesi	10
2.6 Tuş tarama işaretleri	11
2.7 "9" tuşu bağlantısı	11
2.8 Eşzamanlama devresi	12
2.9 Aradevrenin Tüm şeması	13
2.10 Makine dilindeki programın akış şeması	16
3.1 Satır numarası ile beraber bir blok	26
3.2 Koordinat sistemi	26
3.3 Doğrusal hareket	28
3.4 G84 Çevrimleri	30
3.5 G84-kodunda 1 nolu olasılık	31
3.6 Dairesel hareket yörüngeleri	32
3.7 Dairesel hareket olasılıkları	33
3.8 Dış açma örneği	34
3.9 Dış açma kademeleri	34
3.10 G78 çevrim olasılıkları	35
3.11 Koordinat sistemi seçme	37
4.1 Konik tapa parçasının CNC-programı	41
4.2 Konik tapa parçası için yörüngeler	42
4.3 Dairesel tornalama örneği programı	44
4.4 Dairesel tornalama örneği için yörüngeler	45
4.5 King parçası CNC-programı	47
4.6 King parçası için yörüngeler	48

1. GİRİŞ

Bilgisayar ve bilgisayar kontrollü-destekli üretim teknolojisinin durmaksızın gelişmesi, endüstri ile yakından ilgili herkes için bilgisayarın yalnızca bilgi saklamak ya da işlemek için değil aynı zamanda bu bilgiler ile üretime doğrudan yön vermek için kullanabilmeyi bir zorunluluk haline getirmiştir. Artık bilgisayar kullanmayan endüstriyel üretim kuruluşu yok gibidir. İnsanın yalnızca "gözlemci" olarak bulunduğu oldukça büyük üretim-montaj hatları vardır.

Makinanın önceden belirlenen hareketleri bu program dahilinde yüzlerce kez hatasız tekrar etmesi artık alışılmış, bu makine, tezgah ya da robotların üretim hattında kullanılması özel bir hal olmaktan çıkmıştır. Elbette ki bu sistemlerin öğrenilmesi eğitim konusu problemi oluşturmaya başlamıştır.

CNC-tezgahlarına bir giriş niteliğinde olan eğitim amaçlı CNC-tezgah simulatörü her ne kadar endüstriyel tezgahlarla kıyaslanamayacak kadar ucuzsa da yine de oldukça pahalı hasarlara uğrayabilmektedir. Özellikle bir "yeni öğrenen" in kullanımında ise, tezgahı sık sık acil durdurma (emergency stop) düğmesine basarak durdurmak, yazılı programları ve ona harcanan emeği heba etmek yerine yazılmış programları kontrol eden bir kontrolör program gerçek bir ihtiyaçtır. Bu bakımdan en kolay kontrol şekli olan gözle görerek kontrol şeklinin basit bir kişisel bilgisayarda ekran olanaklarından faydalanarak program haline getirilmesi amaçlanmıştır. Ayrıca kişisel bilgisayarda tezgah programı için harcanan zamanın program tezgaha aktarılırken yeniden harcanmaması için bir iletişim aradevresi tasarlanması ve bu devreyi işler hale getiren bir iletim programının hazırlanması amacın içindedir.

Amacımızın gereği hazırlanan program kullanıcıya tezgah takine yakın bir programlama şekli sağlamalı, kullanıcı tezgah başına geçtiğinde kolaylıkla ona hakim olabilmelidir. Eğitici program tezgaha ait komutları kullanıcıya

tanıtıcı olmalı ayrıca kullanıcının hazırladığı programları saklama olanağı vermelidir. Bu amaçla CNC-tezgahtarının G-komutları 3. bölümde açıklanmıştır.

G-komutlarıyla CNC-programlamasının öğretilmesi amacıyla

1. Konik tapa
2. Dairesel tornalama örneği
3. King

iş parçalarının tornalama akış diyagramları ve programları verilmiştir.

Kişinin kendi hazırladığı programlar ile parçanın işleniş şekli ekrandan takip edilebilmekte böylece yanlış komutların parçanın işlenişinde ne hatalara sebep olabileceğini göstermek mümkün olmaktadır.

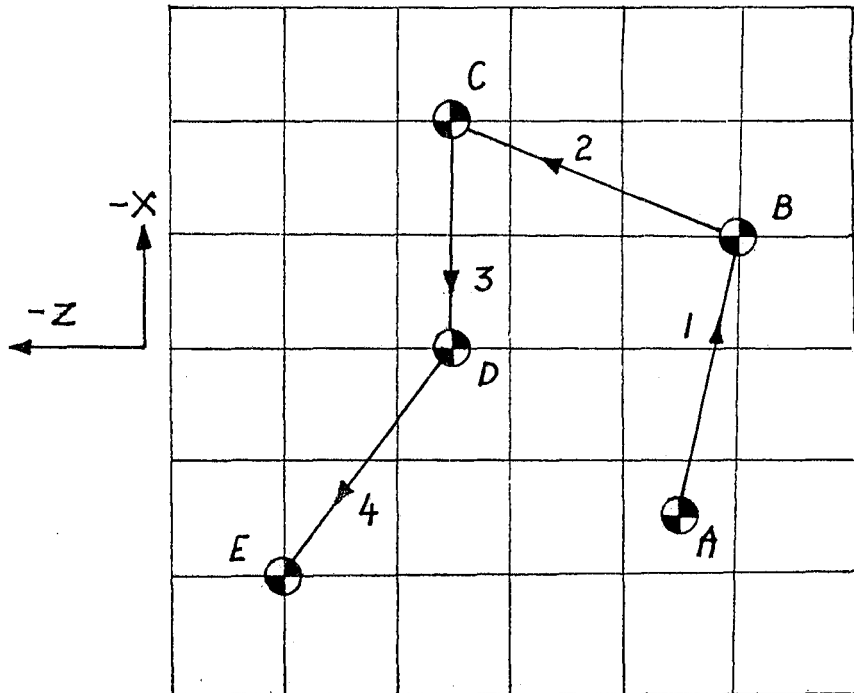
Artımlı Ve Kartezyen Koordinat Sistemleri : Bu kısımda CNC-torna tezgahında kullanılan iki koordinat sistemi hakkında kısaca bilgi verilmiştir.

CNC-torna tezgahı ve diğer endüstriyel CNC-iş makineleri alışılacağı iki koordinat sistemini kullanabilmektedir. Torna kalemi hareket düzlemindeki noktaların belirlenmesi, iki sistem arasında farklılıklar göstermektedir. Bu sistemler

1. Artımlı sistem
2. Kartezyen koordinat sistemi

dir.

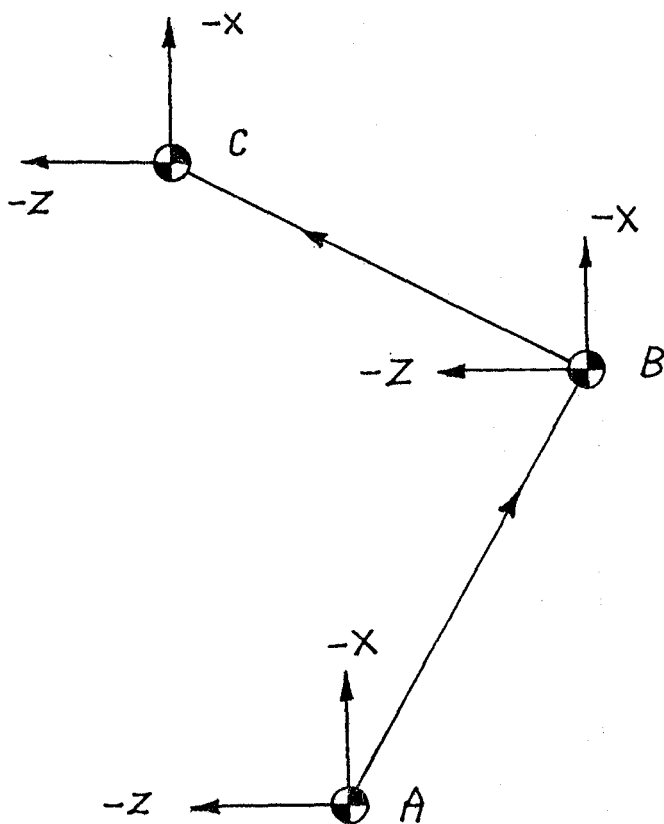
Artımlı sistem : Torna kaleminin istenilen hız ve şekilde gideceği yer o anda bulunulan noktaya göre verilir. Bir sonra ki hareket de yeni gelinen noktaya göredir. Şekil 1.1'deki hareketleri yaptıracak nokta koordinatları aşağıda verilmiştir.



Sekil 1.1 Hareket yolu örneği

	X	Z
1	-2.5	0.5
2	-1	-2.5
3	2	0
4	2	-1.5

Artımlı sistemde her hareket sonunda yeni bir koordinat merkezi belirleniyor denilebilir. A-noktasından B-noktasına gidişi ele alalım. Burada eksen yönleri sabit olmak şartı ile B-noktasının X ve Z bilgileri A-noktası sıfır olacak şekilde C-noktasının koordinatları ise B-noktası sıfır olacak şekilde verilir. Şekil 1.2'de bu sistemler gösterilmiştir.



Şekil 1.2 Artımlı sistemde koordinat sistemi merkezleri

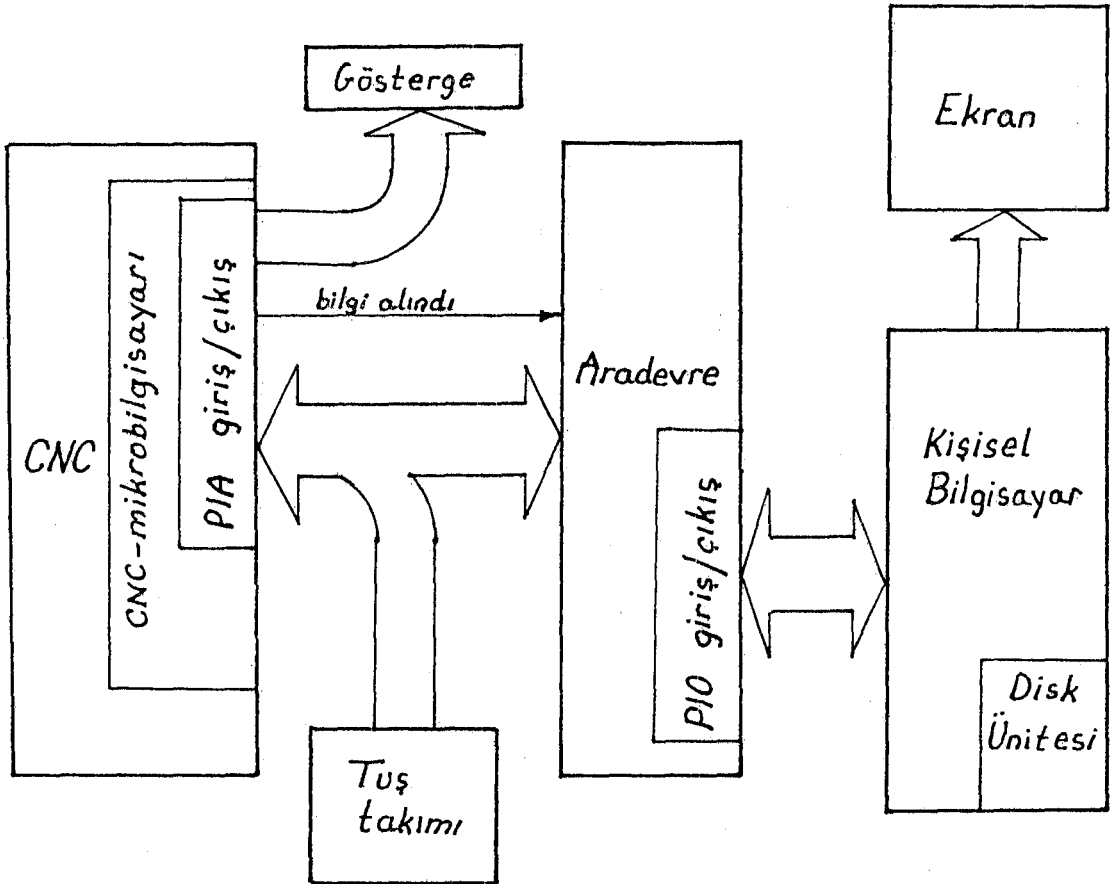
Kartezyen koordinat sistemi : Bu sistemde, birkez sıfır noktası belirlendikten sonra değiştirilmediği sürece her noktanın koordinatı belirlenen sıfır noktasının oluşturduğu sisteme göre verilir. Şekil 1.1'deki hareketler için, A-noktası koordinat sisteminin merkezi olarak belirlendiğinde her harekette verilecek X ve Z koordinatı aşağıdadır.

	X	Z
1	-2.5	0.5
2	-3.5	-2
3	-1.5	-2
4	0.5	-3.5

Artımlı ve kartezyen sistemler aynı iş parçası için birbirlerinin yerine geçebilirler. Ancak uygulamada kullanım yerleri farklılıklar göstermektedir. Teknik resimlerinde ölçüleri bir yere dayalı olarak verilen iş parçaları için kartezyen sistemin kullanılması kolaydır. Ayrıca boyutları büyük iş parçalarının üretiminde de kartezyen koordinat sistemi daha uygundur. Artımlı sistem ise birçok durumda kartezyen sistemden daha kolay olmasına rağmen bir noktanın değişmesi halinde buna bağlı olarak birçok noktanın değişmesini gerektirir.

2. İLETİŞİM KARTI VE PROGRAMI

Kişisel bilgisayar ile CNC-torna tezgahı arasında ilişki kurmak için tasarlanan bağlantı şekil 2.1'deki blok diyagram halinde özetlenebilir.



Şekil 2.1 Aradevrenin bağlantıları

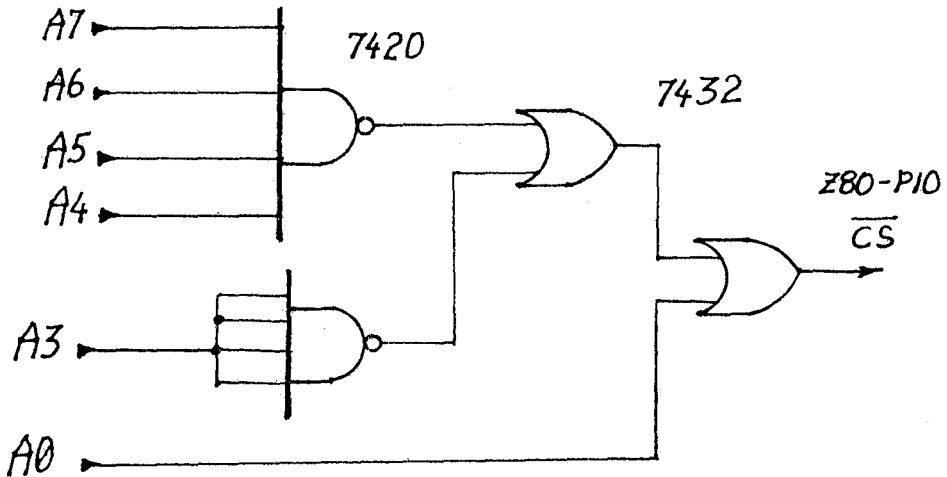
CNC-mikrobilgisayarında 6502-mikroişlemci ve 6821 giriş/çıkış birimi, kişisel bilgisayarda ise Z80-mikroişlemcisi vardır. Buna göre aradevredeki giriş/çıkış birimi doğrudan Z80'in adres, bilgi ve kontrol hatlarına bağlanacağından Z80'e uyumlu 8420 (Z80-PIO) giriş/çıkış birimi kullanılmıştır.

Kişisel bilgisayardan CNC'ye bilgi akışı bilgiyi gönderip cevap bekleme şeklindedir. Gönderilen bilgi CNC-mikrobilgisayarına kişisel bilgisayarda hazırlanan CNC-programının yazılması için basılması gereken tuşların bilgisidir. CNC-mikrobilgisayarının tuş takımında bulunan bir tuşa basmak ile bu tuş takımının okunması için kullanılan giriş/çıkış biriminin girişlerine bu tuşa ait bilgilerin konulması farksızdır.

Bu bölümde aradevrenin yapısı ve bağlantıları açıklanmıştır.

2.1 Devre Şeması

Kişisel bilgisayarda hazırlanan CNC-programlarının CNC-bilgisayarı hafızasına aktarılması için tasarlanan devre Z80-PIO ve yardımcı parçaları içeren bir I/O kartıdır. Kart üzerindeki Z80-PIO, kişisel bilgisayarın bütün adres ve bilgi hatları ve kontrol işaretlerinin verildiği bir konnektörden alınan gerekli hatlar ile XXF8-XXFF adreslerine yerleştirilmiştir. Adresleme için bir SN74LS20 ve bir SN74LS32 kapı tümleşik devreleri kullanılmıştır. Bu bağlantı şekil 2.2'de verilmiştir.



Sekil 2.2 PIO için adres seçici devre

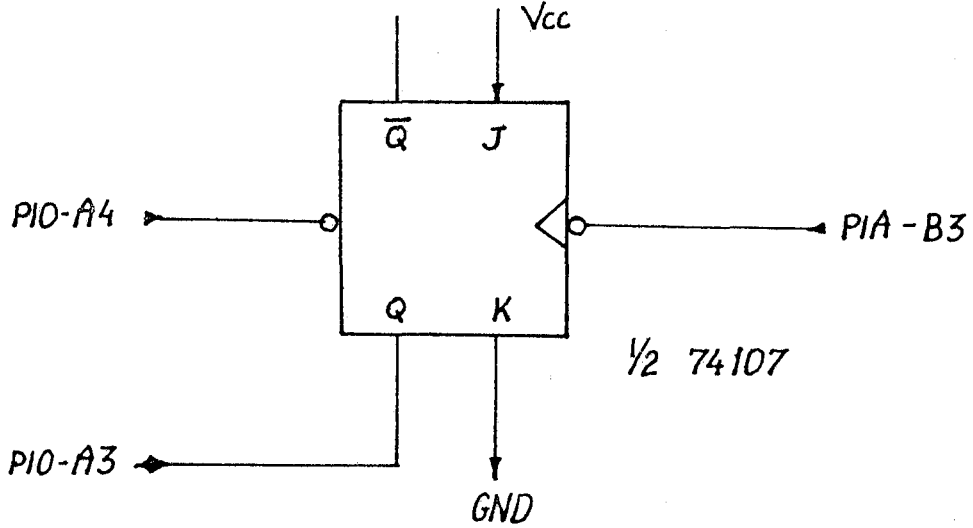
A1 ve A2 adres hatları PIO'nun C/D ve B/A kontrol girişlerine uygulanmıştır. Yani A ve B portlarının adreslemesi aşağıdaki gibidir.

FAFB	A portu
FAFA	A kontrol
FAFC	B portu
FAFE	B kontrol

Adresleme devresinde FAXX elde edilmemiştir. Ancak FAXX haricindeki adreslerde kişisel bilgisayarın kendine ait port ve elemanlar bulunduğu için iletişim devresindeki PIO'yu seçerken A8-A15 adres hatlarında FA sayısının bulunması gerekir. İletişim programında da port seçilirken FAXX adresleri kullanılmıştır.

CNC-bilgisayarı ile iletişim sağlamak için, bu mikro-bilgisayarda tuşların okunduğu porta bilgi konulmaktadır. Mikrobilgisayar bu portu tuşa basılmış gibi okumakta gerekli işlemleri yaptıktan sonra nümerik gösterge kontrol devresi olan ICM7218 tümleşik devresinin MODE girişi için portun B3 çıkışında bir artı darbe sinyali oluşturmaktadır. Bu işaret iletişim devremiz için mikrobilgisayarın verilen bilgiyi aldığını gösteren bir geri besleme olarak işlem görmektedir. Kişisel bilgisayardaki iletişim programının "bilgi alındı" işaretini A3 girişinden kolaylıkla okuyabilmesi için bir RS-FF set edilmekte, işaret okunduktan sonra bu FF program tarafından A4 çıkışından sıfırlanmaktadır. RS-FF aslında bir JK-FF'tur ve bağlantısı şekil 2.3'te verilmiştir.

Z80-PIO ve 6502-PIA arasındaki bağlantıyı sağlamak için 6502 mikrobilgisayara ilave edilen konektörün bağlantıları şekil 2.4'de verilmiştir. Mikrobilgisayar tuş takımının matris düzenlemesi de tablo halinde şekil 2.5'tedir.



Şekil 2.3 "Bilgi alındı" bağlantısı

1. GND
2. PIA0 : Giriş portu. Tuş takımı satırları
3. PIA1 : (Aşağıda açıklanmıştır).
4. PIA2 : .
5. PIA3 : .
6. PIA4 : .
7. PIA5 : .
8. PIA6 : .
9. PIA7 : .
10. PIB0 : Çıkış portu. Tuş takımı sütunları.
11. PIB1 : .
12. PIB2 : .
13. PIB3 : Çıkış portu. ICM7218'in MODE girişi.

Şekil 2.4 ilave edilen konektör bağlantıları.

Sütun						
Satır	PIB2		PIB1		PIB0	
PIA0	14		08	8	00	0
PIA1	13	-X	09	9	01	1
PIA2	12	+X	0A	-	02	2
PIA3	15	+Z	0F	INP	03	3
PIA4	16	-Z	0E	DEL	04	4
PIA5	10	H/C	0D	REV	05	5
PIA6	11	STR	0C	FWD	06	6
PIA7			0B		07	7

Her sütunda ilk rakamlar tuşun hex. değerini ikinciler tuş üzerindeki işaret ve görevini gösterir.

