

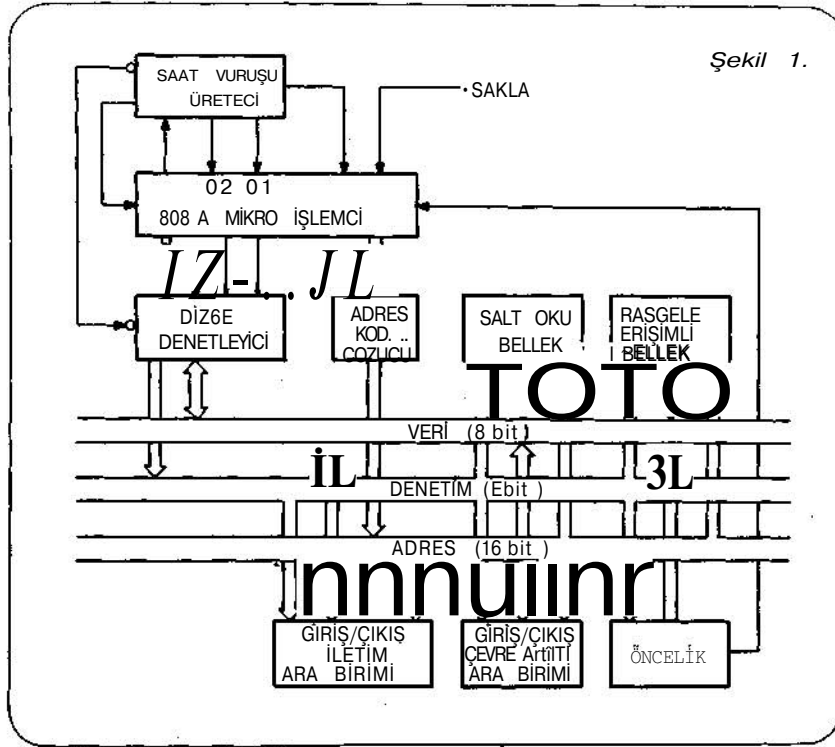
MİKROİŞLEMCİLER ve UYGULAMALARI

Orta çapta tümleşik (*Medium Scale Integration, MSI*) Transistor-Transistor Mantığı (TTM) uygulamalarının yaygınlaşmasını izleyen 1970 lerde geniş çapta tümleşim (*Large Scale Integration, LSI*) uygulamaları, metal oksit yarıiletken (*MOS*) teknolojisi önemli atılımlar sağladı. Tek yonga (*chip*) üzerine giderek daha büyük salt oku bellekler (*Read Only Memory, ROM*), rasgele erişimli bellekler (*Random Access Memory, RAM*) ve mikroişlemciler (*mikroprocessor*) yerleştirilmesi; bilgisayar terimlerinin, uygulamalarının, tekniklerinin, elektroninin -hemen hemen- tüm dallarına yayılmasına yol açtı.

Temel olarak bir mikroişlemci, bilgisayardaki merkezi işlem biriminin (*Central Processing Unit, CPU*) önemli işlerini yüklenebilecek, 4,8,16 bit gibi boyutlarda verileri işleyecek, -çoğunlukla salt oku bellek üzerine kaydedilmiş- bir programı yürütecek bir tümleşik devredir. Mikroişlemciler

- . Ucuz, geniş ölçüde seri yapım ürünü olan
- . Dizgedeki tümleşik devre sayısını önemli ölçüde azaltan
- . Program değişikliğiyle görevi de değiştirilen, böylece hiçbir elektronik dizgede bulunamayan esnekliğe sahip
- . Mühendislik tasarım, yapım sürelerini çok azaltan

önemli bir olanaktır. Buna karşılık yeni bir elektronik eğitimi gerektirmesi, yazılım olarak bazı programlama dillerinin öğrenilmesi, sayısal elektronik ve içindeki "bilgisayar mimarisi" (*computer architecture*) konusu gibi konuları tasarımcı için gerekli kılması da önemli bir sorundur. Mikroişlemcili dizgenin tasarımı, program yazılması, benzeşimi (*simulation*) sırasında genellikle bir bilgisayar -yada minibilgisayar- gereklidir. 1980 e dek mikroişlemciyi temel alan dizgelerin satışının 15 milyar TL sini aşacağını tahmin edildiğini, birçok özel iletişim, sUreç de-



Bunların arasında belki de en önemli olan genel amaçlı yazmaçlarla komut yazmacıdır. Sekiz bitlik komut yazmacı dört bitlik veriyi taşıyabilmektedir. Bazı komutlar 16 bit uzunluğundadır, bellekte iki kat fazla yer tutar, işleme konmaları da iki kat süre alır. 4004 ün 45 tane iş yapan komutu bir tane de hiçbir iş yapmayı yalnızca belirli bir zaman geçirip, zamanlama programlarında kullanılan komutu vardır.

