

**İ.T.Ü.
Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü**

**MİKROBİLGİSAYAR
LABORATUVARI
DENEY RAPORU**

Deney No: 2
Deney Adı: Örnek Programlar
Deney Tarihi: 17/10/2003
Grup: C5
Deneyi Yapanlar:
040000810 Zeynel Abidin Sezer
040000606 Ferhat Karakoç

Deneyi Yaptıran Öğretim Elemanı: Melike Erol

Amaç:İTÜ-Eğit için yararlı programların yazılmasıyla, program yazma ve geliştirme yeteneğinin geliştirilmesi.

Deney: Üç ana kısımdan oluşan deneyin ilk iki kısmını (Bellek alanı sınaması ve ara açma.) başarıyla tamamladık deneyin üçüncü kısmı olan blok aktarma ile ilgili olan programı makineye yazdık fakat hataları düzeltmek için zamanımız yeterli olmadı. Deneyin zamanında tamamlanamamasında bazı buyrukların makinede gerektiği gibi çalışmaması etkili olmuştur.

Donanım Sınaması İle İlgili Açıklamalar:

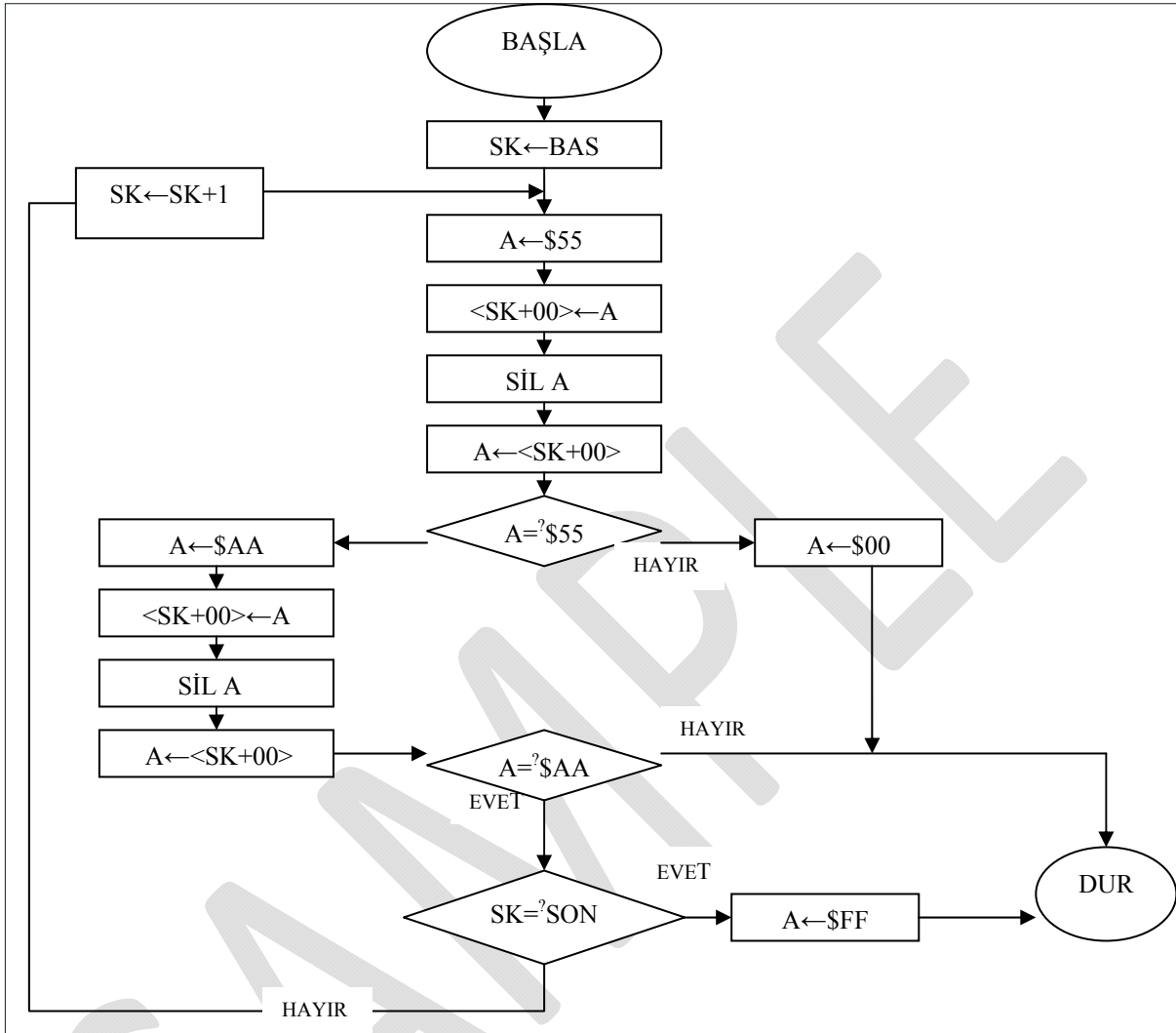
- Bir bellek alanını (8 bitlik sözcük uzunluğu) sınamak için o alana \$55 ve \$AA rakamları yazılır ve okunur. Böylece her bir hücreye '0' ve '1' yazılmış ve her bir hücreden '0' ve '1' okunmuş olur.
- Bizim yazdığımız programla \$0000'dan \$FFFF'e kadar tüm bellek alanını sınamayız. Çünkü belirtilen alanda bizim programımızda yer alıyor. Bu yazma ve okuma işlemini programımızın olduğu alana da uygularsak o alandaki programı kaybederiz dolayısıyla program çalışmaz. Ancak, bilgisayarın ve bizim programımızın çalışabileceği asgari programların olduğu bölgelerde "Zaten bellek alanı mevcuttur." kabulünü yaparsak sınamış sayılabiliriz.

