

MikBil

Bu kitap kapsamında, okuyucuya, önce mikroişlemci kavramı öğretilmeye çalışılmıştır. Bu aşamada, öğrenmeyi kolaylaştırmak ve okuyucuyu belli bir mikroişlemciye yönelmemek için soyut bir mikroişlemci üzerinde çalışılmıştır.

Okuyucunun, soyut mikroişlemci üzerinde deney yapabilmesini sağlamak üzere bir sınamaya yazılımı (simulasyon yazılımı) hazırlanmıştır. PC üzerinde, Windows ortamında çalışan bu yazılım sayesinde, okur Örnek MİB'nin tüm özelliklerini sınamaya olanağı bulacaktır.

Sınamaya yazılımı kullanılarak;

- u Adresleme yöntemleri öğrenilebilir
- u Buyruklara ilişkin tüm özellikler, örneğin
 - u Aktarma buyruklarının işlevleri,
 - u Aritmetik buyruklar
 - u Mantıksal buyruklar,
 - u Tüm işlem buyrukları,
 - u Öteleme ve döndürmeye ilişkin buyruklar
 - u Karar verme işlemi
 - u Dallanma ve bağlanma

buyrukları denenebilir.

- u Altprogram,
- u Kesme
- u Giriş/çıkış işlemleri

irdelenebilir.

2 - MikBil

19.1 MikBil'in Özellikleri

MikBil adını verdiğimiz sınamaya yazılımı, Windows ortamında çalışmaktadır. Programın tek bir parça olarak oluşturulmuştur ve program adı MikBil.exe dir. MikBil, Windows ortamında istenilen bir yere kopya edip çalıştırabilir.

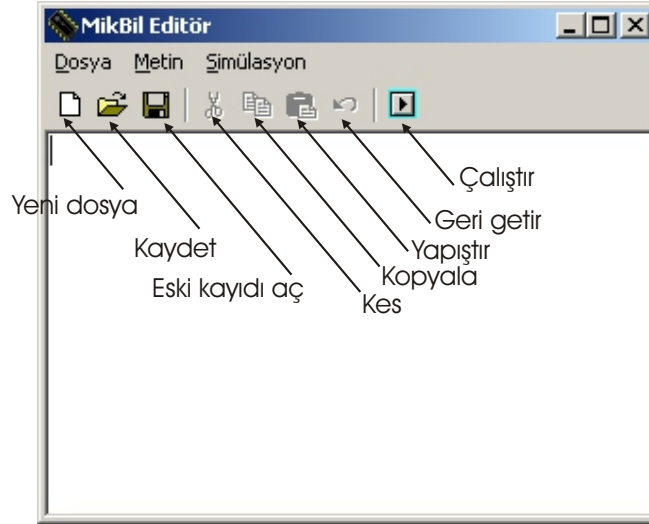
MikBil temelde iki kısımdan oluşmaktadır:

- u Yazım kısmı ve
- u Sınama kısmı

Bu program kısımlarına ilişkin özellikler sırasıyla anlatılmıştır.

19.1.1 MikBil Yazım Programı

MikBil çalıştırıldığında, kullanıcının karşısına, yazım ekranı gelir. Şekil-19.1'de bu ekran gösterilmiştir.



Şekil-19.1: MikBil yazım ekranı

19.1 MikBil'in Özellikleri - 3

Yazım kısmı, kullanıcıya simgesel dilde program yazma olanağı sunar. Bir başka deyişle kaynak programın yazılmasına ortam sağlar. Yazım programı, Windows ortamında çalışan Notepad yazım programına benzer özellikler içermektedir. Dosya seçeneği seçildiğinde, yeni dosya açma, eski bir dosyayı yükleme, üzerinde çalışılan dosyayı kaydetme seçenekleri kullanıcıya sunulur. Şekil-19.1'de düğmelerin görevleri gösterilmiştir.