Mikrobilgisayar Yapıları

Temel işlem birimi olarak mikroişlemci kullanılan dizgeler birkaç firma tarafından gerçekleştirilip kart üzerine yerleştirilmiş olarak veya başka özel yapılar içinde piyasaya sürülmüştür. Kuşkusuz kullanıcının gereksinimlerini karşılayacak biçimde birçok tümeşik devreden oluşan dizgeler de kurulabilir. Bunlara örnek olarak Şekil 1 ve 2'de Intel'in 8080 dizgesi ve Motorola'nın 6800 dizgesi verilmiştir. Birçok 8 bitlik dizgeler için klasikleşmiş bir yapı olarak veri, Denetim ve Adres yollarını bu örneklerde belirgin olarak görmekteyiz. Salt oku belleklerini çıkış uçları Veri yoluna, giriş uçları ise Adres yoluna bağlanmıştır. Rasgele erişimli bellekler veriyi Adres yolunda alıp mikroistemciye Veri yolu

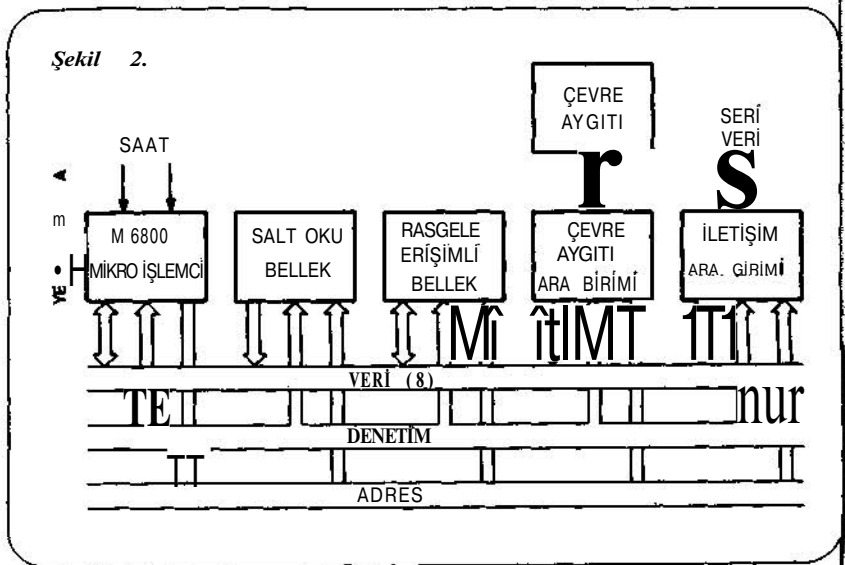
netimi, nitelik denetimi, tezgah denetimi, tıbbi elektronik uygulamaları, ulaşım araçları denetimi gibi konularda mikroişlemcileri göreceğimizi belirtmek sanırız gelişimin yönü konusunda bir fikir vermede yeterlidir.

Günümüzde 4,8,12,16 bitlik mikroişlemciler yapan firmalar 21 tane ve pazarı eşit olmayan bir şekilde paylaşmaktadırlar. (Intel, National, Rockwell, Texas Instruments, Fairchild, Electronic Arrays, Mostek, Advanced Micro Devices, Nippon Electric Company, Siemens, Motorola, RCA, Signetics, Scientific Micro Systems, Zilog, Intersil, Toshiba, General Instruments, Western Digital, Pana Facom, Monolithic Memories)

Fabrikaların özdevininde 1974 de 760 mikro bilgisayar kullanılması karşın bu sayının 1979 da 7050, 1984 de 23 400 dolayına çıkacağı hesaplanmaktadır.

Bir mikro işlemcinin nasıl bir iç yapıya sahip olduğunu anlamak için Intel 4004 örnek olarak incelenebilir. Intel 4004, beş ayrı bölümden oluşur:

- D Adres yazmacı ve adres artırma olanağı ile bir depo
- 2) 16 tane 4 bitlik yazmaç (indeksleme için ve genel amaçlı olarak kullanılabilen türden)
- 3) 4 bitlik bir Aritmetik ve Mantık birimi
- 4) 8 bitlik bir komut kaydedicisi ve kod çözücü
- 5) Çevre devreleri



sayısal elektronik

üzerinden yollarlar. Çeşitli giriş/çıkış birimleri (teletype, ekran, özel elektronik cihazlar, ısı, hız vb. sezicileri, motorlara kumanda uçları) ise gerekli arabirimler (*interface*) ile ana yollara bağlanmışlardır.

Mikroişlemcilerin Programlanması

Temel olarak bir mikroişlemcinin belirli bir dizi işlemi yapabilmesi için kendi anlayacağı bir komut dizisi ile yüklenmesi gerekmektedir. Bilgisayarlardaki makine diline benzeyen bu dilin kullanıcı tarafından öğrenilmesi, programların bu dilde yazılması olanaksız denecek kadar zordur. Bilgisayarlarda olduğu gibi kurgu (*assembly*) dili ve daha yüksek düzeyli, kullanıcıya yönelik dillerle program yazımı gerçekleşir.