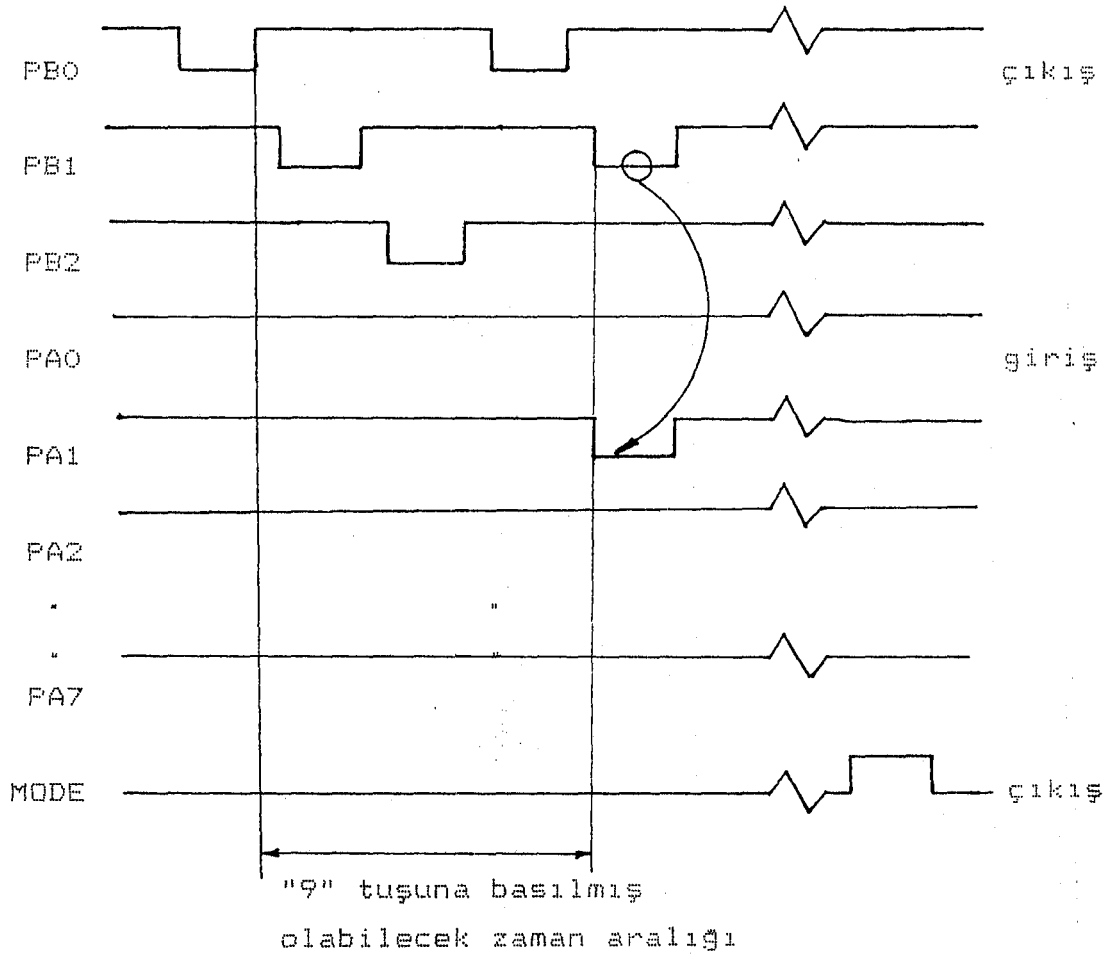
Şekil 2.5 Tuş takımının matris düzenlemesi.

Mikrokomputerin Tuşları Taraması :

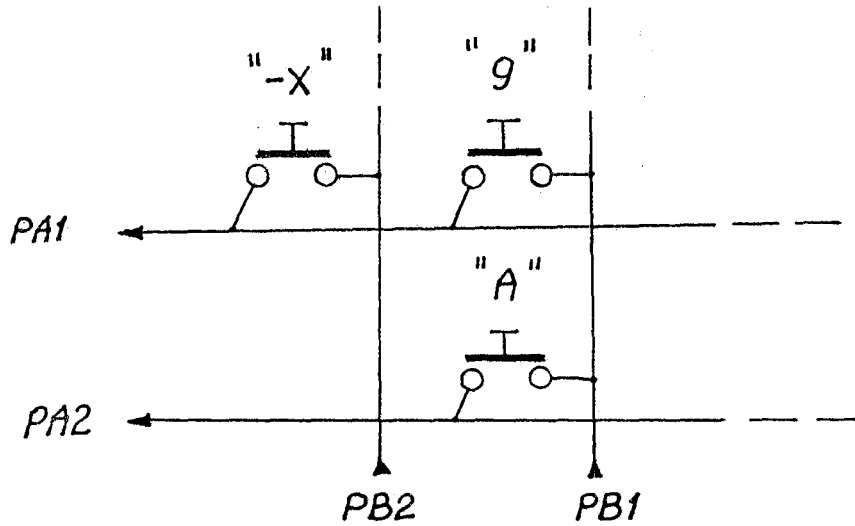
Mikrokomputer kartında tuş takımı ve nümerik gösterge ile ilgilenen giriş/çıkış Ünitesi olan MC6821 PIA-tümleşik devresi fiziksel olarak 9000 adresine yerleştirilmiştir. PIA'nın PB portu çıkış olarak programlanır ve tuşları PBO'dan PB2'ye tarar. Eger herhangi bir tuşa basılmış ise işaret PA portundan okunur. Bunun için de PA0-PA7 bacakları giriş olarak programlanır. Buradaki işaret şekilleri şekil 2.6'da gösterilmiştir. "9" tuşuna basıldığı varsayılmıştır.

PIA'nın tarama çıkışları olan PBO-2'deki işaretlerle eşzamanlı olarak kişisel bilgisayardan gelen bilgileri PIA'nın PA0-7 girişlerine koyabilmek için bir eşzamanlama devresi ve bir 8-bit 3-konumlu sürücü kullanılmıştır. Devre şekil 2.8'dedir.

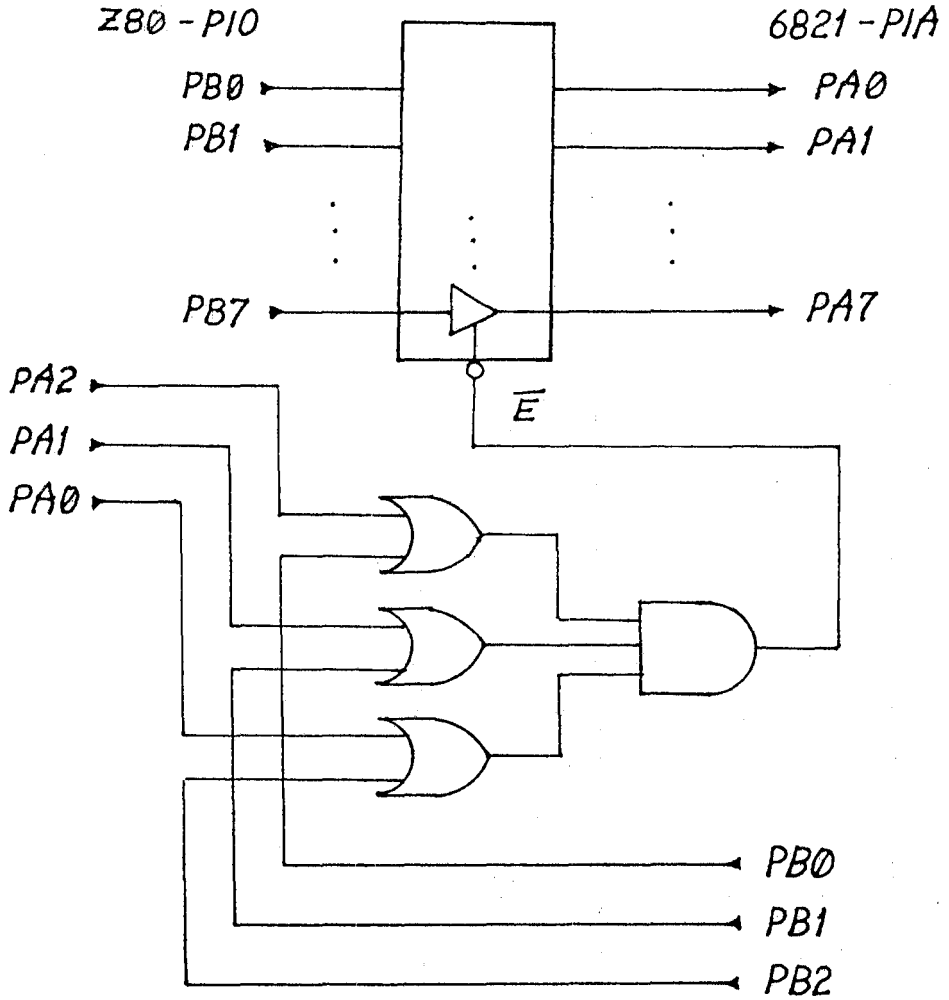
İletişim programı tarafından ZBO-PIO'nun PA ve PB portlarına belirtilecek tuşun satır ve sütun bilgisi konur. Bu bilgiler "9" tuşu için PB=11111101 ve PA=101'dir.



Şekil 2.6 Tuş tarama işaretleri.



Şekil 2.7 "9" tuşu bağlantısı.



Sekil 2.8 Eşzamanlama devresi.

6821-PIA'nın tarama çıkışları olan PB0-2'de 101 işareti görüldüğünde eşzamanlama devresi tarafından 3-konumlu sürücünün \bar{E} girişine 0-sıfır gider ve Z80-P10'nun PB portundaki satır bilgisi 6821-PIA'nın PA portuna iletilir. Böylece "9" tuşuna ait olan satır ve sütun bilgileri mikrobilgisayar tarafından elde edilmiş olur ve basılan tuşa ait işlemleri yaptıktan sonra nümerik göstergedeki düzeltme için PB3'e işaret gönderir. Bu da tuşun alındığına dair bir geri beslemedir.

İletişim kartının tüm şeması şekil 2.9'da verilmiştir.

2.2 İletişim Programı

Aradevreyi kullanarak CNC-torna tezgahı mikrobilgisayarı ile iletişimi sağlamak üzere makine dilinde hazırlanan program dört kısımdan oluşur. Bu kısımlar ve programın kullanıldığı bazı özel adresler aşağıda açıklanmıştır. Akış şeması şekil 2.10'da verilmiştir.

PORTGİR: Z80-PIO'nun PA ve PB portlarını giriş olarak programlar. Eğer mikrobilgisayara bilgi gönderilmiyorsa yani bekleme durumunda ise portlar giriş olarak bırakılır. Bunun için PA ve PB kontrol yazaçlarına 4F sayısı gönderilir.

PORTÇIK: Z80-PIO'nun PA ve PB-portlarını çıkış olarak programlar. Bu durumda CNC'ye bilgi gönderilebilir. Portları çıkış olarak programlamak için PA kontrol yazacına önce FF sonra da 58 gönderilir. Hangi bitlerin çıkış olacağı 58 bilgisi ile bildirilmektedir. PA portuna da FF gönderilerek tuşa basılmadığı belirtilir. PB-kontrol yazacına 0F yazılarak çıkış haline getirilir ve FF yazılır.

TUŞBAS: Bu program A200 adresindeki tuş numarasını kullanarak TABLO tablosundan tuşun satır ve sütun bilgilerini bulur. Bilgileri PA ve PB-portlarına koyar.

BAK: RS-FF çıkışını okuyarak mikrobilgisayarın verilen bilgileri alıp almadığını kontrol eder. Bu işlem bir çevrim içindedir. Yani yaklaşık 500 kez bu kontrolü yapar. Eğer cevap yok ise A201 adresine 05 yazar. Cevap geldikten sonra portlardaki tuş bilgilerini siler.

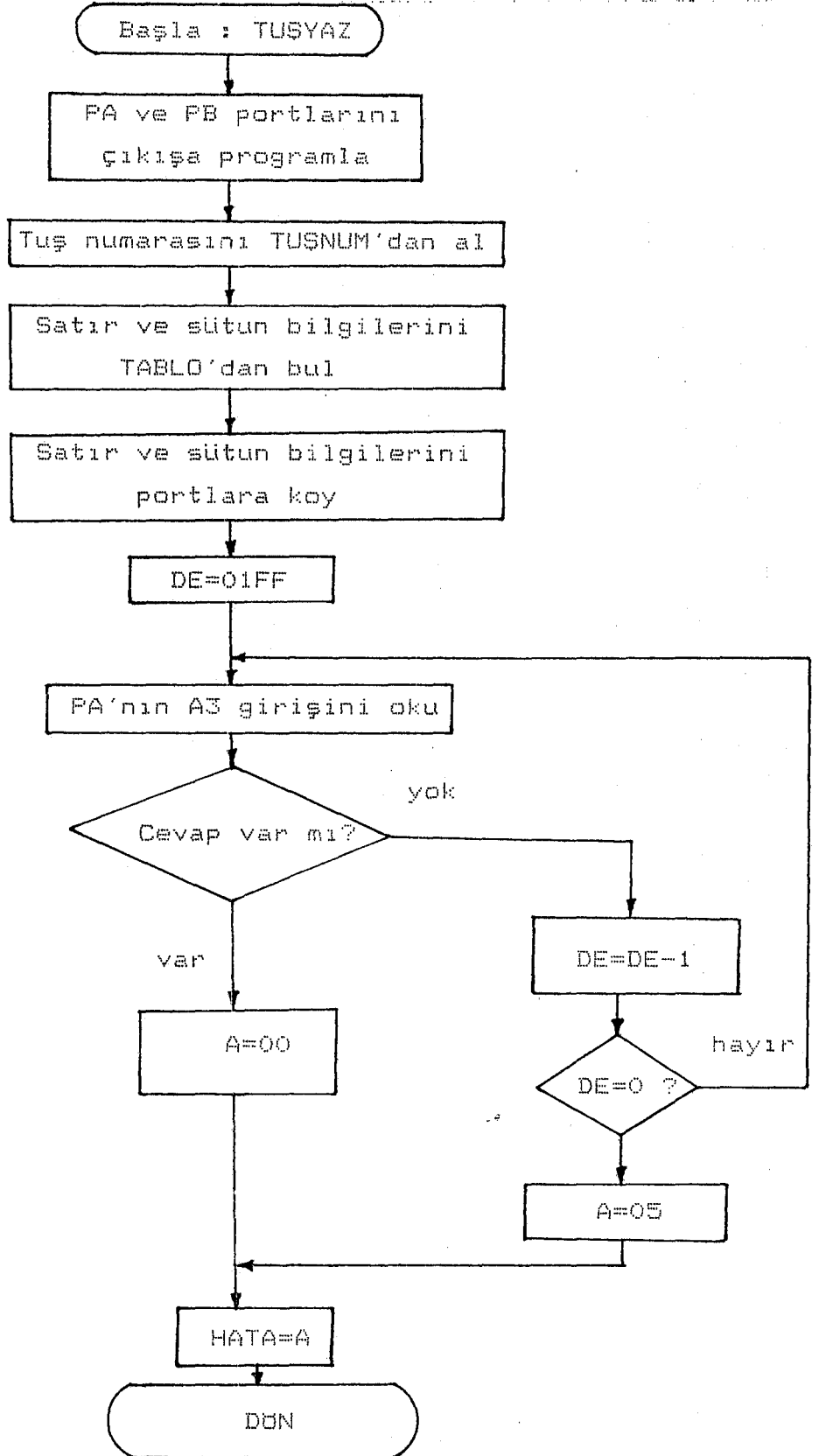
TUŞSİL: PA ve PB portlarına FF bilgisini koyar. Tuşlara basılı olmadığı belirtilir.

TABLO: Mikrobilgisayarın tuş takımındaki tuşlara ait satır ve sütun bilgilerini tablo halinde içerir. TUSBAS programı tarafından kullanılır. Tablo A060 adresinden başlar ve toplam 24 tuş için 48-bytes'lık yer kaplar.

TUŞNUM: Satır ve sütun bilgileri gönderilecek tuşun numarasının yerleştirildiği yerdir. Numara BASIC program tarafından bu adrese (A200) yerleştirilir.

HATA: BAK programında CNC mikrobilgisayarından cevap 500 kontrolden sonra da gelmemişse bu adrese (A201) 05 sayısı yazılır.

BASIC programın yukarıda açıklanan alt programları kullanım şekli ve sırası akış diyagramından anlaşılabilir.



Şekil 2.10 Makine dilindeki programın akış şeması.

* RUTIN : TUSYAZ PROGRAMI *

; TUSNUM : #A200 ; YAZAN : EROL SEKE
 ; HATA : #A201
 ; TABLO : #A060

```

A000 01 FA FA PORTGIR LD BC,#FAFA ;A PORTUNU GIRIS YAP
A003 3E 4F LD A,#4F
A005 ED 79 OUT (BC),A
A007 0E FC LD C,#FE
A009 ED 79 OUT (BC),A
A00B C9 RET
A00C 01 FA FA PORTCIK LD BC,#FAFA ;A VE B PORTLARINI
A00F 11 58 LD DE,#FF58 ;CIKIS YAPAR
A012 ED 51 OUT (BC),D
A014 ED 59 OUT (BC),E
A016 0E FB LD C,#FB
A018 ED 51 OUT (BC),D
A01A 0E FE LD C,#FE
A01C 3E 0F LD A,#0F
A01E ED 79 OUT (BC),A
A020 0E FC LD C,#FC
A022 ED 51 OUT (BC),D
A024 C9 RET
A025 21 61 A0 TUSBAS LD HL,#TABLO ;PORTLARA A200 ADRE-
A028 3A 00 A2 LD A,(A200) ;SINDE NUMARASI BULU-
A02B CB 27 SLA ;NAN TUSUN BILGILERI-
A02D 85 ADD A,L ;NI PORTLARA GONDERIR
A02E 6F LD L,A
A02F 7E LD A,(HL)
A030 01 FC FA LD BC,#FAFC
A033 ED 79 OUT (BC),A
A035 2B DEC HL
A036 7E LD A,(HL)
A037 11 FF 01 LD DE,#01FF ;BEKLEME SAYISI

```

```

A03A 0E F8 LD C, #F8
A03C 2E DF LD L, #DF
A03E ED 69 OUT (BC), L
A040 ED 79 OUT (BC), A
A042 ED 78 BAK IN A, (BC) ; CNC'DEN CEVAP BEKLE
A044 E6 08 AND #08
A046 20 08 JR NZ, DEVAM
A048 1D DEC E
A049 20 F7 JR NZ, BAK
A04B 15 DEC D
A04C 20 F4 JR NZ, BAK
A04E 3E 05 LD A, #05
A050 18 02 JR TUSSIL
A052 3E 00 DEVAM LD A, #00
A054 32 01 A2 TUSSIL LD HATA, A ; PORTLARI YUKSEK EM-
A057 3E FF LD A, #FF ; PEDANSLI HALE GETIR
A059 ED 79 OUT (BC), A ; HATA VARSA YAZ
A05B 0E FC LD C, #FC
A05D ED 79 OUT (BC), A
A05F C9 RET

A060 FE FE FE FD TABLO ; %%%%%%%%%
A064 FE FB FE F7 ; TUŞ BİLGİLERİ TABLOSU
A068 FE EF FE DF ; 23 TUŞ İÇİN ÇIKIŞ KODLARI
A06C FE BF FE 7F ; BULUNUR.
A070 FB FE FB FD ; BU PROGRAM A200 ADRESİNDE
A074 FB FB FB F7 ; NUMARASI BULUNAN TUŞUN SATIR
A078 FB EF FB DF ; VE SUTUN BİLGİLERİNİ CNC PORTU
A07C FB BF FB 7F ; GİRİŞLERİNE YAZAR VE CEVAP
A080 FD FE FD FD ; BEKLER. 6502-MİKROBİLGİSAYAR-
A084 FD FB FD F7 ; DAN CEVAP GELMEZSE A201 ADRE-
A088 FD EF FD DF ; SİNE 05 YAZAR, AKSİ HALDE
A08C FD BF FD 7F ; A201 ADRESİNE 00 YAZAR.

```

SON

C+%%%%%%%%

;

2.3 BASIC Dilinde Egitici Program

Bu bölümde kişisel mikrobilgisayarın ve BASIC'in grafik olanakları kullanılarak hazırlanan eğitimci programın açıklamaları verilmiştir. Programın her bir yardımcı altprogramına iki harfli küçük komutlarla gidilmektedir. BASIC programının listesi EK-1'dedir. Kullanılan küçük komutlar aşağıda açıklanmıştır.

Program Listesi Alma : LS komutu ile CNC- dilinde yazılan programın listesi ekrana alınabilir. İstenilen satırdan itibaren listeyi almak için LS'den sonra satır numarası yazılır, aksi halde sıfır numaralı satırdan itibaren liste verir. Liste almayı durdurmak için boşluk tuşuna (space bar) basılır. İki kere bu tuşa basıldığında işlem akışı liste alma altprogramından döner.

Program Yazma : ED komutu ile CNC- programı yazma altprogramı çağırılır. ED komutundan sonra satır numarası yazılmamış ise NO'dan itibaren program yazılacağı kabul edilir. Program yazma şu şekildedir. ED komutu ile beraber ekrana

NO G

gelir.

G-kodu yazıldıktan ya da yazılıysa düzeltildikten sonra Return tuşuna basılır.

NO G01 X

ekrana gelir. Aynı şekilde X, Z, F ve CM* sütunlarına bilgiler yazılır ya da düzeltilir. Yazılan G-kodu CNC'de geçersiz ise program G-kodunu, eskisini silerek tekrar ister. X Z ve F-bilgileri de geçerli değerlerin dışında ise program bilgileri tekrar ister. Program X, Z ve F sütunlarına G-koduna göre gerekli ise gelir. Örneğin G00 için F sütununa gelmez ve F-bilgisi istemez.

CM sütunundaki bilgi yazıldıktan sonra program bir alttaki satıra geçer.

*:Açıklama sütunu. X, Z ve F bölüm-3'te açıklanmıştır.

N1 G

gelir.

Daha önce yazılmış bir satırda düzeltme yapılıyor ise düzeltilmeyecek sütunlar için yalnızca Return tuşuna basılır. Programlama altprogramından çıkmak için G sütununda OE yazılır.

Program Yazmaya Devam Etme : RP komutu ile programın en son satırından bir sonraki satırdan başlamak üzere program yazma altprogramına geçilir.

Boş Satır ilave Etme : IN komutu ile, komutla beraber verilen satır ve sonraki satırları bir satır ileriye atar. Komut ile numarası verilen satırın G sütununa G21 yazar. G21-kodu boş satır anlamına gelir. Bölüm 3'te açıklandığı gibi bu satırda CNC tezgahı hiç bir iş yapmaz.

Satır Silme : DL komutu ile komutun yanında numarası verilen satırı siler. Bu satırdan sonraki satırları bir satır yukarıya yazar.

