

13. MİKROİŞLEMCİLİ SİSTEM DONANIMI VE YAZILIMI GELİŞTİRME ARAÇLARI

Mikroişlemcili sistemler geliştirilirken, tasarlanan mikroişlemci temelli sistemin donanımını ve yazılımını gerçekleştirmek, fiziksel dünyaya aktarmak amacıyla tasarlanmış donanım ve yazılımlar kullanılır.

Bu sistemler kullanıcının mikroişlemcili sistemin uygulama amacına yönelik geliştirilmesine olanak sağlayan yine kendisi de mikroişlemci temelli sistemlerdir.

Mikroişlemcili Sistem Geliştirme Sistemleri üç ana grupta toplanabilirler.

Donanım

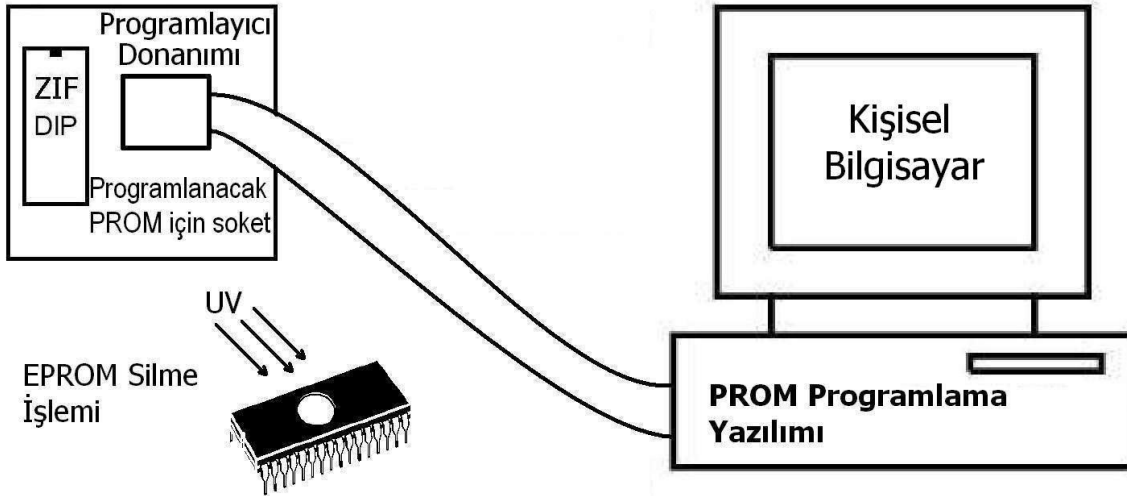
Yazılım

Donanım +Yazılım

13.1. Mikroİşlemcili Sistem Donanımı Geliştirme Araçları

Mikroİşlemcili sistem donanımı geliştirilirken genellikle kullanıcı tarafından tasarlanan mikroİşlemci temelli sistemin donanımı gerçekleştirilerek geliştirme amaçlı kullanılır.

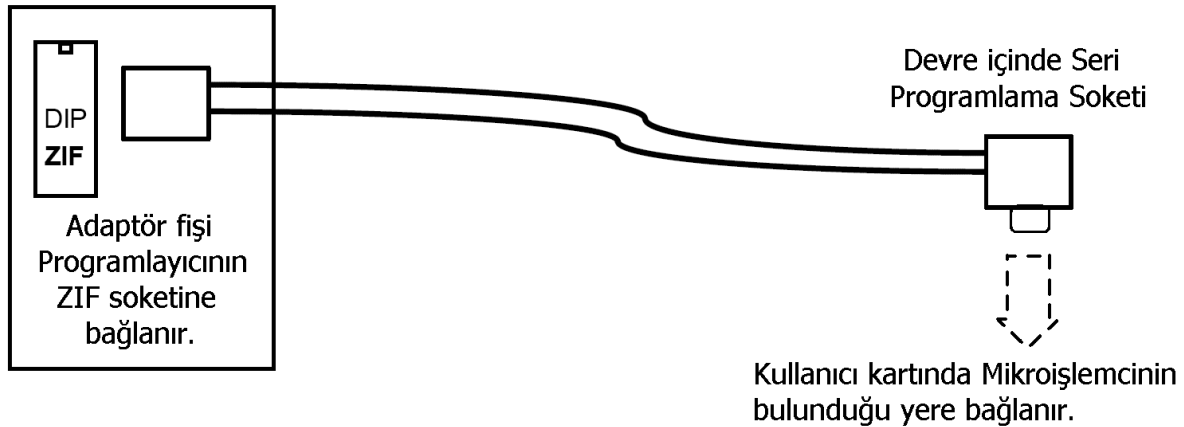
Tasarlanan programın işlem, işlenen ve diğer kodları, bir kişisel bilgisayarda oluşturulur ve sistemin program belleği (Flaş ROM, EPROM vb.) tümleşik devresine yazılarak programlanır.



