LAMPIRAN

```
;2416 26f 3d5 3d5 479 = 32aa

PWM     BIT     p1.0

data     equ     $60

        org     rom
        Mov     TMOD,#$21

loop:   Mov     TH0,#0
        Mov     TL0,#0
        jnb     PWM,*           ;_-
        setb    TR0
        jb      PWM,*

        mov      R0,#Data

        clr     TR0
        mov     A,TH0
        cjne    A,#$20,$+3
        Jc      loop
lebih_besar
        lcall    tampil_Timer

        jnb     PWM,*           ;atas-bawah
        setb    TR0
        jb      PWM,*
        clr      TR0
        lcall    tampil_Timer

        jnb     PWM,*           ;maju-mundur
        setb    TR0
        jb      PWM,*
        clr      TR0
        lcall    tampil_Timer

        jnb     PWM,*           ;miring
        setb    TR0
        jb      PWM,*
        clr      TR0
        lcall    tampil_Timer

        jnb     PWM,*           ;putar
        setb    TR0
        jb      PWM,*
        clr      TR0
        lcall    tampil_Timer

        mov      R1,#5
        mov      R0,#Data
        mov      R3,#0
        mov      r4,#0

ambil_lagi:
        mov      A,@R0
        push     A
        inc      R0
        lcall    ascii_out
        mov      A,@R0
        mov      R5,A

        clr      C
        add      A,R3
        mov      R3,A
        pop      A
        addc     A,R4
        mov      R4,A
```

```
        inc     R0
        mov     A,R5
        lcall   ascii_out
        lcall   space_code

        djnz    R1,ambil_lagi

        mov    A,#'='
        lcall  serial_out
        lcall   space_code

        mov     A,R4
        lcall   ascii_out
        mov     A,R3
        lcall   ascii_out

        mov    A,#$D
        lcall  serial_out
        mov    A,#$A
        lcall  serial_out
        ajmp   loop

tampil_Timer:
        mov    A,TH0
        MOV     @R0,A
        inc     R0

        mov    A,TL0
        MOV     @R0,A
        inc     R0
        Mov    TL0,#0
        Mov    TH0,#0
        ret
```

```
                                REMOTE hasil tuning
;R0 = alamat low indirect RAM
;R1 = alamat high indirect RAM
;R2 = pengulangan jumlah word copy SEE-RAM
;R5 = pointer data Control ke ? pada interupt
;R6 = no high low PWM pada interupt
;R7 = Alamat SEE
;B = Alamat SEE
;A = data hasil baca SEE
;DataSEE = data tulis ke SEE

PWM                     BIT     P2.7
SCL                     BIT     P1.0
SDA                     BIT     P1.1

Port_Keypad             EQU     $80     ;P0

Alamat_SEE              EQU     0
Control_SEE             EQU     $A0
increment               EQU     $10
increment_naik          EQU     $40

; EQU data PWM
Stabil_Off              EQU     -$248   ;263 ok
Stabil_Maju             EQU     -$4D0   ;3D5 ok
Stabil_Miring           EQU     -$3A0   ;3D5 ok
Stabil_Putar            EQU     -$45D   ;479 ok

Start_High              EQU     -$2408 ;2424 ok
Max_Control             EQU     $3223

;2424 263 3d5 3d5 479 = 32aa
;2408 248 3BB 3BB 45D = 3223

Start_Low               EQU     -$F4    ; 10E ok

Atas_Low                EQU     -$104    ; 121 ok
Maju_Low                EQU     -$106    ; 123 ok
Miring_Low              EQU     -$108    ; 125 ok
Putar_Low               EQU     -$108    ; 123 ok

Stop_High               EQU     -$1AD   ; 1C7 ok
Stop_L1                 EQU     -$10A   ; 125 ok
Stop_L2                 EQU     -$113   ; 12C ok
Stop_L3                 EQU     -$114   ; 12E ok
Stop_L4                 EQU     -$116   ; 130 ok
Stop_L5                 EQU     -$F8    ; 111 ok

Min_High                EQU     -$1B4   ; (FE5C) 1CF
Max_High                EQU     -$598   ; (FA58) 5D3

; EQU data tombol keypad
Naik                    EQU     $F6     ; *
Turun                   EQU     $F5     ; 0
Maju                    EQU     $EE     ; 7
Mundur                  EQU     $ED     ; 8
Miring_Ki               EQU     $DE     ; 4
Miring_Ka               EQU     $BE     ; 1
Putar_Ki                EQU     $DD     ; 5
Putar_Ka                EQU     $BD     ; 2

Reset_NT                EQU     $F3     ; #
Reset_MM                EQU     $EB     ; 9
Reset_Miring            EQU     $DB     ; 6
Reset_Putar             EQU     $BB     ; 3

Tuning_Naik             EQU     $F2     ; #*
Tuning_Turun            EQU     $F1     ; #0
Tuning_Maju             EQU     $EA     ; 97
```

```
                                    REMOTE hasil tuning
Tuning_Mundur               EQU     $E9       ; 98
Tuning_Miring_Ki            EQU     $DA       ; 64
Tuning_Miring_Ka            EQU     $9A       ; 61
Tuning_Putar_Ki             EQU     $99       ; 35
Tuning_Putar_Ka             EQU     $B9       ; 32

Isi_SEE                     EQU     $AB       ; 93

.DATA
        Org     $60
Data_Control                Ds      8       ; $60-$67
Slave_AddrSEE               Ds      1       ; $68
DataSEE                     Ds      1       ; $69


.CODE
        org     0
        ljmp    Start

        Org     $B                  ; Interupt Timer 0
        ljmp    Ganti_PWM

Start:
        clr     TR0
        clr     TR1
        mov     TMOD,#$11           ; timer 0 & 1 = mode 1 (16 bit)

        clr     TF1                 ; reset timer auto reload
        mov     TH1,#$FD
        mov     TL1,#0

        clr     TF0                 ; set timer PWM
        mov     TH0,#0
        mov     TL0,#0
        mov     R6,#0
        setb    TR0                 ; aktifkan Timer0
        mov     IE,#$82             ; enable interupt Timer 0 (EA, ET0)
        setb    TR1

Cek_Keypad_OV:
        mov     A,Port_Keypad
        orl     A,#$80
        cjne    A,#$FF,Keypad_Ditekan