Programı Silme : ER komutu ile, yazılan ya da disket ünitesinden yüklenen bütün program silinir.

Disketten Program Yükleme : LO komutu ile program yükleme altprogramına geçilir. Altprogram disketten yüklenecek CNC-programının adını

PROGRAM İSMİ :

yazarak ister.

Örnek olarak TORNA isimli programın ismi yazılır.

Return tuşundan sonra yükleme programı ekrana

YUKLEME...! LUTFEN BEKLEYİNİZ

yazar ve programı disketten bilgisayar hafızasına aktarır. Eger yüklenen programda programın bittigini belirten END komutu açıklama sütununda yok ise ekrana

TORNA .CNC PROGRAMINDA END KOMUTU YOK

yazılır. Yükleme bittikten sonra ekrana

TORNA .CNC YUKLENDİ

bildirimi yazılır.

Diskete Program Saklama : SA komutu ile CNC-dilinde yazılan programın daha sonraki kullanımlar için manyetik diskte saklama amacıyla program saklama altprogramına geçilir. Yüklemede olduğu gibi burada da programın hangi isim altında saklanacağı sorulur, ekrana

PROGRAM ISMI :

yazılır. İsim yazılıp Return tuşuna basıldıktan sonra ekrana

SAKLAMA.! LUTFEN BEKLEYİNİZ

yazılır. Saklama işlemi bittikten sonra ekrana

PROGRAM TORNA .CNC ADI ALTINDA SAKLANDI

PROGRAM UZUNLUGU 36

yazılır.

Eğer saklanan programda END komutu yok ise asıl programdan sonra N159'a kadar olan boş satırlar da diskete kaydedilir. Böylece program diskette, olduğundan daha büyük yer işgal eder.

Hem yükleme hem de saklama alt programları CNC-programının program tipi olan .CNC işaretini kendisi koyar. Eğer program ismi bilgisayar tarafından istenildiğinde isim içinde (.) var ise ekrana

PROGRAM ISMI GECERSİZ

yazılır.

Disketteki Programların Listesi : CA komutu ile diskette CNC-dilinde yazılmış programların listesi ekrana gelir.

Disketteki Programı Silme : KL komutu ile disketteki bir CNC-programı silinebilir. Ekrana

PROGRAM ISMI :

yazılır. Eğer program silmekten vazgeçilirse Q yazılır.

Ekrana

PROGRAM SİLİNMEDİ

yazılır. Silinecek programın ismi yazılıp Return tuşuna basıldıktan sonra disketteki program silinir ve ekrana

TORNA .CNC SİLİNDİ

yazılır. Program ismi içinde (.) olmamalıdır.

Ekranı Silme : CL komutu ile ekran silinir.

Yazıcıdan Liste Alma : LP komutu ile, G-kodları ile yazılan program yazıcı ile kağıda geçirilebilir.

Torna Kaleminin İzlediği Yol :DO, ET ve DR komutları ile CNC-programının torna kalemine izlettiği yol ekrana çizilir. DO ile satırlar ekrana yazılmadan, ET ile satırlar ekrana yazılarak çizim yapılır. DR ile her satırda tuşa basılması beklenir.

Orijin Seçme : OR komutu ile kalem yolu altprogramı için başlama noktası seçilir.

Ekranında Parça İşlenişi : RN komutu ile ekranında parça işleme altprogramına geçilir. Bu programda önce ekrana çizilecek ham parçanın boyutları istenir.

R? çap

L? uzunluk

olmak üzere, ekran noktası cinsinden boyutlar yazılır. Daha sonra tezgah aynası, parça ve torna kalemi ekrana çizilir. Torna kalemi programa göre hareket etmeye başlar. Eğer tezgah kütüphanesinde olmayan bir komut ile karşılaşırsa ekranın altına

GECERSİZ G-KOMUTU C/Q

yazılır. C tuşuna basılırsa program bir alt satırdan devam eder. Q tuşuna basılırsa parça işleme altprogramından çıkarılır. Bazı G-kodları tezgah kütüphanesinde olmasına rağmen

eđitim programında denenemezler. Bunlarla karşılaşıldığında ekrana

G24 KOMUTU DENEMEK İCİN GECERSİZDİR

yazılır. C tuşuna basıldığında altprogramdan çıkılır. İşlem sırasında G20-program bekleme komutu ile karşılaşıldığında ekran altına

PROGRAM BEKLIYOR

yazılır. C tuşuna basıldığında program kaldığı satırdan itibaren devam eder.

Program G22-kodu ile karşılaştığında el ile işleme altprogramına geçer.

El ile işleme : HO komutu ile ekrandaki torna kalemine yön tuşları ile hareket veren altprograma geçilir. C tuşuna basıldığında bu altprogramdan çıkılır.

CNC-El Operasyonu : HC komutu ile kişisel mikrobilgisayarın tuş takımından CNC-torna tezgahına kumanda etmeyi sağlayan altprograma geçilir. Bu işlem için aradevrenin bağlanmış olması gerekir.

(Dikkat : Tezgah ve bilgisayar çalışır durumda iken aradevreyi bağlamak sakıncalıdır.)

Mikrobilgisayardan CNC'ye Program iletimi : TR komutu ile, kişisel mikrobilgisayarda yazılmış olan ya da disketten yüklenen CNC-programının CNC-tezgahı hafızasına aradevre yardımı ile iletimi sağlanır. Bu altprogram, makine dilinde yazılmış olan TUSYAZ programını kullanır. CNC-programının iletilişi tezgah göstergesi ya da monitörden izlenebilir.

İletim sırasında kişisel bilgisayar ekranına TIMOUT yazılabilir. Bu uyarı, iletim esnasında şebeke gerilimindeki bir parazit, gerilim düşümü olduğunda ya da kişisel bilgisayardaki hatalı bir CNC-kodunun iletiminde tezgahın cevap vermemesi ya da hata mesajı vermesi ile yazılır. Hata monitörden de izlenebilir.

Tezgahtaki Programı Çalıştırma : ST komutu ile tezgahtaki program çalıştırılır. Tezgah üzerindeki Start tuşu ile aynı amaçlıdır.

3. G KOMUTLARI

Bircok endüstri tipi CNC tezgahlarında kullanılan G-komutları birkac deęişiklik dışında eğitim amaçlı CNC tezgahında da vardır. Bu bölümde CNC-torna tezgahına ait G-komutları örneklerle kısaca anlatılmıştır. CNC programlama dilinde komutlar ve yardımcı bilgiler, bloklar halinde yazılırlar. Bloklara satır da denilebilir. Bir bloktaki bilgiler şekil 3.1'de gösterilmiş ve tek tek açıklanmıştır.

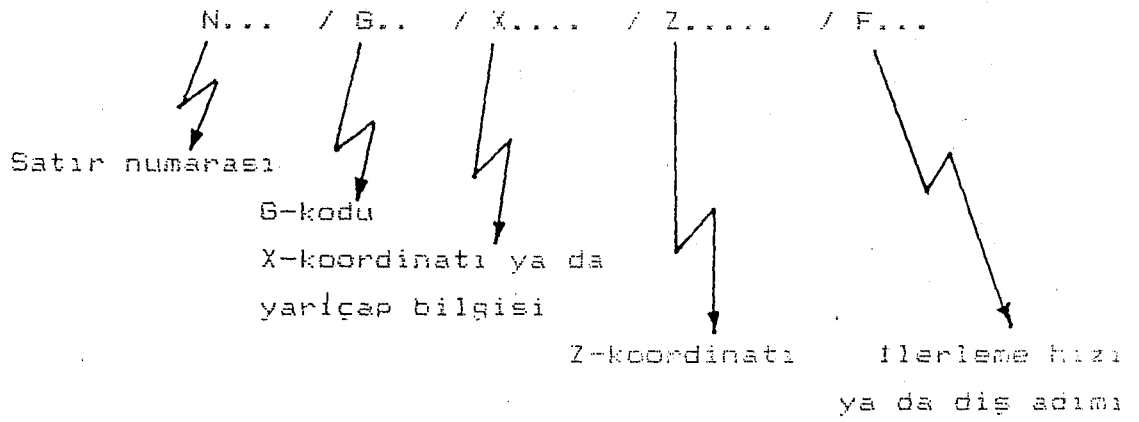
1. G-kodu : Torna tezgahı kaleminin yapacağı hareketin tipi bu kodla verilir (dairesel, doğrusal, kapalı çevrim vs.). Bazı G-kodlarında kalem hiçbir hareket yapmaz. Bunlar mikrobilgisayarda koordinat sistemi bildirmek, işlemi durdurmak ya da bekletmek gibi amaçlar için kullanılırlar. G-kodları iki rakamlı olarak yazılırlar (00, 21, 84 gibi). G-kodunun belirttiği işleme göre, gerekliyse bloka X, Z, F bilgileri yazılır. Bu bilgiler şekil 3.2'deki koordinat sistemini temel alırlar.

2. X-bilgisi : X-koordinatı doğrultusundaki hareket miktarı bilgisidir. Dairesel hareketlerde yarıçap/çap bilgisi olarak işlem görür. Koordinat bildiriliyorsa sıfır noktasının X-koordinatını bildirir. Eger çevrimli bir hareket ise G-koduna göre +/-X yönlerindeki hareketleri öncelikle X-bilgisi kısmına yazılan olmak üzere yapar. Bu kısma mm. cinsinden 0 - +/-59.99 mm. arasında bilgi yazılabilir.

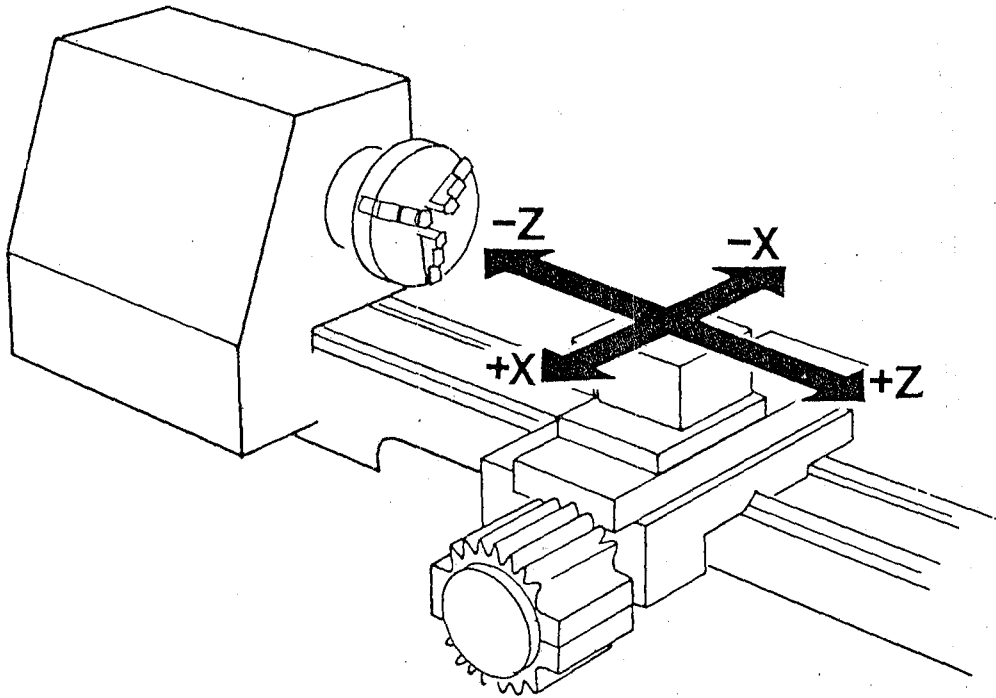
3. Z-bilgisi : Z-koordinatı doğrultusundaki hareket miktarını belirtir. Dairesel hareketlerde bu bilgi verilmez. Koordinat sistemi bildiriliyor ise sıfır noktasının Z-koordinatını bildirir. Çevrimli bir hareket ise +/-Z yönlerindeki hareketleri öncelikle Z-bilgisi kısmına yazılan olmak üzere yapar. Verilen bilgiler 0 - +/-399.99 mm. arasında olmalıdır.

4. F-bilgisi : Torna kaleminin ilerleme hızı bu bilgi kısmında mm/dk ya da mm/devir olarak verilir. Burada 0 - 499 mm/dk arasındaki ilerleme hızları geçerlidir.

Şimdi CNC-torna tezgahında kullanılan G-komutlarını



Sekil 3.1 Satır numarası ile beraber bir blok



Sekil 3.2 Koordinat sistemi

basitlik sırası ile inceleyelim.

G21 - BOS SATIR

Program hazırlanırken ve/veya yazarken daha sonra ilave edilebileceği tahmin edilen bloklar için boş satırlar bırakılabilir. Bu iş için de G21-komutu kullanılır. Mikro-bilgisayar bu satırda hiçbir işlem yapmadan bir sonraki satıra geçer. G21-komutu X, Z, F bilgileri içermez.

N15 G21 - - -

G22 - PROGRAM SONU

Mikro-bilgisayara işletimdeki programın bittiğini ve bundan sonra bir işlem yapılmayacağını belirtmek için kullanılır. Programın sonunda kullanılabileceği gibi herhangi bir satırında da kullanılabilir. X, Z ve F bilgileri yazılmaz.

N16 G22 - - -

Bir program içinde G22 bulunmuyor ise o program işletilemez.

G20 - BEKLEME

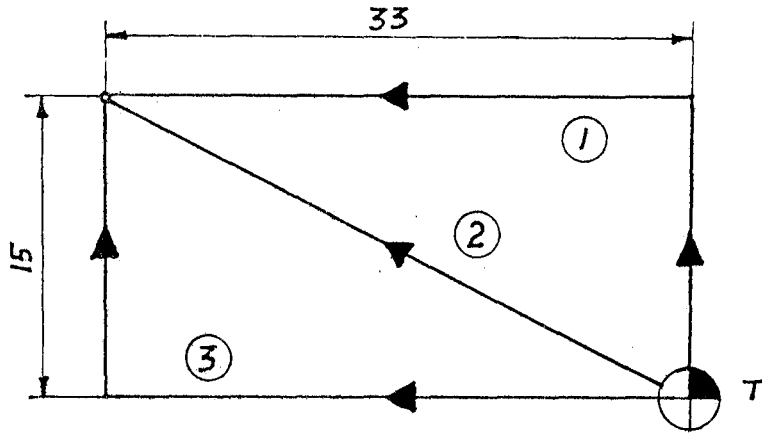
Tezgahta bir programın işletilmesi sırasında herhangi bir işlemden sonra kalem hareketinin "başla" işaretine kadar durması istenirse bir sonraki satıra G20-kodu yazılır. Parça üzerinden ölçü alınacağı, parçanın kontrol edileceği ya da torna aynasının devir sayısının değiştirileceği zamanlarda kullanılır. Yapılacak işlem bittiğinde "start" tuşuna basılarak programın kaldığı yerden devam etmesi.

sağlanır.

N18 G20 - - - :Program burada bekler.

G00 - DOGRUSAL HIZLI HAREKET

Torna kaleminin bulunduğu noktadan hareket sınırları içerisindeki diğer bir noktaya hızla gitmesi istendiğinde G00-kodu kullanılır. Hızlı hareket sırasında bir talaş alma -parça işleme düşünülmemektedir. Hareket aksenlere paralel ya da köşegen üzerinden olabilir. Şekil 3.3'de bir örnek verilmiştir.



Şekil 3.3 Doğrusal hareket.

1. Yol

N05 G00 X-1500 Z0

N06 G00 X0 Z-3300

2. Yol

N07 G00 X-1500 Z-3300

Üçüncü yol olarak ta alttaki dik kenarlar kullanılabilir. Şekil 3.2'deki koordinat sistemi burada da

geçerlidir. Aslında bu sistemdeki X ve Z eksenlerinin yönleri her zaman sabittir, ancak sıfır noktası değişkendir. Eğer örnekteki gibi "artımlı" sistem kullanılıyor ise her bir başlangıç noktası bir sistem sıfırıdır. Yani örnekte gidilen noktadan geri dönmek için,

```

.
.
NOB G00 X1500 Z3300
.
.

```

bloğu kullanılmalıdır. "Kartezyen" sistemde ise,

```

.
.
NOB G00 X0 Z0
.
.

```

bloğu ile kalem eski yerine yani sıfır noktasına döner. "Artımlı" ve "kartezyen" sistemler arasındaki farklar bölüm-1'de anlatılmıştır.

G01 - HIZI VERİLEN DOĞRUSAL HAREKET

G01 komutu ile torna kaleminin istenilen noktaya istenilen hızla gitmesi sağlanılır. G00'dan farkı F-feed kısmında hızın verilmesidir. İlerleme hızı mm/devir ya da mm/dakika olarak verilebilir. Bu olanaklar G94 ve G95'te açıklanmıştır.

```

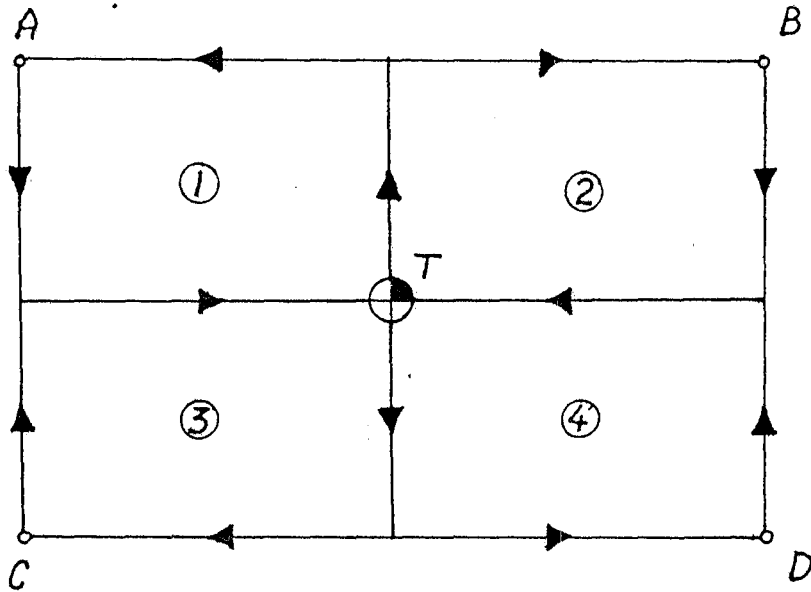
.
.
NO7 G00 X-500 Z0
NO8 G01 X0 Z-2500 F20
.
.

```

program kesiti ile önce -X yönünde 5 mm hızlı ilerleme yapılmış daha sonra NO8'de "G01 X0 Z-2500 F20" komutu ile -Z yönünde 25 mm ilerlenmiştir. İlerleme hızı ise 20 mm/dk dir.

G84 - DİKDÖRTGEN ÇEVİRİM

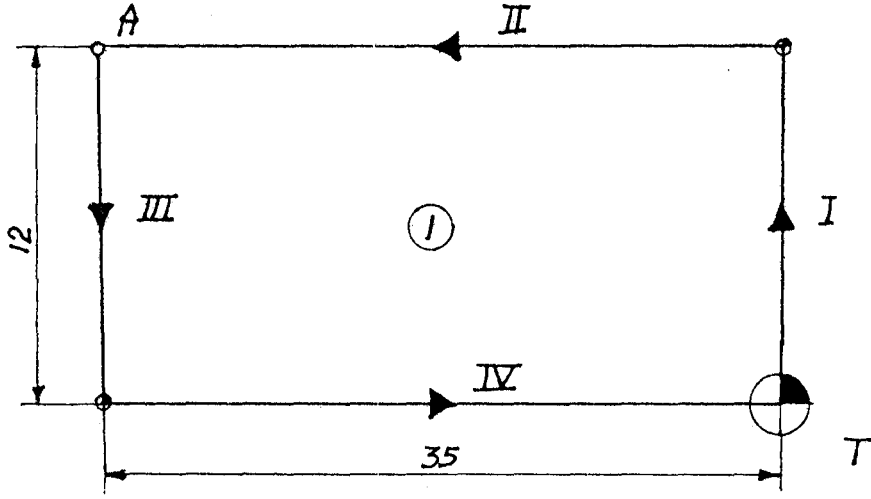
Eksenlere paralel olarak tornalama-talaş alma birçok kez yapılacaksa G84-çevrim kodunun kullanılması daha ekonomik olur. Bu komut dört hareketten oluşur. Olasılıklar şekil 3.4'de gösterilmiştir.



Sekil 3.4 G84 çevrimleri.

Sekil 3.4'de T-noktası başlangıç noktasıdır. Bu noktaya gelen ve giden hareketler G00-kodu ile yapılıyor gibi diğer hareketler de G01-kodu ile yapılıyor gibi düşünülebilir. Şekildeki dört olasılık G84-kodu ile verilen X ve Z bilgilerinin işaretleri ile seçilir. Örneğin 1.nolu olasılık için hareket şekil 3.5'de verilmiştir.

Seçilen kontur için X ve Z bilgisi olarak A, B, C veya D noktasının T'ye göre koordinatı verilir.



Sekil 3.5 G84-kodunda 1.nolu olasılık.

