



# *ADuC 84 I $\mu$ -denetleyicisi*

***Ayhan Yüksel***

***(Son güncelleme: 06.03.2013 – Berat Doğan)***

***Tıbbi Enstrumantasyon Tasarım & Uygulamaları***

***(06.03.2013)***

# Sunum Planı

## Mikrodenetleyici

### ADuC841 mikrodenetleyicisi

#### ADuC 841 kullanımı

- Uygulama devresi
- Program derlemesi

#### Bellek

- Dâhilî bellek
- Program belleği
- Veri belleği
- XRAM

#### 8052 programlama dili

- Saklayıcılar
- Komutlar

#### Örnek

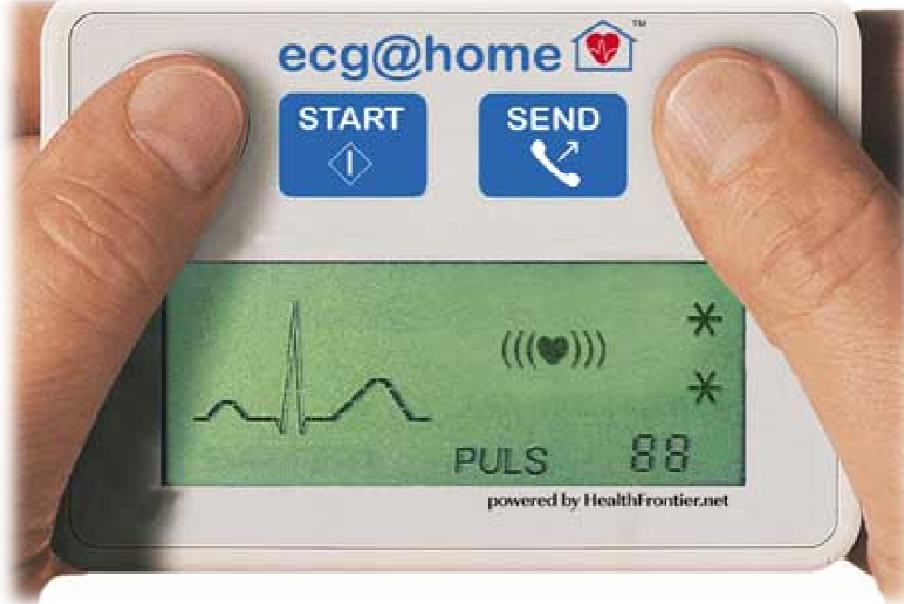


## $\mu$ -denetleyici

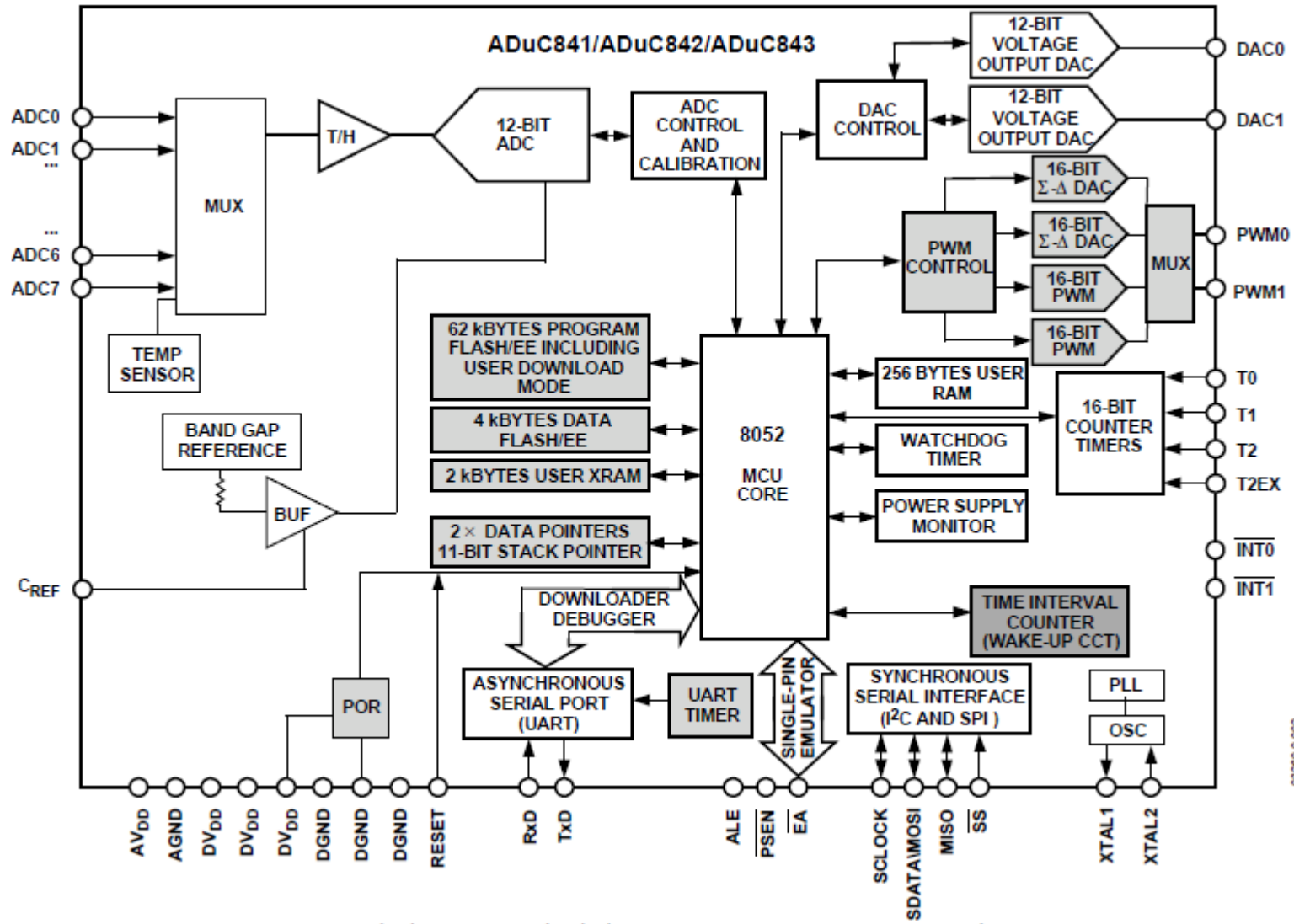
Mikro ( $\mu$ ) denetleyici üzerinde bir mikroişlemciye ek olarak çeşitli çevre birimlerini (ADC, DAC, UART... vs.) bulunduran bir entegre devredir.

