endüstride mikroişlemci kullanımına bir örnek

Kaya YAZGAN

UDK: 621.3.049. 771.14

ÖZET

Bu uygulamada tanıtılan denetim dizgesi mikroişlemcili dizge tasarımı için ön bilgiler vermektedir. Oldukça genel amaçlı bir donanım seçilmiş; ilişkin yazılım da akış çizimi ve komut dizisi aşamalarında verilmiştir.

SUMMARY

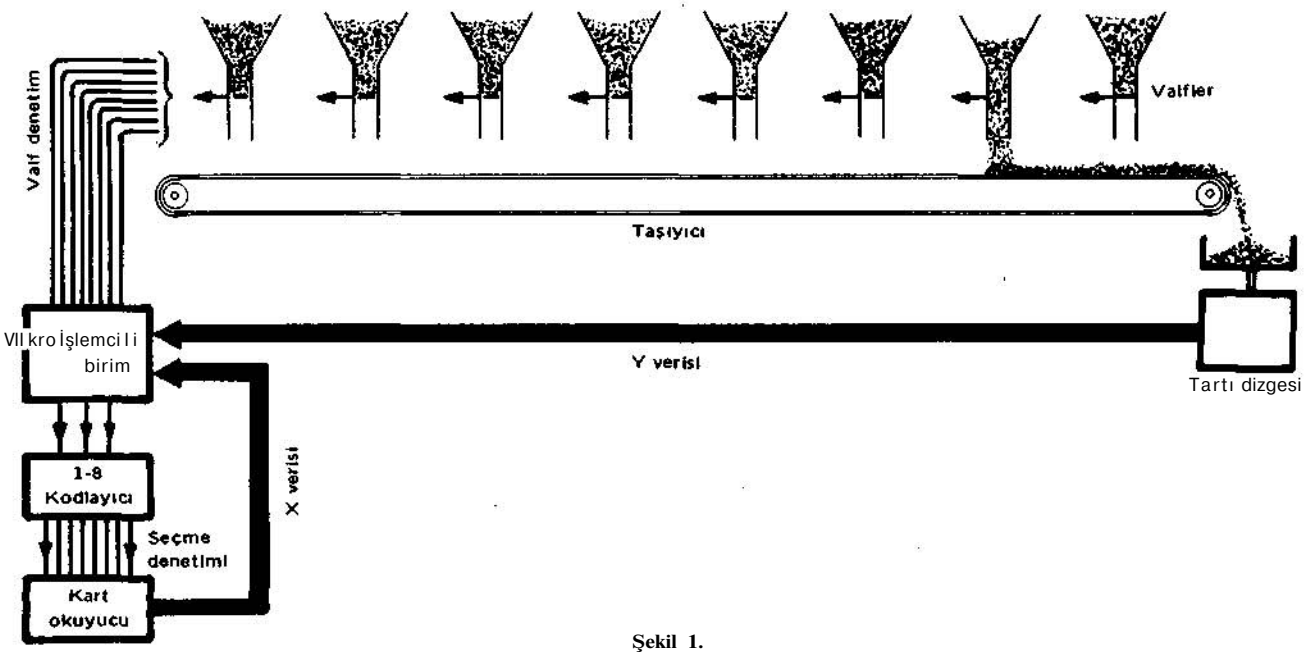
*The microprocessor control system described in this application note contains some basic information on the design of microprocessor systems. The harduıare chosen is quite general-purpose, related softıvare is given at the levels of flow-**chart and** instruction list.*

Endüstri alanında alışılagelmiş denetim yöntemlerinin yerini almaya başlıyan mikroişlemcili dizgelere bir örnek Şekil 1'de görülen tartılı boşaltma dizgesidir. içlerinde farklı hammaddeler bulunan 8 huninin boşalması valflerin açılmasıyla sağlanmakta, taşıyıcı bandın sonunda ise bir tartma aygıtı yer almaktadır. Amaç her huniden ard arda istenen miktarda hammadde dökmektir.