N18 G84 X-1200 Z-3500 F40

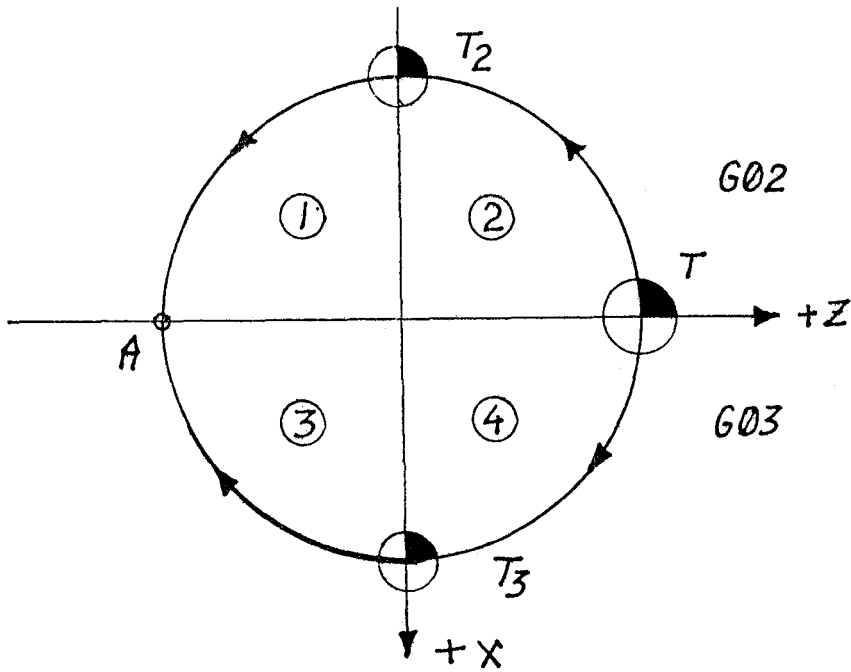
bloğu verilebilir. I. ve IV. hareketler G00 ile II. ve III. hareketler G01 ile yapılıyorymuş gibi düşünülür ve ilerleme hızı da 40 mm/dk dir. Bu işlemi G00 ve G01 kodlarını kullanarak yapılacak olursa

I	N18	G00	X-1200	Z0	
II	N19	G01	X0	Z-3500	F40
III	N20	G01	X1200	Z0	F40
IV	N21	G00	X0	Z3500	

program kesitinin yazılması gerekecektir. İkisi arasında hiçbir fark yoktur. Sonuçta kalem başlangıçtaki yerine döner. İkinci bir talaş alma-tornalama işlemi için hazır olur.

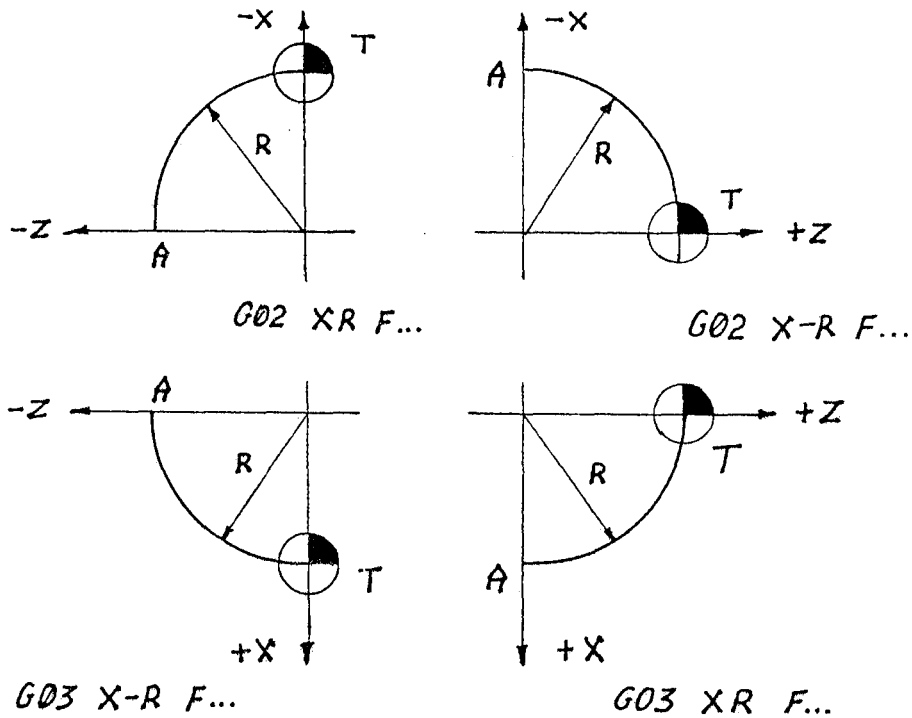
G02 - G03 - DAİRESEL HAREKETLER

Eğitim amaçlı CNC-torna tezgahında dairesel tornalamalar için bir kolaylık olarak bir dairenin dört çeyrek parçası için iki ayrı kod kullanılmıştır. Tezgahın zayıf tarafı olarak bir dairenin istenilen bir yay parçası üzerinde hareketini bitirememesi gösterilebilir. Yani torna kaleminin hareketi 90° yi tamamlayacak şekilde belirlenmiştir. Şekil 3.6'da bu dört olasılık belirtilmiştir.



Şekil 3.6 Dairesel hareket yörüngeleri.

Torna kalemi 90° yi tamamladığı için yalnızca X-bilgisinin verilmesi yeterlidir. Buradaki X-bilgisi hem yarıçapı belirtir hem de işareti her bir koda ait (G02 veya G03) iki çeyrek dairenin (1 veya 2, 3 veya 4) hangisinin seçildiğini gösterir. Z yönündeki hareket ise daima -Z doğrultusundadır. Buna göre her bir çeyrek daire için Şekil 3.7'deki örnekler verilebilir.



Sekil 3.7 Dairesel hareket olasılıkları

R'nin alabileceği değerler sınırlıdır. Mümkün olan yarıçaplar 0.25, 0.50, 1, 2, 3, ..., 59 mm gibi ilk ikisi hariç tamsayılardır.

G78 - G33 - DIŞ AÇMA

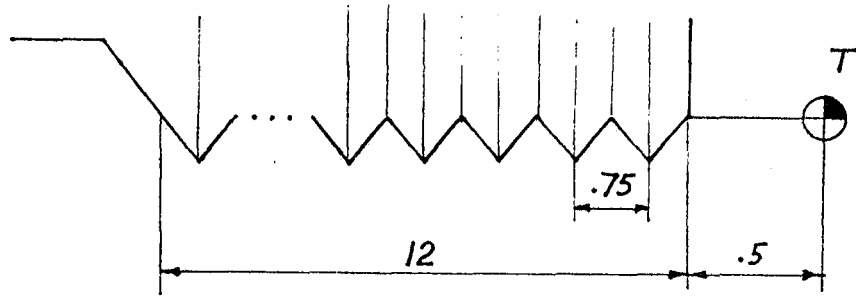
Eğitim amaçlı CNC-torna tezgahında G33-kodu ile değişik ölçülerde diş açmak ta mümkündür. Diş açmak için (dişlerin standartlara uygun olması için) çok önemli olan devir sayısı ile kalem arasındaki eşzamanlılık, ayna gerisindeki optik duyargalardan mikrobilgisayara eşzamanlama işareti alınarak oldukça hassas bir şekilde sağlanmıştır. Tezgahta diş açmakta gerekli bilgiler Z ve F kısımlarında verilir. Z kısmında diş uzunluğu, F kısmında da diş adımı $mm \times 100$ olarak verilir.

Sekil 3.8'deki dişi açmak için yazılması gereken komut

N13 G33

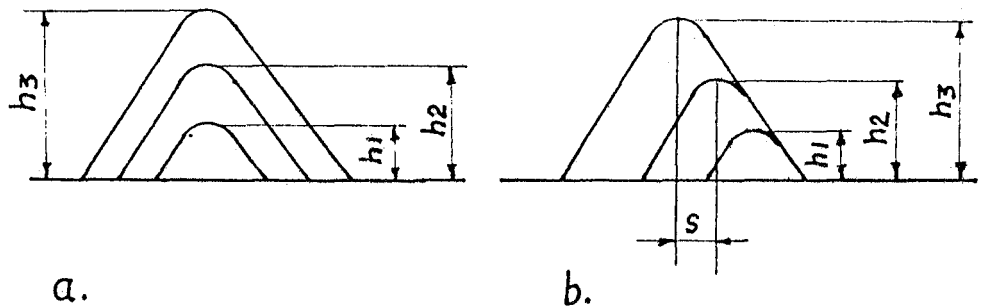
Z-1250 F75

tir.



Sekil 3.8 Diş açma örneği

Diş derinliği bir defada istenilen ölçüye getirilemeyebilir. Kalem tekrar T-noktasına getirip -X yönünde bir miktar kaydırarak kademe kademe istenilen derinliğe inilebilir. Şekil 3.9'da birkaç kademeli diş açma durumunda izlenebilecek iki yol için dişin durumu gösterimiştir.



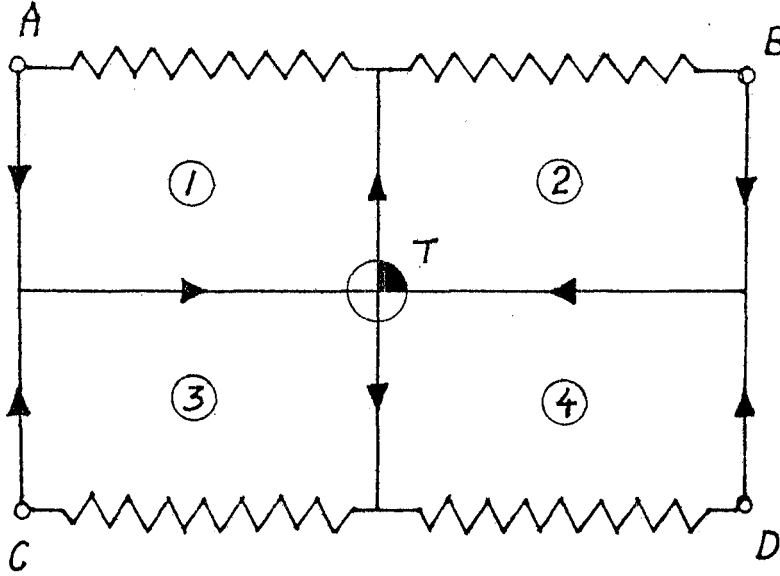
Sekil 3.9 Diş açma kademeleri.

Sekil 3.9-b'deki diş açma şekli için her kademede kalem -Z yönünde s kadar, -X yönünde h_2-h_1 yada h_3-h_2 kadar T-noktasından ötelemek gerekir.

Eğer dişin birçok kademede açılması gerekiyorsa G33 yerine G78-kodunun kullanılması daha ekonomik olur.

G78-kodu bir diř açma çevrimidir. G64 çevrimine benzetilebilir, ancak ilerleme hızları bakımından farklıdır.

G78 olasılıkları Őekil 3.10'da gösterilmiřtir.



Őekil 3.10 G78 çevrim olasılıkları.

Diř açma çevrim kodu G78 ile birlikte istenilen diř sınıfına göre A, B, C veya D-noktasının koordinatı ve diř adımı verilir. Burada bu deęişik hareketleri Őöyle sınıflandırılabilir.

1. Diř saę vida
2. Diř sol vida
3. İç saę diř
4. İç sol diř

Őekil 3.10'da verilen konturlarda T-noktasına yaklařan iki doęru parçası hızlı hareketlerle (G00) herbir olasılığa ait diđer 3 doęru parçaları ise F kısmında programlanan diř adımı ile gerçekteřir.

G90 - KARTEZYEN KOORDINAT SİSTEMİ

Programlama şekli ve sistemi "artımlı" ise "kartezyen" sisteme geçmek için G90-kodu kullanılır. G90-komutu kalıcıdır. Yani G91 ile değiştirilmediği sürece geçerlidir. X, Z ve F bilgileri gerektirmez.

N10 G90 - - -

G90-komutu ile o anda bulunulan nokta kartezyen koordinat sisteminin merkezi olarak kabul edilir. Bundan sonraki işlemler ve koordinatlar bu nokta sıfır noktası alınacak şekilde yapılır. G90-komutu G91-komutu ile değiştirilebilir.

G92 - KARTEZYEN KOORDINAT SİSTEMİ SEÇME

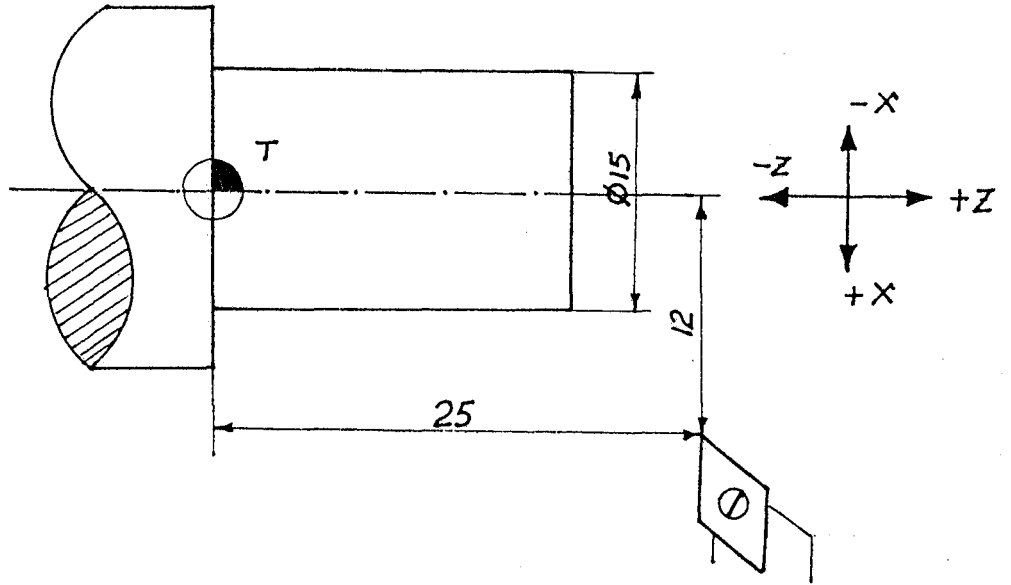
G90-kodu gibi bir kartezyen koordinat sistemi kurmak için kullanılır. Ancak mümkün olan her nokta sistemin merkezi olarak seçilebilir. G90'da ise yalnızca o anda bulunulan yer sıfır noktası olabilir di. G92 ile iş parçası ya da ayna üzerinde veya içinde bir nokta merkez olarak seçilebilir. Bunun için bulunulan noktanın seçilen noktaya göre koordinatları* X ve Z kısımlarında bildirilir.

Şekil 3.11'deki kalem ve iş parçasına göre T-noktasını koordinat sisteminin merkezi olarak seçmek için

N09 G92 X2400 Z2500

bloğu kullanılır.

* Çap programlama kullanıldığından X-koordinatı 2 ile çarpılır.



Sekil 3.11 Koordinat sistemi seçme.

Parça eksenini X için sıfır kabul edilirse kalem ucu daima torna çapı verilmiş şekilde tornalayacaktır. Yani X-koordinatı daima iki ile çarpılmalıdır.

G91 - ARTIMLI SİSTEM

Tezin hazırlanmasında kullanılan sistem olan artımlı sistem (incremental) her hareketin başlangıcında bulunan yeri sıfır noktası kabul eder ve ulaşılabilecek noktaya ait koordinat ya da yarıçap bilgisi bu sıfır noktasına göre verilir. Yani her koordinat bir öncekine bağımlıdır (relative).

G91-komutu X ve Z bilgisi gerektirmez. G92 ya da G90-komutları ile değiştirilmediği sürece sistem artımlı sistem olarak kalır.

N11 G91 - - -

Bir CNC programı içerisinde G90, G91 ve G92 komutları her zaman her yerde kullanılabilir. Eğitim amaçlı CNC-torna

tezgahı ilk açılışında ya da mikrobilgisayar reset edildiğinde artımlı sistem seçilmiş olur, G91-komutunu yazmaya gerek kalmaz.

G94 - G95 - İLERLEME HIZI BİRİMİ

Bu komutlarla torna kaleminin ilerleme hızı mm/dk (inch/dk) ya da mm/devir (inch/devir) olarak seçilebilir. X Z ve F bilgisi gerektirmez. Değiştirilmediği sürece geçerli kalırlar. Tezgah ilk açılışında ya da mikrobilgisayar reset edildiğinde G94, yani mm/dk (inch/dk) sistemi geçerlidir.

N07 G95 - - - ;mm/devir seçilir.

N08 G94 - - - ;mm/dk seçilir.

Programın herhangi bir ya da birçok yerinde kullanılabilirler.

G94 seçilmiş ise;

hassasiyet 1 mm/dk ya da 1/10 inch/dk
geçerli F 2-499 mm/dk ya da 2-199 (0.2-19.9 inch/dk)

G95 seçilmiş ise;

hassasiyet 1/1000 mm/devir ya da 1/10000 inch/devir
geçerli F 2-499 (0.002-0.499 mm/devir) ya da 2-199
(0.0002-0.0199 inch/devir).

G24 - YARIÇAP PROGRAMLAMA SİSTEMİ

G90 ve G92-kodları kullanılarak kartezyen sisteme geçildiğinde X değerleri daima tam çap olarak veriliyordu. Bu ise ancak sıfır noktası parça ekseni üzerinde seçildiği zaman yararlı oluyordu. Eğer yarıçap programlama sistemi kullanılmak isteniyorsa yani X'in gerçek değeri yazılmak isteniyorsa programın en başında (N00) G24-komutu kullanılmalıdır. Böylece sıfır noktası herhangi bir yerde alındığında X

değerlerini iki ile çarpmaya gerek kalmaz. G24-komutu program içinde değiştirilemez yani, eğer yarıçap programlama sistemi seçilirse program içinde tekrar çap programlamaya geçilemez.

```
.  
.   
N00 G24  
N01 G92 (G90) - - -
```

Bu kodlar kullanılarak hazırlanılan örnek programlar ekler kısmındadır. Ayrıca DIN ve ISO standartlarının belirlediği kodlar tablo halinde İngilizce olarak yine ekler kısmında verilmiştir.

4. CNC-DE PARÇA İŞLEME ÖRNEKLERİ

Bu bölümde CNC eğitim programının ve CNC-tezgahının kullanılmasının öğretilmesini kolaylaştırıcı üç örnek program ve programlara ait parçaların şekilleri verilmiştir. Her programın açıklaması yapılmış, program satırları da şekil üzerinde işaretlenerek takip edilmesi kolaylaştırılmıştır. Her üç program da artımlı sistemde yazılmıştır.

4.1. CNC-Program Örneği-1 :KONİK TAPA

Şekil 4.1'deki 19-satırlık CNC-parça programının torna kalemine izlettiği yol ve işlem sonucunda ortaya çıkan parça şekil 4.2'de gösterilmiştir. Programın işlem basamakları şöyledir.

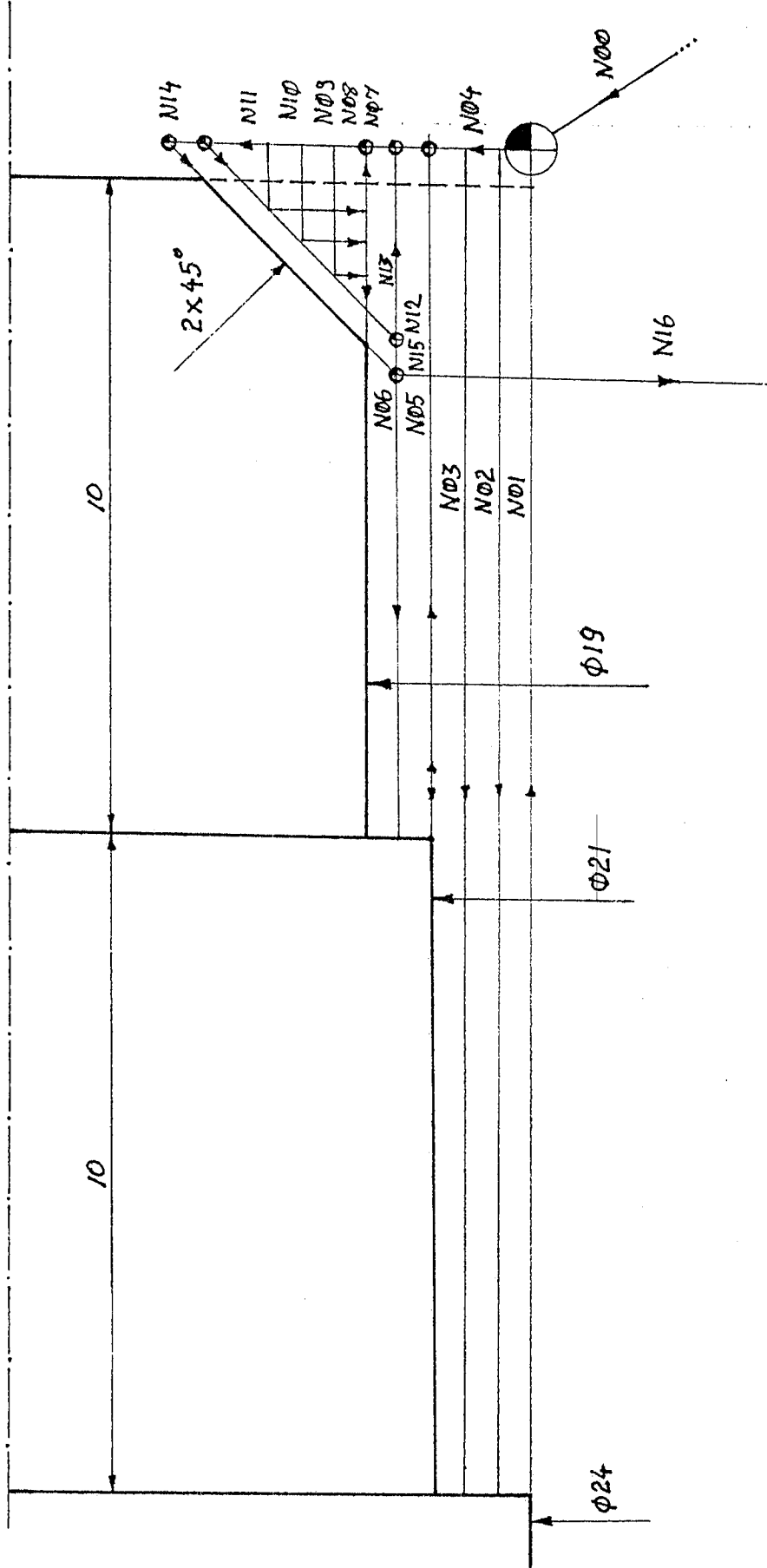
N00 satırında torna kalemi iş parçasına Z-yönündeki uzaklığı 0.5 mm. oluncaya kadar hızlı hareketle yaklaşmaktadır. 1, 2 ve 3. satırlarda birinci basamağı uç kademedeki G84 çevrim hareketi ile tornalamaktadır. Üçüncü işlemin sonunda kalem birinci hareketin başında bulunduğu yere gelir. İkinci kademenin daha hızlı tornalanması için N4 satırında kalem parçaya 1.5 mm. yaklaştırılır. N5 ve N6'da ikinci basamak tornalanır. N7'den N11'e kadar olan kısım konik işleme için hazırlık ve tornalamalardır. Şekil 4.2'de N8, N9 ve N10'daki hareketler ile konik tornalanacak kısım kaba olarak oluşturulur. Daha sonra parça ucundan aynaya doğru bir konik tornalama yapılır. Bu işlem N12 satırında G01 ile yapılmıştır. Aynı şekilde ikinci konik tornalama hareketi de N15'te yapılır.