Yazım programı, Notepad yazım programının yetenekleri içinde bulunan, metin düzenleme özelliklerinin tümünü kapsamaktadır. Metnin belli bir kısmı, kesilebilmekte, kesilen kısım bir başka yere kopyanabilmektedir.

Simgesel dilde program yazmanın, kendisine özgü kurallarının olduğu 11. Bölümde anlatılmıştı. MikBil, 11. Bölümde tanımlanan kurallara uygun, hatta daha esnek olarak, program yazmaya olanak sağlamaktadır. MikBil'in yazım kalıbı Şekil-19.2'de gösterilmiştir.

Etiket	Komut	İşlenen	Açıklama

Şekil-19.2: MikBil simgesel dil yazım kalıbı

MikBil etiket ve işlenen boyunu esnek tutma olanağı vermektedir. Ancak, çok uzun etiket ve işlenen boyu kullanılması önerilmemektedir. Yazım kalıbındaki alanlar arası geçişler için sekme tuşu kullanılması önerilir.

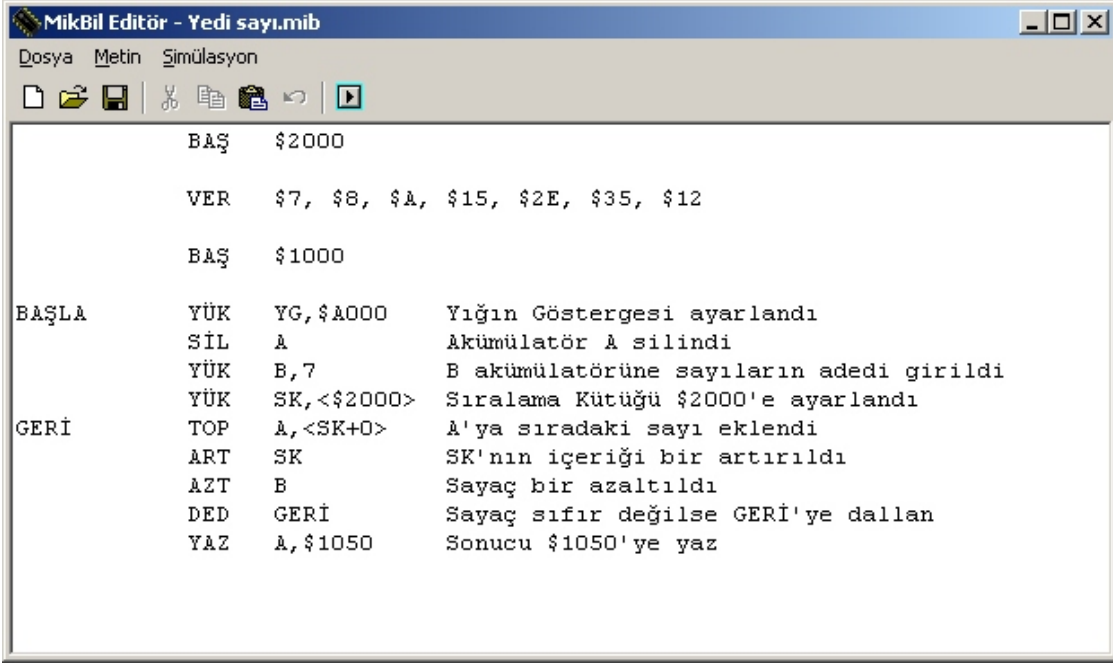
Yazılacak programın başlangıç noktasını belirlemek için BAŞLA etiketinin kullanılması zorunludur. Programın son noktasını belirlemek için SON buyruğunun kullanılmasına gerek yoktur.

'*' işareti ile başlayan satırlar açıklama satırı olarak kabul edilir. Açıklama kısmı için ";" ya da ":" kullanmaya gerek yoktur. Açıklamanın yerinin sekme ile belirlenmesi yeterlidir.

MikBil yazım programı kullanılarak gerçekleştirilmiş bir örnek program Şekil-19.3'te gösterilmiştir.