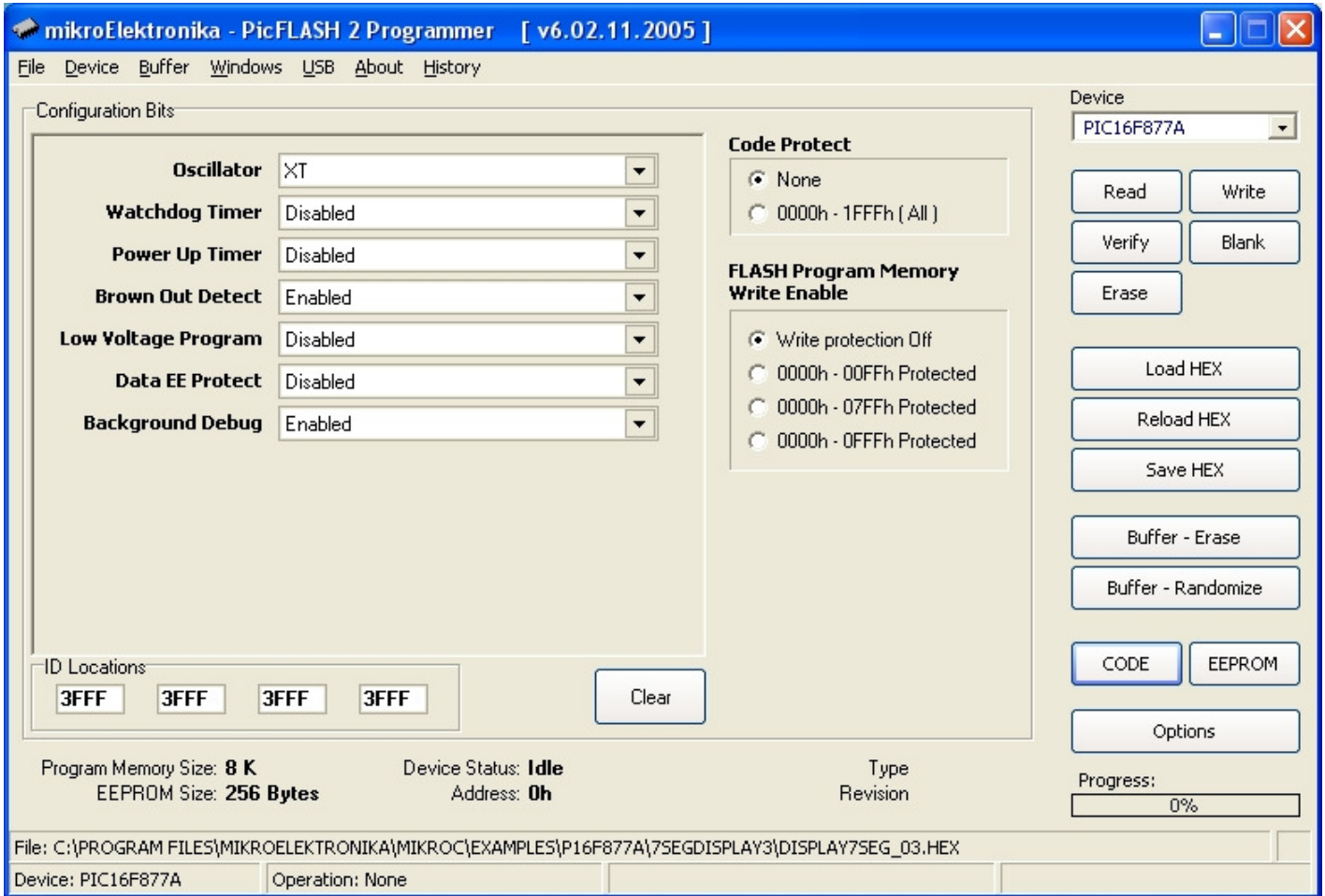
Şekil 13-1 EPROM tümleşik devrelerinin programlanması

Günümüzde yaygın olarak kullanılan mikroişlemcili sistemler, flaş tipinde program belleğini tümleşik olarak içinde bulunduran mikrodenetleyicilerdir. Bu tip bellekler mikrodenetleyici tümleşik devresinin uygulama devresinin içinde programlanmasına olanak verecek şekilde üretilmişlerdir.