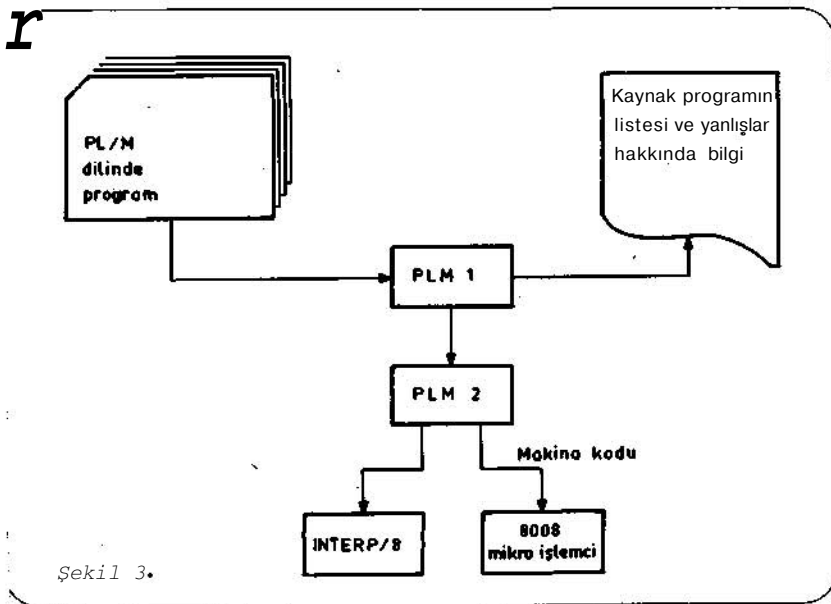
Programlama açısından bilgisayarlardan farklı olan tek nokta bilgisayarlarda derleme (*compilation*) işleminin, makine diline çevrim işleminin de bilgisayarların belleğindeki, derleyici (*compiler*) ve kurgu programlarını kullanarak, bilgisayar içinde gerçekleştirilmesine karşılık, mikroişlemcili dizgelerde genellikle bu işlemlerin bir minibilgisayarda yapılıp sonuçta elde edilen makine dilinin mikrobilgisayarın belleğine yerleştirilmesidir.

Baslık	Komut	Açıklama
TEST	SHL B	B nin adresini yükle
	LAM	B yi birikeçe yükle
	SHL A	A nin adresini yükle
	CPM	A ile B yi karşılaştır
LI	JFC LI B	A ile B yi karşılaştır
	LAM	A yi birikeçe yükle
	SHL C	C nin adresini yükle
	LMA EKD	Birikeçtekini C ye yükle

Şekil 4.

Intel firmasının geliştirdiği 8008 mikroişlemcisine özellikle uygun bir dil olan PL/M dilinin makine koduna çevrilmesinin PLM1 ve PLM2 programları ile gerçekleştirilmesi Şekil 3'de özetlenmiştir. Interp/8 ise büyük bilgisayarda mikroişlemcinin benzeşimini (*simulation*) sağlamaktadır. Interp/8 in verdiği çıkışlar 8008 in çıkışlarının aynı olduğundan mikroişlemcinin yapması gereken tüm işlemlerin yapılıp yapılamadığı sınanabilir.

Kurgu dilinde yazılmış olan Şekil 4'deki program bir mikroişlemcinin nasıl programlanacağı konusunda vereceğimiz ilk örnek olacaktır. A, B ve C nin bellekte üç ayrı yer belirleyen simge (*symbol*) olduğunu varsayalım. A ve B nin hangisinde daha büyük bir sayı varsa onun C ye yerleştirilmesi programımızın amacıdır. Bu programda geçen kurgu dili komutlarının anlamları:



Şekil 3.

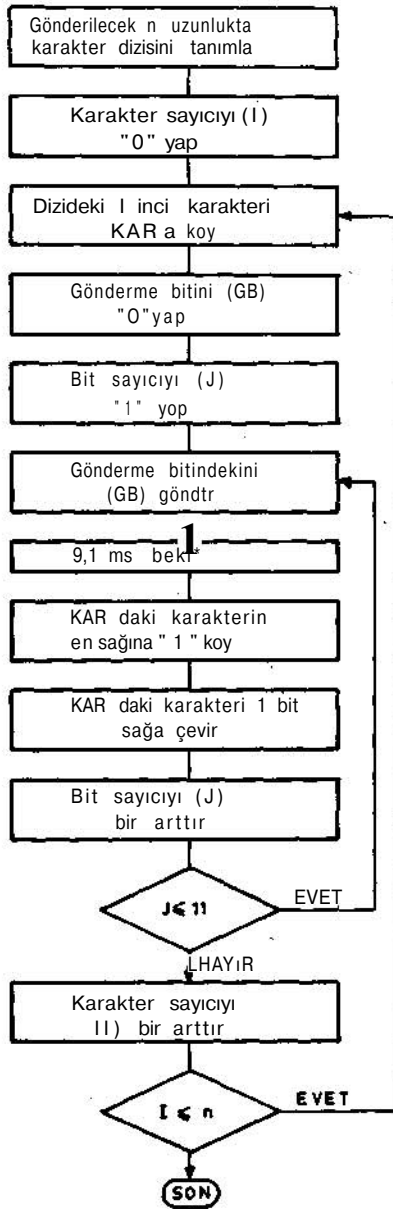
- SHL (simge) : Simgenin adresini, adres yazmacına yükle
- LAM : Adres yazmacında adresi bulunan yerdeki sayıyı birikeçe (*accumulator*) al
- CPM : Birikeçteki sayı ile adres yazmacında adresi bulunan yerdeki sayıyı karşılaştır.
- JFC (başlık) : Karşılaştırma sonucunda birikeçteki küçük veya eşitse programın (başlık) la adlandırılmış yerine git, aksi durumda devam et
- LMA : LAM nın tersi
- END : Program bitti olarak sıralanırsa Şekil 4'deki programın nasıl çalıştığı anlaşılır.

Görüldüğü gibi oldukça kolay bir işi yaptırmak için bile kurgu dilinde birkaç satırlık programlar yazmak gerekmektedir. Daha yüksek düzeyde olan PL/M gibi bir dilde ise Şekil A'deki programın yaptığı iş tek komutla yapılabilir.