$\mu$  -denetleyiciler çok çeşitli yerlerde kullanılabilir:

- LCD ekran, tuş takımı olan çoğu cihazda,
- Otomobillerde (motor, fren sistemi vs.)
- Uzaktan kontrollü cihazlarda,
- Dijital kamera, cep telefonu, yazıcı, mp3 çalar,
- Çamaşır makinesi, bulaşık makinesi,
- **Medikal cihazlar (hasta takip cihazları, ölçüm cihazları)**



# µ-denetleyici

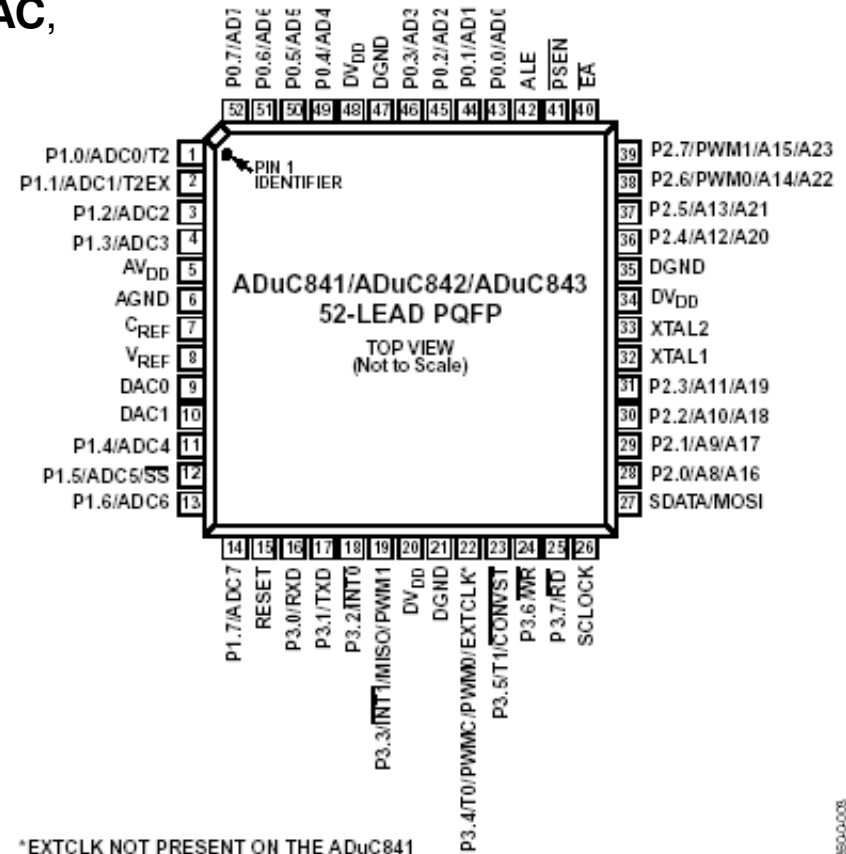


ADuC 841 iç yapısı

# ADuC 841 mikrodenetleyicisi

## ADuC 841,

- 52 bacaklı SMD kılıf içerisinde,
- 8052 uyumlu **8-bit mikroişlemci**,
- yüksek performanslı **12 bit ADC/DAC**,
- Flash/EE/program/veri belleği bulunduran bir mikrodenetleyicidir.



# ADuC 841 mikrodeneleyicisi

## ADuC841,

- 8 kanal 12 bit çözünürlüğe sahip **ADC**,
- 2 kanal 12 bit çözünürlüklü **DAC**,
- güç kaynağı göstergesi ve
- bandgap reference\* gibi analog özelliklere de sahiptir.

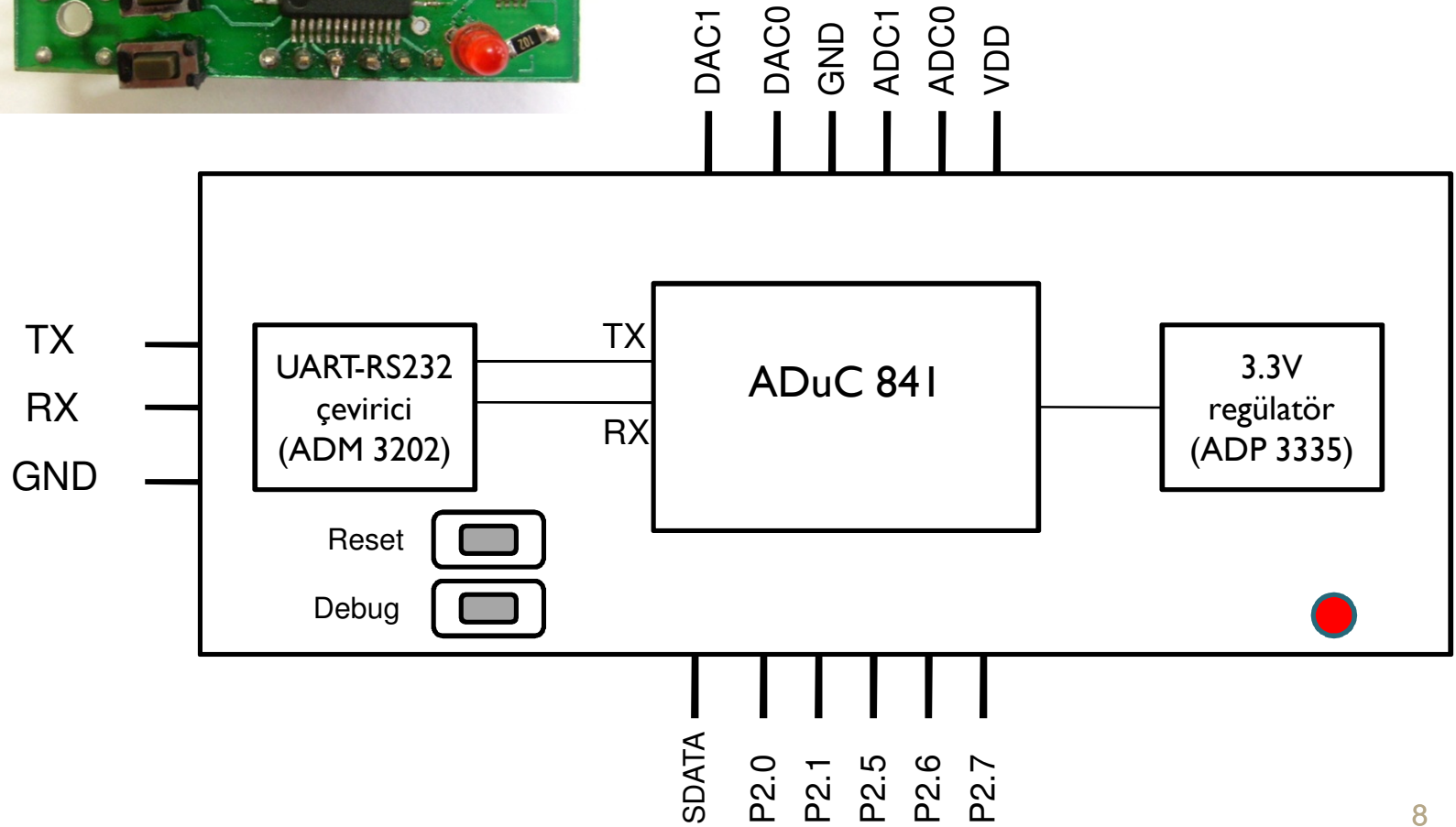
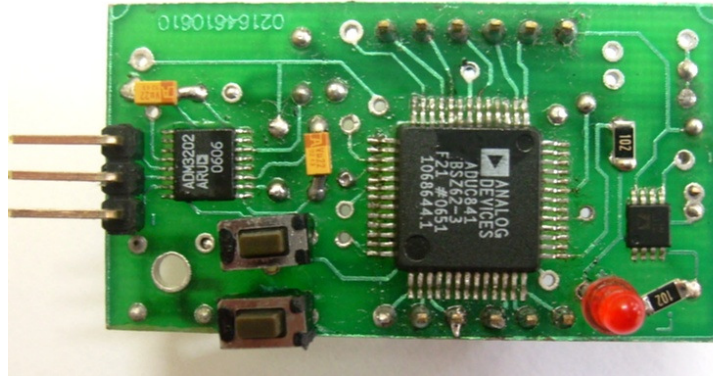
\* Bandgap referansları, A/D çeviriciler, DRAM'ler, flash hafıza devreleri gibi CMOS entegre devrelerinde, yüksek doğrulukları ve sıcaklıktan bağımsız olmaları nedeniyle sıklıkla kullanılmaktadır.