Yazım işlemi tamamlanmış bir programı makine diline çevirmek için "Çalıştır" düğmesine basılmalıdır. Eğer programda, yazım yanlışı varsa, yanlışı pencerenin alt kısmında kullanıcıya bildirilir. Sınama kısmına geçilebilmesi için, program kaynak kodunun yanlışız olması gerekir.

4 - MikBil



Şekil-19.3: MikBil Yazım programı kullanılarak gerçekleştirilmiş bir program

19.1.1.1 Sözde Komutlar

MikBil'in tanıdığı sözde komutlar aşağıda arasıyla tanıtılmıştır:

EŞT

Bir değişkenin eşit olduğu değeri tanımlar. Bu değer bir veri ya da adres olabilir. Kullanım şekli şöyledir:

```
DEĞİŞKEN    EŞT    $1000
```

BAŞ

Kendisinden sonra gelen sözde ya da gerçek komutun yerleşeceği bellek alanının başlangıç adresini belirtir. Örneğin;

19.1 MikBil'in Özellikleri - 5

BAŞ \$1000 satırından sonra, program başlıyor ise, derleme işleminden sonra programın yazılmaya başlayacağı bellek gözünün adresi \$1000 olacaktır.

BAŞ \$1000 sözde komutundan sonra, VER sözde komutu ile veriler tanımlanmış ise, bu durumda, verilerin bellekte yazılmaya başlayacağı adresi belirtir.

Program içinde, birden çok BAŞ tanımlanabilir. Ancak, bu tanımlamalar yapılırken mantık hatasına düşülmemelidir.

Şekil-19.3'e dikkatle bakıldığında BAŞ sözde komutu iki satırda görülmektedir. İlk satırda görülen BAŞ sözde komutu, yazılacak olan verilerin nereye yazılacağı belirtmek için kullanılmıştır. üçüncü satırda görülen BAŞ sözde komutu, programın bellekte yerleşeceği ilk adresi göstermektedir.

VER

Belleğe yerleştirilecek verileri tanımlamak üzere kullanılır. Belleğe yazılacak veriler, birbirinden " , " ile ayrılmalıdır. VER sözde komutunun kullanılış biçimi şöyledir:

VER \$12, \$12, \$AA

Şekil-19.2'de verilen örnekte, VER sözde komutu ile yedi sayının belleğe nasıl yazıldığı gösterilmiştir.

YER

Salt adres belirtmeksizin, veriler için bellekte yer ayırmaya yarar. Kullanış biçimi aşağıdaki gibidir.

DEĞER-1	YER	1	; 1 sekizli alan, kullanılmayacak
DEĞER-2	YER	2	; 2 sekizli alan, kullanılmayacak
YEDEK	YER	UZUNLUK	; 8 sekizli alan (UZUNLUK = 8)

SON

Çevirme işleminin sona ereceği noktayı gösterir. Bu komut yazılmaz ise, çevirici, yazılmış olan tüm satırları işleme katar.

6 - MikBil

19.1.1.2 İşlenen Üzerinde İşlemler

İşlenen üzerinde aritmetik ve mantıksal işlemler yapmak kullanıcıya kolaylık sağlamaktadır. Bu nedenle, MikBil çeviricisine bu tür özellikler eklenmiştir. Etiketler üzerinde yapılabilecek işlemler örnekleriyle birlikte aşağıda sıralanmıştır:

- Toplama (+)
- Çıkarma (-)
- Çarpma (*)
- Bölme (/)
- VE (&)
- VEYA (|)
- YADA (^)

```
DEĞER    EŞT 3 + 4
YÜK A,<BELLEK-3>
ELEK     EŞT ELEK-1 & ELEK-2
KATSAYI  EŞT İLK_DEĞER + YENİ_DEĞER/3
YÜK A,3*5
```

Aritmetik işlemlerde, işlem sırası soldan sağa doğrudur.