Veri akışı seri olduğu için bu tip programlama devrelerine “devre içi seri programlayıcı” (In Circuit Serial Programming, ICSP) adı verilir.



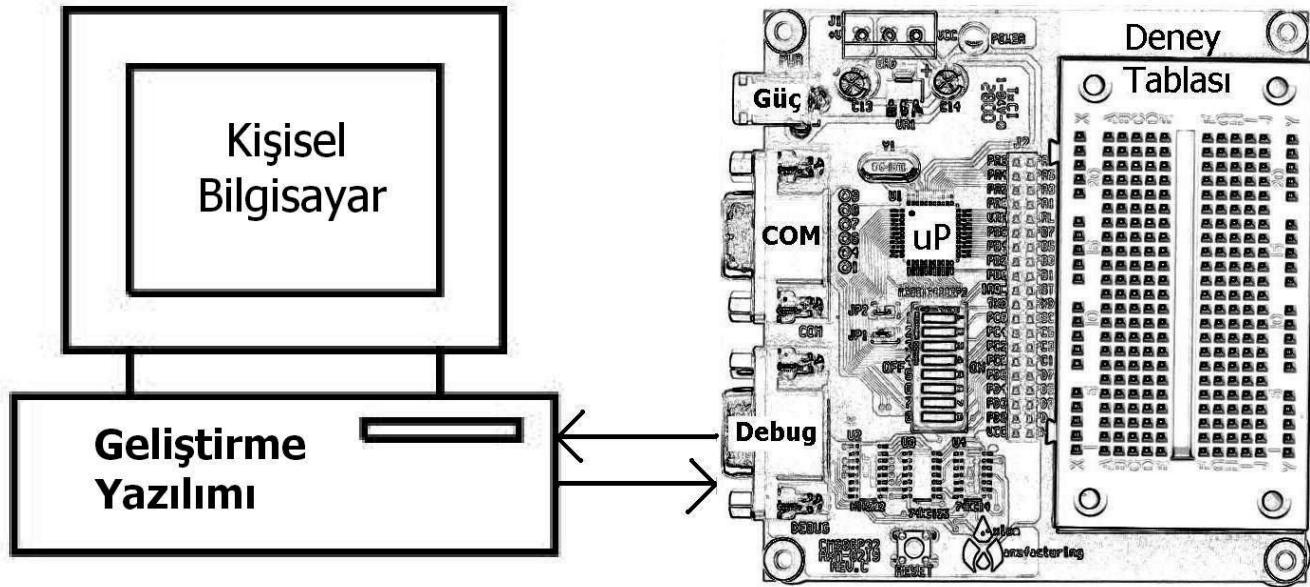
Şekil 13-2 Devre içi seri programlama donanımı



Şekil 13-3 Devre içi seri programlama yazılımı

Eğitim amaçlı veya ekonomik mikroişlemcili veya mikrodenetleyicili ürün geliştirmelerinde ise geliştirmesi yapılacak mikroişlemciye özel genel amaçlı çalışabilen mikroişlemci temelli donanımlar kullanılır.

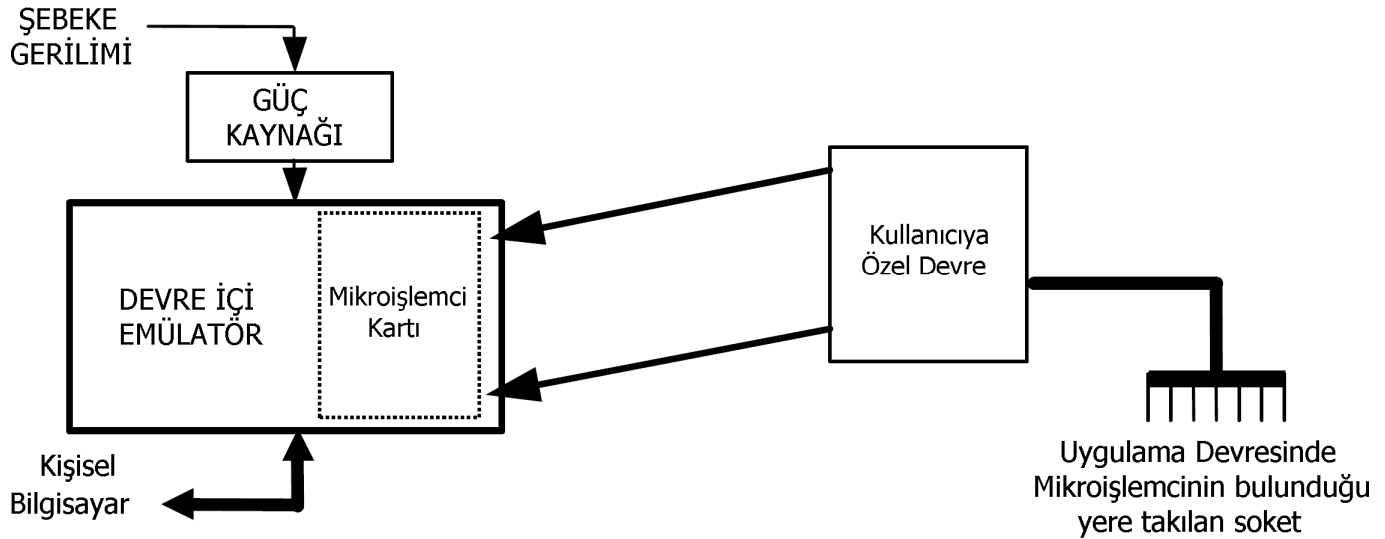
Genel amaçlı mikroişlemci sistem donanımı geliştirme araçlarına “Mikroişlemci Geliştirme Sistemi”, (Microprocessor Development System) adı verilir.