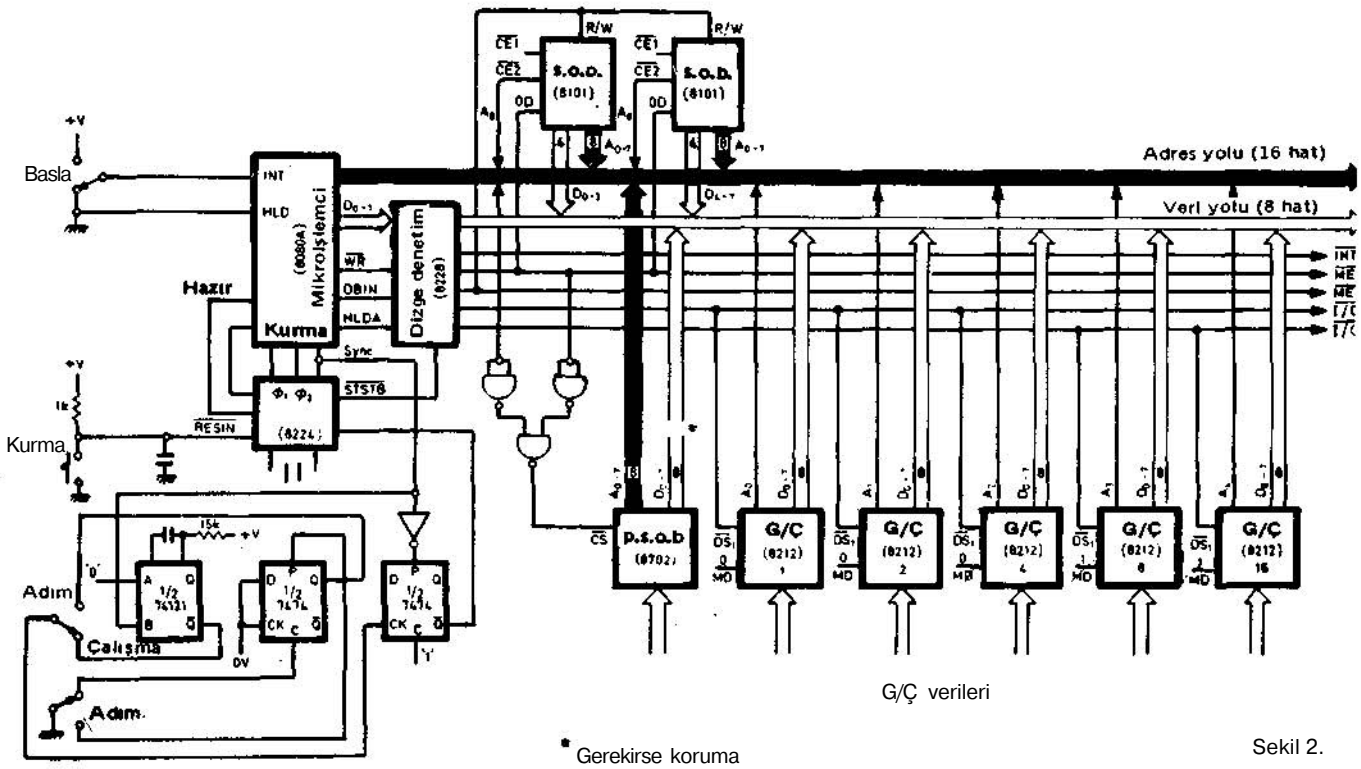
Intel 8080 mikroişlemcisini temel alan denetim biriminin öbek şeması da, Şekil 2'de verilmiştir. Kart okuyucuya yerleştirilecek karta herbir huninin boşaltması istenen miktar üç ondalıklı bir sayı olarak delinmekte; kart, okuyucuya yerleştirildikten sonra kullanıcı başlama düğmesine basıp ilk valfı açmaktadır. Denetim dizgesi tartı biriminden gelen • yine 3 ondalık basamaklı - sayı ile kart okuyucudan gelen o hunideki boşaltma miktarına ilişkin veriyi karşılaştırıp, istenen sayıya ulaşılınca huninin valfını kapatmakta ve bir sonraki huniye geçmektedir. Bütün hunilerden istenen miktarda hammadde boşaltılınca sistem durmakta ve yeni bir komut beklemektedir. Şekil 1'de dizgenin taslağı verilmiştir. Kuşkusuz taşıyıcıkayış şekilde gösterildiği kadar uzun değildir, üzerinde biriken hammadde hiçbir zaman tartıyı etkileyecek büyüklüğe ulaşmamaktadır. Ayrıca tartıdan sonraki boşaltma düzeni de şekilde yer almamıştır.