N16 ve N17'de torna kalemi iş parçasından uzaklaşır ve program işletilmezden önceki yerine gelir. N18'de de G22 komutu ile program durdurulur.

N0	G00	X	-500	Z	-450	F		:TORNA KALEMI IS PARCA
N1	G84	X	-50	Z	-2050	F	30	:SINA YAKLASIYOR.
N2	G84	X	-100	Z	-2050	F	30	:ILK TORNALAMALAR.
N3	G84	X	-150	Z	-2050	F	30	:
N4	G00	X	-150	Z	0	F		:IKINCI BASAMAK ICIN
N5	G84	X	-50	Z	-1050	F	25	:YAKLASIR.
N6	G84	X	-100	Z	-1050	F	20	:
N7	G00	X	-100	Z	0	F		:KONIK TORNALAMA DN
N8	G84	X	-50	Z	-200	F	30	:HAZIRLIK LARI
N9	G84	X	-100	Z	-150	F	30	:
N10	G84	X	-150	Z	-100	F	30	:
N11	G00	X	-250	Z	0	F		:
N12	G01	X	300	Z	-300	F	30	:KONIK TORNALAMA
N13	G00	X	0	Z	300	F		:
N14	G00	X	-350	Z	0	F		:
N15	G01	X	350	Z	-350	F	20	:IKINCI KONIK TORNA
N16	G00	X	700	Z	0	F		:KALEM UZAKLASARAK
N17	G00	X	0	Z	800	F		:YERINE DONER
N18	G22	X		Z		F		:END

Sekil 4.1 Konik tapa parçasının DNC-programı

10 : 1



Şekil 4.2. Konik tapa parçası için yörengeler.

4.2 CNC-Program Örneği-2 :DAİRESEL TORNALAMA

Dairesel tornalama işlemini yapmadan önce tornalananca kışmın yaklaşık dairesel bir şekle getirilmesi için tornalama çevrimleri ve doğrusal hareket kodları kullanılmıştır. N0 satırında torna kalemı parçaya hızla yaklaşır. N1, N2 ve N3 satırlarında G84 çevrimi ile bir miktar tornalandıktan sonra N4 satırında kalem 1.5 mm. -X yönünde ilerler. Bu hareket daha sonraki uç tornalama çevriminde hız sağlamak için yapılır. Aynı amaçlı yaklaşma hareketi N8 ve N10'da da yapılır. N11'deki son çevrim hareketi bittikten sonra dairesel şekil yaklaşık olarak elde edilir. N12 ve N13'te dairesel hareketin başlangıç noktasına gelinir ve N14'te dairesel tornalama yapılır.

İç dairesel kısım ve konik yüzey için, bölge önce N16 dan N18'e kadar olan satırlarda G01 doğrusal tornalama komutları ile boşaltılır, N30 ve N31'de son tornalama ve N32 de iç dairesel yüzeyin tornalaması yapılır.

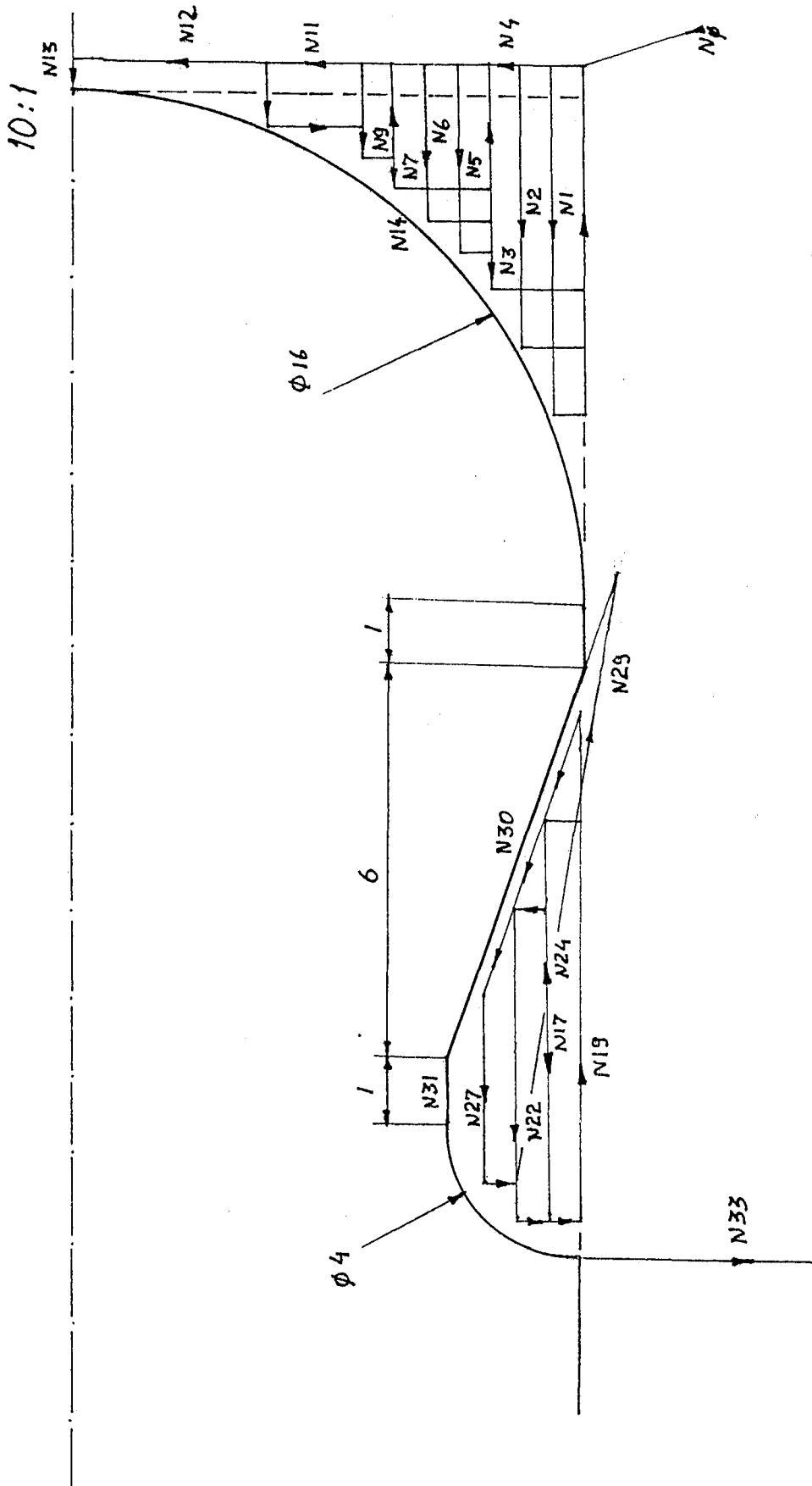
N33 ve N34'te kalem program işletilmezden önceki yerine döner.

```

N00 G00 X -500 Z -450 F ;KALEM PARÇAYA YAKLAŞIYOR
N01 G84 X -50 Z -550 F 40 ;DAİRESEL TORNALAMA İCİN
N02 G84 X -100 Z -450 F 40 ;N11'E KADAR HAZIRLIK
N03 G84 X -150 Z -350 F 40 ;YAPILIYOR
N04 G00 X -150 Z 0 F ;
N05 G84 X -50 Z -300 F 40 ;
N06 G84 X -100 Z -250 F 40 ;
N07 G84 X -150 Z -200 F 40 ;
N08 G00 X -150 Z 0 F ;
N09 G84 X -50 Z -150 F 40 ;
N10 G00 X -50 Z 0 F ;
N11 G84 X -150 Z -100 F 30 ;
N12 G00 X -450 Z 0 F ;
N13 G01 X 0 Z -50 F 40 ;KALEM PARÇAYA TEGET
N14 G03 X 800 Z F 20 ;DAİRESEL TORNALAMA
N15 G01 X 0 Z -150 F 30 ;KONİK KISIM İCİN
N16 G01 X -50 Z -150 F 40 ;HAZIRLIKLAR
N17 G01 X 0 Z -650 F 40 ;
N18 G01 X 50 Z 0 F 50 ;
N19 G00 X 0 Z 650 F ;
N20 G01 X -50 Z 0 F 60 ;İC DAİRE İCİN
N21 G01 X -50 Z -150 F 40 ;HAZIRLIKLAR
N22 G01 X 0 Z -500 F 40 ;
N23 G01 X 50 Z 0 F 50 ;
N24 G00 X 0 Z 500 F ;
N25 G01 X -50 Z 0 F 60 ;
N26 G01 X -50 Z -150 F 40 ;
N27 G01 X 0 Z -300 F 40 ;
N28 G01 X 50 Z 0 F 60 ;
N29 G00 X 200 Z 950 F ;KONİK KISMIN TORNA-
N30 G01 X -250 Z -750 F 20 ;LANMASI
N31 G01 X 0 Z -100 F 20 ;
N32 G02 X 200 Z F 20 ;İC DAİRENİN TORNALAN-
N33 G00 X 500 Z 0 F ;MASİ.KALEMİN PARÇADAN
N34 G00 X 0 Z 2300 F ;UZAKLAŞMASI.
N35 G22 X Z F ;END

```

Şekil 4.3 Dairesel tornalama örneği programı



Şekil 4.4 Dairesel tornalama örneği için yönlükler.

4.3 CNC-Program Örneği-3 :KING

King parçası dört konik yüzeyden meydana gelir. Konik yüzeylerin ve plato bölgelerininin yaklaşık koni haline getirilmesi için N1'den N5'e kadar olan satırlarda G84 ile boşaltma yapılmış N6'da koni oluşturulmuştur. Aynı işlem birinci koni için N8'den N13'e kadar, üçüncü ve dördüncü koniler için N17'den N22'ye kadar olan satırlarda da yapılmıştır.

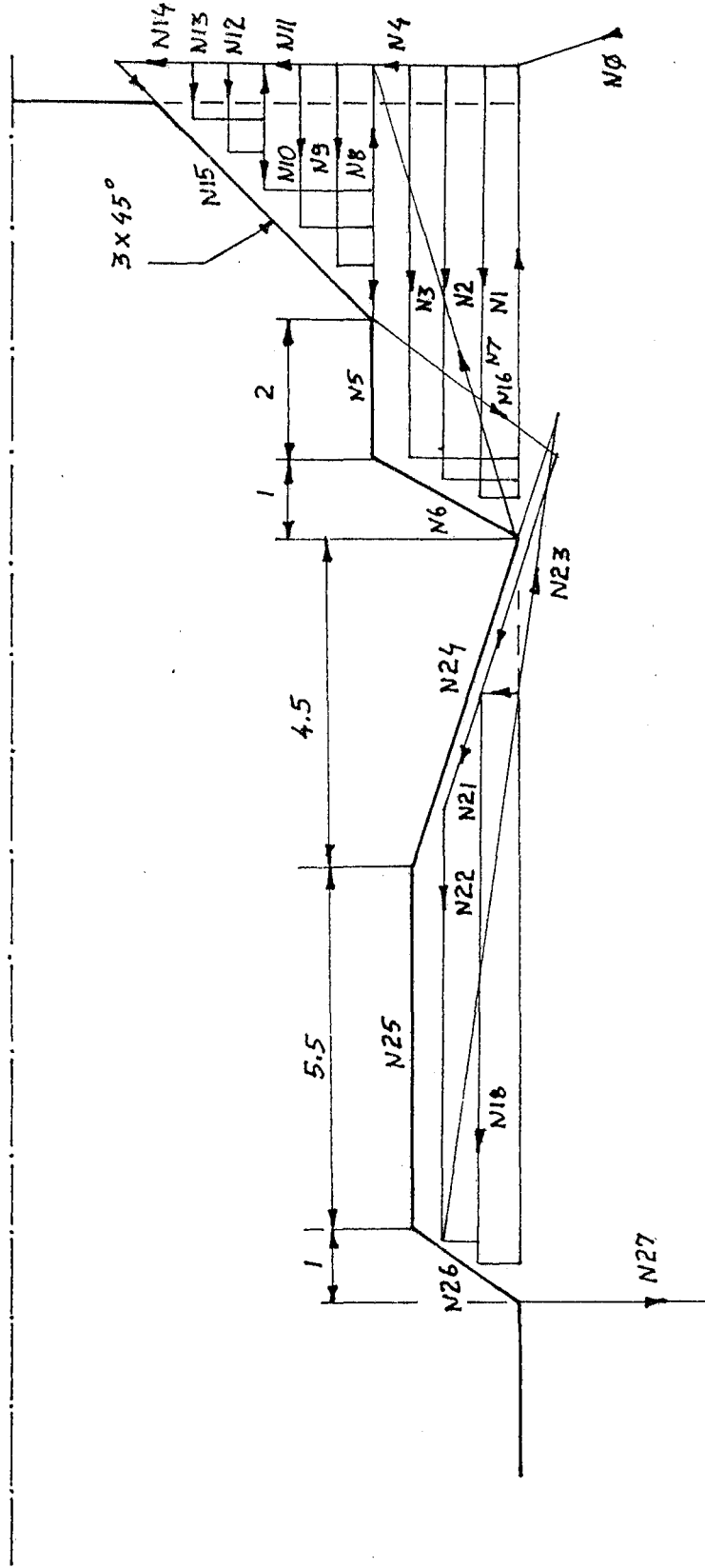
```

N00 G00 X -500 Z -450 F      ;ÜN KONI İÇİN HAZIRLIK
N01 G84 X  -50 Z  -600 F 40  ;TORNALAMALARI
N02 G84 X -100 Z  -580 F 40  ;
N03 G84 X -150 Z  -550 F 40  ;
N04 G00 X -200 Z    0 F      ;
N05 G01 X   0 Z  -550 F 30  ;
N06 G01 X 200 Z  -100 F 30  ;İKİNCİ KONİK YÜZEYİN
N07 G00 X -200 Z   650 F      ;TORNALANMASI
N08 G84 X  -50 Z  -280 F 30  ;
N09 G84 X -100 Z  -230 F 30  ;
N10 G84 X -150 Z  -180 F 30  ;
N11 G00 X -150 Z    0 F      ;
N12 G84 X  -50 Z  -130 F 30  ;
N13 G84 X -100 Z   -80 F 30  ;
N14 G00 X -200 Z    0 F      ;
N15 G01 X 350 Z  -350 F 30  ;BİRİNCİ KONİK YÜZEY
N16 G00 X 250 Z  -200 F      ;
N17 G01 X -100 Z  -300 F 40  ;ÜÇÜNCÜ KONİK YÜZEY İÇİN
N18 G01 X   0 Z  -850 F 40  ;HAZIRLIK TORNALAMALARI
N19 G00 X  50 Z   850 F      ;
N20 G01 X  -50 Z    0 F 40  ;
N21 G01 X  -50 Z  -150 F 40  ;
N22 G01 X   0 Z  -680 F 40  ;
N23 G00 X 150 Z 1180 F      ;
N24 G01 X -200 Z  -650 F 40  ;ÜÇÜNCÜ KONİK YÜZEY.
N25 G01 X   0 Z  -550 F 40  ;PLATO KISMI
N26 G01 X 150 Z  -100 F 40  ;DÖRDÜNCÜ KONİK YÜZEY
N27 G00 X 500 Z    0 F      ;KALEM PARÇADAN UZAKLASIR
N28 G00 X   0 Z 2200 F      ;..
N29 G22 X      Z      F      ;END

```

Sekil 4.5 King parçası CNC-programı.

10:1



Şekil 4.6 King parçası için yörüngeler

5. SONUÇ VE TARTIŞMA

Basit bir kişisel mikrobilgisayarda hazırlanan program CNC-tezgah eğitimi gören kişilere oldukça yararlı olabilir. Bu program sayesinde tezgahın ve kullanıcının çeşitli arıza ve hasarlardan korunması sağlanmıştır. Tezgahın olmadığı hallerde de kişisel bilgisayarda CNC-programlama dili üzerinde çalışmak mümkündür. Çünkü CNC-parça programı bilgisayarda yazılabilmekte ve tezgah olmaksızın grafik ekran üzerinde işletilebilmektedir. Bilgisayarın grafik ekranı üzerinde denenen ve doğru olduğuna karar verilen CNC-parça programı bir aradevre ve program aracılığı ile kolaylıkla CNC-torna tezgahına aktarılabilir. Böylece program hatalarından doğan ve programın CNC-torna tezgahına elle yazılması sırasında zaman kayıplarından tasarruf edilebilmektedir. Hazırlanan programlar, kişisel bilgisayarın üzerindeki manyetik disk ünitesi ile saklanabilmekte CNC-torna tezgahı üzerindeki manyetik band ünitesinin yaşlılığından ve yaptığı hatalardan korunulmuş olmaktadır.

Program daha da geliştirilerek tezgah kütüphanesinde olmayan komutlar da eklenebilir. Örneğin birkaç komutla yapılabilecek işlemler bir komut haline getirilebilir. Bu kısım, CNC-programı torna tezgahına aktarılırken tezgah kütüphanesinde bulunan komutlarla oluşturulur. Böyle komutlara örnek olarak pah kırma, iki kademeli son tornalama verilebilir.

Benzer bir program CNC-freze için de hazırlanabilir. Ancak bu durumda kullanılacak kişisel bilgisayarın grafik olanaklarının daha ileri düzeyde olması gerekmektedir. Çünkü CNC-freze tezgahının üç boyutlu parça işlemesi ancak nokta sayısı yüksek olan bir ekranda doyurucu ve yararlı olarak izlenebilir. Burada da, ekrandaki görüntünün üç boyutlu perspektif bir görünüş mü yoksa, parçanın üç boyutlu görünüşü mü daha yararlı olacağı da bir tartışma konusu olabilir. İster CNC-torna ister CNC-freze tezgahı olsun eğitim programının BASIC değil de makine dilinde hazırlanması

halinde daha hızlı olacağı beklenir.

CNC-torna tezgahını kullanacak olanlara programın faydalı olacağı umulmaktadır.

KAYNAKLAR

1. Compact 5-CNC el kitabları
EMCO, 1982-Hallein
2. AMSTRAD CPC-6128 personal computer el kitabı
1984-Essex
3. Z80-Assembly Language Programming
Lance A. Leventhal, 1979-Berkeley
4. Numerical Control and Computer-Aided Manufacturing
Roger S. Pressman, 1977-Newyork

KISALTMALAR

CNC : Computer Numeric Controlled
PIO : Peripheral Input Output
I/O : Input/Output
PIA : Peripheral Interface Adapter
FF : Flip-Flop
RS-FF : Set-Reset Flip-Flop
E : Enable
CA : Kontrol A
CB : Kontrol B
Dir : Direction
CS : Chip Select
Hex.: Hexadecimal number

CNC : Bilgisayar denetimli iş makinası
PIO : Z80 ailesinde giriş/çıkış birimi
I/O : Giriş/çıkış birimi
PIA : 6800 ailesinde giriş/çıkış birimi
FF : çıkışı iki konumlu bir bitlik yazaç
RS-FF : Set-Reset tipi FF
E : İzin verme işaret girişi
CA : A portunu denetleyen yazaç
CB : B portunu denetleyen yazaç
Dir : Yön belirten işaret girişi
CS : Tümleşik devreyi aktif hale getiren işaret girişi
Hex.: Onaltılık sayı sisteminde yazılmış bir sayı

EKLER

1. BASIC program listesi.
2. ZBO komut listesi.
3. Devre elemanlarının katalog bilgileri.
4. Standard CNC G ve M-kodları.

```

10 MODE 2:MEMORY C9FFF:LOAD "KEYASM.BIN":CALL CA000
20 NN=159:DX=100:DZ=400:mg=0.1
30 DIM G$(NN),X$(NN),Z$(NN),F$(NN),CM$(NN),A$(26)
40 VPS=1:GOSUB 3350
50 IF INL=1 THEN RETURN
60 PRINT TAB(2)"N"L;
70 LOCATE 7,VPS:INPUT "G",GI$
80 IF GI$="" THEN GI#=G$(L):LOCATE 8,VPS:PRINT G$(L);
90 IF GI$="" THEN GOTO 1180
100 IF GI$="00" THEN 300 ' HIZLI HAREKET
110 IF GI$="01" THEN 370 ' HIZI VERILEN DOGRUSAL HAREKET
120 IF GI$="02" THEN 440 ' SAAT YONUNUN TERSINE DAIRESEL HAR
EKET
130 IF GI$="84" THEN 580 ' EKSENLERE DIK DIKDORTGEN TORNALAM
A CEVRiMi
140 IF GI$="03" THEN 510 ' SAAT YONUNDE DAIRESEL HAREKET
150 IF GI$="20" THEN 590 ' BEKLEME
160 IF GI$="21" THEN 600 ' BOS SATIR
170 IF GI$="22" THEN 610 ' PROGRAM SONU
180 IF GI$="90" THEN 660 ' KARTEZYEN KOORDINAT SiSTEMi SECME
190 IF GI$="91" THEN 670 ' ARTIMLI SiSTEM SECME
200 IF GI$="92" THEN 680 ' KARTEZYEN KOORD. SiS. SECME VE YE
R BiLDiRME
210 IF GI$="33" THEN 690 ' DiS ACMA
220 IF GI$="78" THEN 700 ' DiS ACMA CEVRiMi
230 IF GI$="24" THEN 720 ' KARTEZYEN SiSTEMDE YARICAP PROG.
SECME
240 IF GI$="26" THEN 710 ' TOOL OFFSET
250 IF GI$="94" THEN 730 ' iLERLEME HIZI MM/DAKiKA
260 IF GI$="95" THEN 740 ' iLERLEME HIZI MM/DEViR
270 IF INL=1 THEN RETURN
280 IF GI$="0E" THEN LL=L:GOTO 1180
290 LOCATE 7,VPS:PRINT " " " ";GOTO 70
300 G$(L)="00"
310 GOSUB 950'GET X
320 GOSUB 1010'GET Z
330 LOCATE 36,VPS:PRINT"F"