Şekil 13-4 Mikroişlemci Geliştirme Sisteminin Blok Diyagramı

Araştırma, geliştirme gerektiren mikroişlemcili veya mikrodenetleyicili ürün geliştirmelerinde ise geliştirmesi yapılacak mikroişlemciye özel çalışabilen mikroişlemci temelli donanımlar kullanılır.

Bu tür özel amaçlı mikroişlemci sistem donanımı geliştirme araçlarına “Devre İçi Donanım Benzetici, Emülatör”, (In Circuit Emulator, ICE) denir.



Şekil 13-5 Devre İçi Donanım Benzetici sisteminin blok diyagramı.

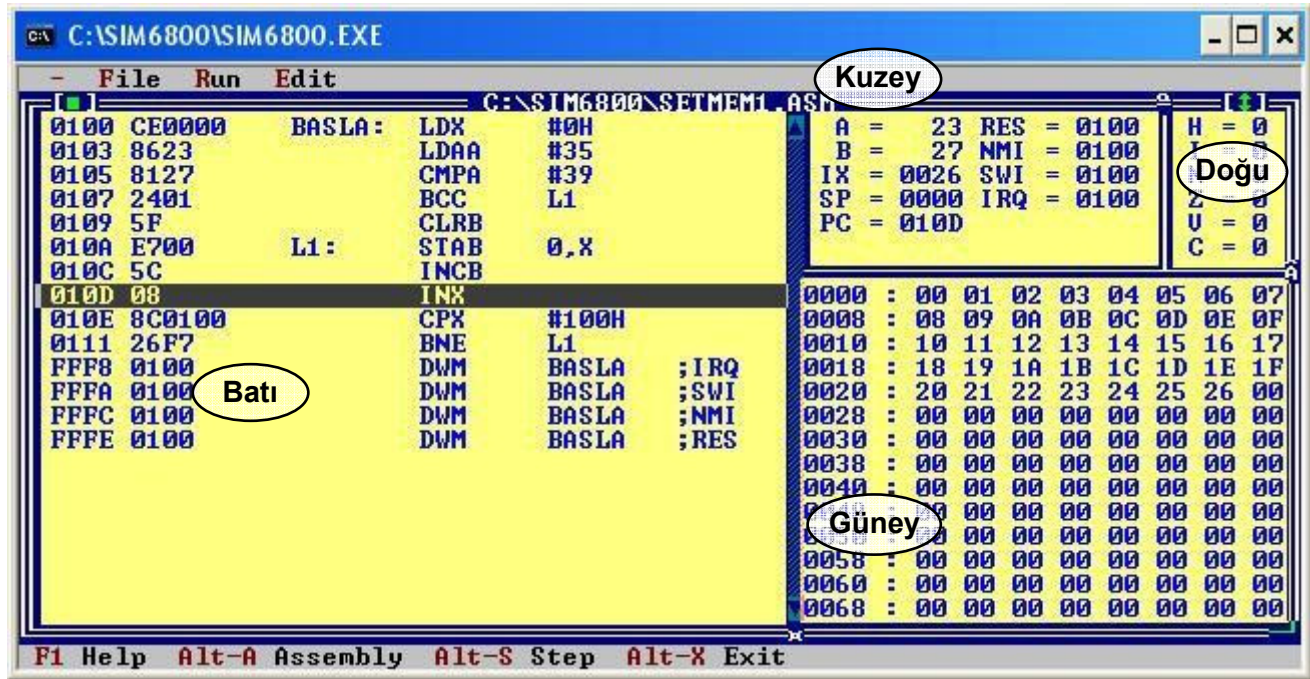
Bu sistemler geliştirilecek mikroişlemcinin uç fonksiyonlarını benzeterek, uygulama devresi içine takılan soketle aktaran, bağımsız veya bir kişisel bilgisayara bağlı olarak çalıştırabilen sistemlerdir.