Mikroişlemci dizgesi kurulurken özellikle ilerde beklenebilecek büyümeler, program değişiklikleri düşünülmüş hatta bu uygulama için gerekli olmasa da bir tane yaz/oku bellek (RAM) yerleştirilmiştir. Şimdi denetim diz-

***Kaya** Yazgan, ASELSAN*



Şekil 1.



Sekil 2.

gesini oluşturan öğeleri teker teker ele alıp seçin lerinin nasıl yapıldığını ve temel özelliklerini belirtelim.

Merkez işlem Birimi

Programın gerektirdiği matematik ve mantık işlemlerini gerçekleştiren birimdir. Bu uygulamada daha önce değinildiği gibi Intel'in oldukça yaygın kullanılan 8080 mikro işlemcisi seçilmiştir. Bu mikroişlemci 8'li bir veri yolu ***(Bus),*** 16 lı adres yolu ve bir dizi denetim iletkeni ile diğer birimlerle ilişki kurmaktadır. Diğer birimlere gönderilen ya da onlardan gelen veriler, veri yolundan iletilir. Bunların gideceği ya da geleceği adresler ise adres yolundaki bilgi ile belirlenir. 16 iletken ile 2^{*a} = 65 536 adet 8 bitlik bilgiyi adresleme olanağı vardır. (Bu kapasite örnek uygulamamızda kullanılmamaktadır.) 8080 içinde işlemleri yöneten aritmetik - mantık biriminin dışında 8 tane 8 bitlik yazmaç ***(register)*** vardır: B, C, D, E, H, L, Bellek, Birikeç adını taşıyan bu yazmaçların makina dilindeki kodlan yukarıdaki sırayla "000" dan "111 "e dek uzanmaktadır.

Dizge Denetim ve Saat

Şekil 2'de Merkez İşlem Biriminin hemen yanındaki 8228 Dizge Denetim Birimi, özellikle bellek ve Giriş/Çıkış Birimlerini denetlemek amacıyla kullanılmaktadır. 8080'in gereksinim duyduğu iki evreli saat ise 8224 Saat Birimi tarafından üretilmektedir.

Yaz/Oku Bellek

Daha önce değinildiği gibi tanıttığımız uygulamada bu tür bir belleğe gerek yoksa da, ilerde belirli bilgileri

ÇİZELGE 1

Şekil 2'deki Kısaltmaların Anlamları

$\overline{CE1}.\overline{CE2}$	Yonga engelsiz
R/W	Oku/Yaz girişi
OD	Çıkış engelli
INT	Kesme istemi
$\overline{INTA}$	Kesme değerlendirme
HLD	Tut
WR	Yaz çıkışı
DBIN	Veri yolu giriş konumu (Dizge denetleyicisine veri yolunun giriş konumunda olduğu bilgisi)
HLDA	Tut değerlendirme
$\overline{CS}$	Yonga seçme girişi
$\overline{DSi}$	Aygıt seçme girişi
MD	Konum
$\overline{MEM~R}$	Bellek oku
$\overline{MEMW}$	Bellek yaz
$\overline{I/OR}$	Giriş/Çıkış oku
$\overline{I/OW}$	Giriş/Çıkış yaz

Evrilmiş isimler, im '0' konumundayken işlevin yapılacağını göstermektedir.

yazıp saklayıp sonradan işleyen bir denetim uygulaması istendiğinde yeni donanım eklemekten kaçınmak için iki adet 8101 Yaz/Oku bellek yerleştirilmiştir. Bunların herbiri 4 bit 256 sözcüklü belleklerdir. Adresleri ortaktır, verinin yarısı (D „ _J bir 8101 de. diğer yarısı ($D_{4\ \_7}$) ise diğer 8101 de saklanmaktadır. Adresleri $A_{o\ \_7}$ yolundan, belirlenmekte ve ayrıca A_g iletkeni de bu iki yonganın seçilmesini sağlamaktadır (Çizelge 1). Okuma/Yazma konuır(seçin,i dizge denetiminin oluşturduğu MEMR, MEMW komutlarıyla yapılmaktadır.