# ADuC 841 mikrodeneleyicisi

## ADuC841,

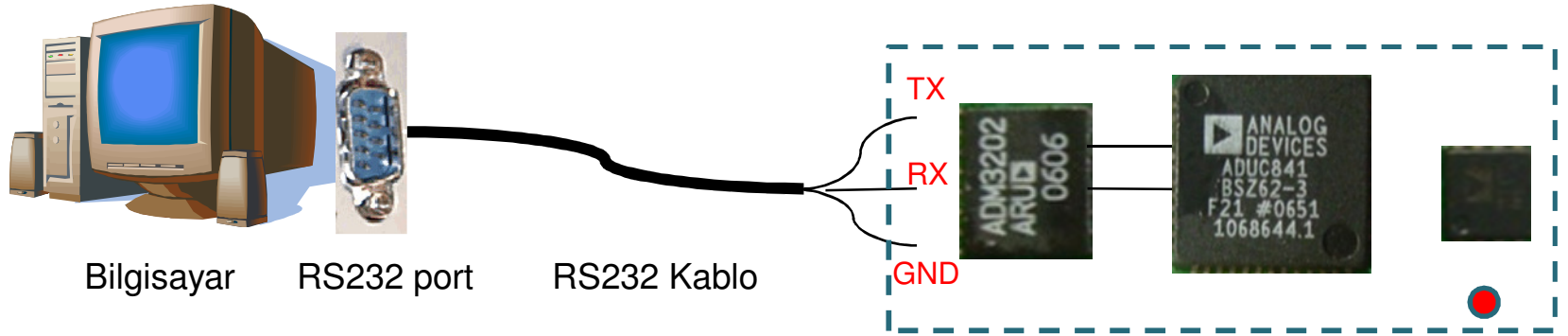
- TIC (time interval counter),
- WDT (watchdog timer),
- 3 adet timer/counter ve
- 2 seri I/O portu da çip üzerinde bulunmaktadır (SPI/UART) .
  
- Fabrika yazılımı; devre üzerinde seri yükleme, debug mod(UART ile) ve tek pin emulasyon modunu (EA pini ile) desteklemektedir.
- Besleme gerilimi (3V veya 5V)
- Düzgün çalışma sıcaklığı aralığı: (-40°C --- +85°C)