19.1.2 MikBil Sınama Programı

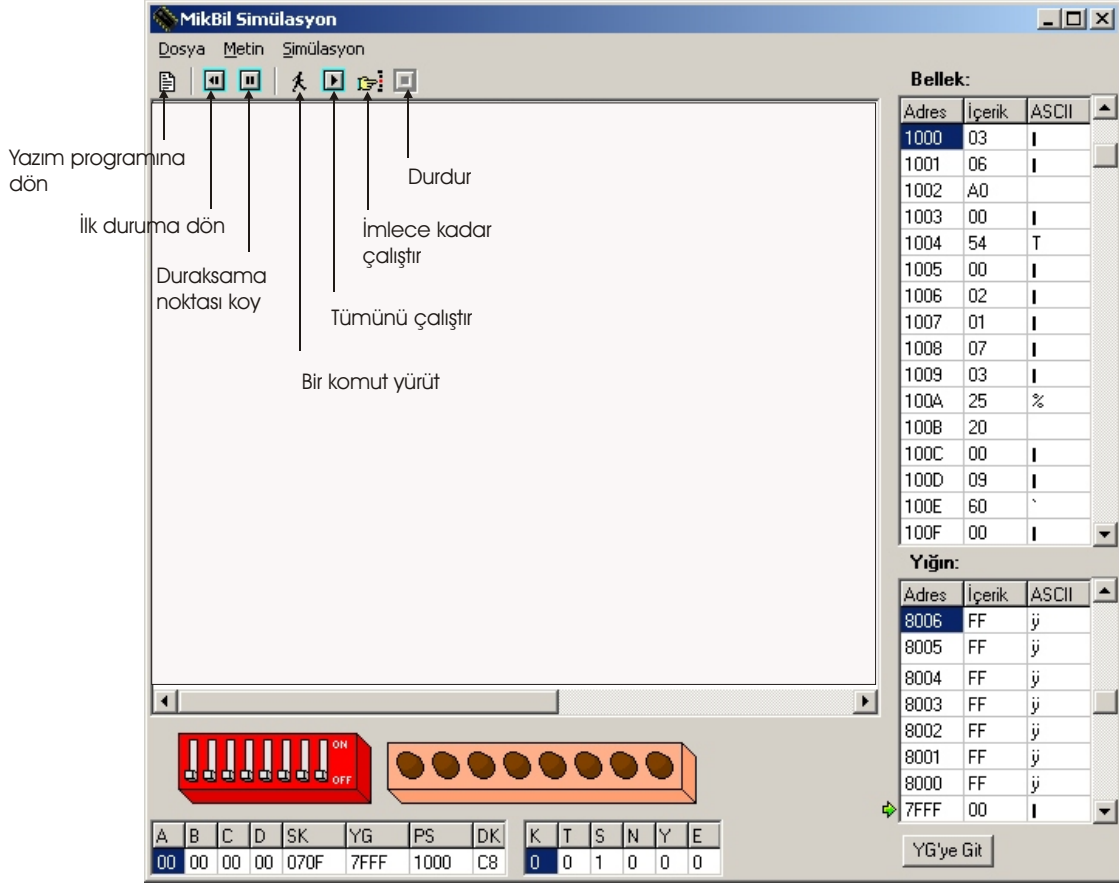
Kaynak program yazıldıktan sonra, çevirici program tarafından, makine koduna çevrilir. Makine koduna çevrilmiş olan program denenmeden çalıştırılabileceği gibi adım adım çalıştırılarak denenebilir.

Yazım programından, sınama programına geçmek için, Simülasyon sekmesinde, "Çalıştır" seçeneğinin seçilmesi gerekir. Çalıştır seçeneği seçildiğinde, çevirici kaynak programı çözülmeye başlar. Bu süreçte, yazım hatalarını kullanıcıya gösterir. Program hatasız ise, sınama aşamasına geçer. Sınama penceresi Şekil-19.4'te gösterilmiştir. Bu şekil üzerinde, düğmelerin görevleri belirtilmiştir.