13.2. Mikroişlemcili Sistem Yazılımı Geliştirme Araçları

Mikroişlemcili sistem yazılımı geliştirilirken kullanıcı tarafından tasarlanan, Mikroişlemci dilinde yazılan çevirici dili programı makine diline çeviren çeviriciler, C ve BASIC gibi yüksek seviyeli dillerden makine diline çeviren derleyiciler (compiler) gibi yazılımlar kullanılır.

Mikroişlemci temelli sistem programının kişisel bilgisayar ortamında çalışmasını sağlayan programlara “Benzetim, Simülatör” (Simulator) programı denir.

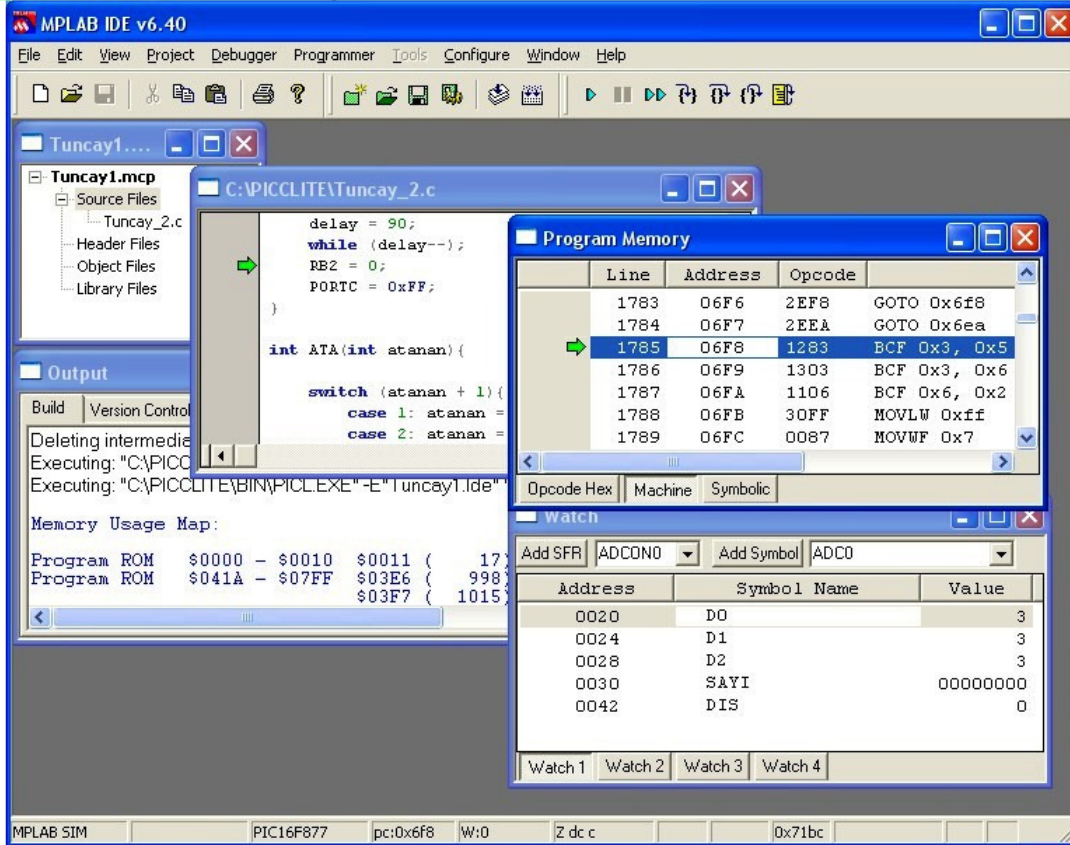
Bu yazılımlar tasarlanan programın bir kişisel bilgisayar ortamında çalışmasını, incelenmesini, hata ayıklanmasını (debug) sağlar.