Programlanabilir Salt Oku Bellek

Dizgedeki 8702 Programlanabilir Salt Oku bellek ise programın saklandığı 8 bit 256 sözcüklü bir bellektir. Yonga Seçimi ($\overline{CS}$) girişini besleyen geçitlerin oluşturduğu mantık yardımıyla yapılmaktadır. Yaz Oku bellek seçilmemiş ve bellekten okuma komutu gelmişse 8702'deki program okunmaktadır.

Kart Okuyucu

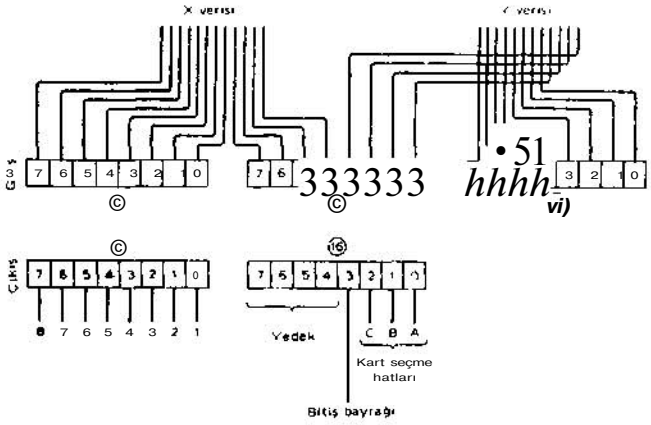
Kart okuyucu herbir valf denetimi için üç ondalık basamaklı bir sayı okuyup bunu ikili kodlanmış ondalık *(binary coded decimal)* yapısı içinde çıkış ucunda hazırlamaktadır. Bir kart üzerine yanyana dizilmiş 8 tane üç ondalık basamaklı sayıdan hangisini okuyacağını anlaması için ise 8 tane giriş ucundan birinin " 1 " olması gereklidir. Bu iş için 8212 Giriş/Çıkış birimlerinden biri ayrılabilirdi; ancak, bu bilgi 3 ikili basamakla taşınabildiğinden ve bir kod açıcı, 8212'den çok daha ucuz olduğundan Şekil 1 ve 2'de görüldüğü gibi 8212'nin 3 biti kullanılmış ve bir kod açıcı eklenmiştir. Kod açıcı, girişindeki ikili bilginin belirlediği çıkış ucunu " 1 " yapmakta böylece kart okuyucu da hangi sayıyı okuyacağını belirlemektedir.

Valf Denetimi

Denetim Dizgesi açısından valflerin denetimi çok basit bir yapı içinde çözülmüştür. Dizge, programın belirli bir noktasında ilgili 8212'ye bir çıkış yaptırmakta ve ilgili valfı açtırmaktadır, ölçme sonucunun karttan okunan değere ulaşması durumunda ise yeni bir çıkış yaptırılıp valf kapatılmaktadır.

Giriş/Çıkış Birimleri

Kart Okuyucu ve Valf Denetimi sırasında değindiğimiz 8212 Giriş/Çıkış birimlerinin bizim ömeğimizdeki kullanımları son derece yalındır. Kesme *(Interrupt)* yapmamakta, üç tanesi hep çıkış birimi, ikisi de hep giriş birimi olarak kullanılmakta yani konumları değişmemektedir.