# ADuC 841 $\mu$ -denetleyicisi uygulama devresi





## Uygulama devresi-bilgisayar bağlantısı

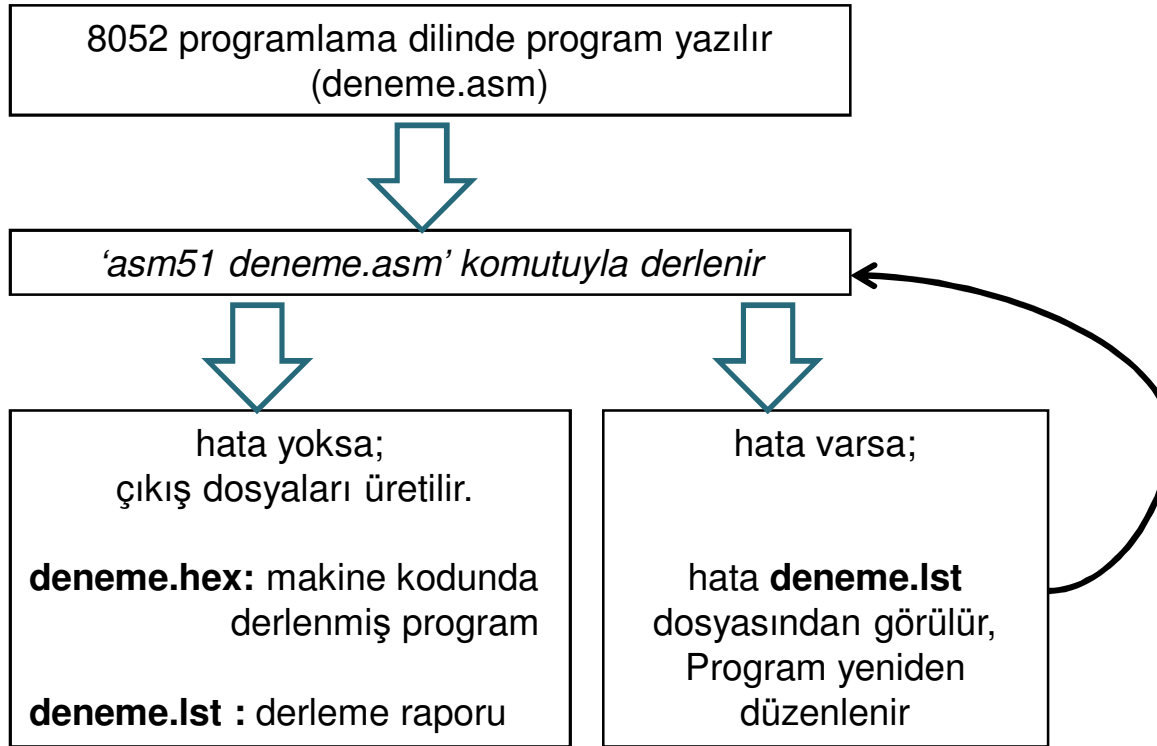


Uygulama devresi, RS232-UART çeviricisi ile bilgisayara bağlanır.

### Bilgisayar-denetleyici bağlantısı,

- Seri port üzerinden veri alış-verişi
- ADuC 841 denetleyicisine program yüklenmesi
- ADuC 841'in program koşturmasına müdahale (debugging) amaçlarıyla yapılır.

# Program Derleme (.asm)



```
c:\aduc\ASM51>asm51 deneme.asm
8051 Cross-Assembler, Version 1.2h
(c) Copyright 1984, 1985, 1986, 1987, 1988, 1989, 1990
by MetaLink Corporation
First pass
Second pass
ASSEMBLY COMPLETE, 0 ERRORS FOUND

c:\aduc\ASM51>
```

deneme.asm	29.09.2003 19:32	ASM Dosyası	2 KB
DENEME.LST	17.02.2009 18:26	LST Dosyası	4 KB
DENEME.HEX	17.02.2009 18:26	HEX Dosyası	1 KB