        jbc     TF1,Copy_SEE_ke_RAM
        sjmp    Cek_Keypad_OV

Copy_SEE_ke_RAM:
        clr     TR1
        mov     R7,#Control_SEE         ; alamat SEE
        mov     B,#Alamat_SEE+2         ; alamat SEE

        mov     R0,#Data_Control+2      ; alamat RAM
        mov     R1,#Data_Control+3      ; alamat RAM
        mov     R2,#3                   ; pengulangan jumlah byte yg dicopy

copy_SR_lagi:
        lcall   baca_See8b
        inc     B
        push    A
        lcall   baca_See8b
        inc     B

        mov     IE,#0               ; Disable interupt Timer 0 (EA, ET0)
        mov     @R1,A
        pop     A
        mov     @R0,A
        mov     IE,#$82             ; enable interupt Timer 0 (EA, ET0)
```

```
        inc     R0
        inc     R0
        inc     R1
        inc     R1
        djnz    R2,copy_SR_lagi

Delay_Autoreload:
        mov     B,#0
        mov     A,#0
Tunda:  djnz    A,*
        djnz    B,tunda

Enable_Timer:
        clr     TF1
        mov     TH1,#0
        mov     TL1,#0
        setb    TR1
        ljmp    cek_Keypad_OV

;---------------------------------- tombol naik
;*'---------------------------------
Keypad_Ditekan:
        clr     TR1                     ; timer auto-reload off
        mov     R7,#Control_SEE         ; alamat SEE
        cjne    A,#Naik,Cek_Turun
        mov     R0,#Data_Control+1
        lcall   cek_RAM_Max             ; cek min/max
        jnc     Naik_lagi
        ljmp    Enable_Timer            ; out of range, tdk ada perubahan data
Naik_lagi:
        mov     R0,#Data_Control        ; in range, ubah data RAM
        mov     R1,#Data_Control+1
        lcall   Kurangi_RAM_naik
        ljmp    Delay_Keypad

;---------------------------------- tombol turun '0'
----------------------------------
Cek_Turun:
        cjne    A,#Turun,Cek_Maju
        mov     R0,#Data_Control+1
        lcall   cek_RAM_Min             ; cek min/max
        jc      Turun_lagi

        ljmp    Enable_Timer            ; out of range, tdk ada perubahan data
Turun_Lagi:
        mov     R0,#Data_Control        ; in range, ubah data RAM
        mov     R1,#Data_Control+1
        lcall   tambahkan_RAM_naik
        ljmp    Delay_Keypad

;---------------------------------- tombol Maju '7'
----------------------------------
Cek_Maju:
        cjne    A,#Maju,Cek_Mundur
        mov     R0,#Data_Control+3
        lcall   cek_RAM_Min             ; cek min/max
        jc      Maju_lagi
        ljmp    Enable_Timer            ; out of range, tdk ada perubahan data
Maju_Lagi:
        mov     R0,#Data_Control+2      ; in range, ubah data RAM
        mov     R1,#Data_Control+3
        lcall   tambahkan_RAM
        ljmp    Delay_Keypad

;---------------------------------- tombol Mundur '8'
----------------------------------
Cek_Mundur:
        cjne    A,#Mundur,Cek_MKiri
```

```
        mov     R0,#Data_Control+3
        lcall   cek_RAM_Max
        jnc     Mundur_lagi
        ljmp    Enable_Timer            ; out of range, tdk ada perubahan data
Mundur_lagi:
        mov     R0,#Data_Control+2
        mov     R1,#Data_Control+3
        lcall   Kurangi_RAM
        ljmp    Delay_Keypad

;--------------------------------- tombol Miring Kiri '4'
---------------------------------
Cek_MKiri:
        cjne    A,#Miring_Ki,Cek_MKanan
        mov     R0,#Data_Control+5
        lcall   cek_RAM_Max
        jnc     MKiri_lagi
        ljmp    Enable_Timer            ; out of range, tdk ada perubahan data
MKiri_lagi:
        mov     R0,#Data_Control+4
        mov     R1,#Data_Control+5
        lcall   Kurangi_RAM
        ljmp    Delay_Keypad

;--------------------------------- tombol Miring Kanan '1'
---------------------------------
Cek_MKanan:
        cjne    A,#Miring_Ka,Cek_PKiri
        mov     R0,#Data_Control+5
        lcall   cek_RAM_Min
        jc      MKanan_lagi
        ljmp    Enable_Timer            ; out of range, tdk ada perubahan data
MKanan_Lagi:
        mov     R0,#Data_Control+4
        mov     R1,#Data_Control+5
        lcall   tambahkan_RAM
        ljmp    Delay_Keypad

;--------------------------------- tombol Putar Kiri '5'
---------------------------------
Cek_PKiri:
        cjne    A,#Putar_Ki,Cek_PKanan
        mov     R0,#Data_Control+7
        lcall   cek_RAM_Min
        jc      P_Kiri_lagi
        ljmp    Enable_Timer            ; out of range, tdk ada perubahan data
P_Kiri_Lagi:
        mov     R0,#Data_Control+6
        mov     R1,#Data_Control+7
        lcall   tambahkan_RAM
        ljmp    Delay_Keypad

;--------------------------------- tombol Putar Kanan '2'
---------------------------------
Cek_PKanan:
        cjne    A,#Putar_Ka,Cek_ResetNT
        mov     R0,#Data_Control+7
        lcall   cek_RAM_Max
        jnc     PKanan_lagi
        ljmp    Enable_Timer            ; out of range, tdk ada perubahan data
PKanan_lagi:
        mov     R0,#Data_Control+6
        mov     R1,#Data_Control+7
        lcall   Kurangi_RAM
        ljmp    Delay_Keypad

;--------------------------------- tombol OFF Putaran '#'
---------------------------------
```

```
Cek_ResetNT:
        cjne    A,#Reset_NT,Cek_RMM
Reset_HighLow:
        mov     B,#Alamat_SEE           ; alamat SEE
        mov     R0,#Data_Control        ; alamat RAM
        mov     R1,#Data_Control+1      ; alamat RAM
        mov     R2,#2                   ; pengulangan jumlah Word yg dicopy
        lcall   copy_SEE_RAM
        ljmp    Delay_Keypad

;------------------------------ tombol Reset Maju Mundur '9'
------------------------------
Cek_RMM:
        cjne    A,#Reset_MM,Cek_RMiring
        mov     B,#Alamat_SEE+2         ; alamat SEE
        mov     R0,#Data_Control+2      ; alamat RAM
        mov     R1,#Data_Control+3      ; alamat RAM
        mov     R2,#2                   ; pengulangan jumlah Word yg dicopy
        lcall   copy_SEE_RAM
        ljmp    Delay_Keypad

;-------------------------------- tombol Reset Miring '6'
--------------------------------
Cek_RMiring:
        cjne    A,#Reset_Miring,Cek_RPutar
        mov     B,#Alamat_SEE+4         ; alamat SEE
        mov     R0,#Data_Control+4      ; alamat RAM
        mov     R1,#Data_Control+5      ; alamat RAM
        mov     R2,#2                   ; pengulangan jumlah Word yg dicopy
        lcall   copy_SEE_RAM
        ljmp    Delay_Keypad

;-------------------------------- tombol Reset Putar '3'
--------------------------------
Cek_RPutar:
        cjne    A,#Reset_Putar,Cek_TNaik
        mov     B,#Alamat_SEE+6         ; alamat SEE
        mov     R0,#Data_Control+6      ; alamat RAM
        mov     R1,#Data_Control+7      ; alamat RAM
        mov     R2,#2                   ; pengulangan jumlah Word yg dicopy
        lcall   copy_SEE_RAM
        ljmp    Delay_Keypad

;------------------------------ tombol Tuning Naik '#' dan '*'
------------------------------
Cek_TNaik:
        cjne    A,#Tuning_Naik,Cek_TTurun
        mov     B,#Alamat_SEE+1         ; alamat SEE
        lcall   cek_SEE_Max             ; cek min/max
        jnc     Naik_SEE_lagi
        ljmp    Delay_Keypad            ; out of range, tdk ada perubahan data
Naik_SEE_lagi:
        mov     B,#Alamat_SEE           ; in range, ubah data RAM
        lcall   Kurangi_SEE
        ljmp    Delay_Keypad

;------------------------------ tombol Tuning Turun '#' dan '0'
------------------------------
Cek_TTurun:
        cjne    A,#Tuning_Turun,Cek_TMaju
        mov     B,#Alamat_SEE+1         ; alamat SEE
        lcall   cek_SEE_Min             ; cek min/max
        jc      Turun_SEE_lagi
        ljmp    Delay_Keypad            ; out of range, tdk ada perubahan data
Turun_SEE_lagi:
        mov     B,#Alamat_SEE           ; in range, ubah data RAM
        lcall   Tambahkan_SEE
        ljmp    Delay_Keypad
```

```
;------------------------------ tombol Tuning Maju '9' dan '7'
------------------------------
Cek_TMaju:
        cjne    A,#Tuning_Maju,Cek_TMundur
        mov     B,#Alamat_SEE+3         ; alamat SEE
        lcall   cek_SEE_Min             ; cek min/max
        jc      Maju_SEE_lagi
        ljmp    Delay_Keypad            ; out of range, tdk ada perubahan data
Maju_SEE_lagi:
        mov     B,#Alamat_SEE+2         ; in range, ubah data RAM
        lcall   Tambahkan_SEE
        ljmp    Delay_Keypad

;------------------------------ tombol Tuning Mundur '9' dan '8'
------------------------------
Cek_TMundur:
        cjne    A,#Tuning_Mundur,Cek_TMKiri
        mov     B,#Alamat_SEE+3         ; alamat SEE
        lcall   cek_SEE_Max             ; cek min/max
        jnc     Mundur_SEE_lagi
        ljmp    Delay_Keypad            ; out of range, tdk ada perubahan data
Mundur_SEE_lagi:
        mov     B,#Alamat_SEE+2         ; in range, ubah data RAM
        lcall   Kurangi_SEE
        ljmp    Delay_Keypad

;------------------------------ tombol Tuning Miring Kiri '6' dan '4'
------------------------------
Cek_TMKiri:
        cjne    A,#Tuning_Miring_Ki,Cek_TMKanan
        mov     B,#Alamat_SEE+5         ; alamat SEE
        lcall   cek_SEE_Max             ; cek min/max
        jnc     MKiri_SEE_lagi
        ljmp    Delay_Keypad            ; out of range, tdk ada perubahan data
MKiri_SEE_lagi:
        mov     B,#Alamat_SEE+4         ; in range, ubah data RAM
        lcall   Kurangi_SEE
        ljmp    Delay_Keypad

;------------------------------ tombol Tuning Miring Kanan '6' dan '1'
------------------------------
Cek_TMKanan:
        cjne    A,#Tuning_Miring_Ka,Cek_TPKiri
        mov     B,#Alamat_SEE+5         ; alamat SEE
        lcall   cek_SEE_Min             ; cek min/max
        jc      MKanan_SEE_lagi
        ljmp    Delay_Keypad            ; out of range, tdk ada perubahan data
MKanan_SEE_lagi:
        mov     B,#Alamat_SEE+4         ; in range, ubah data RAM
        lcall   Tambahkan_SEE
        ljmp    Delay_Keypad

;------------------------------ tombol Tuning Putar Kiri '3' dan '5'
------------------------------
Cek_TPKiri:
        cjne    A,#Tuning_Putar_Ki,Cek_TPKanan
        mov     B,#Alamat_SEE+7         ; alamat SEE
        lcall   cek_SEE_Min             ; cek min/max
        jc      PKiri_SEE_lagi
        ljmp    Delay_Keypad            ; out of range, tdk ada perubahan data
PKiri_SEE_lagi:
        mov     B,#Alamat_SEE+6         ; in range, ubah data RAM
        lcall   Tambahkan_SEE
        ljmp    Delay_Keypad

;------------------------------ tombol Tuning Putar Kanan '3' dan '2'
------------------------------
```

```
Cek_TPKanan:
        cjne    A,#Tuning_Putar_Ka,Copy_Flash_SEE
        mov     B,#Alamat_SEE+7         ; alamat SEE
        lcall   cek_SEE_Max             ; cek min/max
        jnc     PKanan_SEE_lagi
        ljmp    Delay_Keypad            ; out of range, tdk ada perubahan data
PKanan_SEE_lagi:
        mov     B,#Alamat_SEE+6         ; in range, ubah data RAM
        lcall   Kurangi_SEE

Delay_Keypad:
        mov     R0,#2
        mov     B,#0
        mov     A,#0
Dly:    djnz    A,*
        djnz    B,Dly
        djnz    R0,Dly
        ljmp    Enable_Timer

;--------------------------- tombol Copy FLASH - SEE '9' dan '3'
---------------------------