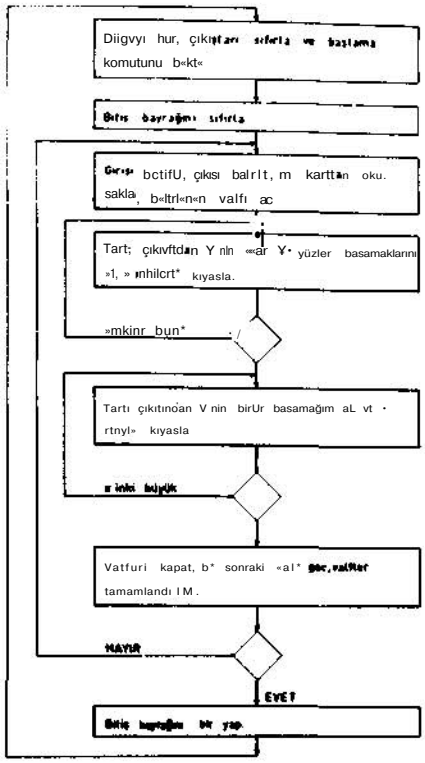


Sekil 3.

Herbiri farklı birer adres iletkenine bağlı olduklarından (1, 2, 4, 8, 16 numaraları ve A_O, A_{1f} A_2, A_3, A_4 e bağlanmaları) adreslen meleri çok kolaydır. Dizge tasarımlanırken oldukça pahalı olan 8212 lerin sayısını enaza indirmek amaçlandığından Şekil 3'de görüldüğü gibi X ve Y verilerini (yani kart okuyucu ve tartı verilerini) iki adet Giriş/Çıkış birimine yerleştirmek yolu seçilmiştir. Her biri 3 tane ikili kodlanmış onlu, (3x4 = 12) bitlik veri iletken X ve Y iletkenlerinin en önemli *(mostsignificant)* ikişer basamağı (onlar ve yüzler basamakları) 1 ve 4 adresli Giriş/Çıkış birimlerine verilirken; her ikisinin ve en önemsiz *(least significant)* birer basamağı (birler basamağı) 2 adresli Giriş/Çıkış birimine verilmiştir. Örneğin kart okuyucu 246 okuyor ve tartı sonucu belirli bir anda 139 ise 1 nolu birimde 24; 4 nolu birimde 13; 2 nolu birimde ise 69 gözlenecektir.

Akış Çizgesi

Şekil 4'de çizilen akış çizgesi programın yazılması için gerekli mantıksal diziyi oluşturmaktadır. Başlama düğmesine basılmasıyla "Bitiş bayrağı" kaldırılır, okunacak kart sütunu belirlenir karttaki bilgi saklanır, ilgili valf açılır. Tartı aygıtından gelen bilgi saklanan veriyle sürekli karşılaştırılarak belirlenen miktarda hammaddenin akması sağlanır. Saklı sayıya tartı sonucunun ulaşması ile huninin valfi kapatılıp bir sonraki huniden boşaltılacak hammadde miktarını belirlemek amacıyla kart okuyucunun yeni bir sütun okuması sağlanır. Tüm hunilerden boşalma sağlanınca bitiş bayrağı yerleştirilip dizge yeniden bekleme durumuna geçer.



Şekil 4.

Bitiş bayrağı Şekil 3'de gösterildiği gibi 16 adresli Giriş/Çıkış biriminde tek bitlik bir bilgidir. Giriş/Çıkış birimlerinin sayısını en aza indirmek için X ve Y bilgilerinin ikişer parçaya bölündüğüne değinilmişti, işte bu bölünmenin gereği olarak önce tartı sonucunun (Y verisinin) onlar ve yüzler basamakları, karttan okunan sayının (X verisinin) onlar ve yüzler basamağına ulaşıncaya dek üstteki çevrim içinde program beklemektedir. Bu sağlandıktan sonra X ve Y verilerinin birler basamağı koşulu sağlayıncaya dek de alttaki çevrimde beklenmektedir.

X ve Y verilerinin birler basamaklarını saklayan 8 adresi! Giriş/Çıkış biriminden istenen basamağın seçilmesi "FO" ya da "OF" onaltılı ***(hexodecimol)*** sayılarıyla maskelenmesi (bu öritekte VE'leme) ile gerçekleştirilmektedir. Bilindiği gibi "FO" ikili tabanda "1111 0000" dizisini "OF" ise "0000 1111" dizisini oluşturmaktadır.