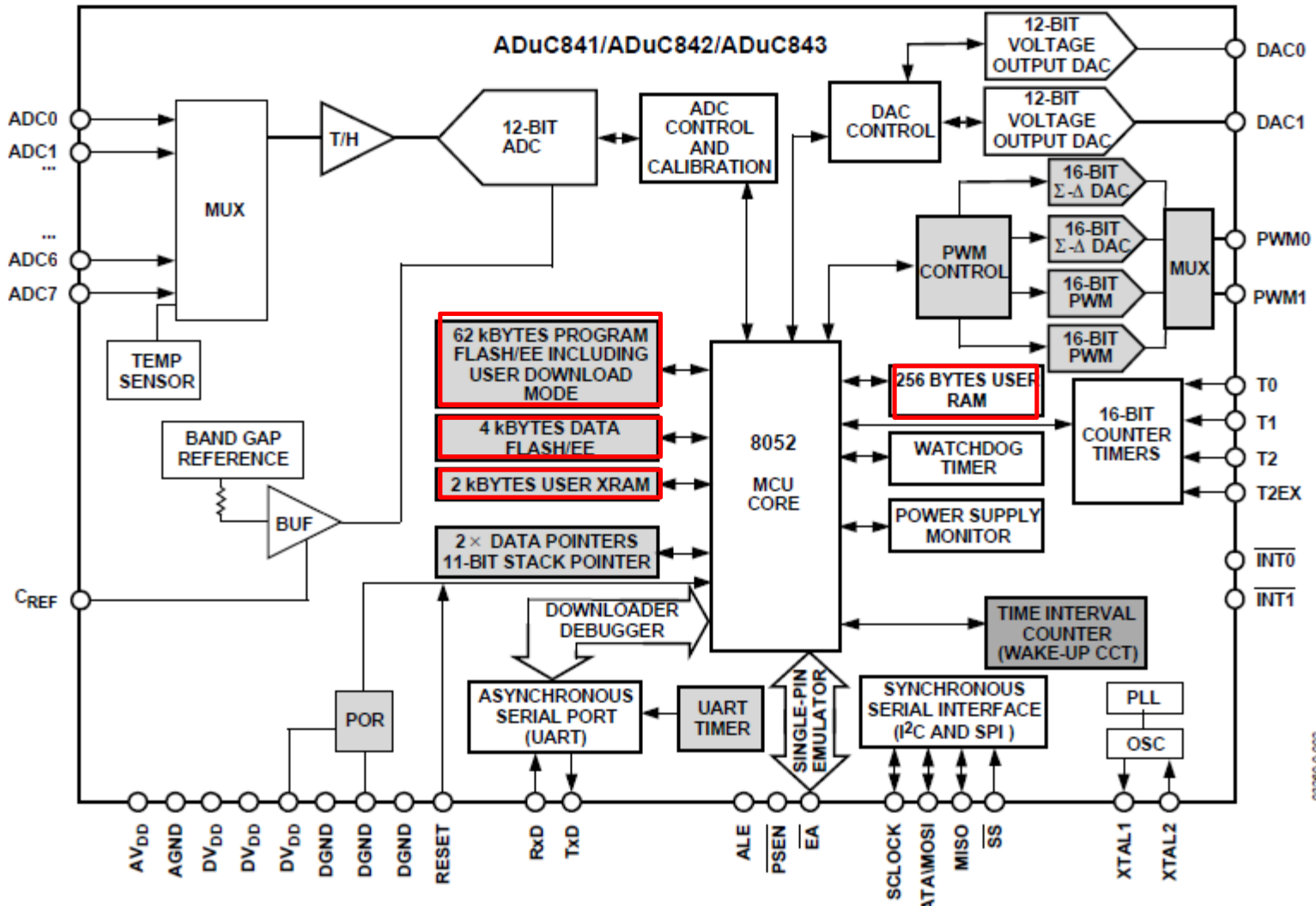
# BELLEK

**Genel Amaçlı Dâhilî RAM:** 256 Byte

**Program Belleği:** 62 kByte elektriksel olarak silinebilen bellek

**Veri Belleği:** 4 kByte elektriksel olarak silinebilen bellek

**XRAM:** 2 kByte



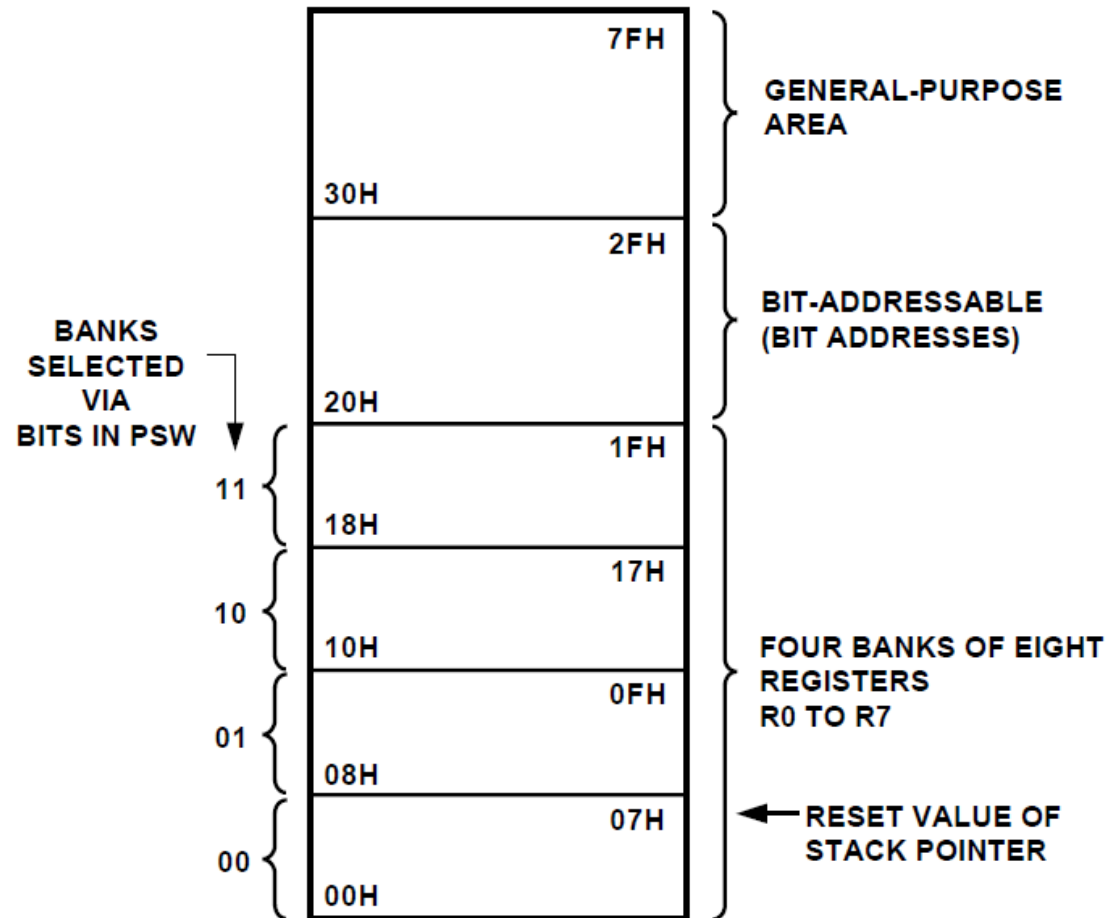
## Dahili RAM:

- Erişimin en hızlı olduğu bellek bölgesidir.
- Alt 128 byte'lık bölüme doğrudan veya dolaylı olarak erişim yapılabilirken, üst 128 byte'lık bölüme erişim yalnızca dolaylı olmalıdır.
- Dâhilî belleğin 20H-30H arasındaki bölümü **bit adreslenebilir** bölümdür. Buradaki her bitin adresi olup, tek bit gerektiği durumlarda kullanılabilir.
- 8052 işlemcisi 4 adet saklayıcı (register) bloğu içermektedir (RegBank 0-3) Bu bloklardan herhangi biri SFR'ler (special function register) ile seçilebilir.

IRAM Addr									Description
00	R0	R1	R2	R3	R4	R5	R6	R7	Reg. Bank 0
08	R0	R1	R2	R3	R4	R5	R6	R7	Reg. Bank 1
10	R0	R1	R2	R3	R4	R5	R6	R7	Reg. Bank 2
18	R0	R1	R2	R3	R4	R5	R6	R7	Reg. Bank 3
20	00	08	10	18	20	28	30	38	Bits 00-3F
28	40	48	50	58	60	68	70	78	Bits 40-7F
30	General User RAM & Stack Space (80 bytes, 30h-7Fh)								General IRAM
7F									
80	Special Function Registers (SFRs) (80h - FFh)								SFRs
:									
:									

<http://www.8052.com/tutmemor.phtml>

## Dahili RAM:



*Lower 128 Bytes of Internal Data Memory*

PSW: Program Status Word (SFR)

## Özel İşlevli Saklayıcılar (SFRs):

- Özel işlevli saklayıcılar:
  - Giriş/Çıkış (I/O) portlarına erişim
  - Seri porta yazma veya seri porttan okuma
  - Baud rate ayarlama
  - Zamanlayıcıları ayarlama kontrol etme
  - Kesmeleri düzenleme

gibi işlevleri yerine getirirler.

80	PE	SP	DPL	DPH				PCON	87
88	ICON	IMOD	TL0	TL1	TH0	TH1			8F
90	P1								97
98	SCON	SBUF							9F
A0	P2								A7
A8	IE								AF
B0	P3								B7
B8	IP								B9
C0									C7
CA									CF
D0	PSW								D7
D8									DF
E0	ACC								E7
E8									EF
F0	D								F7
F8									FF

Blue background are I/O port SFRs  
Yellow background are control SFRs  
Green background are other SFRs

<http://www.8052.com/tutmemor.phtml>

## Program Belleği:

- Program belleğinde 8052'nin koştığı program bulunmaktadır.
- ADuC841 denetleyicisi 62 KB elektriksel silinebilir flash bellek kullanmaktadır.
- Bu belleğe programın oluşturulması esnasında sabit veriler koyulabilir.
- **MOVC** komutu ile kod belleğinden bu şekilde yazılmış bir veri okunabilir.
- Program belleği, programın koşması sırasında da okunup yazılabilir.

## XRAM:

- ADuC841 denetleyicisi, 2 Kbyte harici bellek (XRAM) sağlamaktadır.
- XRAM, harici adres alanının alt 2 Kbyte'lık kesiminde bulunmaktadır.
- Bu bellek bölgesine erişmek için CFG841 biti "1" olarak ayarlanmalıdır (*MOV CFG841,#01H*).
- Bu belleğe **MOVX** komutu ile erişilir.