```

```

340 GOSUB 1120'GET COMMENT
350 IF VPS<25 THEN VPS=VPS+1
360 L=L+1:GOTO 50
370 G$(L)="01"
380 GOSUB 950'GET X
390 GOSUB 1010'GET Z
400 GOSUB 1060'GET F
410 GOSUB 1120'GET COMMENT
420 IF VPS<25 THEN VPS=VPS+1
430 L=L+1:GOTO 50
440 G$(L)="02"
450 GOSUB 950'GET X
460 LOCATE 24.VPS:PRINT "Z"
470 GOSUB 1060'GET F
480 GOSUB 1120'GET COMMENT
490 IF VPS<25 THEN VPS=VPS+1
500 L=L+1 :GOTO 50
510 G$(L)="03"
520 GOSUB 950'GET X
530 GOSUB LOCATE 24.VPS:PRINT "Z"
540 GOSUB 1060'GET F
550 GOSUB 1120'GET COMMENT
560 IF VPS<25 THEN VPS=VPS+1
570 L=L+1:GOTO 50
580 G$(L)="84":GOTO 380
590 G$(L)="20":GOTO 620
600 G$(L)="21":GOTO 620
610 G$(L)="22"
620 LOCATE 14.VPS:PRINT"X"
630 GOSUB 1120'GET COMMENT
640 IF VPS<25 THEN VPS=VPS+1
650 L=L+1:GOTO 50
660 G$(L)="90":GOTO 620
670 G$(L)="91":GOTO 620
680 G$(L)="92":GOTO 310
690 G$(L)="33":LOCATE 14.VPS:PRINT "X"
700 G$(L)="78":GOTO 380

```

```

710 G$(L)="26":GOTO 310
720 G$(L)="24":GOTO 620
730 G$(L)="94":GOTO 620
740 G$(L)="95":GOTO 620
750 ' PROGRAM LISTESi ALMA
760 FOR L=KL TO NN
770 PRINT TAB(2)"N";TAB(3)L;TAB(8)"G"+G$(L);TAB(13)"X"+X$(L)
;TAB(21)"Z"+Z$(L);TAB(30)"F"+F$(L);TAB(39)";"CM$(L)
780 IF L=ENDL THEN RETURN
790 IF INKEY#="" THEN 810
800 NEXT L
810 FOR I=1 TO 20:NEXT
820 IK#=INKEY#
830 IF IK#="" THEN RETURN
840 IF IK#<>"" THEN KL=L+1:GOTO 760
850 GOTO 820
860 ' PRINTER CIKISI ALMA
870 IF CD<>0 THEN ENDL=CD
880 FOR L=0 TO NN
890 IF G$(L)="" THEN PRINT #8,:RETURN
900 PRINT #8,TAB(2)"N";TAB(3)L;TAB(8)"G"+G$(L);TAB(13)"X"+X#
(L);TAB(21)"Z"+Z$(L);TAB(30)"F"+F$(L);TAB(39)";"CM$(L)
910 IF L=ENDL THEN RETURN
920 NEXT L
930 RETURN
940 '99999999X GIRISI999999999
950 LOCATE 14,VPS:INPUT "X",XI#
960 IF XI#="" THEN LOCATE 15,VPS:PRINT X$(L);:RETURN
970 IF ABS(VAL(XI#))>5999 THEN LOCATE 14,VPS:PRINT"
";:GOTO 950
980 X$(L)=RIGHT$(" "+XI#,6)
990 RETURN
1000 '99999999Z GIRISI999999999
1010 LOCATE 24,VPS:INPUT "Z",ZI#
1020 IF ZI#="" THEN LOCATE 26,VPS:PRINT Z$(L);:RETURN
1030 IF ABS(VAL(ZI#))>39999 THEN LOCATE 25,VPS:PRINT"
";:GOTO 1010

```

```

1040 Z$(L)=RIGHT$("          "+ZI$,6)
1050 RETURN
1060 '99999999F GIRISI9999999999
1070 LOCATE 36,VPS:INPUT "F",FI$
1080 IF FI$="" THEN LOCATE 37,VPS:PRINT F$(L);:RETURN
1090 IF ABS(VAL(FI$))>499 THEN LOCATE 36,VPS:PRINT"
";:GOTO 1070
1100 F$(L)=RIGHT$("          "+FI$,3)
1110 RETURN
1120 '99999999ACIKLAMA GIRISI99
1130 LOCATE 43,VPS:INPUT ";",CI$
1140 IF CI$="" THEN LOCATE 44,VPS:PRINT CM$(L):RETURN
1150 IF LEN(CI$)>25 THEN LOCATE 43,VPS:LOCATE 69,VPS:PRINT"
";:GOTO 1130
1160 CM$(L)=CI$
1170 RETURN
1180 INPUT"",CW$
1190 INL=0
1200 CC#=LEFT$(UPPER$(CW$),2):CO=INT(VAL(MID$(CW$,3,2)))
1210 IF CC#="LS" THEN KL=CO:ENDL=NN:GOSUB 760:PRINT "Ok":GOT
O 1180
1220 IF CC#="LP" THEN KL=CO:ENDL=NN:GOSUB 860:PRINT"Ok":GOTO
1180
1230 IF CC#="RP" THEN PRINT: VPS=VPOS(#0):L=LL:GOTO 50
1240 IF CC#="ED" THEN L=CO:PRINT: VPS=VPOS(#0):GOTO 50
1250 IF CC#="CL" THEN CLS:PRINT"Ok":GOTO 1180
1260 IF CC#="TR" THEN GOSUB 1470:CALL CA000:PRINT "Ok":GOTO
1180
1270 IF CC#="IN" THEN GOSUB 1740:PRINT"Ok":GOTO 1180
1280 IF CC#="DL" THEN GOSUB 1830:PRINT "Ok":GOTO 1180
1290 IF CC#="ST" THEN GOSUB 1950:PRINT "Ok":GOTO 1180
1300 IF CC#="HC" THEN GOSUB 2000:PRINT "Ok":GOTO 1180
1310 IF CC#="DO" OR CC#="BT" OR CC#="DR" THEN GOSUB 2210:PRI
NT"Ok":GOTO 1180
1320 IF CC#="ER" THEN GOSUB 1430:PRINT"Ok":GOTO 1180
1330 IF CC#="SA" THEN UUSER,1:GOSUB 2640:UUSER,0:PRINT "Ok":
GOTO 1180

```

```

1340 IF CC#="LO" THEN UUSER,1:GOSUB 2790:UUSER,0:PRINT "Ok":
GOTO 1180
1350 IF CC#="CA" THEN UUSER,1:CAT:UUSER,0:PRINT "Ok":GOTO 11
80
1360 IF CC#="KL" THEN UUSER,1:GOSUB 2560:UUSER,0:PRINT "Ok":
GOTO 1180
1370 IF CC#="OR" THEN GOSUB 2130:PRINT "Ok":GOTO 1180
1380 IF CC#="HO" THEN GOSUB 3120:MODE 2:PRINT "Ok":GOTO 1180
1390 IF CC#="RN" THEN GOSUB 3590:MODE 2:PRINT"Ok":GOTO 1180
1400 IF CC#="" THEN 1180
1410 PRINT CW#+ "?"
1420 GOTO 1180
1430 ' PROGRAMI SiL
1440 ERASE G#,X#,Z#,F#,CM#
1450 DIM G$(NN),X$(NN),Z$(NN),F$(NN),CM$(NN)
1460 RETURN
1470 ' PROGRAMI CNC'YE TRANSFER ET
1480 CALL CA00C:KO=24:GOSUB 1690:KO=21:GOSUB 1690:GOSUB 1690
:KO=25:GOSUB 1690
1490 ' BLOKLARI GONDER
1500 FOR NL=STL TO ENDL
1510 TG#=G$(NL):KO=15:GOSUB 1690
1520 IF TG#="20" OR TG#="21" OR TG#="22" OR TG#="90" OR TG#="
91" OR TG#="94" OR TG#="24" OR TG#="95" THEN GOSUB 1580:GOT
O 1560
1530 IF TG#="02" OR TG#="03" THEN GOSUB 1580:TG#=X$(NL):GOSU
B 1580:TG#=F$(NL):GOSUB 1580:GOTO 1560
1540 IF TG#="00" OR TG#="92" THEN GOSUB 1580:TG#=X$(NL):GOSU
B 1580:TG#=Z$(NL):GOSUB 1580:GOTO 1560
1550 IF TG#="01" OR TG#="84" THEN GOSUB 1580:TG#=X$(NL):GOSU
B 1580:TG#=Z$(NL):GOSUB 1580:TG#=F$(NL):GOSUB 1580
1560 IF UPPER$(LEFT$(CM$(NL),3))="END" THEN RETURN
1570 NEXT NL
1580 ' GXZF CIKISI
1590 SG=0
1600 FOR D=1 TO LEN(TG#)
1610 NM#=MID$(TG#+ " ",D,1)

```

```

1620 IF NM$="" OR NM$=" " THEN 1650
1630 IF NM$="-" THEN SG=1:GOTO 1650
1640 IF VAL(NM$)<>0 OR NM$="0" THEN KO=VAL(NM$):GOSUB 1690
1650 NEXT D
1660 IF SG=1 THEN KO=10:GOSUB 1690 'EKSi iSARETi
1670 KO=11:GOSUB 1690 'iNF TUSU
1680 RETURN
1690 ' TUS YAZICI PROGRAM
1700 POKE CA200,KO
1710 CALL CA025
1720 FOR DLY=1 TO 150:NEXT DLY
1730 RETURN
1740 ' ARAYA BOS SATIR SOKMA
1750 GOSUB 1910
1760 FOR SHL=MXL TO CO STEP -1
1770 G$(SHL+1)=G$(SHL):X$(SHL+1)=X$(SHL):Z$(SHL+1)=Z$(SHL):F
$(SHL+1)=F$(SHL):CM$(SHL+1)=CM$(SHL)
1780 NEXT SHL
1790 G$(CO)="21":X$(CO)="":Z$(CO)="":F$(CO)="":CM$(CO)=" "
1800 L=CO:INL=1:VPS=VPDS(#0):GOSUB 60
1810 IF GI$="DE" THEN INL=0:GOSUB 1830:RETURN
1820 CO=CO+1:GOTO 1740
1830 ' BiR SATIR SiL VE SATIRLARI YENiDEN NUMARALA
1840 GOSUB 1910
1850 FOR SHL=CO TO MXL
1860 G$(SHL)=G$(SHL+1):X$(SHL)=X$(SHL+1):Z$(SHL)=Z$(SHL+1):F
$(SHL)=F$(SHL+1):CM$(SHL)=CM$(SHL+1)
1870 NEXT SHL
1880 G$(MXL)="":X$(MXL)="":Z$(MXL)="":F$(MXL)="":CM$(MXL)=" "
1890 MXL=MXL-1
1900 RETURN
1910 ' PROGRAMIN SONUNU BUL
1920 FOR MXL=0 TO NN-1
1930 IF UPPER$(LEFT$(CM$(MXL),3))="END" THEN RETURN
1940 NEXT MXL:RETURN
1950 ' PROGRAM CALISTIR
1960 CALL CA00C

```

```

1970 KO=21:GOSUB 1690:GOSUB 1690
1980 KO=22:GOSUB 1690:CALL CA000:RETURN
1990 ' EL iLE ( MANUAL ) KUMANDA ETME
2000 CALL CA000
2010 FOR I=1 TO 26:READ A$(I):NEXT I
2020 DATA 0,1,2,3,4,5,6,7,8,9,-,I,D,R,F,G,ö,P,/,L,;,H,S,A,Q,
W
2030 A$=INKEY$
2040 IF A$="E" THEN RESTORE :RETURN
2050 IF A$="" THEN 2030
2060 FOR J=1 TO 26
2070 IF A$=A$(J) THEN 2100
2080 NEXT J
2090 GOTO 2030
2100 POKE CA200,J-1
2110 CALL CA00C:CALL CA025:CALL CA000
2120 GOTO 2030
2130 ' ORiJiN BELiRLEME
2140 LOCATE 1,1:INPUT "X(0-399),Z(0-599) :",OX,OZ
2150 IF OX>399 OR OX<0 OR OZ>599 OR OZ<0 THEN LOCATE 1,1:PRI
NT SPACE$(60):GOTO 2140
2160 ORIGIN OZ,OX:PLOT 0,0,13
2170 PRINT "Y/N"
2180 IK$=INKEY$
2190 IF IK$="" THEN 2180
2200 IF IK$="N" OR IK$="n" THEN LOCATE 1,1:PRINT SPACE$(60):
GOTO 2140 ELSE RETURN
2210 ' PROGRAMI EKRAK UZERiNDE DENE
2220 ORIGIN OZ,OX:MAG=0.1:TL=0
2230 GOSUB 1910:ENDL=MXL:CLS:PLOT 0,0,13
2240 IF CC$<>"DR" THEN LOCATE 2,2:PRINT"N"TL:LOCATE 7,2:PRIN
T"G"+G$(TL):LOCATE 14,2:PRINT"X"+X$(TL):LOCATE 24,2:PRINT"Z"
+Z$(TL):LOCATE 33,2:PRINT"F"+F$(TL):LOCATE 38,2:PRINT";"+CM$
(TL)
2250 FOR TL=0 TO ENDL
2260 IF CC$="DR" THEN 2320
2270 IF CC$<>"BT" THEN 2290

```

```

2280 IF INKEY(47)<>0 THEN 2280
2290 LOCATE 1,1:PRINT SPACE$(160)
2300 LOCATE 2,1:PRINT"N"TL:LOCATE 7,1:PRINT"G"+G$(TL):LOCATE
  14,1:PRINT"X"+X$(TL):LOCATE 24,1:PRINT"Z"+Z$(TL):LOCATE 33,
  1:PRINT"F"+F$(TL):LOCATE 38,1:PRINT";"+CM$(TL)
2310 LOCATE 2,2:PRINT"N"TL+1:LOCATE 7,2:PRINT"G"+G$(TL+1):LO
  CATE 14,2:PRINT"X"+X$(TL+1):LOCATE 24,2:PRINT"Z"+Z$(TL+1):LO
  CATE 33,2:PRINT"F"+F$(TL+1):LOCATE 38,2:PRINT";"+CM$(TL+1)
2320 XD=VAL(X$(TL)):ZD=VAL(Z$(TL)):FD=VAL(F$(TL))
2330 IF G$(TL)="00" OR G$(TL)="01" THEN GOSUB 2400
2340 IF G$(TL)="84" THEN GOSUB 2430
2350 IF G$(TL)="02" THEN GOSUB 2480
2360 IF G$(TL)="03" THEN GOSUB 2520
2370 IF G$(TL)="33" THEN GOSUB 2950
2380 NEXT TL:RETURN
2390 FOR DL=1 TO 300:NEXT DL:RETURN
2400 ' 600,601 DENEMELERi
2410 DRAWR MAG*ZD,-MAG*XD,13
2420 RETURN
2430 ' 684 DENEMELERi
2440 DRAWR 0,-MAG*XD,13:GOSUB 2390
2450 DRAWR MAG*ZD,0,13 :GOSUB 2390: DRAWR 0,MAG*XD,13:GOSUB
  2390
2460 DRAWR -MAG*ZD,0,13
2470 RETURN
2480 ' 602 DENEMELERi
2490 DEG:RD=XD*MAG
2500 IF RD>0 THEN :MOVER 0,-RD:FOR ACI=90 TO 180 :XR=RD*COS(
  ACI):YR=RD*SIN(ACI):PLOT R XR,YR,13:MOVER -XR,-YR:NEXT ACI:MO
  VER -RD,0:RETURN
2510 MOVER RD,0:FOR ACI=0 TO 90:XR=-RD*COS(ACI):YR=-RD*SIN(A
  CI):PLOT R XR,YR,13:MOVER -XR,-YR:NEXT ACI:MOVER 0,-RD:RETURN
2520 ' 603 DENEMELERi
2530 DEG:RD=XD*MAG
2540 IF RD>0 THEN:MOVER -RD,0:FOR ACI=0 TO 90:XR=RD*COS(ACI)
  :YR=-RD*SIN(ACI):PLOT R XR,YR,13:MOVER -XR,-YR:NEXT ACI:MOVER
  0,-RD:RETURN

```

```

2550 MOVER 0,-RD:FOR ACI=0 TO 90:XR=RD*SIN(ACI):YR=RD*COS(AC
I):PLOTR XR,YR,13:MOVER -XR,-YR:NEXT ACI:MOVER RD,0:RETURN
2560 ' PROGRAMI DiSKETTEN SiL
2570 INPUT "PROGRAM iSMi:",FILENAME#
2580 IF LEN(FILENAME#)>8 THEN 2650
2590 IF FILENAME#="Q" THEN PRINT "PROGRAM SiLiNMEdi":RETURN
2600 FOR CHF=1 TO 8:IF MID$(FILENAME#,CHF,1)="." THEN PRINT
"PROGRAM iSMi GECERSiZ.!":GOTO 2650:NEXT CHF
2610 UERA,FILENAME#+".CNC"
2620 PRINT FILENAME#+".CNC SiLiNdi"
2630 RETURN
2640 ' BiLgi DOSYASI OLARAK DiSKETE SAKLAMA
2650 INPUT "PROGRAM iSMi:",FILENAME#
2660 FILENAME#=LEFT$(FILENAME#+",8)
2670 FOR CHF=1 TO 8
2680 IF MID$(FILENAME#,CHF,1)="." THEN PRINT "PROGRAM iSMi G
ECERSiZ.!":GOTO 2650
2690 NEXT CHF
2700 PRINT "SAKLAMA.! LUTFEN BEKLEYiNiZ"
2710 OPENOUT FILENAME#+".CNC"
2720 GOSUB 1910
2730 FOR SL=0 TO MXL
2740 WRITE #9,G$(SL),X$(SL),Z$(SL),F$(SL),CM$(SL)
2750 NEXT SL
2760 PRINT "PROGRAM ";FILENAME#;".CNC ADI ALTINDA SAKLANDI"
2770 PRINT "PROGRAM UZUNLUGU ";SL
2780 CLOSEOUT:RETURN
2790 ' BiR CNC PROGRAMI YUKLEME
2800 INPUT "PROGRAM iSMi :",FILENAME#
2810 FILENAME#=LEFT$(FILENAME#+",8)
2820 FOR CHF=1 TO 8
2830 IF MID$(FILENAME#,CHF,1)="." THEN PRINT "PROGRAM iSMi G
ECERSiZ.!":GOTO 2800
2840 NEXT CHF
2850 PRINT "YUKLEME.! LUTFEN BEKLEYiNiZ"
2860 OPENIN FILENAME#+".CNC"
2870 FOR LL=0 TO NN

```

```

2880 INPUT #9, "", G$(LL), X$(LL), Z$(LL), F$(LL), CM$(LL)
2890 IF UPPER$(LEFT$(CM$(LL), 3))="END" THEN 2920
2900 NEXT LL
2910 PRINT FILENAME$+".CNC PROGRAMINDA END KOMUTU YOK"
2920 PRINT FILENAME$+".CNC YUKLENDİ"
2930 CLOSEIN
2940 RETURN
2950 ' G33 DENEMELERİ
2960   XD=YPOS
2970 IF FD>ABS(ZD) THEN 2400
2980 PIT=FD*MAG
2990 IF ZD<0 THEN GOTO 3060
3000 ZM=ZD*MAG-PIT
3010 FOR CD=0 TO ZM STEP PIT
3020 DRAWR PIT/2, -2*PIT/3, 13: DRAWR PIT/2, 2*PIT/3, 13
3030 NEXT CD
3040 DRAWR MAG*ZD-CD, XD-YPOS, 13
3050 RETURN
3060 ZM=ZD*MAG+PIT
3070 FOR CD=0 TO ZM STEP -PIT
3080 DRAWR -PIT/2, -2*PIT/3, 13: DRAWR -PIT/2, 2*PIT/3, 13
3090 NEXT CD
3100 DRAWR MAG*ZD-CD, XD-YPOS, 13
3110 RETURN
3120 ' EL İLE KUMANDA
3130 GOSUB 3230: GOSUB 3490
3140 GOSUB 3400
3150 A$=INKEY$
3160 IF A$="Q" AND EXR=1 THEN EXR=0: GOTO 3620
3170 IF A$="C" OR a$="c" THEN RETURN
3180 IF A$=CHR$(240) THEN ty=ty+2: GOSUB 3560
3190 IF A$=CHR$(241) THEN ty=ty-2: GOSUB 3580
3200 IF A$=CHR$(242) THEN tx=tx-2: GOSUB 3520
3210 IF A$=CHR$(243) THEN tx=tx+2: GOSUB 3540
3220 GOTO 3140
3230 MODE 1
3240 LOCATE 1, 1: INPUT "R="; R

```

```

3250 LOCATE 1,1:INPUT "L=";L
3260 CLS
3270 wx=100:wy=200-R/2:wym=wy+R
3280 PLOT wx,wy,13:DRAW wx+L,wy
3290 DRAW wx+L,wy+R:DRAW wx,wy+R
3300 DRAW wx,wy:MOVE wx+2,wy+2:FILL 13
3310 PLOT wx,40,3:DRAW wx,360:DRAW wx-60,360:DRAW wx-60,40:DR
RAW wx,40:MOVE wx-2,42:FILL 3
3320 PLOT wx+50,wy-10,3:DRAW wx,wy-50:DRAW wx,wy+R+50:DRAW w
x+50,wy+R+10:DRAW wx+50,wy-10:MOVE wx+2,wy:FILL 3
3330 PLOT wx+2,wy-30,0:DRAW wx+2,wy+R+30
3340 RETURN
3350 STE=20:m1f=1:m1d=INT(ste/3)+1:m1e=INT(ste/1.5)+1
3360 DIM s(STE,2)
3370 DEG:FOR i=1 TO STE:s(i,1)=200+160*COS(aci):aci=aci+180/
STE:NEXT i
3380 FOR i=2 TO STE:s(i,2)=s(i-1,1):NEXT i:s(1,2)=42
3390 RETURN
3400 m1f=m1f+1:IF m1f=STE+1 THEN m1f=1
3410 m1c=m1f:GOSUB 3460
3420 m1d=m1d+1:IF m1d=ste+1 THEN m1d=1
3430 m1c=m1d:GOSUB 3460
3440 m1e=m1e+1:IF m1e=ste+1 THEN m1e=1
3450 m1c=m1e:GOSUB 3460
3460 lc=s(m1c,1):ec=s(m1c,2)
3470 PLOT 96,lc,13:DRAWR -44,0:PLOT 96,ec,3:DRAWR -44,0
3480 RETURN
3490 tx=MG*1520+L:ty=wy-MG*500
3500 PLOT tx,ty,3:DRAWR 0,-30:DRAWR 26,-16:DRAWR 0,30:DRAWR
-26,16:MOVER 2,-6:FILL 3:RETURN
3510 ' -Z HAREKET
3520 PLOT tx,ty,3:DRAWR 0,-30:DRAWR 26,-16:PLOTR 2,0,0:DRAWR
0,30:DRAWR -26,16:PLOTR 2,-1:DRAWR 25,-15:PLOT tx,wym+2,0:DR
RAW tx,wym+wy-ty:RETURN
3530 ' +Z HAREKET
3540 PLOT tx,ty,3:DRAWR 26,-16:DRAWR 0,-30:PLOTR -2,0,0:DRAW