Programlama

Ele alman uygulama için gereken küçük programı yazmak için Çizelge 2'de verilen birkaç komutu tanımak yeterlidir. Sıçrama ***(jump)*** komutlarından sonra gelen ilk

ÇİZELGE 2

örnek Programda Geçen Komutların Tanıtımı

MVI, A	Bir sonraki satırdaki veriyi birikece yükle
OUT	Çıkış yap
El	Giriş engelsiz
HLT	Dur
MVI,D	D yazacına al
MOVA, D	Veriyi D'den A'ya (Birikece) al
İN	Giriş yap
ANI	Bir sonraki satırdaki veriyle birikeçtekini VE'le
RRC	Birikeçteki sayıyı bir bit sağa kaydır.
CMPL	L yazacındaki sayıyla birikeçtekini karşılaştır
JM	Bir önceki işlem sonucu eksi ise alttaki adrese sıçra
DCRD	D yazacındaki sayıyı bir eksilt
JNC	Bayrak sıfırsa aşağıdaki adrese sıçra
Dİ	Kesme engelli
JMP	Aşağıdaki adrese sıçra(koşulsuz)

onaltı tabanlı kod sıçranacak adresin son parçasını, ikinci onaltı tabanlı kod ise adresin ilk parçasını oluşturmaktadır. Örneğin "JMP, 1E, 00" komutu yerine getirilirken program koşulsuz olarak "001 E" adresine geçer.

RRC komutu ise biri keç teki bilgiyi sağa doğru kaydırmaya yarar. Birikecin sağ ucundan bir bit çıkınca kaybolmaz ve tek bitlik özel bir bayrak birikecinde saklanır. Örneğin JNC komutu bu bitin sınanmasında kullanıl?. Eğer bu bit "0" ise sıçra komutunun verdiği adrese sıçranır, aksi durumda devam edilir. Dikkat edilirse Çizelge 3'de verilen programda "000A" ve "000B" adreslerindeki komutta E yazacına 80 H (H, 801n onaltılı tabanda olduğunu belirtiyor, iki tabanında yazılırsa: 1000 0000) yerleştiriliyor. Ardından "32" adresli komutta sağa bir kayma sağlanıyor. Sekiz sağa kaymadan sonra bayrakta 1 gözlenmekte ve görevin tamamlanmış olduğu böylece belirlenmektedir.

Sonuç

Bir mikroişlemcili denetim dizgesi örneği olan bu uygulama, mikroişlemcinin endüstrideki kullanım olanaklarını göstermesi yönünden ilginçtir. Daha önce değinildiği gibi Yaz/Oku bellek kuttanım zorunluluğu yokken 256 sözcüklü bir Yaz/Oku bellek kullanılmıştır. Salt oku bellek olarak seçilen 256 sözcüklü belleğe de yalnızca 62 sözcüktük bir program yerleştirilmiştir. Kısacası kurulu olan dizge bu uygulamanın gerektirdiğinden daha büyük olanaklara sahiptir.

Programlama elden geldiğince basit tutulmuş, prop-amlanabilir salt oku bellek olarak seçilen 8702 yerine önce bir yaz oku bellek yerleştirilip provam geliştirilmiş ardından 8702'ye geçilmiştir.

Niikroişlemci olarak seçilen 8080, yaklaşık 450-550 nsan erişim süreli bellekler için tasarımlanmıştır ve 850 nsan' ye dek uzanan erişim süreleri bir üst sınır oluşturmaktadır. Ucuz olması nedeniyle bu uygulamada 1,3 /isan erişim süreli 8702 kullanılmış, bu nedenle de Şekil 2'de 'saat, altındaki iki duraklarla *(flip-flop)* yavaşlatılmıştır^ Katalog, 8702'nin çıkışının eksi gerilim düzeylerinden korunmasının gereğini belirttiği için diyot-direnç sınırlaması ile şekilde yıldız ile işaretli bölgede koruma yapılmıştır.