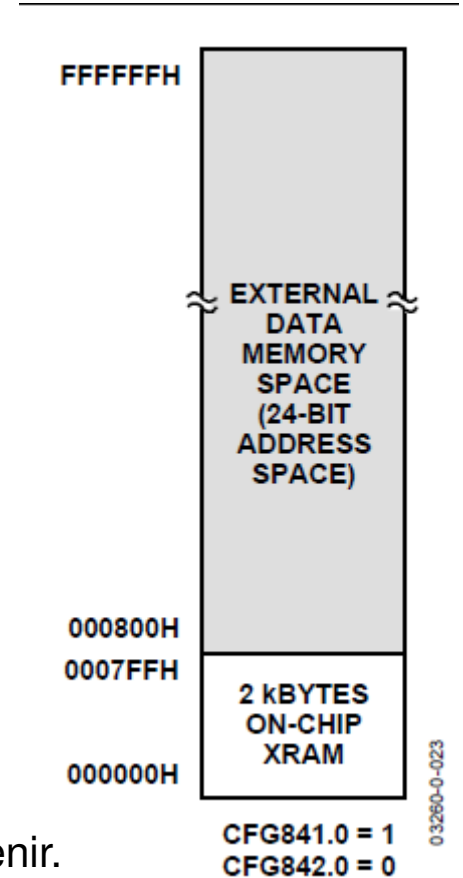
## MOVX

*MOVX A,@DPTR*

'Data pointer'ın gösterdiği adresteki veri A saklayıcısına yüklenir.

*MOVX @DPTR,A*

'A' in içeriği "Data pointer"ın gösterdiği adrese yüklenir.





## 8052 Programlama Dili

- ADuC841, yapısında 8052 tabanlı mikroişlemci çekirdeği bulundurmaktadır.
- 8052 işlemci 20 milyon işlem/saniye (MIPS) hızında çalışabilmektedir. Çoğu işlem tek çevrimde tamamlanabilmektedir.
- Genel olarak komut yapısı aşağıdaki gibidir:

**İŞLEM hedef, kaynak** (İşlem sonucu **hedef**'e yazılmaktadır. )

# 8052 Programlama Dili

## Saklayıcılar

- **Akümülatör ("A" ya da "ACC"):** Bütün "operand" gerektiren işlemlerde kullanılabilir. 8 bitlik bir saklayıcıdır.
- **B saklayıcısı ("B"):** Akümülatör ile çarpma ve bölme işlemlerinde kullanılır. Bunun yanında, genel amaçlı bir saklayıcı gibi de kullanılır.
- **Genel saklayıcılar (R0, R1, R2, R3, R4, R5, R6, R7):** Çoğu işlemde operand olarak kullanılabilirler.

Yalnız R0 ve R1, dolaylı adreslemede kullanılır.

Bellekte bulunan saklayıcı gözleri arasında geçiş yapılabilir.

## 8052 Komut Seti

Cebirsel komutlar	Mantıksal komutlar	Veri Aktarım komutları	İkililer üzerine uygulanan komutlar	Dallanma komutları	Diğer
ADD	ANL	MOV	CLR	JMP	NOP
ADDC	ORL	MOVC	SETB	RET	
SUBB	XRL	MOVX	CPL	RETI	
INC	CLR	PUSH	ANL	CALL	
DEC	CPL	POP	ORL	CJNE	
MUL	SWAP	XCH	MOV	JC	
DIV	RL	XCHD		JNC	
DA	RLC			JZ	
	RR			JNZ	
	RRC			DJNZ	
				JNB	
				JB	
				JBC	

## 8052 Komut Seti- MOV

Biçim: **MOV hedef , kaynak**

**MOVA,R1** : R1 saklayıcısının içeriğini A'ya kopyalar (**Saklayıcı adresleme**)



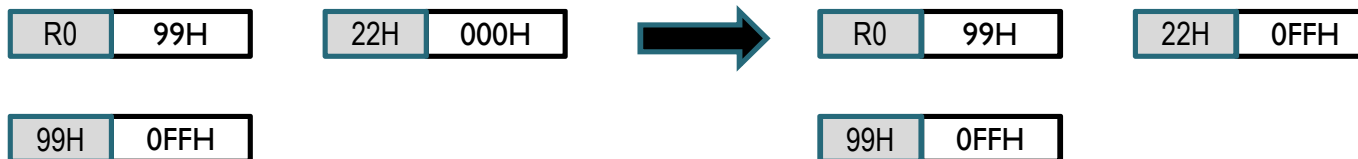
**MOV R7, #07H** : Sabit bir değeri R7 saklayıcısına yazar (**İvedi Adresleme**)



**MOV A,38H** : Bir adresin (00H-7FH) içeriğini A'ya kopyalar (**Doğrudan Adresleme**)

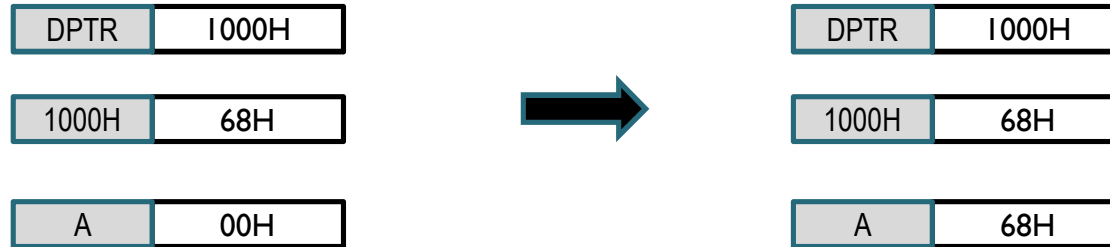


**MOV 22H,@R0** : R0 veya R1 saklayıcılarının gösterdiği adresin içeriğini 22H adresine kopyalar (**dolaylı adresleme**)



## 8052 Komut Seti- MOV

*MOVX A,@DPTR* : Harici belleğe 16 bitlik DPTR saklayıcısı kullanılarak erişilir. Erişim dolaylı olarak yapılmaktadır. DPTR'nin içeriğinin gösterdiği adresten bir byte'ı akümülatöre yükler.