DONANIM SINAMA PROGRAMI

BAŞLA	YÜK	SK,<\$0000>	FE 00	00	0010-0011-0012	başlangıç adresini SK' ya yükle
DEVAM	YÜK	A, \$55	86 55	0013-0014	A' ya \$55 yükle	
	YAZ	A,<SK+00>	A7 00	0015-0016	sıradaki belleğe yaz	
	SİL	A	4F	0017-		
	YÜK	A,<SK+00>	A6 00	0018-0019	sıradaki belleği oku	
	KAR	A,\$55	81 55	001A-001B		
	DED	HATA	26 15	001C-001D	okuma yada yazma yapılmadıysa programı hata ile bitir	
	YÜK	A,\$AA	86 AA	001E-001F	A' ya \$AA yükle	
	YAZ	A,<SK+00>	A7 00	0020-0021	sıradaki belleğe yaz	
	SİL	A	4F	0022-		
	YÜK	A,<SK+00>	A6 00	0023-0024	sıradaki belleği oku	
	KAR	A,\$AA	81 AA	0025-0026		
	DED	HATA	26 0A	0027-0028	okuma yada yazma yapılmadıysa programı hata ile bitir	
	ART	SK	O8	0029-	sıradaki belleğe geç	
			BC 00			
	KAR	SK,<\$0002>	02	002A-002B-002C		
	DED	DEVAM	26 E4	002D-002E	istenen bellek kümesi kontrol edildiye programı bitir	
	YÜK	A, \$FF	86 FE	002F-0030	hata' ya geçilmedi. A' ya \$FF yükle. (bellek kümesi kullan	
	DHZ	BİTTİ	20 02	0031-0032-	programı bitir	
HATA	YÜK	A, \$00	86 00	0033-0034	A' ya \$00 yükle. (bellek kümesi kullanılamaz)	
BİTTİ	DUR		3F	0035-		

Program başlangıçta DKE komutu da kullanılarak yazılmıştı. Fakat bu komut düzgün çalışmadığından program yukarıdaki şekilde yeniden düzenlendi. Bu nedenle sınanılması istenen bellek kümesinin son elemanı sınanmamaktadır.

Donanım Sınama Programına ilişkin akış diyagramı:



Yaptığımız çözümde A' nın FF olması donanımın kullanılabilir olduğunu gösteriyor.

Sonuç: Bu deneyde, belirtilen adreslerde *yaz/oku* bellek alınının mevcut olup olmadığı nasıl sınırlar onu öğrendik. Ayrıca bazı küçük programların mühendislik uygulamalarında bizleri nasıl büyük zahmetlerden kurtarabileceğini ve meselelere mühendislik bakış açısıyla bakabilmeyi (kısmen) öğrendik.

