

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



**MİKROBİLGİSAYAR**  
**LABORATUVARI**  
**DENEY RAPORU**

**Deney No** : 3  
**Deney Adı** : Altprogram ve Yığın İşlemleri  
**Deney Tarihi** : 20.10.2011  
**Grup** : 10  
**Deneyi Yapanlar** : 040080153 Serkan Güler  
040080322 Osman Boyacı  
040090533 Abdullah Aydeğer

**Deneyi Yaptıran Araştırma Görevlisi :** Hasan Kıvrak

## Deneyin İçeriği

Deneyde MC6802 mikroişlemcisine ait yığın yapısı incelenmiş olup, altprogram çağrılarını yapılarak yığın kullanımı yapılmıştır. Deneyde öncelikle yığın yapısını inceleyebilmek için Altprogram Yığın Etkileşimi adlı deney gerçekleştirilmiş olup, daha sonra 4 Bitlik Sayıların Çarpımı adlı deney sayıların çarpımını alt programda yapacak şekilde yani yığın kullanacak şekilde yazılmıştır. Son olarak ise Rekürsif Çalışan Altprogram adlı deneyde fonksiyon verilen parametreye göre verilen parametreye kadar olan sayıların toplamını bulup buna 1 eklemektedir.

## Deneyler ve Sonuçları

- *Altprogram – Yığın Etkileşimi*

	Etiket	Komut	Makine Kodu
1		LDS \$5F00	8E 5F 00
2		TSX	30
3		JSR ALT1	BD 40 50
4		SWI	3F
5			
6	ALT 1	TSX	30
7		JSR ALT2	BD 40 60
8		RTS	39
9			
10	ALT2	TSX	30
11		RTS	39

Şekil 1. Altprogram-Yığın Etkileşimi Kodları

Adım	Sıralama Kütüğü	Program Sayacı
1	0	4003
2	5F01	4004
3	5F02	4050
4	5EFF	4051
5	5EFF	4060
6	5EFD	4061
7	5EFD	4054
8	5EFD	4007

Şekil 2. Program çalıştırılırken adım adım SK ve PS değerleri

Bu program çalıştırıldıktan sonra öncelikle YG'yi SK'ya atama işlemini gerçekleştiriyor. Daha sonra alt program çağrısı yapılıyor. Bu altprogramda YG'nin yeni değeri SK'ya atanıyor ve diğer bir altprogram çağrısı yapılıyor. Bu 2.altprogramda da YG SK'ya atanıp RTS komutuyla altprogramlardan geri dönülüyor.

- **4 Bitlik Sayıların Çarpımı**

	Etiket	Komut	Makine Kodu	Açıklama Satırı
1		LDS #003F	8E 00 3F	
2		LDAA 0000H	B6 00 00	
3		PSHA	36	
4		LDAA 0001H	B6 00 01	
5		PSHA	36	
6		JSR ÇARP	BD 40 50	
7		INS	31	
8		INS	31	
9		SWI	3F	
10				
11	ÇARP	LDAA \$00	86 00	A akümülatörüne toplam yazılacağı için 0 yüklenir
12		STAA M[11]	97 11	Ve A akümülatörüyle işlem yapılacağı için bir yere yazılır
13		LDAA \$04	86 04	4 bit için çarpma işlemi yapılacağından
14		STAA M[10]	97 10	10 adresine 4 yazılır
15		INS	31	Yığından veri çekmek için 2 kez YG değeri artırılır
16		INS	31	
17		PULB	33	Yığından veri B akümülatörüne çekilir
18		PULA	32	Ardından 2. veri A'ya çekilir.
19		ANDA \$0F	84 0F	Yığına atılan verilerin yalnızca ilk 4bitiyle işlem yapmak
20		ANDB \$0F	C4 0F	için çekilen veriler 0F ile VE'lenir
21	BAŞ	LSRA	44	Burda A sağa kaydırılır
22		BCC İLERİ	24 07	Elde oluşmamış ise İLERİ dallanılır
23		PSHA	36	Dallanma olmazsa A yığına atılır
24		LDAA M[11]	96 11	A akümülatörü 11 adresinden okunur
25		ABA	1B	A'ya A+B toplamı yazılır
26		STAA M[11]	97 11	A akümülatörünün içeriği 11 adresine geri yazılır
27		PULA	32	A yığından çekilir
28	İLERİ	ASLB	58	Burda B akümülatörü sola kaydırılır
29		DEC M[10]	7A 00 10	10 adresindeki döngünün dönmesi gereken miktar 1 azaltılır
30		BEQ ALTDON	27 02	10 adresi sıfırlanmışsa ALTDON'e dallanılacak
31		BRA BAŞ	20 EE	Döngü BAŞ'tan devam edecek
32	ALTDON	DES	34	YG 4kez azaltılacak
33		DES	34	
34		DES	34	
35		DES	34	
36		RTS	39	Altprogramdan geri dönülecek