Sınama penceresinde;

- Makine kodları ile birlikte kaynak program
- Merkezi işlem birimi içindeki tüm kütüklerin içerikleri,
- Durum Kütüğünün bayrakları
- Seçili bellek alanının içeriği
- Yığın içeriği

19.1 MikBil'in Özellikleri - 7



Şekil-19.4: Sınama penceresi ve bu pencerede görülen düğmelerin görevleri

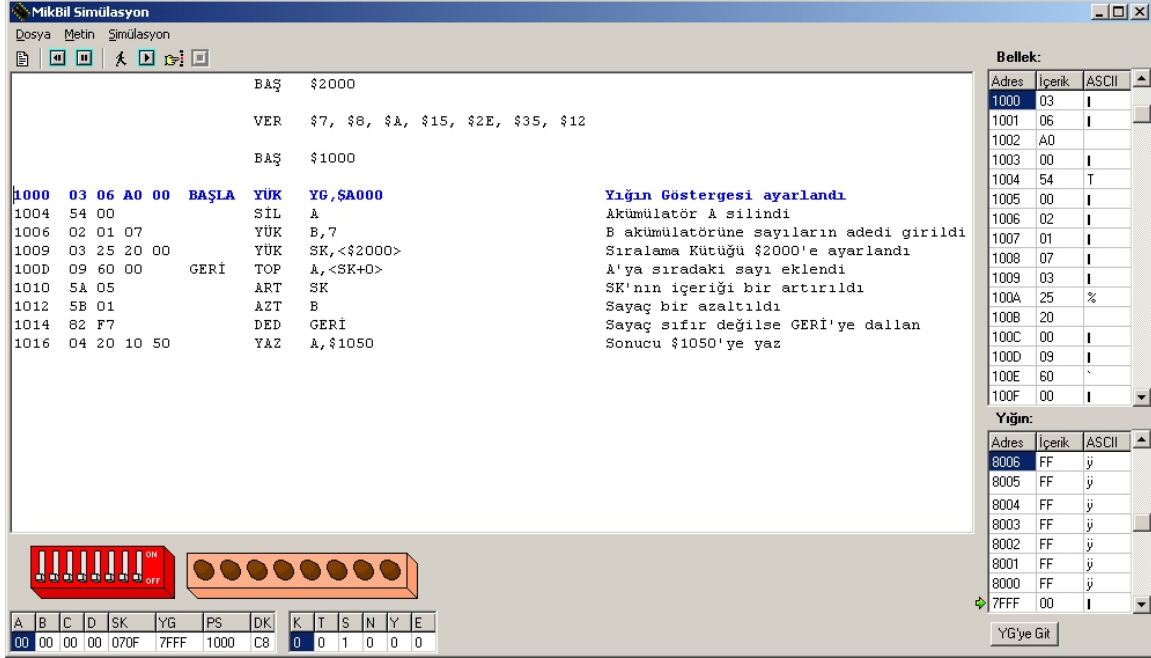
görülür. Makine koduna çevrilmiş olan program

- u Adım adım
- u Durağa kadar ya da
- u Program sonuna kadar

çalıştırılabilir.

Şekil-19.3'te kaynak kodu görülen programın çevrilmiş hali Şekil-19.5'te gösterilmiştir.

8 - MikBil



Şekil-19.5: Örnek bir sinama ekranı

Adım Adım Çalıştırma

Adım adım çalıştırmak için, ilk olarak çalıştırmaya başlanacak program satırının belirtilmesi gerekir. Sinama ekranına gelindiğinde, ilk program satırı seçilmiş durumdadır ve bu satır mavi renktedir.

Adım adı çalıştırma tuşuna basılarak her bir adımdaki gelişmeler izlenebilir. İşlenen adım mavi renkte görülür. Adım adım izleme sürecinde, MİB içindeki kütüklerin aldığı değerler, Durum Kütüğünün bayraklarının değişimi, bellekteki ve yığındaki değişimler izlenebilir.