Copy_Flash_SEE:
        cjne    A,#Isi_SEE,Delay_Keypad

tunggu_lepas:
        mov     A,Port_Keypad
        orl     A,#$80
        cjne    A,#$FF,tunggu_lepas

        mov     DPTR,#Data_Hasil_Tuning         ; alamat Flash
        mov     B,#Alamat_SEE           ; alamat SEE
        mov     R2,#8                   ; pengulangan jumlah Word yg dicopy
Isi_lagi:
        clr     A
        movc    A,@A+DPTR
        mov     datasee,A
        lcall   Tulis_SEE8b
        inc     DPTR
        inc     B
        djnz    R2,Isi_lagi
        ljmp    Reset_HighLow

;------------------------------------------------------------------------------
-------
copy_SEE_RAM:
        lcall   baca_See8b
        inc     B
        push    A
        lcall   baca_See8b
        mov     IE,#0           ; Disable interupt Timer 0 (EA, ET0)
        mov     @R1,A
        pop     A
        mov     @R0,A
        mov     IE,#$82         ; enable interupt Timer 0 (EA, ET0)
        ret

;##############################################################################
########
Kurangi_RAM_naik:
        mov     A,@R0
        clr     C
        subb    A,#Increment_naik
        sjmp    kurang

Kurangi_RAM:
        mov     A,@R0
        clr     C
        subb    A,#Increment
```

```
kurang:
        push    A

        push    0
        inc     R0
        mov     A,@R0
        pop     0

        subb    A,#0
simpan_ke_RAM:
        mov     IE,#0           ; Disable interupt Timer 0 (EA, ET0)
        mov     @R1,A
        pop     A
        mov     @R0,A
        mov     IE,#$82         ; enable interupt Timer 0 (EA, ET0)



        ret


Tambahkan_RAM_naik:
        mov     A,@R0
        add     A,#Increment_naik
        sjmp    tambah


Tambahkan_RAM:
        mov     A,@R0
        add     A,#Increment
tambah:
        push    A

        push    0
        inc     R0
        mov     A,@R0
        pop     0
        addc    A,#0
        sjmp    simpan_ke_RAM

;#########################################################################
########
Kurangi_SEE:
        lcall   baca_See8b
        clr     C
        subb    A,#Increment
        mov     datasee,A
        lcall   Tulis_SEE8b
        inc     B
        lcall   baca_See8b
        subb    A,#0
        mov     datasee,A
        lcall   Tulis_SEE8b
        ret

Tambahkan_SEE:
        lcall   baca_See8b
        add     A,#Increment
        mov     datasee,A
        lcall   Tulis_SEE8b

        inc     B
        lcall   baca_See8b
        addc    A,#0
        mov     datasee,A
        lcall   Tulis_SEE8b
        ret
```

```
;#########################################################################################
cek_RAM_Min:
        cjne    @R0,#Min_High/$100,out_RAM_min
        dec     R0
        cjne    @R0,#Min_High,out_RAM_min
out_RAM_min:
        ret

Cek_RAM_Max:
        cjne    @R0,#Max_High/$100,out_RAM_max  ; tdk sama(<)    C=0  sama C=1
        dec     R0
        cjne    @R0,#Max_High,out_RAM_max       ; lebih kecil(<) C=0  >=   C=1
out_RAM_max:
        ret


cek_SEE_Min:
        lcall   baca_See8b
        cjne    A,#Min_High/$100,out_SEE_min
        dec     B
        lcall   baca_See8b
        cjne    A,#Min_High,out_SEE_min
out_SEE_min:
        ret

Cek_SEE_Max:
        lcall   baca_See8b
        cjne    A,#Max_High/$100,out_SEE_max    ; tdk sama(<)    C=0  sama C=1
        dec     B
        lcall   baca_See8b
        cjne    A,#Max_High,out_SEE_max         ; lebih kecil(<) C=0  >=   C=1
out_SEE_max:
        ret

;=========================================================================================
;=========================================================================================
;=========================================================================================
;=========================================================================================
;=========================================================================================
;=========================================================================================
Ganti_PWM:
        push    PSW
        push    A
        push    B
        push    DPH
        push    DPL
        push    1

        mov     A,R6
        mov     DPTR,#Lompat
        jmp     @A+DPTR

Lompat: ajmp    H_Start_PWM
        ajmp    L_Start_PWM
        ajmp    H_AwalControl
        ajmp    L_AtasBawah
        ajmp    H_Control
        ajmp    L_MajuMundur
        ajmp    H_Control
        ajmp    L_Miring
```

```
        ajmp    H_Control
        ajmp    L_Putar
        ajmp    H_Stop_PWM
        ajmp    L_Stop1
        ajmp    H_Stop_PWM
        ajmp    L_Stop2
        ajmp    H_Stop_PWM
        ajmp    L_Stop3
        ajmp    H_Stop_PWM
        ajmp    L_Stop4
        ajmp    H_Stop5_PWM
;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_Stop5:
        clr     PWM
        mov     TL0,#Stop_L5
        mov     TH0,#Stop_L5/$100
        mov     R6,#0
        ljmp    selesai_Skip

;1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
111
H_Start_PWM:
        setb    PWM
        mov     TL0,#Start_High
        mov     TH0,#Start_High/$100

Selesai_Timer:
        inc     R6
        inc     R6
selesai_Skip:
        pop     1
        pop     DPL
        pop     DPH
        pop     B
        pop     A
        pop     PSW
        reti

;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_Start_PWM:
        clr     PWM
        mov     TL0,#Start_Low
        mov     TH0,#Start_Low/$100
        ljmp    selesai_Timer

;1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
111