PL/M Dilinde Örnek Program

Teleks, Teletype gibi yazılı bir metnin vuruşlar dizisi şeklinde gönderilmesi/alınması için kullanılan bir aygıtın mikroişlemcilerle gerçekleştirilmesi örneğini ele alalım. ASCII kodu ile belleğe yüklenen karakterlerin her birinin gönderilmesinden önce bir tane "0" in gönderilmesi, 8 bitten oluşan karakterin sağ baştan başlayarak gönderilmesi, karakter sonunda iki tane "1" gönderilmesi ve her bitten sonra 9,1 ms lik bir ara verilmesi istensin. Şekil 5'de görülen akış çizimi ve Şekil 6'daki PL/M dilindeki program, mikroişlemciye gerekli komutlar dizisini sağlar.

Şekil 6'da verilen PL/M dilinin FORTRAN, BASIC gibi dillere çok benzediği ilk bakışta görülmektedir. "DO LOOP" ların satırbaşlarının kaydırılmasıyla sınırlandırıldığı, TIME adlı



Şekil 5.

Şekil 6.

```

    DECLARE DRI DATA ('DENEME PROGRAMI')
    (KAR, I, J, GB) BYTE ;
    DO 1=0 TO LAST (DIZI)J
    KAR=DIZI (I) ;
    GB="0" ;
    DO 0=1 TO 11 ;
    OUTPUT (O)-GB ;
    CALL TIME (91) ;
    GB= KAR AND 1 ;
    KAR=ROR (KAR OR 1, 1) ;
    END ;
    END ;
  
```

programın "CALL TIME (91)" komutuyla çağrılıp 9,1 ms lik bekleme sağladığı, "KAR AND 1" ile KAR içinde en sağdaki bitin alındığı, "KAR OR 1" ile KAR içinde en sağdaki bitin "1" yapıldığı dikkat edilmesi gereken noktalarlardır.

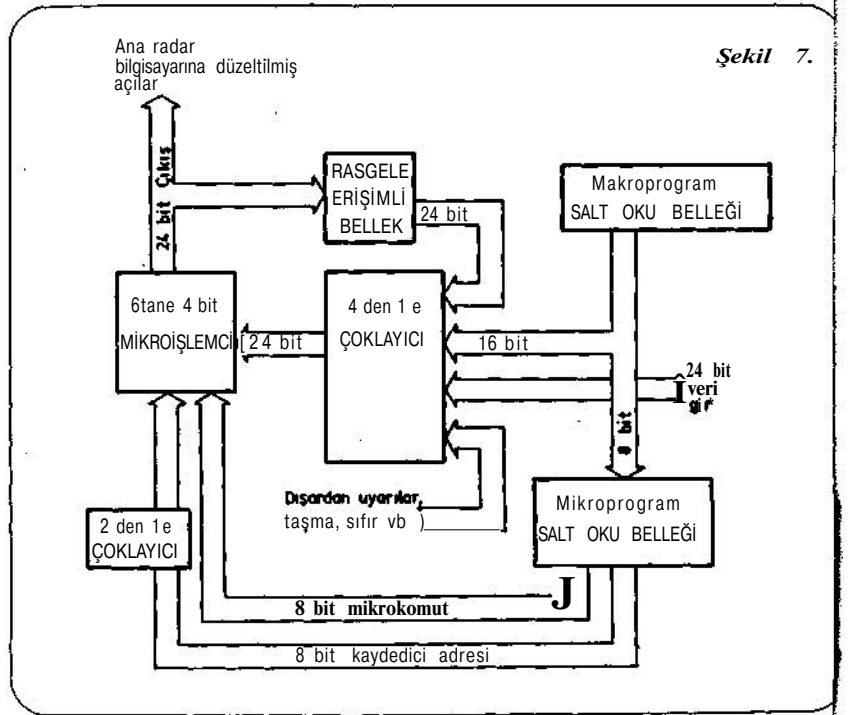
Mikroişlemci Uygulamaları
Mikroişlemcilerin yer aldığı uygulamalar genellikle oldukça karmaşık olan dizgenin çok iyi bilinmesini, akış çiziminin, programın, belleklerin boyutlarının, çevre aygıtlarının özelliklerinin ayrıntılarıyla incelenmesini gerektirmektedir. Burada sıralanan örnekler bu tür bir çalışmanın sergilenmesinden çok mikroişlemcilerin ne gibi alanlarda kullanılabildiğini belirtmek amacıyla anlatılmışlardır.

Radar Konum Çeviricisi
Gemilerdeki radarlar değişken olan konum koordinatlarını, değişmez bir dayanak koordinat dizgesine çevirmek zorundadır. Radarın bağlı olduğu bilgisayar bu işi yapabilir. Ne var ki 5 ms de bir çevrimin yapılmasının, 24 bitlik bir doğrulukla sinüslerin kosinüslerin

hesaplanmasının gerekliliği gözönüne alınırsa bilgisayarın merkezi işlem birimi üzerine çok büyük bir yük olacağı, büyük bir olasılıkla bilgisayarın başka bir iş yapamayacağı anlaşırlır.

Bir mikrobilgisayar için ise çok uygun olan bu görev 6 tane 4 bitlik mikroişlemcinin koşut (paralel) bağlanması, mikroprogram ve makroprogram taşıyan iki ayrı salt oku bellek kullanılması ile gerçekleştirilebilir (Şekil 7). Salt oku belleklerden birine yerleştirilen mikroprogram genel amaçlı bir komutlar dizisidir. Temel aritmetik işlemlerini yapar. Diğer salt oku bellek içindeki makroprogram ise özellikle bu uygulama için yazılmış olan bir programdır ve konum veren uç açının geçmiş değerlerinden, sinüs kosinüslerinden (Taylor açılımı ile elde edilir), değişim hızını veren türevden yararlanarak gerçek değerleri hesaplamak için gereken komut dizisinden oluşur. Rasgele erişimli bellek açılarının geçmiş değerlerinin saklandığı yer olarak kullanılmaktadır.