Durağa Kadar Çalıştır

Program boyları büyüyünce, adım adım çalıştırma güçlük oluşturabilir. Bu nedenle, programa duraksama noktası ekleme yeteneği katılmıştır. Duraksama noktası koymak için önce, imleç duraksamanın konacağı satıra getirilir ve bu satır işaretlenir.

Ardından "Duraksama Noktası Ekle" düğmesine basılarak işlem tamamlanır. Duraksama noktası konmuş olan satırın rengi kırmızıya döner.

Duraksama noktasına kadar çalıştır düğmesine basılarak, programın belli bir kısmı, sınamadan çalıştırılır. Ardından, istenirse, adım adım sınamaya devam edilir.

Tamamının Çalıştır

Programın tamamı çalıştırmak istendiğinde, çalıştır düğmesine basmak yeterlidir.

19.1.2.1 Giriş/Çıkış İşlemleri

Sınama yazılımı, PIA sınamasına da olanak vermektedir. Bu amaçla, sanal iki PIA programa eklenmiştir. Sanal PIA, 9. Bölümde tanıtılan PIA'nın tüm özelliklerini kapsamaktadır. PIA'nın birimleri sembolik isimler ile belirtilenmiştir.

PIA - A

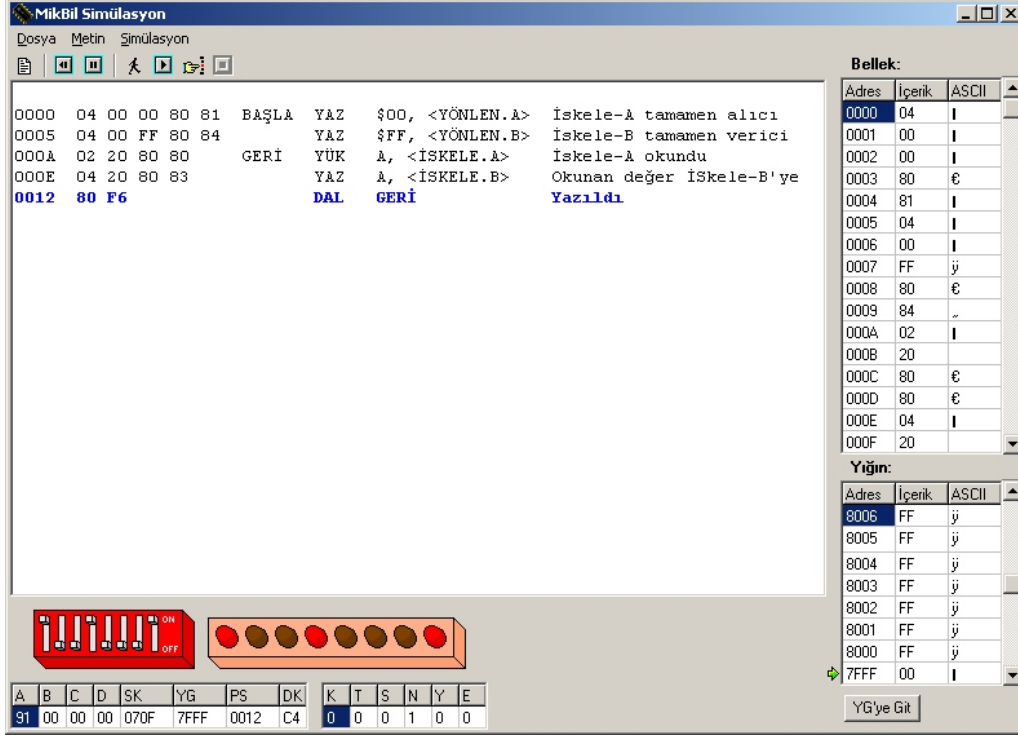
◇ İskele	İSKELE.A
◇ Yönlendirici	YÖNLEN.A
◇ Durum/Denetim	DURDEN.A