Ara Açma:

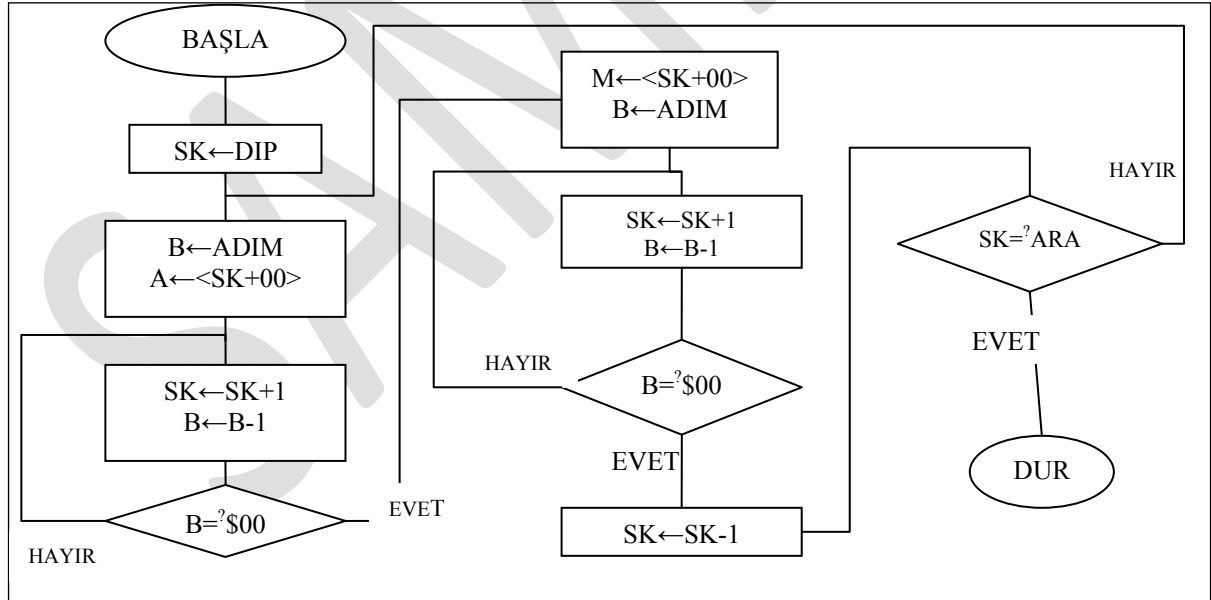
Amaç:

Bellekte var olan bir programa parça eklemek için programın yeniden yazılması yerine istenilen yerde ara açılmasını sağlayan ara açma programını yazmak ve test etmek.

ARA AÇMA PROGRAMI

BAŞLA	YÜK	SK, <\$0002>	FE 00	02	4000-4001-4002	SK' ya son adresi yükle
GERİ	YÜK	B, <\$0006>	F6 00 06	4003-4004-4005	B' ye adım sayısını yükle	
	YÜK	A, <SK+00>	A6 00	4006-4007	ötelenen olan kümedeki son elemanı a' ya yükle	
GERİ1	ART	SK	O8	4008		
	AZT	B	5A	4009		
	KAR	B, \$00	C1 00	400A-400B		
	DED	GERİ1	26 FA	400C-400D		
	YAZ	A, <SK+00>	A7 00	400E-400F	sıradaki elemanı adım sayısı kadar öteye yaz	
	YÜK	B, <\$0006>	F6 00 06	4010-4011-4012	B' ye adım sayısını yükle	
GERİ2	AZT	SK	O9	4013		
	AZT	B	5A	4014		
	KAR	B, \$00	C1 00	4015-4016		
	DED	GERİ2	26 FA	4017-4018		
	AZT	SK	O9	4019	ötelenen olan kümenin sıradaki ötelenecek son elemanı	
			BC 00			
	KAR	SK, <\$0004>	04	401A-401B-401C	ARA' ya gelindi mi?	
	DEB	GERİ	2E E4	401D-401E	gelinmediyse devam et	
	DUR		3F	401F		

Bu programın akış diyagramı şu şekildedir:



Blok Aktarma

Amaç:

Belleğin belli bir bölgesinde bulunan veri kümesinin başka bir yere aktarılması.

Bu amaca yönelik tek bir algoritma tam olarak çözüm oluşturmadı. Çünkü varış yerinin kopyalanacak yerden ileride veya geri olma durumunda veri kaybı olabilmektedir.