Şekil 7'de denetim imleri dışındaki önemli veri akış yolları



ları gösterilen dizge 10x11,25 santimetre boyutunda 38 baskı devre kartından oluşmaktadır. Hızının yaklaşık olarak 2 MHz (çevrim süresi 500 ns) olduğu, yazmaçlar arası işlemlerin 1 ps de, 24 bit çarpmanın 14 ps de gerçekleştiği belirtilebilir.

{Maşazalar için
Hesaplayıcı-Kasa

Elektromekanik hesaplayıcı-kasaların yerini almak üzere geliştirilen elektronik modeller üzerinde de mikroişlemcilerin önemli bir katkısı gözlenmektedir. Gelişimi incelemek için önce "mikroişlemci öncesi" bir elektronik modelin, "Data Terminal Systems" adlı ABD firmasının "DaCap 44" ünün özelliklerini belirtelim.

"DaCap 44" bağımsız çalışabilen, programlanan bir aygıttır. 1971'de geliştirilmiş ve 48 000 TL dolayında satılmıştır. Aritmetik/Mantık birimi, sekiz rasgele erişimli bellek, altı salt oku bellek, kü-

çük ve orta çapta tümleşik bazı tranzistor-tranzistor inantığı yongalarından oluşmaktadır. Bunların tümü ile güç düzenleyicisi, saat, yazıcının (printer) sürme mantığına ait devreler, kasa çekmecesinin bobinlerinin, sayısal gösterenin ve klavyenin sürme devreleri bir kartın üzerine yerleştirilmiştir. İkinci kartta klavyenin anahtarları ve klavye matrisinin tarama mantığı vardır. Üçüncü kart ise ışık yayan diyotlardan oluşan gösterge ve sürgü (latch) devrelerini kapsamaktadır.

Eöyle bir aygıtın mikroişlemcili bir yapı içinde gerçekleştirilmesi ise çok büyük ek olanaklar sağlamıştır (Şekil 8). 1973'de aynı firmanın geliştirmeye başladığı 300 serisi hesaplayıcı-kasaların "DaCap 44" den üstün oldukları yönler şöyle sıralanabilir:

- Maliyet yarıya inmiş 24 000 TL'ye satılabilir aygıtlar yapılmaya başlanmıştır.

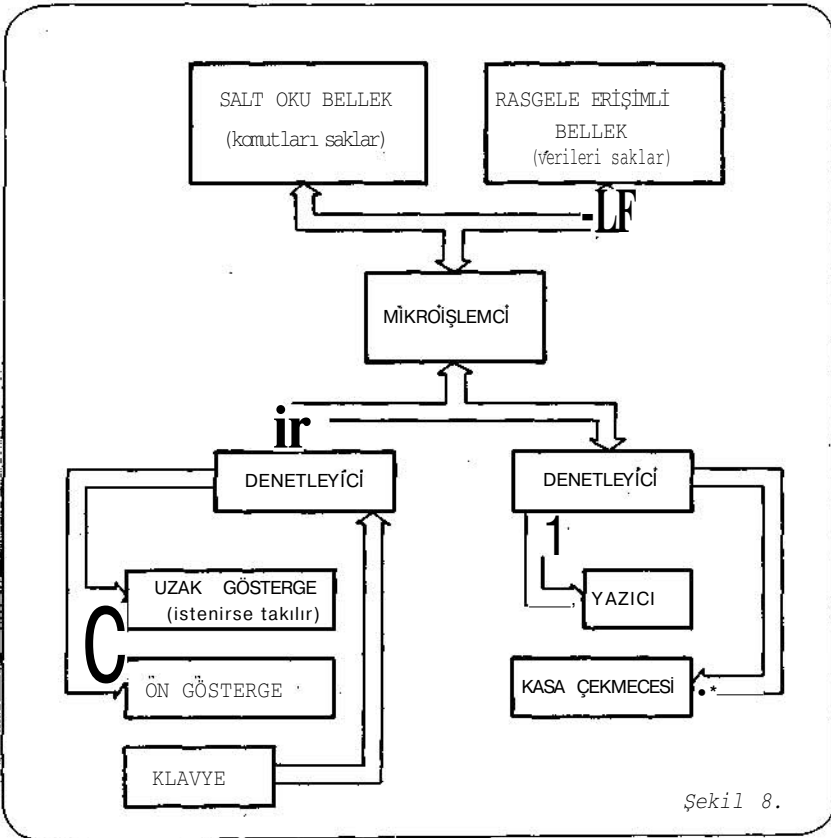
- Daha az sayıda tümleşik devreyle yapılması nedeniyle yedek parça, onarım sorunları azalmıştır.
- Aynı nedenle aygıt küçülmüş "DaCap 44" de dışarda duran piller 300 serisinde aygıtın içine yerleştirilebilmiştir (Elektrik kesilmesinde 1 saat çalışması, 24 saat belleğini koruması istenmektedir).

Hangi mikroişlemcinin seçilmesi gerektiği konusunda ise en önemli ölçüt çevre birimi denetleyicilerinin sayısıdır. 2 tane denetleyiciye sahip Rockwell PPS 4 bu bakımdan üstünlük göstermektedir. Denetleyicinin biri gösterge ve klavye ile ilgiliyken diğeri yazıcı ve kasa çekmecesini çalıştırmaktadır (Şekil 8).