Şekil 3. 4Bitlik sayıların çarpımı için gerekli kodlar

Program yazılan açıklama satırıyla anlatılmaya çalışılmıştır. Burada föylerde verilmiş olan algoritma gerçekleştirilmiştir. Bu algoritma Şekil 4 te gösterilmiştir.

```

par1 ve par2 carpilacak sayilar
sonuc := 0
for basamak sayisi kadar
    par1'i saga kaydir
    if elde
        sonuc := sonuc + par2
    endif
    par2'yi saga kaydir
endfor

```

Şekil 4. 4Bitlik sayıların çarpımı için kullanılan algoritma

- **Rekürsif Çalışan Altprogram**

	Etiket	Komut	Makine Kodu	Açıklama Satırı
1		LDS \$0050	8E 00 50	
2		LDAA #04	86 04	
3		PSHA	36	
4		JSR FACT	BD 40 50	
5		INS	31	
6		SWI	3F	
7				
8	FACT	PSHB	37	
9		STX 0000H	FF 00 00	
10		LDAB 0000H	F6 00 00	
11		PSHB	37	
12		LDAB 0001H	F6 00 01	
13		PSHB	37	
14		TSX	30	
15		LDAB 5,X	E6 05	Buradan itibaren bizim hazırladığımız kodlar yer almaktadır
16		CMPB \$00	C1 00	Eğer B akümülatörü sıfır ise
17		BEQ İLERİ	27 09	İLERİ dallanılacak
18		DEC B	5A	Dallanma olmaz ise, B bir azaltılacak
19		ABA	1B	A akümülatörüne A+B atanır
20		PSHB	37	B yığına atılır
21		JSR FACT	BD 40 50	Altprogram yeniden çağrılır
22		INS	31	SK 1 arttırılır
23		BRA F-SON	20 01	Koşulsuz olarak F-SON'a dallanılır
24	İLERİ	INCA	4C	Eğer B akümülatörü sıfır ise, A 1 arttırılır
25	F-SON	PULB	33	Buradan itibaren hazır bulunan kodlar yer almaktadır
26		STAB 0001H	D7 01	Altprogramdan dönüş kodlarını içermektedir
27		PULB	33	
28		STAB 0000H	D7 00	
29		LDX	DE 00	
30		PULB	33	
31		RTS	39	

Şekil 5. Rekürsif Çalışan Altprogram

Bu programda ise başta A akümülatörüne yüklenmiş olan değere kadar olan sayılar toplanıp bunlara 1 eklenerek tekrar A akümülatörüne yazılır. Programda B akümülatörü sıfır olana kadar A ile B toplanıp A ya yazılmaktadır. Yani  $A += B$  yapılmaktadır. Burada B akümülatörü bu işlemin her yapılışında bir azaltılmış olmaktadır. Sonuç olarak altprogramdan döndüğünde gerekli değer A akümülatöründen okunabilmektedir.

Altteki resimde ise rekürsif olarak çalışan programımızın takibi görülmektedir. Listenin fazla uzamaması için A akümülatörüne programın başında 2 değeri verilmiş olup böylece rekürsif fonksiyonumuz 2 kere çağrılmıştır.

Adım	Yığın Göstergesi	Program Sayısı	ACC A
1	0050	4003	4
2	0050	4003	2
3	004F	4006	2
4	004D	4050	2
5	004C	4051	2
6	004C	4054	2
7	004C	4057	2
8	0048	4058	2
9	0048	4056	2
10	004A	405C	2
11	004A	405D	2
12	004A	405F	2
13	004A	4061	2
14	004A	4063	2
15	004A	4064	2
16	004A	4065	3
17	0049	4066	3
18	0047	4050	3
19	0046	4051	3
20	0046	4054	3
21	0046	4057	3
22	0043	4058	3
23	0043	4056	3
24	0044	405C	3
25	0044	405D	3
26	0044	405F	3
27	0044	4061	3
28	0044	4063	3
29	0044	4064	3
30	0044	4065	3
31	0043	4066	3
32	0041	4050	3
33	0040	4051	3
34	0040	4054	3
35	0040	4057	3
36	003F	4058	3
37	003F	4056	3
38	003E	405C	3
39	003E	405D	3
40	003E	405F	3
41	003E	4061	3
42	003E	4063	3
43	003E	406D	4

Şekil 6. Rekürsif programın takibi