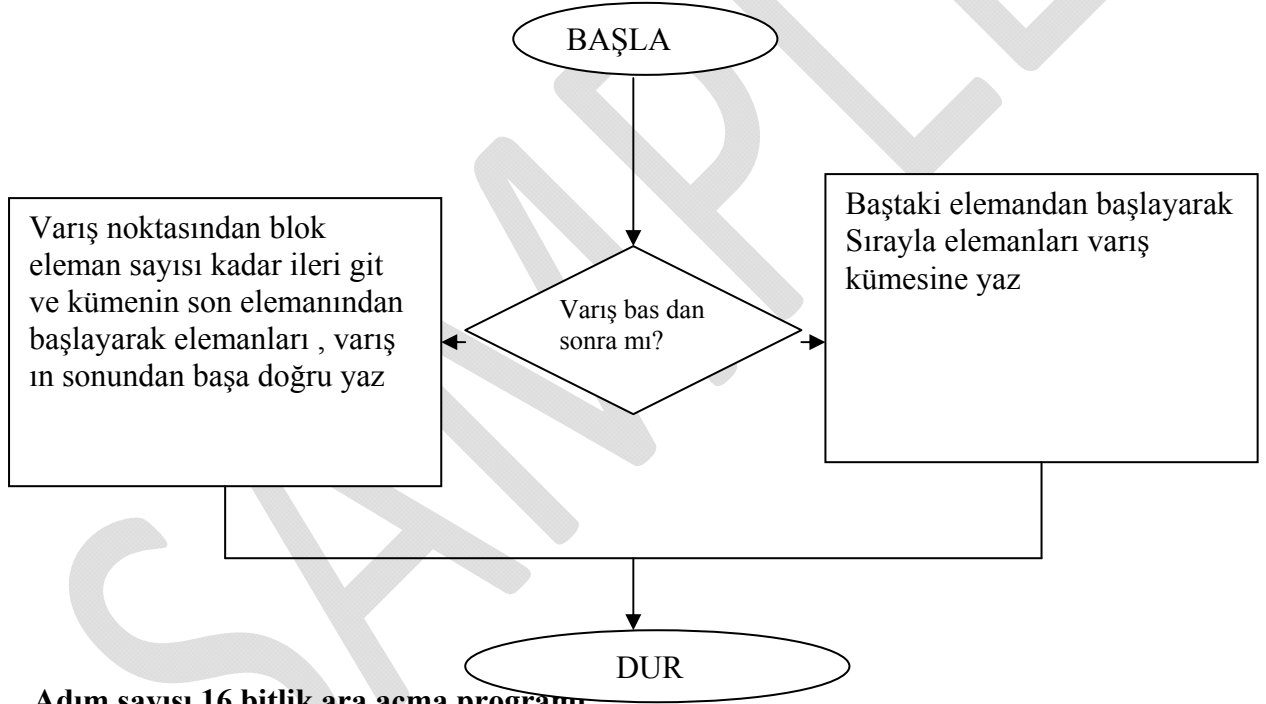
Bu nedenle bir programın içinde iki algoritma kullanılmıştır.