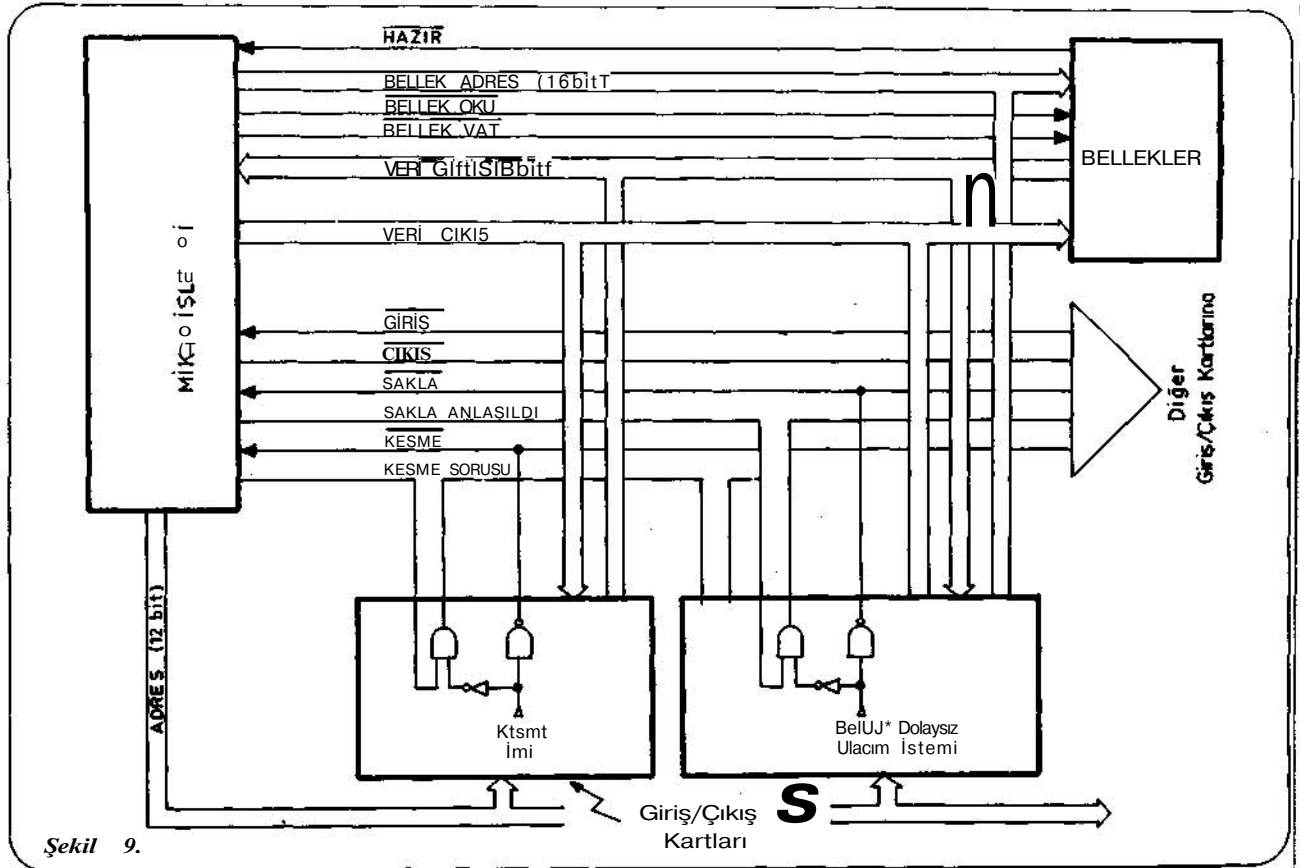
Süreç Denetim

Çeşitli amaçlarla tasarımılanan süreç denetim uygulamalarının en zor ve masraflı yönlerinden biri mikroişlemci için verimli bir programın hazırlanmasıdır. Bundan kaçınmanın yolu ise programlamanın öneminin az olduğu, donanıma dayalı bir dizge kurmaktır. Şekil 9'da böyle bir dizgenin belirgin özellikleri gösterilmiştir. Dizgede mikroişlemci olarak Intel 8080 seçilmiştir. Otuz tane giriş/çıkış kartıyla sürecin çeşitli yerlerinden bilgiler alınmakta, gerekli komutlar gönderilmektedir. Şekil 9'da soldaki giriş/çıkış kartında bir veri iletimi yapılacağını varsayalım. Kartın oluşturduğu "kesme imi" (interrupt signal) KESME iletkeni aracılığıyla bu istemi mikroişlemciye ulaştırır. Mikroişlemci ise KESME SORUSU yoluyla hangi kartın istemde bulunduğunu bulur. Dikkat edilirse "Kesme imi" olan karttan ileriye, KESME SORUSU nun iletilmesi engellenmiştir.