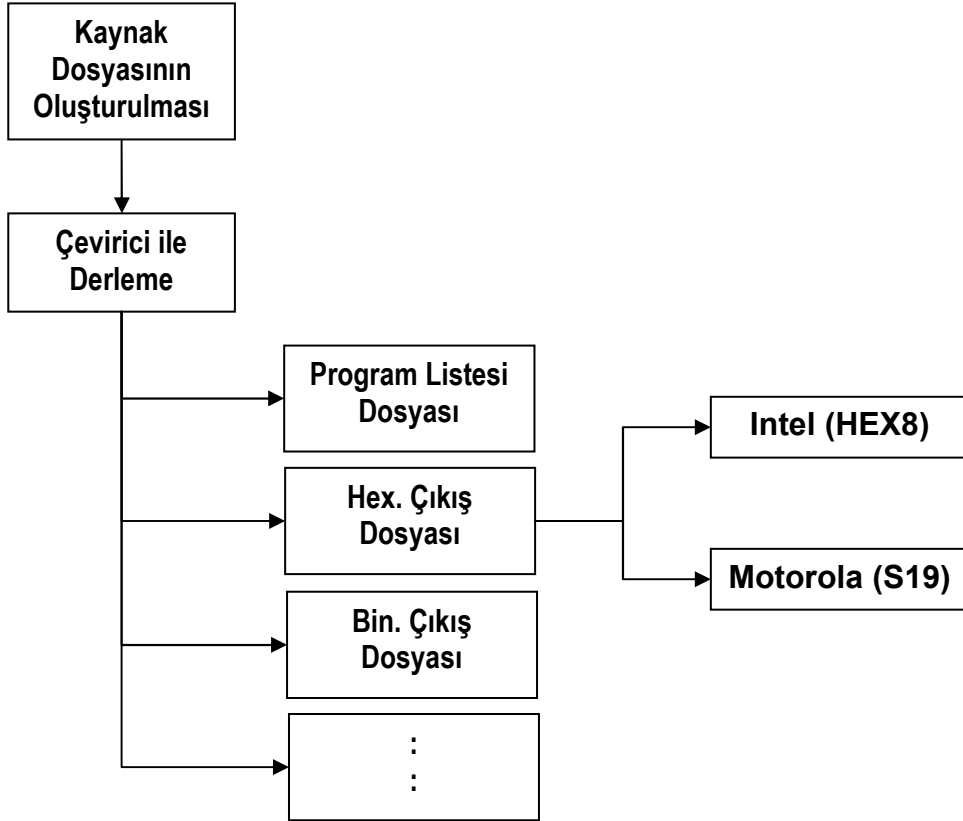
Şekil 13-6 6800 Mikroişlemcisi için kullanılan bir simülatör yazılımı

Mikroişlemci komutlarının ve yazılımların çalışmasının daha iyi anlaşılması için bir 6802 mikroişlemci simülatör programı (SIM6802) www.tuncayuzun.com veya www.yildiz.edu.tr/~uzun internet adreslerinden elde edilerek IBM uyumlu kişisel bilgisayarda çalıştırılabilir.

Günümüzde, yaygın olarak kullanılan geliştirme araçları, tümleşik geliştirme ortamlarıdır (Integrated Development Environment, IDE).



Şekil 13-7 Bir mikroişlemci tümleşik geliştirme yazılımının ekran görüntüsü



Şekil 13-9 Çeviricilerin çalışma diyagramı

13.2.7. Onaltılık Çıkış Dosyaları

Onaltılık çıkış dosyası program listesi dosyasında bulunan verileri ve makine dilindeki değerleri saklar.

Programlayıcı cihazı, tümleşik geliştirme ortamı, simülatör programı vs. tarafından program belleğinin programlanması için kullanılır ve genellikle HEX (*.HEX) kelimesini tip kodu olarak kabul eder.

Bu dosya mikroişlemci temelli sistemler arasında veri aktarımını hata kontrolü yaparak sağladığı için yaygın olarak kullanılır.

Kişisel bilgisayarda oluşturulan mikroişlemci yazılımı, programlayıcı modülüne, donanım veya yazılım geliştirme sistemlerine bu biçimdeki dosya şeklinde aktarılır.

Motorola 8-bit biçiminde onaltılık (Hex) çıkış dosyası

Dosya, başlık, veri ve dosya sonu olmak üzere üç ana kısımdan oluşur.

Başlık kısmı birden fazla dosya olduğu zaman kullanılır, ilk satırdır ve “S0” ile başlayan sabit biçimli satırdır.

Veri kısmında satırlar vardır ve satırlar “S1” ile başlar.

Dosya sonu satırı ise son satırdır ve “S9” ile başlayan sabit biçimli satırdır.