```

```

R -26,16:DRAWR 0,30:PLOT tx,wym+2,0:DRAW tx,wym+wym-ty:DRAWR
26,16:RETURN
3550 ' -X HAREKET
3560 PLOT tx,ty,3:DRAWR 26,-16:PLOTR 0,-1:DRAWR -24,16:PLOTR
24,-47,0:DRAWR -26,16:PLOT tx,wym+2,0:DRAW tx,wym+wym-ty:DRA
WR 26,16:LOCATE 22,15:PRINT TY;" ":RETURN
3570 ' +X HAREKET
3580 PLOT tx,ty+2,0:DRAWR 26,-16:PLOTR 0,-32,3:DRAWR -26,16:
PLOT tx,wym+2,0:DRAW tx,wym+wym-ty:LOCATE 22,15:PRINT TY;" ":
RETURN
3590 ' PROGRAMI CALISTIRMA
3600 GOSUB 3230:GOSUB 3490
3610 FOR LR=0 TO NN
3620 LOCATE 1,1:PRINT SPACE$(70):LOCATE 1,1:PRINT TAB(2)"N";
TAB(3)LR;TAB(8)"G"+G$(LR);TAB(13)"X"+X$(LR);TAB(21)"Z"+Z$(LR
);TAB(30)"F"+F$(LR);TAB(39)";"CM$(LR)
3630 IF INKEY#="Q" THEN EXR=1:GOTO 3140
3640 IF G$(LR)="22" OR CM$(LR)="END" THEN 3770
3650 IF G$(LR)="00" OR G$(LR)="01" THEN GOSUB 3780:GOTO 3760
3660 IF G$(LR)="84" THEN GOSUB 4180:GOTO 3760
3670 IF G$(LR)="02" THEN GOSUB 4330:GOTO 3760
3680 IF G$(LR)="33" THEN GOSUB 3780 :GOTO 3760
3690 IF G$(LR)="03" THEN GOSUB 4460:GOTO 3760
3700 IF G$(LR)="20" THEN GOSUB 4590:LOCATE 1,25:PRINT SPACE$
(36):GOTO 3760
3710 IF G$(LR)="21" THEN GOSUB 4590:LOCATE 1,25:PRINT SPACE$
(36):GOTO 3760
3720 IF G$(LR)="90" OR G$(LR)="92" OR G$(LR)="91" OR G$(LR)=
"24" OR G$(LR)="26" OR G$(LR)="94" OR G$(LR)="95" THEN GOSUB
4630:LOCATE 1,25:PRINT SPACE$(36):GOTO 3760
3730 IF G$(LR)="78" THEN GOSUB 4180:GOTO 3760
3740 GOSUB 4680:LOCATE 1,25:PRINT SPACE$(34)
3750 IF DV=1 THEN DV=0:RETURN
3760 NEXT LR
3770 GOTO 3140
3780 ' DOGRUSAL HAREKETLER 600 601
3790 XR=MG*VAL(X$(LR)):ZR=MG*VAL(Z$(LR)):SX=TX+ZR:SY=TY-XR
3800 IF ABS(XR)>ABS(ZR) THEN GOSUB 3820 ELSE GOSUB 3880

```

```
3810 TX=SX:TY=SY:RETURN
3820 ' UXRÜ>UZRÜ
3830 DNM=2*ABS(ZR/XR)
3840 IF XR>=0 AND ZR<0 THEN 3940
3850 IF XR>0 AND ZR>=0 THEN 3970
3860 IF XR<0 AND ZR<=0 THEN 4000
3870 IF XR<=0 AND ZR>0 THEN 4030
3880 ' UZRÜ>UXRÜ
3890 DMM=2*ABS(XR/ZR)
3900 IF ZR<=0 AND XR<0 THEN 4060
3910 IF ZR<0 AND XR>=0 THEN 4090
3920 IF ZR>0 AND XR<=0 THEN 4120
3930 IF ZR>=0 AND XR>0 THEN 4150
3940 TY=TY-2:GOSUB 3570:TX=TX-DNM:GOSUB 3510:GOSUB 3400
3950 XR=XR-2
3960 IF XR>0 THEN 3940 ELSE RETURN
3970 TY=TY-2:GOSUB 3570:TX=TX+DNM:GOSUB 3530:GOSUB 3400
3980 XR=XR-2
3990 IF XR>0 THEN 3970 ELSE RETURN
4000 TY=TY+2:GOSUB 3550:TX=TX-DNM:GOSUB 3510:GOSUB 3400
4010 XR=XR+2
4020 IF XR<0 THEN 4000 ELSE RETURN
4030 TY=TY+2:GOSUB 3550:TX=TX+DNM:GOSUB 3530:GOSUB 3400
4040 XR=XR+2
4050 IF XR<0 THEN 4030 ELSE RETURN
4060 TX=TX-2:GOSUB 3510:TY=TY+DMM:GOSUB 3550:GOSUB 3400
4070 ZR=ZR+2
4080 IF ZR<0 THEN 4060 ELSE RETURN
4090 TX=TX-2:GOSUB 3510:TY=TY-DMM:GOSUB 3570:GOSUB 3400
4100 ZR=ZR+2
4110 IF ZR<0 THEN 4090 ELSE RETURN
4120 TX=TX+2:GOSUB 3530:TY=TY+DMM:GOSUB 3550:GOSUB 3400
4130 ZR=ZR-2
4140 IF ZR>0 THEN 4120 ELSE RETURN
4150 TX=TX+2:GOSUB 3530:TY=TY-DMM:GOSUB 3570:GOSUB 3400
4160 ZR=ZR-2
4170 IF ZR>0 THEN 4150 ELSE RETURN
```

```

4180 ' EKSENLERE DiK TORNALAMA CEVRiMi ' GB4
4190 XR=MG*VAL(X$(LR)):ZR=MG*VAL(Z$(LR))
4200 IF XR>0 AND ZR>0 THEN GOSUB 4280:GOSUB 4320:GOSUB 4260:
GOTO 4300
4210 IF XR<0 AND ZR>0 THEN XR=-XR:GOSUB 4260:GOSUB 4320:GOSU
B 4280:GOTO 4300
4220 IF XR>0 AND ZR<0 THEN ZR=-ZR:GOSUB 4280:GOSUB 4300:GOSU
B 4260:GOTO 4320
4230 IF XR<0 AND ZR<0 THEN XR=-XR:ZR=-ZR:GOSUB 4260:GOSUB 43
00:GOSUB 4280:GOTO 4320
4240 RETURN
4250 ' UP
4260 FOR CR=0 TO XR/2:TY=TY+2:GOSUB 3560:GOSUB 3400:NEXT CR:
RETURN
4270 ' DOWN
4280 FOR CR=0 TO XR/2:TY=TY-2:GOSUB 3580:GOSUB 3400:NEXT CR:
RETURN
4290 ' LEFT
4300 FOR CR=0 TO ZR/2:TX=TX-2:GOSUB 3520:GOSUB 3400:NEXT CR:
RETURN
4310 ' RIGHT
4320 FOR CR=0 TO ZR/2:TX=TX+2:GOSUB 3540:GOSUB 3400:NEXT CR:
RETURN
4330 ' SAATiN TERS YONUNDE DAiRESEL TORNALAMA HAREKETi 602
4340 RC=MG*VAL(X$(LR)):SBT=100*MG
4350 IF RC<0 THEN 4380
4360 IF RC>0 THEN 4420
4370 RETURN
4380 GX=TX+RC:GY=TY
4390 FOR FI=92 TO 0 STEP SBT/RC
4400 TX=GX-RC*SIN(FI):GOSUB 3520:TY=GY-RC*COS(FI):GOSUB 3560
:GOSUB 3400
4410 NEXT FI:TX=GX:TY=GY-RC:RETURN
4420 GX=TX:GY=TY-RC
4430 FOR FI=0 TO 92 STEP SBT/RC
4440 TX=GX-RC*SIN(FI):GOSUB 3520:TY=GY+RC*COS(FI):GOSUB 3580
:GOSUB 3400

```

```
4450 NEXT FI:TX=GX-RC:TY=GY:RETURN
4460 ' SAAT YONUNDE DAİRESEL TORNALAMA HAREKETİ 603
4470 RC=MG*VAL(X$(LR)):SBT=100*MG
4480 IF RC>0 THEN 4510
4490 IF RC<0 THEN 4550
4500 RETURN
4510 GX=TX-RC:GY=TY
4520 FOR FI=0 TO 90 STEP SBT/RC
4530 TX=GX+RC*COS(FI):GOSUB 3520:TY=GY-RC*SIN(FI):GOSUB 3580
:GOSUB 3400
4540 NEXT FI:TX=GX:TY=GY-RC:RETURN
4550 GX=TX:GY=TY-RC
4560 FOR FI=90 TO -2 STEP SBT/RC
4570 TX=GX+RC*COS(FI):GOSUB 3520:TY=GY+RC*SIN(FI):GOSUB 3560
:GOSUB 3400
4580 NEXT FI:TX=GX+RC:TY=GY:RETURN
4590 ' PROGRAMI BEKLETME 620
4600 LOCATE 1,25:PRINT "PROGRAM BEKLİYOR"
4610 IF INKEY$="C" THEN RETURN
4620 GOSUB 3400:GOTO 4610
4630 ' EXTRA G-KOMUTLARI
4640 LOCATE 1,25:PRINT "G";G$(LR);" KOMUTU DENEMEK İCİN GECE
RSİZDİR "
4650 GOSUB 3400
4660 IF INKEY$="C" THEN RETURN ELSE GOTO 4650
4670 ' GECERSİZ G-KOMUTLARI
4680 LOCATE 1,25:PRINT"GECERSİZ G-KOMUTU C/Q"
4690 GOSUB 3400
4700 IF INKEY$="C" THEN DV=0:RETURN
4710 IF INKEY$="Q" THEN DV=1:RETURN
4720 GOTO 4680
10000 PRINT #8,CHR$(C1B);"A";CHR$(1B)
```

INSTRUCTION SET

The following is a summary of the Z80 instruction set showing the assembly language mnemonic and the symbolic operation performed by the instruction. A more detailed listing appears in the Z80-CPU technical manual. The instructions are divided into the following categories:

- 8-bit loads Miscellaneous Group
- 16-bit loads Rotates and Shifts
- Exchanges Bit Set, Reset and Test
- Memory Block Input and Output
- Moves Jumps
- Memory Block Calls
- Searches Restarts
- 8-bit arithmetic Returns
- and logic
- 16-bit arithmetic
- General purpose Accumulator &
- Flag Operations

In the table the following terminology is used.

- b ≡ a bit number in any 8-bit register or memory location
- cc ≡ flag condition code
 - NZ ≡ non zero
 - Z ≡ zero
 - NC ≡ non carry
 - C ≡ carry
 - PO ≡ Parity odd or no over flow
 - PE ≡ Parity even or over flow
 - P ≡ Positive
 - M ≡ Negative (minus)

- d ≡ any 8-bit destination register or memory location
- dd ≡ any 16-bit destination register or memory location
- e ≡ 8-bit signed 2's complement displacement used in relative jumps and indexed addressing
- L ≡ 8 special call locations in page zero. In decimal notation these are 0, 8, 16, 24, 32, 40, 48 and 56
- n ≡ any 8-bit binary number
- nn ≡ any 16-bit binary number
- r ≡ any 8-bit general purpose register (A,B,C, D,E,H, or L)
- s ≡ any 8 bit source register or memory location
- s_L ≡ a bit in a specific 8-bit register or memory location
- ss ≡ any 16-bit source register or memory location
- subscript "L" ≡ the low order 8 bits of a 16-bit register
- subscript "H" ≡ the high order 8 bits of a 16-bit register
- () ≡ the contents within the () are to be used as a pointer to a memory location or I/O port number
- 8-bit registers are A, B, C, D, E, H, L, I and R
- 16-bit register pairs are AF, BC, DE and HL
- 16-bit registers are SP, PC, IX and IY

Addressing Modes implemented include combinations of the following:

Immediate	Indexed
Immediate extended	Register
Modified Page Zero	Implied
Relative	Register Indirect
Extended	Bit

	Mnemonic	Symbolic Operation	Comments
8-BIT LOADS	LD r, s	r ← s	s ≡ r, n. (HL), (IX+e), (IY+e)
	LD d, r	d ← r	d ≡ (HL), r (IX+e), (IY+e)
	LD d, n	d ← n	d ≡ (HL), (IX+e), (IY+e)
	LD A, s	A ← s	s ≡ (BC), (DE), (nn), I, R
	LD d, A	d ← A	d ≡ (BC), (DE), (nn), I, R
16-BIT LOADS	LD dd, nn	dd ← nn	dd ≡ BC, DE, HL, SP, IX, IY
	LD dd, (nn)	dd ← (nn)	dd ≡ BC, DE, HL, SP, IX, IY
	LD (nn), ss	(nn) ← ss	ss ≡ BC, DE, HL, SP, IX, IY
	LD SP, ss	SP ← ss	ss = HL, IX, IY
	PUSH ss	(SP-1) ← ss _H ; (SP-2) ← ss _L	ss = BC, DE, HL, AF, IX, IY
POP dd	dd _L ← (SP); dd _H ← (SP+1)	dd = BC, DE, HL, AF, IX, IY	

	Mnemonic	Symbolic Operation	Comments
EXCHANGES	EX DE, HL	DE ↔ HL	
	EX AF, AF'	AF ↔ AF'	
	EXX	$\begin{pmatrix} BC \\ DE \\ HL \end{pmatrix} \leftrightarrow \begin{pmatrix} BC' \\ DE' \\ HL' \end{pmatrix}$	
	EX (SP), ss	(SP) ↔ ss _L ; (SP+1) ↔ ss _H	ss ≡ HL, IX, IY
MEMORY BLOCK MOVES	LDI	(DE) ← (HL), DE ← DE+1 HL ← HL+1, BC ← BC-1	
	LDIR	(DE) ← (HL), DE ← DE+1 HL ← HL+1, BC ← BC-1 Repeat until BC = 0	
	LDD	(DE) ← (HL), DE ← DE-1 HL ← HL-1, BC ← BC-1	
	LDDR	(DE) ← (HL), DE ← DE-1 HL ← HL-1, BC ← BC-1 Repeat until BC = 0	

Cont'd on next page

17. INSTRUCTION SETS

IN DRAWING NUMBER SEQUENCE

65 Cont'd

IS65 Cont'd

Mnemonic	Symbolic Operation	Comments
MEMORY BLOCK SEARCHES		
CP1	A ← (HL), HL ← HL+1 BC ← BC-1	
CP1R	A ← (HL), HL ← HL+1 BC ← BC-1, Repeat until BC = 0 or A = (HL)	A ← (HL) sets the flags only. A is not affected
CPD	A ← (HL), HL ← HL-1 BC ← BC-1	
CPDR	A ← (HL), HL ← HL-1 BC ← BC-1, Repeat until BC = 0 or A = (HL)	
8-BIT ALU		
ADD s	A ← A + s	
ADC s	A ← A + s + CY	CY is the carry flag
SUB s	A ← A - s	
SBC s	A ← A - s - CY	s ≡ r, n, (HL) (IX+e), (IY+e)
AND s	A ← A ∧ s	
OR s	A ← A ∨ s	
XOR s	A ← A ⊕ s	
CP s	A - s	s = r, n (HL) (IX+e), (IY+e)
INC d	d ← d + 1	
DEC d	d ← d - 1	d = r, (HL) (IX+e), (IY+e)
16-BIT ARITHMETIC		
ADD HL, ss	HL ← HL + ss	} ss ≡ BC, DE, HL, SP
ADC HL, ss	HL ← HL + ss + CY	
SBC HL, ss	HL ← HL - ss - CY	
ADD IX, ss	IX ← IX + ss	ss ≡ BC, DE, IX, SP
ADD IY, ss	IY ← IY + ss	ss ≡ BC, DE, IY, SP
INC dd	dd ← dd + 1	dd ≡ BC, DE, HL, SP, IX, IY
DEC dd	dd ← dd - 1	dd ≡ BC, DE, HL, SP, IX, IY
CP ACC. & FLAG		
DAA	Converts A contents into packed BCD following add or subtract.	Operands must be in packed BCD format
CPL	A ← \overline{A}	
NEG	A ← $\overline{A} + 1$	
CCF	CY ← \overline{CY}	
SCF	CY ← 1	
MISCELLANEOUS GROUP		
NOP	No operation	
HALT	Halts CPU	
DI	Disable Interrupts	
EI	Enable Interrupts	
IM0	Set interrupt mode 0	8080A mode
IM1	Set interrupt mode 1	Call to 0038H
IM2	Set interrupt mode 2	Indirect Call

Mnemonic	Symbolic Operation	Comments
ROTATES AND SHIFTS		
RLC s		
RL s		
RRC s		
RR s		
SLA s		s ≡ r, (HL) (IX+e), (IY+e)
SRA s		
SRL s		
RLD		
RRD		
BIT S, R, & T		
BIT b, s	Z ← s _b	Z is zero flag
SET b, s	s _b ← 1	s ≡ r, (HL) (IX+e), (IY+e)
RES b, s	s _b ← 0	s ≡ r, (HL) (IX+e), (IY+e)
INPUT AND OUTPUT		
IN A (n)	A ← (n)	
IN r, (C)	r ← (C)	
INI	(HL) ← (C), HL ← HL + 1 B ← B - 1	Set flags
INIR	(HL) ← (C), HL ← HL + 1 B ← B - 1 Repeat until B = 0	
IND	(HL) ← (C), HL ← HL - 1 B ← B - 1	
INDR	(HL) ← (C), HL ← HL - 1 B ← B - 1 Repeat until B = 0	
OUT(n), A	(n) ← A	
OUT(C), r	(C) ← r	
OUTI	(C) ← (HL), HL ← HL + 1 B ← B - 1	
OUTIR	(C) ← (HL), HL ← HL + 1 B ← B - 1 Repeat until B = 0	
OUTD	(C) ← (HL), HL ← HL - 1 B ← B - 1	
OUTDR	(C) ← (HL), HL ← HL - 1 B ← B - 1 Repeat until B = 0	

Cont'd on next page

17. INSTRUCTION SETS

IN DRAWING NUMBER
SEQUENCE

IS65 Cont'd

IS65 Cont'd

JUMPS	JP nn	PC ← nn	cc { NZ PO Z PE NC P C M kk { NZ NC Z C ss = HL, IX, IY
	JP cc, nn	If condition cc is true PC ← nn, else continue	
	JR e	PC ← PC + e	
	JR kk, e	If condition kk is true PC ← PC + e, else continue	
	JP (ss)	PC ← ss	
DJNZ e	B ← B - 1, if B = 0 continue, else PC ← PC + e		
CALLS	CALL nn	(SP-1) ← PC _H (SP-2) ← PC _L , PC ← nn	
	CALL cc, nn	If condition cc is false continue, else same as CALL nn	

RETURNS	RST L	(SP-1) ← PC _H (SP-2) ← PC _L , PC _H ← 0 PC _L ← L	
	RET	PC _L ← (SP), PC _H ← (SP+1)	
	RET cc	If condition cc is false continue, else same as RET	
	RET I	Return from interrupt, same as RET	
	RET N	Return from non- maskable interrupt	

2.0 PIO ARCHITECTURE

A block diagram of the Z80-PIO is shown in figure 2.0-1. The internal structure of the Z80-PIO consists of a Z80-CPU bus interface, internal control logic, Port A I/O logic, Port B I/O logic, and interrupt control logic. The CPU bus interface logic allows the PIO to interface directly to the Z80-CPU with no other external logic. However, address decoders and/or line buffers may be required for large systems. The internal control logic synchronizes the CPU data bus to the peripheral device interfaces (Port A and Port B). The two I/O ports (A and B) are virtually identical and are used to interface directly to peripheral devices.

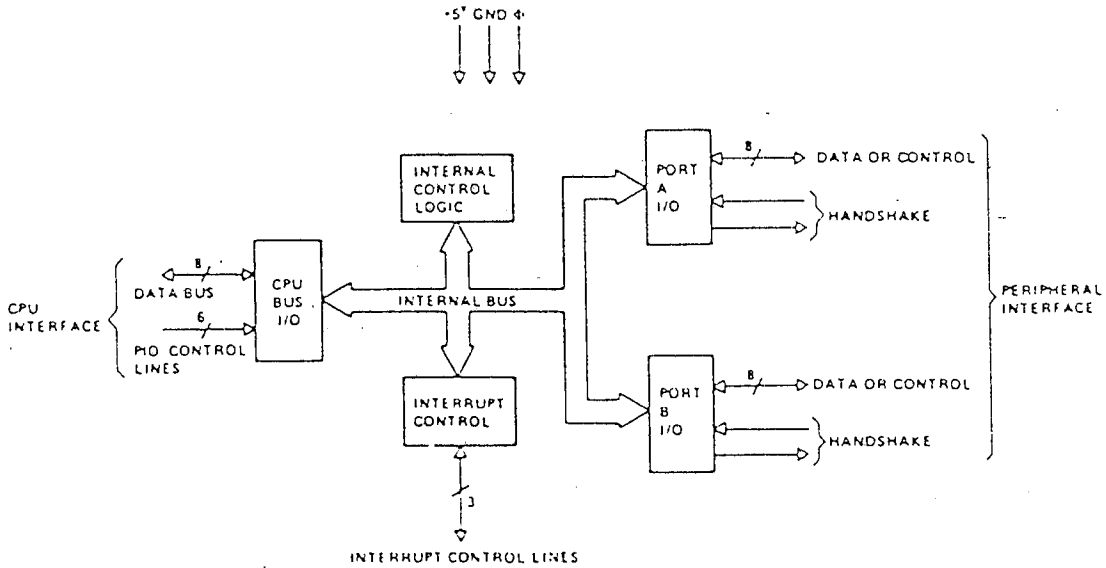


FIGURE 2.0-1
PIO BLOCK DIAGRAM

The Port I/O logic is composed of 6 registers with "handshake" control logic as shown in figure 2.0-2. The registers include: an 8 bit data input register, an 8 bit data output register, a 2 bit mode control register, an 8 bit mask register, an 8 bit input output select register, and a 2 bit mask control register.