Şekil 9'da sağdaki kartın ise "belleğe dolaysız erişim" (direct memory access) yaptığını düşünelim. Bu durumda SAKLA iletkeni etkilemesi ve tıpkı KESME SORUSU gibi her karta birbiri ardınca girip çıkan SAKLA ANLAŞILDI imini beklemesi gerekmektedir. Daha sonra veriyi VERİ ÇIKIŞ'a vermesi ve

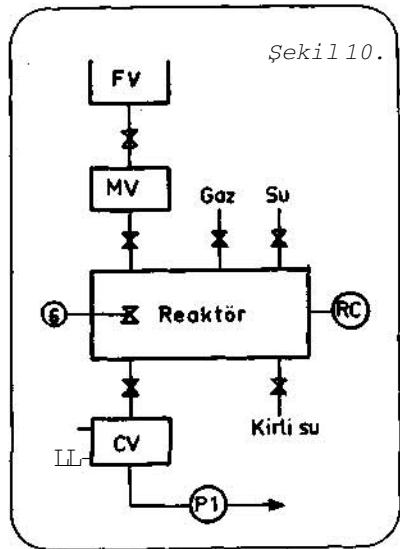


Şekil 8.



Şekil 9.

BELLEK YAZ komutunu göndermesi yada BELLEK OKU komutuyla birlikte veriyi VERİ GİRİŞ'ten alması yeterlidir. Her kartın yapısının birbirinin aynı olduğuna, şekilde iki örnek kartın yalnızca iş gören kısımlarının gösterildiğine değinelim.



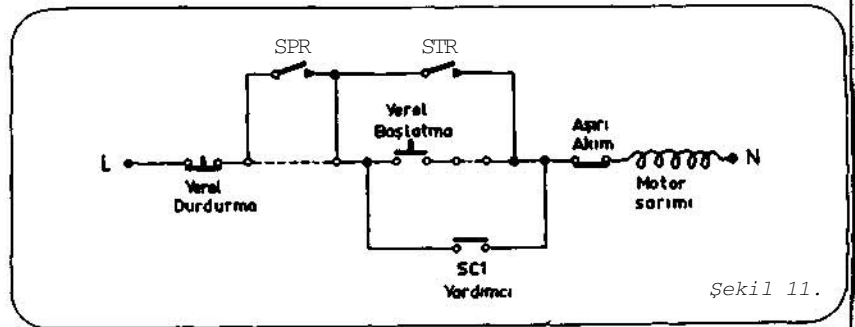
Şekil 10.

Ardarda bağlanmış 30 giriş/çıkış kartı ile oldukça karmaşık süreç denetim uygulamaları kolaylıkla gerçekleştirilebilmektedir. Bu yolun izlenmemesi, ayrıntılı bir programlama ile -diğer bir deyişle- yazılım üzerinde durarak aynı işlemin yapılmasında aşağıdaki seçeneklerle karşılaşmaktadır:

a) 8080 in FORTRAN dilinde yazılmış olan kurgucusu (assembler), oldukça yavaş olduğundan çok bilgisayar zamanı harcar ve altyordam

(subroutine) kullanılmasına olanak vermez.

- "Mikrobilgisayar geliştirme dizgesi" satın alınabilir, fakat bu durumda da sonuç çok uzun sürede alınır.
- Disk işletim dizgesine sahip bir minibilgisayarı olan herkes çok daha verimli olan bir başka yolu izleyebilir. Bu tür minibilgisayarların kendi kurgu (assembler) dillerini kendi makine dillerine çeviren bir makro kurgucuları vardır. 8080 in komut kümesini makro komut-



Şekil 11.

lar olarak tanımlamak yoluyla 8080 in kurgu dilini 8080 in makine diline çevirmek olanaklıdır. Böylece FORTRAN'ı temel alan bir programdan 10 kat daha hızlı ve minibilgisayarın belleğinde diğerinin 1/5 i kadar yer tutan özel bir kurgu dili oluşturulur. Kuşkusuz bu uygulamanın da en zor kısmı bu kurgucunun yazılmasıdır.