Her satırın sonunda hata kontrolü için bir kontrol toplamı bulunur.

Motorola tipindeki 8-bit onaltılık çıkış dosyasındaki her satırın sonunda yer alan kontrol toplamı (**checksum**) aşağıda verilen şekilde hesaplanır.

$$\left(\begin{array}{c} \text{Veri Bayt} \\ \text{değerleri} \end{array} + \begin{array}{c} \text{Adres bayt} \\ \text{değerleri} \end{array} + \begin{array}{c} \text{Uzunluk} \\ \text{değeri} \end{array} \right) = \begin{array}{c} \text{Sonucun} \\ \text{1'e tümleyeni} \end{array}$$

Intel 8-bit biçiminde onaltılık (Hex) çıkış dosyası

Intel biçimli onaltılık dosya, veri ve dosya sonu olmak üzere iki ana kısımdan oluşur. Bu kısımlar, küçük farklar dışında Motorola tipindeki onaltılık çıkış dosyasına benzerdir.

Intel tipindeki 8-bit onaltılık çıkış dosyasındaki her satırın sonunda yer alan kontrol toplamı (checksum) aşağıda verilen şekilde hesaplanır.

$$\left(\begin{array}{c} \text{Veri Bayt} \\ \text{değerleri} \end{array} + \begin{array}{c} \text{Adres bayt} \\ \text{değerleri} \end{array} + \begin{array}{c} \text{Tip ve Uzunluk} \\ \text{değerleri} \end{array} \right) = \begin{array}{c} \text{Sonucun} \\ \text{2'ye tümleyeni} \end{array}$$

Örnek 13-7 Intel 8-bit biçiminde bir onaltılık çıkış dosyasının analizi:

:0ED00000CE00004FA7004C088C010026F7015F

:02FFFE00D00031

:00000001FF

:0E D000 00 CE 00 00 4F A7 00 4C 08 8C 01 00 26 F7 01 5F

	Adres	Tip	Değerler	Kontrol Toplamı
: 0E	D0 00	00	CE00004FA7004C088C010026F701	5F= 00 – 4A1
Uzunluk →			1 2 3 4 5 6 7 8 9 10 11 12 13 14	

:02 FFFE 00 D0 00 31

	Adres	Tip	Veriler	Kontrol Toplamı
: 02	FF FE	00	D0 00	31=– (02+FF+FE+00+D0+00)= 00 – 2CF
Uzunluk →			1 2	

:00 0000 01 FF

	Adres	Tip	Kontrol Toplamı
: 00	00 00	01	FF=– (00+01+00+00)= 00 –1
Uzunluk →			

11. MİKROİŞLEMCİ YAZILIMI

Mikroişlemcinin Programlama Modeli +
Komutların Çalışmasını Belirleyen Adresleme Şekilleri +
Mikroişlemcinin Komut Kümesidir.

11.1. Programlama Modeli (AVR CPU Core)

Şekil 11-1 Mikroişlemcisi Programlama Modeli

11.2. CPU Yazmaçları

Mikroişlemci yazmaçları genel, özel ve durum amaçlı olmak üzere üç ana grupta toplanabilir.

Genel amaçlı yazmaçlar, aritmetik ve mantıksal işlemlerde veya mikroişlemci yazmaçlarıyla bellek arasında veri transfer için kullanılabilir.

Mikroişlemciler, çalışması sırasında bazı işlemlerde özel amaçlı yazmaçları ve yığın gibi özel olarak tanımlanmış bellek alanı kullanır. Özel amaçlı yazmaçlar, mikroişlemcinin çalışmak için gerek duyduğu işlemlerde yalnız özel komutlar tarafından kullanılabilir.

AVR mikrodeneleyicisi içinde 3 tane 16 Bit ve 32 tane 8 Bit yazmaç vardır. Bu yazmaçlardan R0-R31 genel amaçlı, SP, PC ve SREG ise özel amaçlı, SREG ise esas olarak durum amaçlıdır.

11.2.1. Yığın İşaretçisi (SP)

16-Bit Yığın İşaretçisi (Stack Pointer) SP

11.2.2. Program Sayıcısı (PC)

16-Bit Program Sayıcısı (Program Counter) PC

11.2.3. Durum Yazmacı (SR)

8-Bit Durum Yazmacı (Status REGister) SREG

Aritmetik / Lojik işlem sonuçlarını ve mikroişlemci (CPU) durumu ile ilgili bilgi saklar. Bu nedenle durum yazmacı bitleri “bayrak” olarak adlandırılır.