BLOK AKTARMA PROGRAMI

			FE 00		
BAŞLA	YÜK	SK,<\$0000>	00	4000-4001-4002	başlangıç adresini SK ' ya yükle
	SİL	A	4F	4003	
D1	ART	SK	O8	4004	
	ART	A	4C	4005	
			BC 00		
	KAR	SK,<\$0002>	02	4006-4007-4008	
	DEK	D1	2D F9	4009-400A	eleman sayısını belirle
			FE 00	400B-400C-	
	YÜK	SK,<\$0004>	04	400D	varış adresi SK' ya yükle
D2	ART	SK	O8	400E	
	AZT	A	4A	400F	
	KAR	A,\$00	81 00	4010-4011	
	DEB	D2	2E FA	4012-4013	yeni bellek kümesinin son adresini belirle
			FF 00		
	YAZ	SK,<\$0006>	06	4014-4015-4016	yeni bellek kümesinin son adresini 0006 nolu bellek gözünüme
			FE 00		
	YÜK	SK,<\$0000>	00	4017-4018-4019	
			BC 00	401A-401B-	
	KAR	SK,<\$0004>	04	401C	eğer kopyalanacak adres kopyanılanacak yerden büyükse
	DKE	DURUM2	2F 28	401D-401E	durum
					için yazılan diğer algoritmaya git
			FE 00		
	YÜK	SK,<\$0004>	04	401F-4020-4021	
			FF 00		
	YAZ	SK,<\$000A>	0A	4022-4023-4024	varış adresini 000A nolu bellek gözüne kopyala
			FE 00		
	YÜK	SK,<\$0000>	00	4025-4026-4027	
			FF 00	4028-4029-	
	YAZ	SK,<\$0008>	08	402A	baş adresini 0008 nolu bellek gözüne kopyala
			FE 00	402B-402C-	
G1	YÜK	SK,<\$0008>	08	402D	
	YÜK	A,<SK+00>	A6 00	402E-402F	kopyalanacak sıradaki elemanı al
	ART	SK	O8	4030	
			FF 00		
	YAZ	SK,<\$0008>	08	4031-4032-4033	bir sonraki elemanın kopyalanması için
			FE 00		
	YÜK	SK,<\$000A>	0A	4034-4035-4036	
	YAZ	A,<SK+00>	A7 00	4037-4038	alınan elemanı yaz
	ART	SK	O8	4039	
			FF 00	403A-403B-	
	YAZ	SK,<\$000A>	0A	403C	bir sonraki elemanın bir sonraki yeni yere yazılması için
			FE 00	403D-403E-	
	YÜK	SK,<\$0008>	08	403F	
			BC 00		
	KAR	SK,<\$0002>	02	4040-4041-4042	
	DEB	BİTTİ	2E 2A	4043-4044	kopyalama bitti mi?
	DHZ	G1	20 E4	4045-4046	bitmediyse devam
			FE 00		
DURUM2	YÜK	SK,<\$0002>	02	4047-4048-4049	
			FF 00	404A-404B-	
	YAZ	SK,<\$0008>	08	404C	son elemanın adresini 0008 nolu bellek gözüne kopyala
			FE 00	404D-404E-	
	YÜK	SK,<\$0006>	06	404F	
			FF 00		
	YAZ	SK,<\$000A>	0A	4050-4051-4052	yeni yerin son adresini 000A ' ya kopyala
			FE 00		
G2	YÜK	SK,<\$0008>	08	4053-4054-4055	
	YÜK	A,<SK+00>	A6 00	4056-4057	kopyalanacak son elemanı al

AZT	SK	O9	4058	
		FF 00	4059-405A-	
YAZ	SK,<\$0008>	08	405B	daha sonra bir önceki elemanın kopyalanması için
		FE 00	405C-405D-	
YÜK	SK,<\$000A>	0A	405E	
YAZ	A,<SK+00>	A6 00	405F-4060	kopyalanacak son elemanı yaz
AZT	SK	O9	4061	
		FF 00		
YAZ	SK,<\$000A>	0A	4062-4063-4064	daha sonra yeni yerin son elemanından bir önceki yere git
		FE 00		
YÜK	SK,<\$0008>	08	4065-4066-4067	
		BC 00	4068-4069-	
KAR	SK,<\$0000>	00	406A	kopyalama bitti mi?
DEK	BİTTİ	2D 02	406B-406C	
DHZ	G2	20 E4	406D-406E	
BİTTİ	DUR	3F	406F	

Program temel olarak şu şekilde çalışmaktadır:



Adım sayısı 16 bitlik ara açma programı.

16 BİTLİK ARA AÇMA PROGRAMI

BAŞLA	YÜK	SK,<\$0002>	son adresi SK' ya yükle
GERİ	YÜK	AB,<\$0006>	
	YAZ	AB,<\$0010>	adım sayısını 0010' a kopyala
	YÜK	A,<SK+00>	
	YAZ	A,<\$0008>	ötelenecek olan kümenin sıradaki son elemanını 0008'e yaz
GERİ1	ART	SK	
	AZT	<\$0010>	
	YÜK	AB,<\$0010>	
	KAR	AB,\$0000	
	DED	GERİ1	ötelenecek olan sıradaki elmanı adım sayısı kadar ileri yazmak için
	YÜK	A,<\$0008>	
	YAZ	A,<SK+00>	sıradaki eleman adım sayısı kadar ileri yazıldı
	YÜK	AB,<\$0006>	0010 daki adım sayısı silindiği için yeniden yükle

GERİ2 YAZ AB,<\$0010>
AZT SK
AZT <\$0010>
YÜK AB,<\$0010>
KAR AB,\$0000
DED GERİ2
AZT SK
KAR SK,<\$0004>
DEB GERİ
DUR

en son ötelenen elemandan bir önceki elamana ulaşmak için
ara açma bitti mi?

Çalışma mantığı 8 bitlik ara açma programıyla aynı olup, 16 bit olduğundan ek olarak bellek kullanılmıştır.

SAMPLE