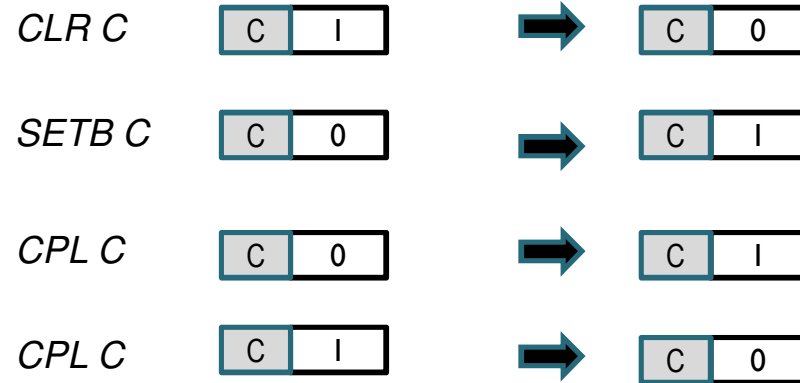
*MOVC A,@A+DPTR*

*MOVC A,@A+PC* : Kod belleğine DPTR veya PC saklayıcısı kullanılarak erişilir. Erişim dolaylı olarak yapılmaktadır. *MOVC*, kod belleğinden bir byte'ı akümülatöre yükler. Byte'ın yükleneceği kod belleğinin adresi Akümülatörün değerini DPTR veya PC ile toplayarak hesaplanır. PC kullanıldığında, PC akümülatörler toplanmadan önce 1 arttırılır.

## 8052 Komut Seti

### CLR, SET, CPL

- CLR: bir biti sıfırlar.
- SETB: bir biti birler.
- CPL: bir biti ters çevirir.



### XCH

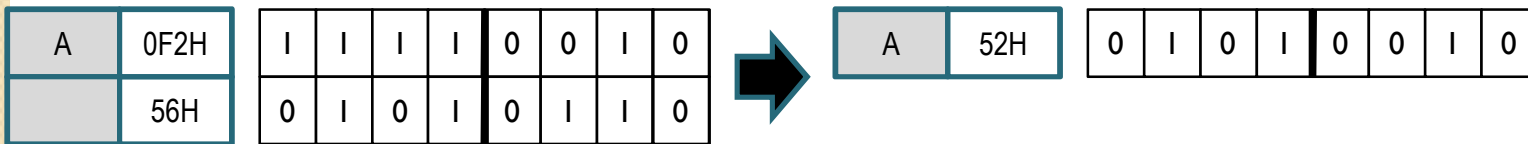
*XCH A, R7*

A ve R7 saklayıcılarının içeriklerini yer değiştirir.

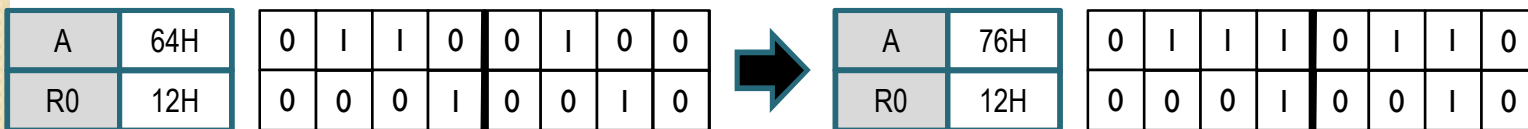


## ANL, ORL

*ANL A,#56H*



*ORL A,R0*



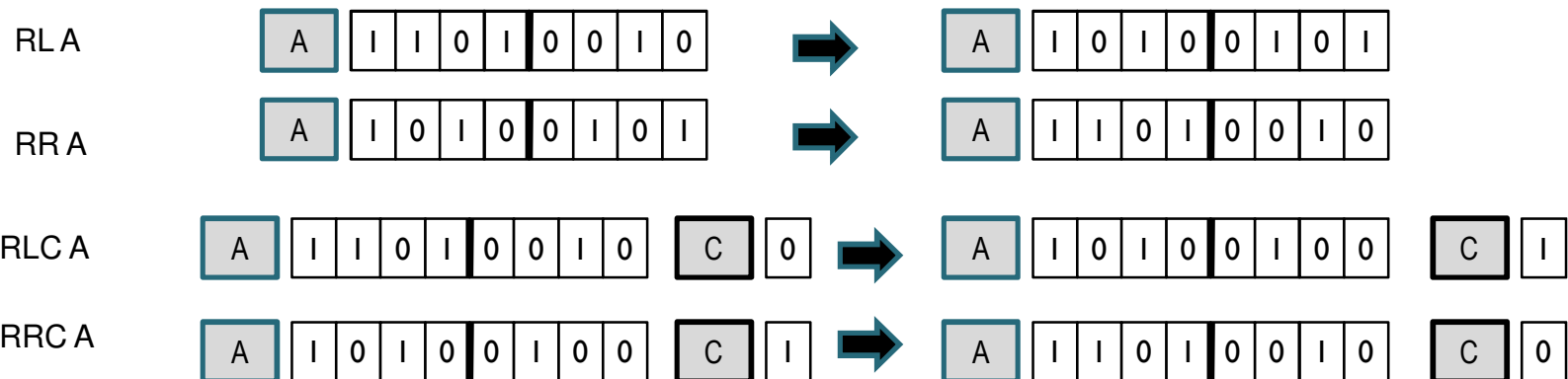
## RL,RLC,RR,RRC

RL: veriyi bir bit sola öteler (en soldaki bit en sağa taşınır)

RR: veriyi bir bit sağa öteler (en sağdaki bit en sola taşınır)

RLC: veriyi elde (C) bitini kullanarak bir bit sola öteler (C biti operand'ın en sağ bitine atanır)

RRC: veriyi elde (C) bitini kullanarak bir bit sağa öteler (C biti operand'ın en sol bitine atanır)





## MOV C

*MOV C A,@A+DPTR*

'Data pointer' in gösterdiği adresin 'A' kadar ötesindeki bir byte A saklayıcısına yüklenir.

### ÖRNEK

IO0010:

```
MOV    DPTR, #TITLE
MOV    A,B
INC    B
MOV C  A,@A+DPTR
JZ     IO0020
CALL   SENDCHAR
JMP    IO0010
```

TITLE:

```
DB 10,10,13,'_____',10,13
DB 'Analog Devices MicroConverter ADuC841',10,13
DB '    UART Demo Routine',10,13
```

## DALLANMA Komutları

**JMP hedef:** Koşulsuz Dallanma

**JZ hedef:** A=0 ise dallan , **JNZ hedef:** A≠0 ise dallan

**DJNZ Rn, rel:** saklayıcıyı bir azalt, Rn ≠0 ise dallan

TEKRAR:	MOV	A,B
	INC	B
	MOVC	A,@A+DPTR
	<b>JZ</b>	<b>CIKIS</b>
	CALL	SENDCHAR
	<b>JMP</b>	<b>TEKRAR</b>
CIKIS:	POP	B
	POP	ACC