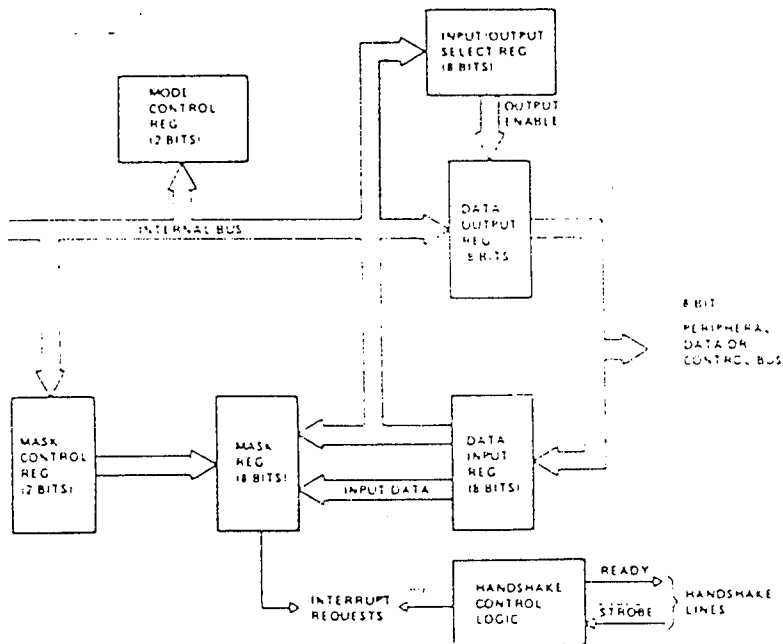


FIGURE 2.0-2
PORT I/O BLOCK DIAGRAM

FIGURE 5.0.2
MODE 1 (INPUT) TIMING

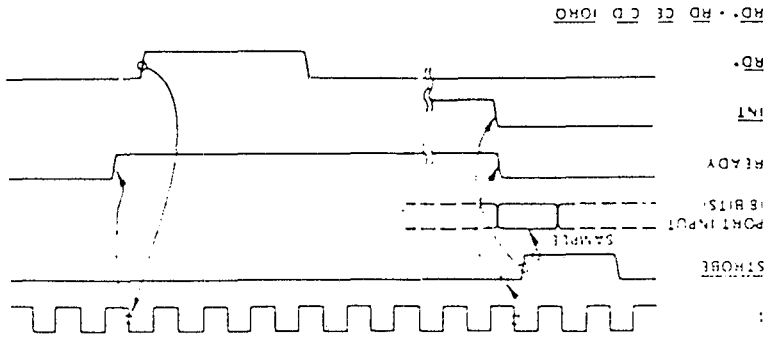


Figure 5.0.2 illustrates the timing of an input cycle. The peripheral initiates this cycle using the strobe line after the CPU has performed a data read. A low level on this line loads data into the port input register and the rising edge of the strobe line activates the interrupt request line (INT) if the interrupt enable is set and this is the highest priority requesting device. The next falling edge of the clock line (φ) will then reset the Ready line to an inactive state signifying that the input register is full and further loading must be inhibited until the CPU reads the data. The CPU will in the course of its interrupt service routine, read the data from the interrupting port. When this occurs, the positive edge from the CPU read signal will raise the Ready line with the next low going transition of φ, indicating that new data can be loaded into the PI0.

5.2 INPUT MODE (MODE 1)

FIGURE 5.0.1
MODE 0 (OUTPUT) TIMING

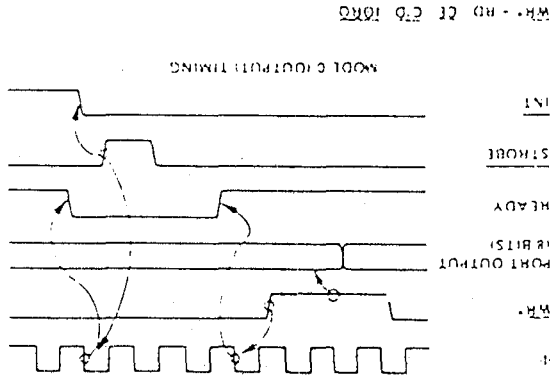


Figure 5.0.1 illustrates the timing associated with Mode 0 operation. An output cycle is always started by the execution of an output instruction by the CPU. The low level of the WR signal is used to latch the data from the CPU data bus into the addressed port's (A or B) output register. The rising edge of the write pulse then raises the Ready flag after the next falling edge of φ to indicate that data is available for the peripheral device. In most systems the rising edge of the Ready signal can be used as a latching signal in the peripheral device if desired. The Ready signal will remain active until a positive edge is received from the strobe line indicating that the peripheral has taken the data. However, the Ready signal will not go inactive until a falling edge occurs on the clock (φ) line. The purpose of delaying the negative transition of the Ready signal until after a negative clock transition is that it allows for a very simple generation scheme for the strobe pulse. By merely connecting the Ready line to the Strobe line, a strobe with a duration of one clock period will be generated with no other logic required. The positive edge of the strobe pulse automatically generates an INT request if the interrupt enable flip flop has been set and this device is the highest priority device requesting an interrupt.

5.1 OUTPUT MODE (MODE 0)

5.0 TIMING

5.3 BIDIRECTIONAL MODE (MODE 2)

This mode is merely a combination of Mode 0 and Mode 1 using all four handshake lines. Since it requires all four lines, it is available only on Port A. When this mode is used on Port A, Port B must be set to the Bit Control Mode. Figure 5.0-3 illustrates the timing for this mode. It is almost identical to that previously described for Mode 0 and Mode 1 with the Port A handshake lines used for output control and the Port B lines used for input control. The difference between the two modes is that, in Mode 2, data is allowed out onto the bus only when the A strobe is low. The rising edge of this strobe can be used to latch the data into the peripheral since the data will remain stable until after this edge. The input portion of Mode 2 operates identically to Mode 1.

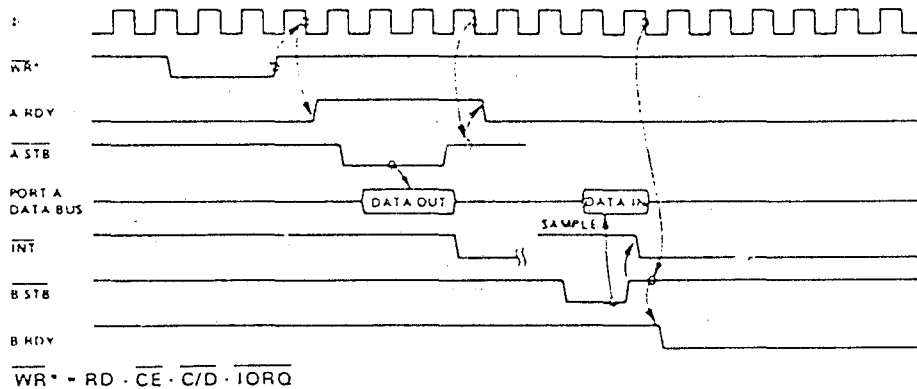


FIGURE 5.0-3
PORT A, MODE 2 (BIDIRECTIONAL) TIMING

The peripheral must not gate data onto a port data bus while $\overline{A STB}$ is active. Bus contention is avoided if the peripheral uses $\overline{B STB}$ to gate input data onto the bus. The PIO uses the $\overline{B STB}$ low level to latch this data. The PIO has been designed with a zero hold time requirement for the data when latching in this mode so that this simple gating structure can be used by the peripheral. That is, the data can be disabled from the bus immediately after the strobe rising edge.

5.4 CONTROL MODE (MODE 3)

The control mode does not utilize the handshake signals and a normal port write or port read can be executed at any time. When writing, the data will be latched into output registers with the same timing as Mode 0.

When reading the PIO, the data returned to the CPU will be composed of output register data from those port data lines assigned as outputs and input register data from those port data lines assigned as inputs. The input register will contain data which was present immediately prior to the falling edge of \overline{RD} .

An interrupt will be generated if interrupts from the port are enabled and the data on the port data lines satisfies the logical equation defined by the 8-bit mask and 2-bit mask control registers.

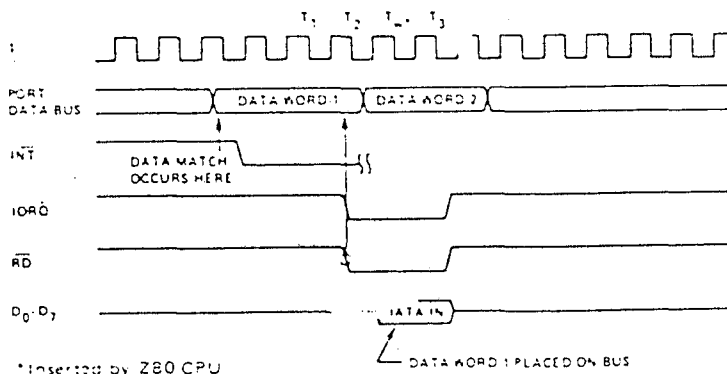


FIGURE 5.0-4

recommended operating conditions

Table with 6 columns for series (54, 54H, 54L, 54LS, 54S) and 3 columns for min, nom, max values. Rows include Supply voltage VCC, High-level output current IOH, Low-level output current IOL, and Operating free-air temperature TA.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

Table with 6 columns for series and 3 columns for min, typ, max values. Rows include VIH, VIL, VI, VOH, VOL, Ii, IiH, IiL, IOS, and ICC.

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions. ‡ All typical values are at VCC = 5 V, TA = 25°C. § Ii = -12 mA for SN54/SN74, -8 mA for SN54H/SN74H, and -18 mA for SN54LS/SN74LS and SN54S/SN74S. * Not more than one output should be shorted at a time, and for SN54H/SN74H and SN54S/SN74S, duration of short-circuit should not exceed 1 second. ** The input clamp voltage specification is effective for Series 54/74 and 54H/74H parts date-coded 7332 or higher.

POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

supply current†

Table with 5 columns: TYPE, IccH (mA), IccL (mA), Icc (mA), and Average per gate (50% duty cycle). Rows list various gate types like '00, '04, '10, '20, '30, 'H00, 'H04, 'H10, 'H20, 'H30, 'L00, 'L04, 'L10, 'L20, 'L30, SN54L30, SN74L30, 'LS00, 'LS04, 'LS10, 'LS20, 'LS30, 'S00, 'S04, 'S10, 'S20, 'S30, 'S133.

switching characteristics at VCC = 5 V, TA = 25°C

Table with 4 columns: TYPE, TEST CONDITIONS, tPLH (ns), and tPHL (ns). Rows list various gate types with their respective test conditions and propagation delay times.

Load circuits and voltage waveforms are shown on pages 148 and 149.

schematics (each gate)

Four circuit diagrams showing internal transistor structures for different gate types: '00, '04, '10, '20, '30; 'H00, 'H04, 'H10, 'H20, 'H30; 'L00, 'L04, 'L10, 'L20, 'L30; and 'S00, 'S04, 'S10, 'S20, 'S30, 'S133.

POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

TEXAS INSTRUMENTS

TEXAS INSTRUMENTS

recommended operating conditions

	54 FAMILY 74 FAMILY	SERIES 54 SERIES 74			SERIES 54LS SERIES 74LS			UNIT
		'32			'LS32			
		MIN	TYP	MAX	MIN	TYP	MAX	
Supply voltage, V_{CC}	54 Family 74 Family	4.5 4.75	5	5.5 5.25	4.5 4.75	5	5.5 5.25	V
High-level output current, I_{OH}				-800			-400	μ A
Low-level output current, I_{OL}	54 Family 74 Family		16				4 8	mA
Operating free-air temperature, T_A	54 Family 74 Family	-55 0		125	-55 0		125 70	$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST FIGURE	TEST CONDITIONS ¹	SERIES 54 SERIES 74			SERIES 54LS SERIES 74LS			UNIT
			'32			'LS32			
			MIN	TYP ²	MAX	MIN	TYP ²	MAX	
V_{IH} High-level input voltage	1, 2		2		2			V	
V_{IL} Low-level input voltage	1, 2							V	
V_I Input clamp voltage	3	$V_{CC} = \text{MIN}, I_I = \text{§}$					-1.5	V	
V_{OH} High-level output voltage	1	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, I_{OH} = \text{MAX}$	54 Family 74 Family	2.4 3.4	3.4	2.5 3.4	3.4	0.7 0.8	V
V_{OL} Low-level output voltage	2	$V_{CC} = \text{MIN}, V_{IL} = V_{IL \text{ max}}, I_{OL} = \text{MAX}$	54 Family 74 Family	0.2 0.2	0.4 0.4	0.25 0.35	0.4 0.5		V
I_I Input current at maximum input voltage	4	$V_{CC} = \text{MAX}, V_I = 5.5 \text{ V}$			1			0.1	mA
I_{IH} High-level input current	4	$V_{CC} = \text{MAX}$		$V_{IH} = 2.4 \text{ V}$ $V_{IH} = 2.7 \text{ V}$	40			20	μ A
I_{IL} Low-level input current	5	$V_{CC} = \text{MAX}, V_{IL} = 0.4 \text{ V}$			-1.6			-0.36	mA
I_{OS} Short-circuit output current*	6	$V_{CC} = \text{MAX}$	54 Family 74 Family	-20 -18	-55 -55	-6 -5	-40 -42		mA
I_{CC} Supply current	7	$V_{CC} = \text{MAX}$	Total, outputs high		15	22	3.1	6.2	
			Total, outputs low		23	38	4.9	9.8	
			Average per gate	$V_{CC} = 5 \text{ V}, 50\% \text{ duty cycle}$	4.75		1.00		

¹For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
²All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^{\circ}\text{C}$.
[§] $I_I = -12 \text{ mA}$ for SN54/SN74¹ and -18 mA for SN54LS/SN74LS¹.
^{*}Not more than one output should be shorted at a time.
¹The input clamp voltage specification is effective for Series 54/74 parts date-coded 7332 or higher.

1272

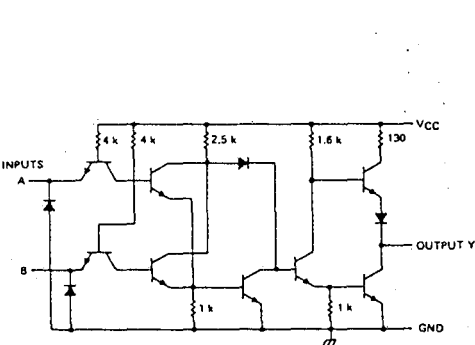
1272

switching characteristics at $V_{CC} = 5 \text{ V}, T_A = 25^{\circ}\text{C}$

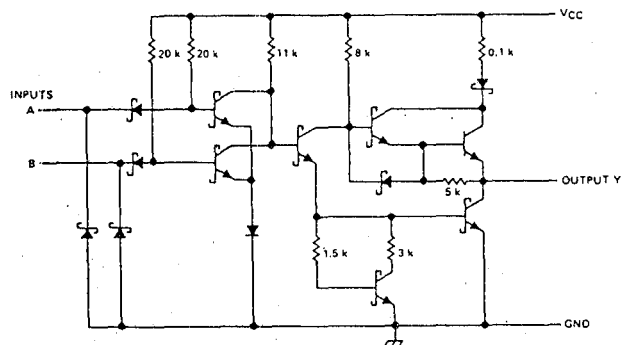
TYPE	TEST CONDITIONS [#]	t_{pLH} (ns)			t_{pHL} (ns)		
		Propagation delay time, low-to-high-level output			Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
'32	$C_L = 15 \text{ pF}, R_L = 400 \Omega$		10	15		14	22
'LS32	$C_L = 15 \text{ pF}, R_L = 2 \text{ k}\Omega$		14	22		14	22

[#]Load circuit and voltage waveforms are shown on pages 148 and 149.

schematics (each gate)



'32 CIRCUITS



'LS32 CIRCUITS

Resistor values shown are nominal and in ohms.

recommended operating conditions

Table with columns for SERIES 54/74, '70, '72, '73, '76, '107, '74, '109, '110, '111, and UNIT. Rows include Supply voltage, VCC; High-level output current, IOH; Low-level output current, IOL; Pulse width, tpw; Input setup time, tsetup; Input hold time, thold; and Operating free-air temperature, TA.

The arrow indicates the edge of the clock pulse used for reference: ↑ for the rising edge, ↓ for the falling edge. electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

Table with columns for PARAMETER, TEST CONDITIONS, '70, '72, '73, '76, '107, '74, '109, '110, '111, and UNIT. Rows include VIH, VIL, VI, VOH, VOL, Ii, IiH, IiL, IOS, and ICC.

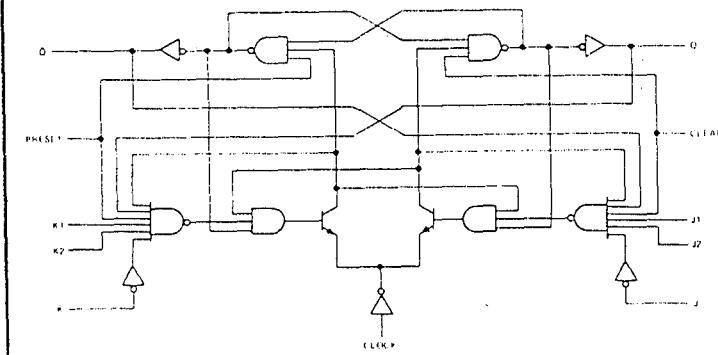
For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions. All typical values are at VCC = 5 V, TA = 25°C. Not more than one output should be shorted at a time. NOTE 1: With all outputs open, ICC is measured with the Q and Q outputs high in turn. At the time of measurement, the clock input is at 4.5 V for the '70, '110, and '111, and is grounded for all the others. *The input clamp voltage specification is effective for Series 54/74 parts date coded 7332 or higher.

switching characteristics, VCC = 5 V, TA = 25°C

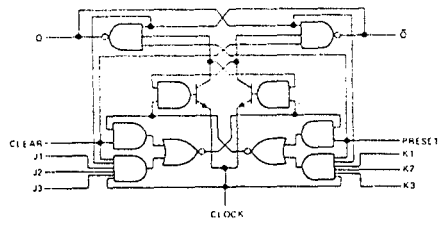
Table with columns for PARAMETER, FROM (INPUT), TO (OUTPUT), TEST CONDITIONS, '70, '72, '73, '76, '107, '74, '109, '110, '111, and UNIT. Rows include fmax, tPLH, tPHL, tPLH, tPHL, tPLH, tPHL.

fmax: maximum clock frequency, tPLH: propagation delay time, low to high level output, tPHL: propagation delay time, high to low level output. NOTE 2: Load circuit open and voltage waveforms are shown on page 148.

functional block diagrams



70-GATED JK WITH CLEAR AND PRESET



72-GATED JK WITH CLEAR AND PRESET

See following pages for: '73-DUAL JK WITH CLEAR; '74-DUAL D WITH CLEAR AND PRESET; '76-DUAL JK WITH CLEAR AND PRESET; 107-DUAL JK WITH CLEAR

'109-DUAL JK WITH CLEAR AND PRESET; '110-GATED JK WITH CLEAR AND PRESET; '111-DUAL JK WITH CLEAR AND PRESET

Preparatory Function	Meaning	Preparatory Function	Meaning
G00	Point to Point, Positioning	G40	Tool Offset Cancel
G01	Linear Interpolation	G41 - G52	Tool Offset
G02	Circular Interpolation Arc clockwise	G53	Linear Shift Cancel
G03	Circular Interpolation Arc counterclockwise	G54 - G59	Linear Shift
G04	Dwell	G60	Positioning Exact 1 (fine)
G05	Parabolic Interpolation	G61	Positioning Exact 2 (med)
G08	Acceleration	G62	Positioning Fast (coarse)
G09	Deceleration	G63	Tapping
G17	XY Plane Selection	G80	Fixed Cycle Cancel
G18	XZ Plane Selection	G81 - G89	Fixed Cycle
G19	YZ Plane Selection	G90	Absolute Dimension
G25 - G29	Permanently unassigned	G91	Incremental Dimension
G33	Thread Cutting, constant lead	G92	Preload Registers
G34	Thread Cutting, increasing lead	G93	Inverse Time, Feed Rate
G35	Thread Cutting, decreasing lead	G94	Feed per Minute
G36 - G39	Permanently unassigned	G95	Feed per Spindle Revolution
		G96	Constant Surface Speed
		G97	Revolution per Minute

Miscellaneous Function	Meaning
M00	Program Stop
M01	Optional (planned) Stop
M02	End of Program
M03	Spindle Clockwise
M04	Spindle Counterclockwise
M05	Spindle Off
M06	Tool Change
M07	Coolant No. 1 ON
M08	Coolant No. 2 ON
M09	Coolant OFF
M10	Clamp
M11	Unclamp
M13	Spindle Clockwise and Coolant ON
M14	Spindle Counterclockwise and Coolant ON

Miscellaneous Function	Meaning
M15	Motion +
M16	Motion -
M19	Oriented Spindle Stop
M30	End of Tape
M31	Interlock Bypass
M36 - M37	Feed Ranges
M38 - M39	Spindle Speed Range
M40 - M45	Gear Changes
M50	Coolant No. 3 ON
M51	Coolant No. 4 ON
M55 - M56	Linear Tool Shift
M60	Workpiece Change
M61 - M62	Linear Workpiece Shift
M71 - M72	Angular Workpiece Shift