H_AwalControl:
        mov     R5,#Data_Control
H_Control:
        mov     R1,5
        setb    PWM
        mov     TL0,@R1
        inc     R1
        mov     TH0,@R1
        inc     R1
        mov     R5,1
        ljmp    selesai_Timer

;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_AtasBawah:
        clr     PWM
        mov     TL0,#Atas_Low
        mov     TH0,#Atas_Low/$100
        ljmp    selesai_Timer
```

```
;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_MajuMundur:
        clr     PWM
        mov     TL0,#Maju_Low
        mov     TH0,#Maju_Low/$100
        ljmp    selesai_Timer

;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_Miring:
        clr     PWM
        mov     TL0,#Miring_Low
        mov     TH0,#Miring_Low/$100
        ljmp    selesai_Timer

;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_Putar:
        clr     PWM
        mov     TL0,#Putar_Low
        mov     TH0,#Putar_Low/$100
        ljmp    selesai_Timer

;1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
111
H_Stop_PWM:
        setb    PWM
H_stop: mov     TL0,#Stop_High
        mov     TH0,#Stop_High/$100
        ljmp    selesai_Timer

H_Stop5_PWM:
        setb    PWM
        ljmp    H_stop
;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_Stop1:
        clr     PWM
        mov     TL0,#Stop_L1
        mov     TH0,#Stop_L1/$100
        ljmp    selesai_Timer

;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_Stop2:
        clr     PWM
        mov     TL0,#Stop_L2
        mov     TH0,#Stop_L2/$100
        ljmp    selesai_Timer

;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_Stop3:
        clr     PWM
        mov     TL0,#Stop_L3
        mov     TH0,#Stop_L3/$100
        ljmp    selesai_Timer

;0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
L_Stop4:
        clr     PWM
        mov     TL0,#Stop_L4
        mov     TH0,#Stop_L4/$100
        ljmp    selesai_Timer
```

```
;=============================================================================
=
Tulis_SEE8b:
        Lcall   Siapkan8bAlamatSEE
        Jc      Wrong_Write8b
        Mov     A,DataSEE
        Lcall   KirimDataSEE
        Jc      Wrong_Write8b
        Lcall   Buat_StopBit
        Ret
Wrong_Write8b:
        Lcall   Buat_StopBit
        Clr     C
        Ljmp    Tulis_SEE8b

;=============================================================================
=
Baca_SEE8b:
        Clr     C
        Lcall   Siapkan8bAlamatSEE
        Jc      Wrong_Read8b
        Lcall   Buat_StartBit           ;Kirim Device Address dengan
        Lcall   ModeBacaSEE
        Jc      Wrong_read8b
        Lcall   BacaDataSEE
        Ret
Wrong_Read8b:
        Lcall   Buat_StopBit
        Clr     C
        Ljmp    Baca_SEE8b

;=============================================================================
=
Siapkan8bAlamatSEE:
        Lcall   KirimDeviceAddress
        Jc      SalahTulisAlamat
        Lcall   Kirim1WordAddress
SalahTulisAlamat:
        Ret

;=============================================================================
=
KirimDataSEE:
        Push    B
        Mov     B,#8
Send8_bitloop
        Rlc     A
        Mov     SDA,C
        Lcall   Pulse_SEE
        Djnz    B,Send8_bitloop
        Pop     B
        Clr     C
        Lcall   Ambil_Ack
        Ret

;=============================================================================
=
Buat_StopBit:
        Clr     SDA
        Setb    SCL
        Setb    SDA
        Clr     SCL
        Ret

Buat_StartBit:
        Setb    SDA
        Setb    SCL
```

```
        Clr     SDA
        Clr     SCL
        Ret

;=============================================================================
=
ModeBacaSEE:
        Push    A                       ;
        Mov     A,R7
        Setb    A.0
        Mov     Slave_AddrSEE,A
        Pop     A
        Mov     A,Slave_AddrSEE
        Lcall   KirimDataSEE
        Ret

;=============================================================================
=
BacaDataSEE:
        Push    B
        Mov     B,#08H
        Clr     A
LoopBacaSEE16b:
        Push    B
        Rl      A
        Setb    SDA
        Setb    SCL
        Clr     C
        Mov     C,SDA
        Mov     A.0,C
        Clr     SCL
        Pop     B
        Djnz    B,LoopBacaSEE16b
        Lcall   Ambil_Ack
        Lcall   Ambil_Ack
        Lcall   Buat_StopBit
        Pop     B
        Ret

;=============================================================================
=
KirimDeviceAddress:
        Lcall   Buat_StartBit
        Push    A
        Mov     A,R7                    ;Device Address
        Lcall   KirimDataSEE            ;
        Pop     A                       ;
        Ret

;=============================================================================
=
Kirim1WordAddress:
        Push    A                       ;
        Mov     A,B                     ;First Word Address
        Lcall   KirimDataSEE            ;
        Pop     A                       ;
        Ret

;=============================================================================
=
Pulse_SEE:
        Push    b
        Setb    SCL
        Clr     SCL
        Pop     B
        Ret

;=============================================================================
```

```
=
Ambil_Ack:
        Clr     C
        Setb    SDA
        Setb    SCL
        Mov     C,SDA
        Clr     SCL
        Ret

;-------------------------------------------------------------------------------
---
Data_Hasil_Tuning:
        DW      Stabil_Off,Stabil_Maju,Stabil_Miring,Stabil_Putar
```