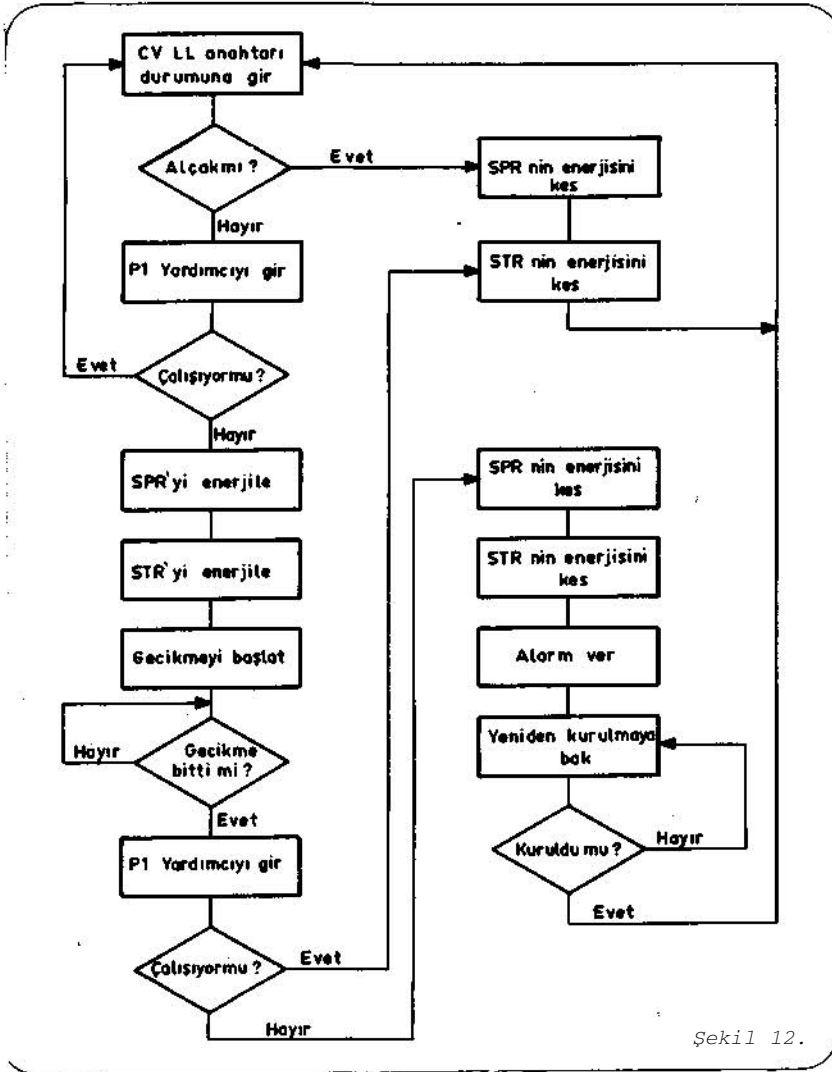
Pompa Denetimi

Bir kimyasal reaksiyonun oluşumu için Şekil 10'daki elektromekanik dizgenin denetimini ele alalım. FV sıvı hammadde deposu, MV ölçme kabı, S karıştırıcı motoru, CV ürünün

ara deposu, P1 ise ürünü depoya basan pompadır. Birkaç reaksiyondan sonra, reaksiyon kazanının su ile yıkanması gerekmektedir. Reaksiyonun tamamlandığı ise RC sezicisi aracılığıyla anlaşılabilir.

Örnek olarak P1 pompasının denetimini incelemek gerekirse Şekil 11'deki bağlantı şeması ve Şekil 12'deki akış çizimi pompanın çalışmasını açıklamaktadır.

Motor sarımına gerilimin yerel olarak verilmesi yada kesilmesi dışında özdevimsel çalışma sırasında ara depodaki sıvı düzeyinin alçalması üzerine SPR ve STR üzerinden motor çalıştırılır. Motorun çalışmasıyla SCI Yardımcı'nın devreyi



Şekil 12.

BASLIK	KOMUT	AÇIKLAMA
PUMP	LAI LLSU INP SUMA	CV deposundaki sıvının alt düzeyini veren sezici LL nin değerini birikece alır
	CPI 001 JFZ SPRF	Bundan 1 çıkartılır Sonuç 0 ise (sıvı alt sınırın altında ise) SPRF ye gidilir, aksi halde devam edilir
	LAI PIAUX INP SUMA CPI 001 JFZ RM	Yardımcının durumu birikere alınır Bundan 1 çıkartılır Sonuç 0 ise (motor çalışıyor) PUMP a gidilir, aksi halde devam edilir
	LAI SPRON OUT GPDA	SPR rölesinin enerjilenmesi için gerekli komutlardır
	LAI STRON OUT GPDA	STR rölesinin enerjilenmesi için gerekli komutlardır
	CALL PDLV	PDLV altıyordamıyla gecikme sağlandıktan sonra bunu izleyen komuta geçilir
	LAI PIAUX INP SUMA CPI 001 JFZ FAIL	Yardımcının durumu birikece alınır Bundan 1 çıkartılır Sonuç 0 ise (motor çalışmıyorsa) FAIL'e gidilir, aksi halde devam edilir
STRF	LAI STROF OUT GPDA	STR rölesinin enerjisi kesilir
	JMP PUMP	PUMP'a gidilir
SPRF	LAI STROF OUT GPDA	STR rölesinin enerjisi kesilir
	JMP STRF	STRF'ye gidilir
FAIL	LAI FLAMP OUT GPDA	Yanlış olduğunu bildiren lamba yakılır
	LAI SPROF OUT GPDA	SPR rölesinin enerjisi kesilir
	LAI STROF OUT GPDA	STR rölesinin enerjisi kesilir
	LAI R2B	Yeniden kuma düğmesinin durumu birikece alınır.
SRZ	INP SUMA CPI 001	Birikecek1 sayıdan 1 çıkartılır
	JFZ SRZ	Sonuç 0 ise (kuma yoksa) SRZ ye gidilir, aksi durumda devam edilir
	LAI FLOF OUT GPDA JUMP PUMP	Yanlış olduğunu bildiren lamba söndürülür PUMP'a gidilir
PDLV	LCI 000 LDI 000	C kaydedicisine 0 konur D kaydedicisine 0 konur
INCD	IND	0 kaydedicisinden sayı 1 arttırılır
	JFZ INCD	0 varsa INOC ye gidilir, aksi halde devam edilir
	DDC	C kaydedicisinden 1 çıkartılır
	JFZ INCD	0 varsa INDC ye gidilir, aksi halde devam edilir
	FRET	Ana programda ayrıldığımız yerden bir sonraki komuta dönülür.

Şekil 13.

kapatmasından sonra STR bırakılabilir. SPR ve STR nin belirli bir süre (4 s kadar) kapalı kalmasına karşın SCI Yardımcı çekmemişse alarm devresine gidilir.

Şekil 13 ise dizgenin Intel 8008 mikroişlemcisinin kurgu dilinde yazılmış programını vermektedir. Görüldüğü gibi pompa denetimi 43 komutlu bir programla gerçekleştirilebilmektedir. Şekil 10'daki tüm sistemin programı ise 800 program komutu ile yazılabilmekte ve 8 bitlik programlanan salt oku bellek içinde 1000-1500 yer tutmaktadır.