**CALL hedef:** bir fonksiyonu çağır

**RET:** çağrılan fonksiyondan geri dön

**RETI:** çağrılan kesmeden geri dön

BLINK:	MOV	A,#010	; set delay length
	CPL	LED	; flash (complement) the red LED
	<b>CALL</b>	<b>DELAY</b>	; call software delay
	<b>JMP</b>	<b>BLINK</b>	; repeat indefinitely
DELAY:	MOV	R5,A	
DLY2:	MOV	R7,#090h	
DLY1:	MOV	R6,#0FFh	
	<b>DJNZ</b>	<b>R6,\$</b>	
	<b>DJNZ</b>	<b>R7,DLY1</b>	
	<b>DJNZ</b>	<b>R5,DLY2</b>	
	<b>RET</b>		

**Not:** (CALL komutu çağrıldıktan sonra mutlaka bir RET komutu çağrılmalıdır)

## MATEMATİKSEL İŞLEMLER

*ADD A,Rn*       $A+Rn \rightarrow A$

*ADDC A,Rn*       $A + Rn + C \rightarrow A$

*SUBB A,Rn*       $A - Rn \rightarrow A$ , (eğer  $Rn > A$  ise  $C=1$ )

*INC A*       $A+1 \rightarrow A$

*DEC A*       $A-1 \rightarrow A$

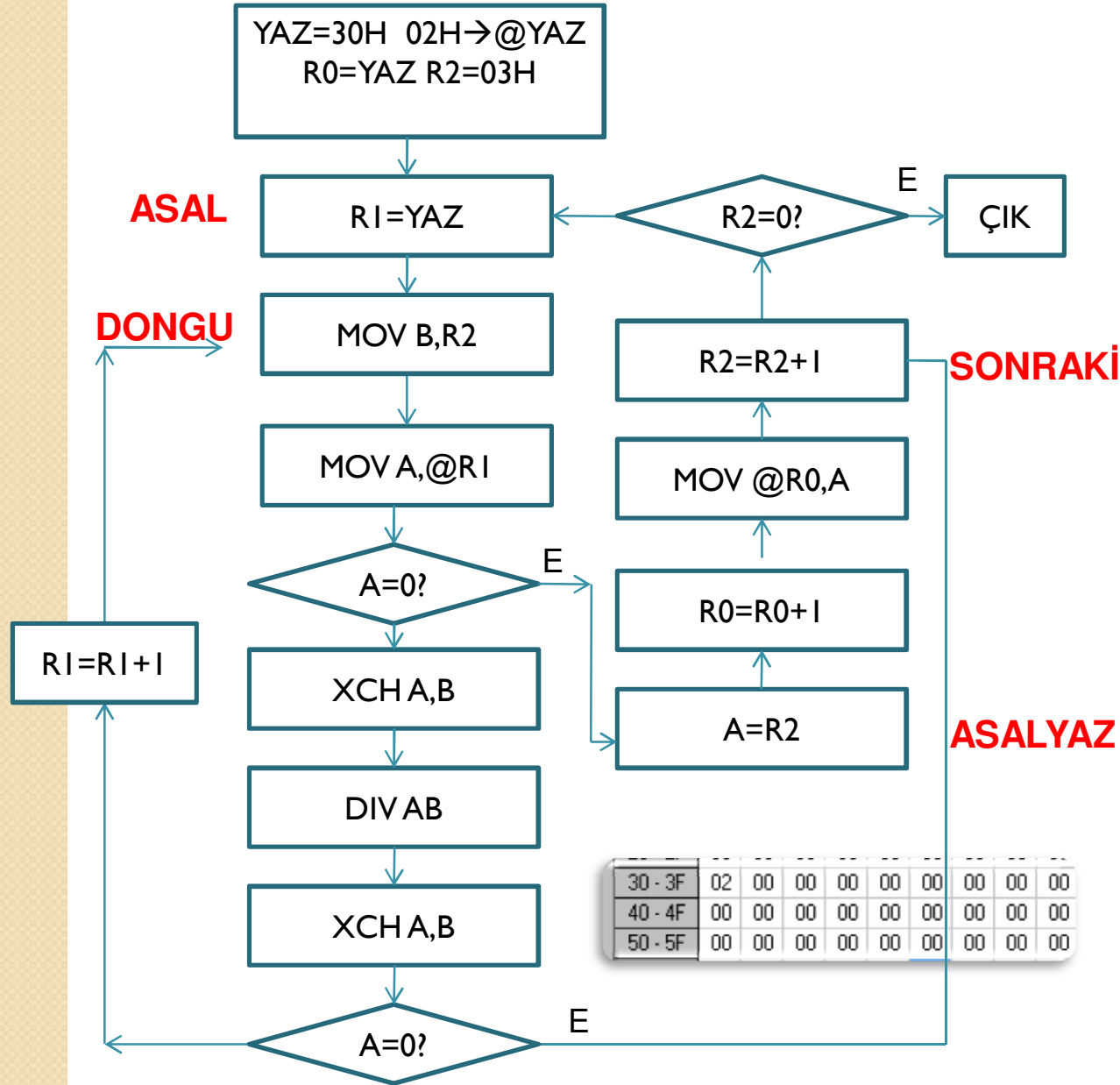
*MUL AB*       $A \times B \rightarrow A, B$  (yüksek anlamlı bayt B saklayıcısına, düşük anlamlı bayt A saklayıcısına yazılır)

*DIV AB*       $A/B \rightarrow A, B$  (Bölüm A'ya, kalan B'ye yazılır)

**NOT:** 8052, 8 bitlik bir programlama dilidir. Bir saklayıcı maksimum 255 (0xFF) değerini alabilir. Eğer işlem sonucu 255'ten büyükse, sonucun 256'ya bölümünden kalan gösterilir.

# Örnek

2-255 arasındaki asal sayılar 30H adresinden başlayarak belleğe yazılacaktır.



YAZ EQU 30H

```
mov 30H, #02H
mov R0, #YAZ
mov R1, #YAZ
mov R2, #03
```

**asal:**

```
mov r1, #YAZ
```

**dongu:**

```
mov B,R2
mov A,@R1
JZ asalyaz
```

```
xch A,B
DIV AB
XCH A,B
JZ sonraki
inc R1
jmp dongu
```

**asalyaz:**

```
mov A,R2
inc R0
mov @R0,A
```

**sonraki:**

```
INC R2
MOV A,R2
JnZ asal
```

```
jmp $
```

## Kaynaklar

- [www.analog.com](http://www.analog.com) - ADuC 841 data sheet
- <http://www.8052.com/set8051.php>

**Dinlediğiniz için Teşekkürler**