Features

- Compatible with MCS-51® Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

Description

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

8-bit Microcontroller with 4K Bytes In-System Programmable Flash

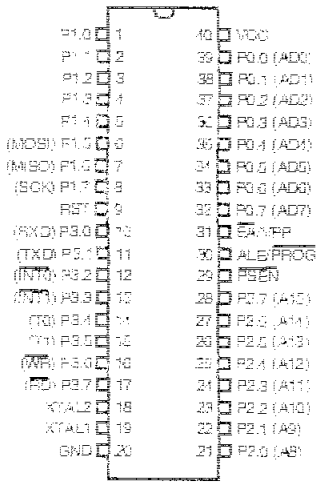
AT89S51

Preliminary

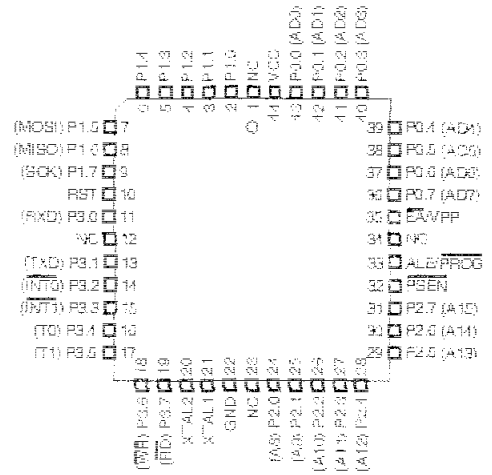
Rev. 2487A–10/01

Pin Configurations

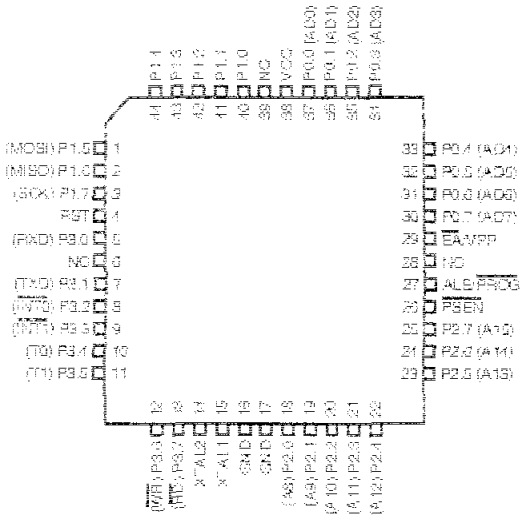
PDIP



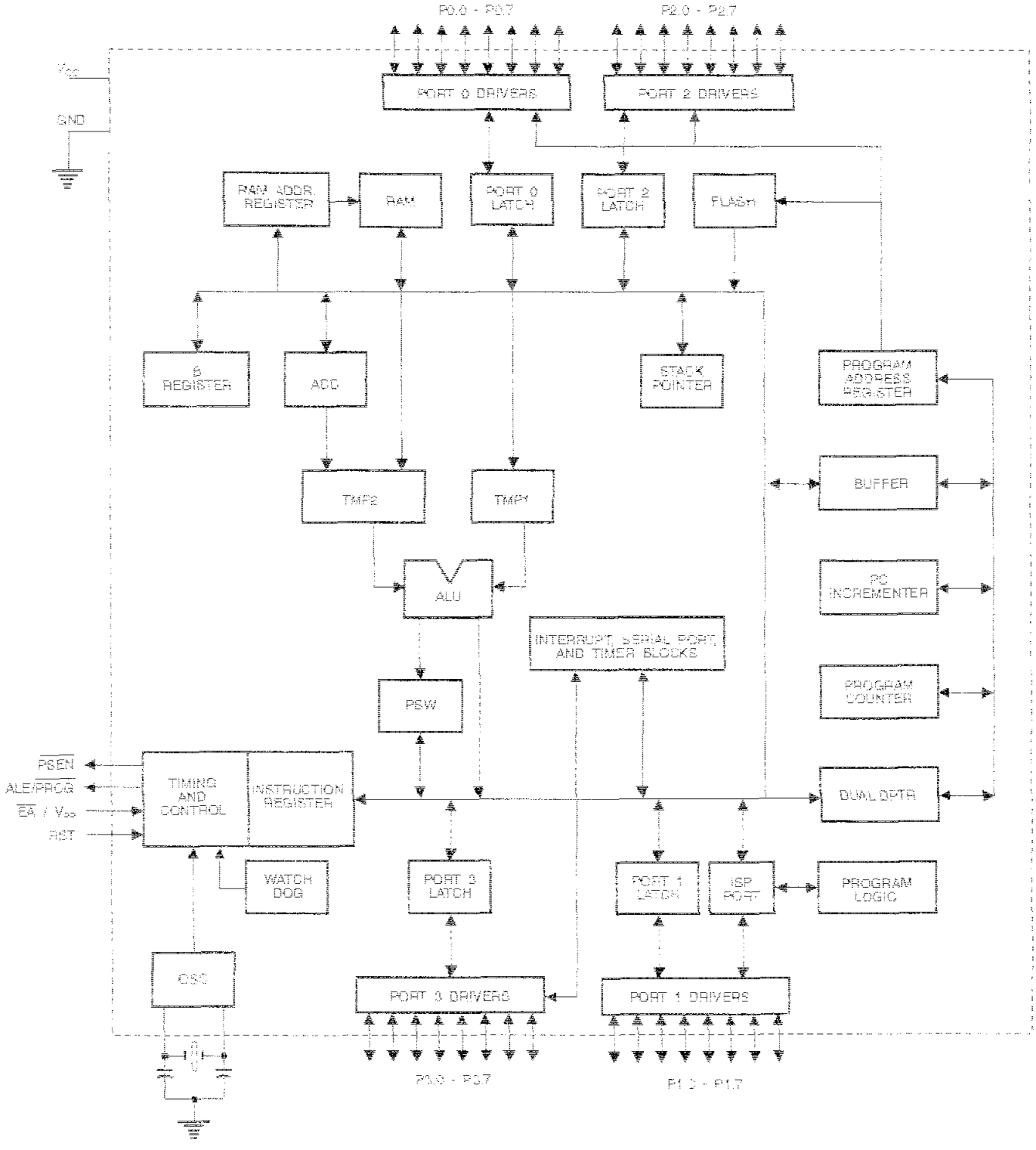
PLCC



TQFP



Block Diagram



Pin Description

VCC Supply voltage.