ÇİZELGE 3

PROGRAM

Adres		Açıklama
0000	MVI, A	Biri keçe 0
0001	O	yüklenir
0002	OUT	8 adresli Giriş/Çıkış biriminden 'O'
0003	8	gönderilir (yani tüm vanaları kapat)
0004	E,	Kesme engeli kaldırılır
0005		beklemeye geçirilir
0006	HLT	16 adresli G/Ç birimine '0' gönderilir
0007	OUT	Kod açıcıya 0 adresi gider ve bitiş
	16	bayrağı kaldırılır
0008	MVI, D	D yazacına 7 yazılır '111 '-'000' değişen bu yazaç kapsamı denetlenen
0009		vana numarısını saklar.
	7	
000A	MVI, E	E yazacına 16 tabanına göre 80 yazılır $(1000\ 0000)_2$
000B	80H	
000C	MOVA.D	Denetlenecek vana numarası birikeçe alınır
000D	OUT	Denetlenecek vana numarası kod
000E	16	açıcıya gönderilir
000F	*T*	1 adresli G/Ç biriminden karttaki
0010		sayının en önemli iki basamağı birikece alınır.
0011	MOVH,A	Bu iki basamak Hyazacına yerleştirilir.
0012	İN	2 adresli G/Ç biriminden karttan ve
0013	2	tartı aygıtından gelen sayıların en önemsiz birer basamağı okunur.
0014	ANI	'1111 0000' ile VE'lenir (karttan
0015	FO	gelen basamak biri keçin solunda kalır)
0016	RRC	
0017	RRC	Birikeçteki sayı 4 bit sağa
0018	RRC	kaydırılır
0019	RRC	
001A	MOVL.A	Karttan geler. en önemsiz basamak L yazacında saklanır
001B	MOVA,E	E'deki sayı (bir biti ? diğerleri 0 olan sayı) birikece taşınır
001C	OUT	8 adresli G/Ç birimince bu sayı gün.
001C	8	c'erilince '1 'e karşı gelen valf açılır.
001E	İN	4 adresli G/Ç birimincen tartı so-
001F	4	nucuna onlar ve yüzler basamakları alınır.
0020	CMPH	Karttan okunan onlar ve yüzler basamaklarından tartıdan okunan onlar ve yüzler basamakları çıkartılır.
0021	JM	Sonuç eksiyse '001 E' adresine
022	1E	sıçranır
0023	00	
0024	İN	2 adresli G/Ç biriminden tartı ve
0025	2	kart verilerinin birler basamakları
0026	ANI	basamağı alınıp tartının en birler
0027	CF	basamağı elde edilir
0028	CMPL	Kart verisinin birler basamağından tartı verisinin birler basamağı çıkarılır
0029	JM	
002A	24	Sonuç eksiyse '0024" adresine
002B	00	sıçranır.
002C	MVI,A	Birikece'0'yüklenir
002D	0	
002E	OUT	8 adresli G/Ç birimine '0' yollanır
002F	8	(yani tüm valfler kapatılır)
0030	DCRD	D yazmacındaki sayı bir eksi İtilir. (bir sonraki valfe geçiş yapılır)
0031	MVA,E	E'deki sayı birikece alınır (tek '1' gerisi '0' olan sayı)
0032	RRC	Birikeçteki sayı bir bit sağa kaydırılır.
0033	MOVE,A	Birikeçteki sayı E'ye taşınır
0034	JNC	Sağa kayma sırasında bayrak sıfır
0035	0C	ise (yani tüm valfler denetlenmemiş
0036	00	ise) 'OOOC'ye gidilir.
037	MVI,A	Birikece 000 1000 alınır
0038	8 H	
0039	OUT	16 adresli G/Ç birimine birikeçteki
003A	16	sayı aktarılır, yani bitiş bayrağı '1' yapılır.
003B	DI	Kesme engellenir
003C	JMP	
003 D	00	Program başa döner
003E	00	

KAYNAKLAR:

J. Skinner, *Using a Microprocessor",* Wireles World, Haziran, 1977, Eylü! 1977.