PIA - B

◇ İskele	İSKELE.B
◇ Yönlendirici	YÖNLEN.B
◇ Durum/Denetim	DURDEN.B

PIA-A'nın girişlerine, pencerenin sol alt köşesinde görülen anahtarlar bağlıdır. Kullanıcı bu anahtarları, fare yardımıyla istediği mantıksal konuma getirebilir. PIA-B'nin çıkışlarına 8 tane LED bağlıdır. PIA-B'nin çıkışlarının alacağı değere göre bu LED'ler yanar ya da söner. Şekil-19.6'da basit bir giriş çıkış program örneği görülmektedir. Bu örnek program, PIA-A'nın girişlerindeki anahtarların değerini okumakta ve okuduğu değerleri PIA-B iskelesine aktarmaktadır.

10 - MikBil



Şekil-19-6: Giriş çıkış deneyine ilişkin örnek

19.1.2.2 Kesme İşlemleri

Sınama yazılımı ile kesme deneyleri de yapılabilmektedir. Bu deneyi yapabilmek için bazı bağlantıları yapmaya ve buna göre koşturmaya gerek vardır.

- u Kesme ile ilgili deneyi yapabilmek amacıyla, P1A-A'nın AL-A çıkışı, P1A-B'nin HAZIR-B girişine, P1A-B'nin AL-B çıkışı P1A-A'nın HAZIR-A girişine bağlanmış kabul edilecektir.
- u Her iki P1A da kesme üretebilmektedir.

Aşağıda verilen örnek programda P1A-A'ya bağlı düğmelerden birisi durum değiştirdikçe, AL-A 0'dan 1'e geçmektedir. HAZIR-B girişi 0'dan 1'e geçtiğinde kesme üretmek üzere koşturulmuştur. Kesme alındığında, kesme hizmet programına daldanılmaktadır.

*** Kesme Deneyi ***
BAŞ \$2000

YAK EŞT \$00
SÖNDÜR EŞT \$01

* BAYRAK LED'lerin yanmakta mı sönmekte mi olduğunu belli ediyor.

BAYRAK YER 1

BAŞLA KEN	Koşullamalar yapılanaya kadar izin yok.
YÜK SK, KESMEHİZ	Kesme hizmet programı adresi
SK'da.	
YAZ SK, \$FFF8	Adres kesme isteği vektörüne yazıldı.
YAZ \$00, <YÖNLEN.A>	İskele-A tamamen alıcı.
YAZ \$FF, <YÖNLEN.B>	İskele-B tamamen verici.
YAZ \$00, <DURDEN.A>	AL-A = 0 ve elle kontrol.
YAZ \$01, <DURDEN.B>	HAZIR-B 1->0 etkin.
YAZ \$00, <İSKELE.B>	Tüm LED'leri söndür.
YAZ YAK, <BAYRAK>	LED'ler yanarak başlayacak.
KİZ	Kesme isteğine izin ver.
YÜK B, <İSKELE.B>	
GERİ YÜK A, <İSKELE.A>	Yeni durumu A'ya oku.
Eskisi B'de.	
KAR A, B	Düğmelerin durumu değişmiş mi?
DEE DEVAM	Değişmemişse kesme üretmeden devam et.
KUR 4, <DURDEN.A>	AL-A 1'e çıktı. (Kesme üretir.)
SİL 4, <DURDEN.A>	AL-A 0'a indi.
DEVAM AKT B, A	Durumu B'de sakla.
DAL GERİ	
KESMEHİZ YÜK A, <BAYRAK>	
KAR A, YAK	
DEE KUR_E	

12 - MikBil

SİL E
DAL İLERİ
KUR_E KUR E
İLERİ SOLD <İSKELE.B>
YÜK A, <İSKELE.B>
SIN A, \$80
DED SÖNECEK
YANACAK YAZ YAK, <BAYRAK>
DÖNK
SÖNECEK YAZ SÖNDÜR, <BAYRAK>
DÖNK