GND Ground.

Port 0 Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. **External pull-ups are required during program verification.**

Port 1 Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2 Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3 Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pull-ups.

Port 3 receives some control signals for Flash programming and verification.

Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/$\overline{PROG}$

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable ($\overline{PSEN}$) is the read strobe to external program memory.

When the AT89S51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

Table 1. AT89S51 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Interrupt Registers: The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the five interrupt sources in the IP register.

Table 2. AUXR: Auxiliary Register

AUXR Address = 8EH Reset Value = XXX00XX0B

Not Bit Addressable

Bit	7	6	5	4	3	2	1	0
	–	–	–	WDIDLE	DISRTO	–	–	DISALE

– Reserved for future expansion

DISALE Disable/Enable ALE

DISALE
Operating Mode

0 ALE is emitted at a constant rate of 1/6 the oscillator frequency

1 ALE is active only during a MOVX or MOVC instruction

DISRTO Disable/Enable Reset out

DISRTO

0 Reset pin is driven High after WDT times out

1 Reset pin is input only

WDIDLE Disable/Enable WDT in IDLE mode

WDIDLE

0 WDT continues to count in IDLE mode

1 WDT halts counting in IDLE mode

Dual Data Pointer Registers: To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

Power Off Flag: The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and rest under software control and is not affected by reset.

Table 3. AUXR1: Auxiliary Register 1

AUXR1

Address = A2H

Reset Value = XXXXXXX0B

Not Bit Addressable

	–	–	–	–	–	–	–	DPS
Bit	7	6	5	4	3	2	1	0

– Reserved for future expansion

DPS Data Pointer Register Select

DPS

0 Selects DPTR Registers DP0L, DP0H

1 Selects DPTR Registers DP1L, DP1H

Memory Organization

MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

Program Memory

If the $\overline{EA}$ pin is connected to GND, all program fetches are directed to external memory.

On the AT89S51, if $\overline{EA}$ is connected to V_{CC}, program fetches to addresses 0000H through FFFFH are directed to internal memory and fetches to addresses 1000H through FFFFH are directed to external memory.

Data Memory

The AT89S51 implements 128 bytes of on-chip RAM. The 128 bytes are accessible via direct and indirect addressing modes. Stack operations are examples of indirect addressing, so the 128 bytes of data RAM are available as stack space.

Watchdog Timer (One-time Enabled with Reset-out)

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 14-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is 98xTOSC, where TOSC=1/FOSC. To make the best use of the WDT, it

should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt, which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S51 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S51 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

UART

The UART in the AT89S51 operates the same way as the UART in the AT89C51. For further information on the UART operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select 'Products', then '8051 Architecture Flash Microcontroller', then 'Product Overview'.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S51 operate the same way as Timer 0 and Timer 1 in the AT89C51. For further information on the timers' operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Interrupts

The AT89S51 has a total of five interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), two timer interrupts (Timers 0 and 1), and the serial port interrupt. These interrupts are all shown in Figure 1.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 4 shows that bit position IE.6 is unimplemented. In the AT89S51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle.

Table 4. Interrupt Enable (IE) Register

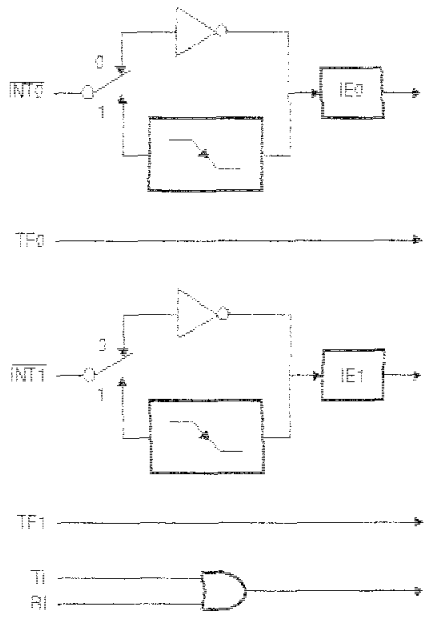
(MSB)							(LSB)
EA	–	–	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved
–	IE.5	Reserved
ES	IE.4	Serial Port interrupt enable bit
ET1	IE.3	Timer 1 interrupt enable bit
EX1	IE.2	External interrupt 1 enable bit
ET0	IE.1	Timer 0 interrupt enable bit
EX0	IE.0	External interrupt 0 enable bit
User software should never write 1s to reserved bits, because they may be used in future AT89 products.		

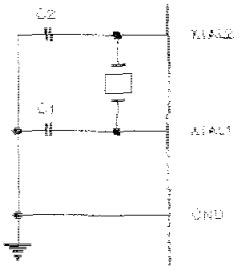
Figure 1. Interrupt Sources



Oscillator Characteristics

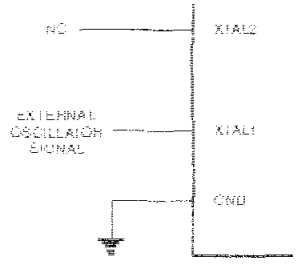
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 3. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 2. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals = 40 pF ± 10 pF for Ceramic Resonators

Figure 3. External Clock Drive Configuration



Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special function registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by activation of an enabled external interrupt into $\overline{INT0}$ or $\overline{INT1}$. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Table 5. Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

The AT89S51 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 6. Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash memory is disabled
3	P	P	U	Same as mode 2, but verify is also disabled
4	P	P	P	Same as mode 3, but external execution is also disabled

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

Programming the Flash – Parallel Mode

The AT89S51 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S51 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S51, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S51, take the following steps:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50 µs. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89S51 features Data Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.0 is pulled low after ALE goes high during programming to indicate $\overline{BUSY}$. P3.0 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(100H) = 51H indicates 89S51
(200H) = 06H

Chip Erase: In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/$\overline{PROG}$ low for a duration of 200 ns - 500 ns.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 500 m.s.