Küresel Kesme İzin (Global Interrupt enable) I

Kopya bit (I copy) T

Yarım Elde (Half carry) H

İşaret (Sign) S

Eksi (Negative) N

Sıfır (Zero) Z

Taşma (oVerflow) V

Elde/Ödünç (Carry) C

Bit 0 Elde/Ödünc (C)

$$\begin{array}{r} 0101\ 0101 \\ + 0101\ 0010 \\ \hline 1010\ 0111 \end{array}$$

C=0

$$\begin{array}{r} 1111\ 1111 \\ + 0000\ 0001 \\ \hline 0000\ 0000 \end{array}$$

C=1

$$\begin{array}{r} 0000\ 0011 \\ + 0000\ 0001 \\ \hline 0000\ 0100 \end{array}$$

C=0

$$\begin{array}{r} 1000\ 0000 \\ - 0000\ 0001 \\ \hline 0111\ 1111 \end{array}$$

C=0

$$\begin{array}{r} 0000\ 1111 \\ - 1111\ 0000 \\ \hline 0001\ 1111 \end{array}$$

C=1

$$\begin{array}{r} 1111\ 1111 \\ - 0000\ 0001 \\ \hline 1111\ 1110 \end{array}$$

C=0

Bit 1 Sıfır (Z)

$$\begin{array}{r} 0101\ 0101 \\ - 0101\ 0101 \\ \hline 0000\ 0000 \end{array}$$

Z=1

$$\begin{array}{r} 1111\ 1111 \\ + 0000\ 0001 \\ \hline 0000\ 0000 \end{array}$$

Z=1

$$\begin{array}{r} 0000\ 0011 \\ + 0000\ 0001 \\ \hline 0000\ 0100 \end{array}$$

Z=0

Bit 2 Eksi (N)

$$\begin{array}{r} 0010\ 0001 \\ + 0101\ 0101 \\ \hline 0111\ 0110 \end{array}$$

N=0

$$\begin{array}{r} 0110\ 1101 \\ + 0010\ 0101 \\ \hline 1001\ 0010 \end{array}$$

N=1

Bit 3 Taşma (V)

$$\begin{array}{r} 0101\ 0101 \\ + 0101\ 0010 \\ \hline 1010\ 0111 \end{array}$$

V=1

$$\begin{array}{r} 1111\ 1111 \\ + 0000\ 0001 \\ \hline 0000\ 0000 \end{array}$$

V=0

$$\begin{array}{r} 0000\ 0011 \\ + 0000\ 0001 \\ \hline 0000\ 0100 \end{array}$$

V=0

$$\begin{array}{r} 1000\ 0000 \\ - 0000\ 0001 \\ \hline 0111\ 1111 \end{array}$$

V=1

$$\begin{array}{r} 0000\ 1111 \\ - 1111\ 0000 \\ \hline 0001\ 1111 \end{array}$$

V=0

$$\begin{array}{r} 1111\ 1111 \\ - 0000\ 0001 \\ \hline 1111\ 1110 \end{array}$$

V=0

Bit 4 İşaret (S)

S-biti her zaman N eksi bayrağı ile V ikiye tümleyen taşıma bayrağı arasında ÖZEL-VEYA işlemi yapılarak ($S = N \oplus V$) elde edilir.

Bit 5 Yarım Elde (H)

0000 0001	0000 1101
+ 0000 0101	+ 0000 0101
<u>0000 0110</u>	<u>0001 0010</u>
<u>H=0</u>	<u>H=1</u>

Bit 6 Kopya Bit (T)

Bu bit, dosya yazmacındaki bir yazmacın bir bitinin BST komutuyla T'ye ve T'deki bitin, dosya yazmacındaki bir yazmacın bir bitine BLD komutuyla kopyalanması için kullanılır.

Bit 7 Küresel Kesme İzin (I)

I-biti "1" yapılırsa, mikroişlemciye kesme isteği (INT0, INT1) uçlarına dışarıdan ve iç birimlerden gelen bütün örtülebilir kesme işaretlerine izin verilir. Daha sonra iç birimler için ayrı kesme izin denetimi ilgili çevre biriminin ayrı denetim yazmaçlarında gerçekleştirilir. Küresel Kesme İzin biti temizlenirse, I="0" yapılırsa, kesmelerin hiçbirine izin verilmez! I-biti, bir kesme meydana geldikten sonra donanım tarafından temizlenir ve sonraki kesmeleri izin vermek için RETI komutu tarafından ayarlanır. I-biti, komut tablosunda açıklandığı gibi, SEI ve CLI komutlarıyla uygulama tarafından da ayarlanabilir.