During chip erase, a serial read from any address location will return 00H at the data output.

Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to V_{CC}. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

Serial Programming Algorithm

To program and verify the AT89S51 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:

   Apply power between VCC and GND pins.

   Set RST pin to "H".

   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction that returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.

Power-off sequence (if needed):

Set XTAL1 to "L" (if a crystal is not used).

Set RST to "L".

Turn V_{CC} power off.

Data Polling: The Data Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 8 on page 18.

Programming Interface – Parallel Mode

Every code byte in the Flash array can be programmed by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Table 7. Flash Programming Modes

Mode	V_{CC}	RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.3-0 Address	P1.7-0 Address
Write Code Data	5V	H	L	(2)	12V	L	H	H	H	H	D_{IN}	A11-8	A7-0
Read Code Data	5V	H	L	H	H	L	L	L	H	H	D_{OUT}	A11-8	A7-0
Write Lock Bit 1	5V	H	L	(3)	12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L	(3)	12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L	(3)	12V	H	L	H	H	L	X	X	X
Read Lock Bits 1, 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
Chip Erase	5V	H	L	(1)	12V	H	L	H	L	L	X	X	X
Read Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	0000	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	51H	0001	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	06H	0010	00H

Notes:
1. Each PROG pulse is 200 ns - 500 ns for Chip Erase.
2. Each PROG pulse is 200 ns - 500 ns for Write Code Data.
3. Each PROG pulse is 200 ns - 500 ns for Write Lock Bits.
4. RDY/BSY signal is output on P3.0 during programming.
5. X = don't care.

Figure 4. Programming the Flash Memory (Parallel Mode)

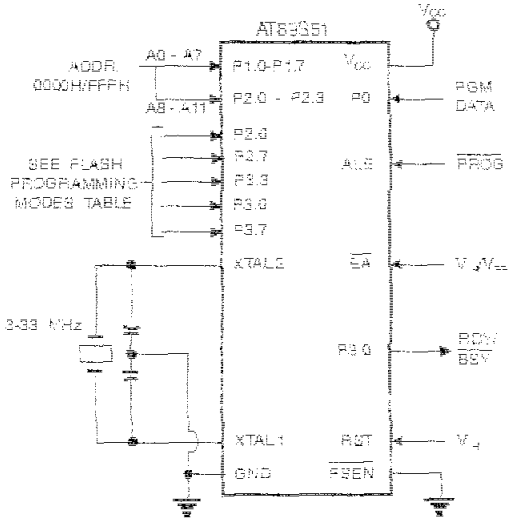
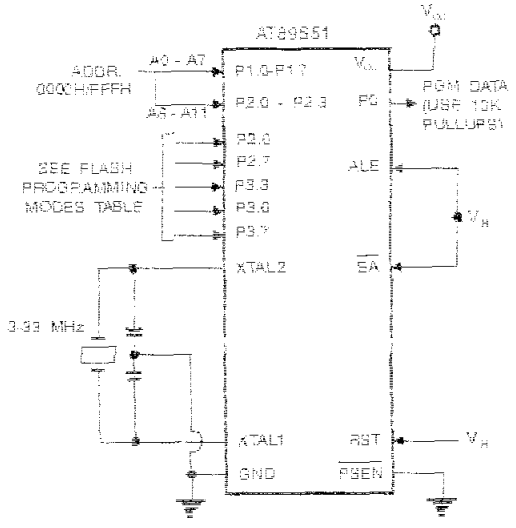


Figure 5. Verifying the Flash Memory (Parallel Mode)



Flash Programming and Verification Characteristics (Parallel Mode)

T_A = 20°C to 30°C, V_{CC} = 4.5 to 5.5V

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	11.5	12.5	V
I_{PP}	Programming Supply Current		10	mA
I_{CC}	V_{CC} Supply Current		30	mA
$1/t_{CLCL}$	Oscillator Frequency	3	33	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	0.2	1	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		50	µs

Figure 8. Flash Programming and Verification Waveforms – Parallel Mode

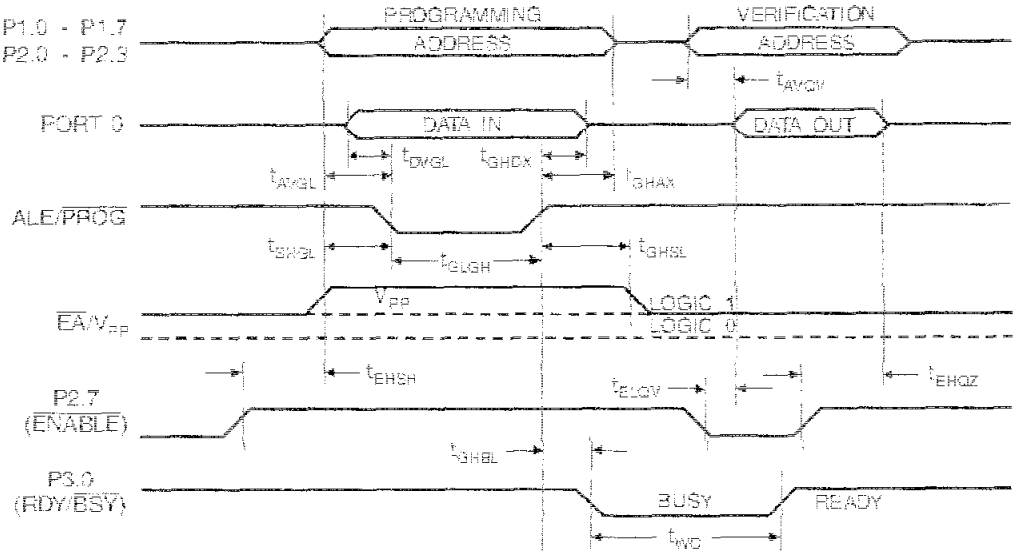
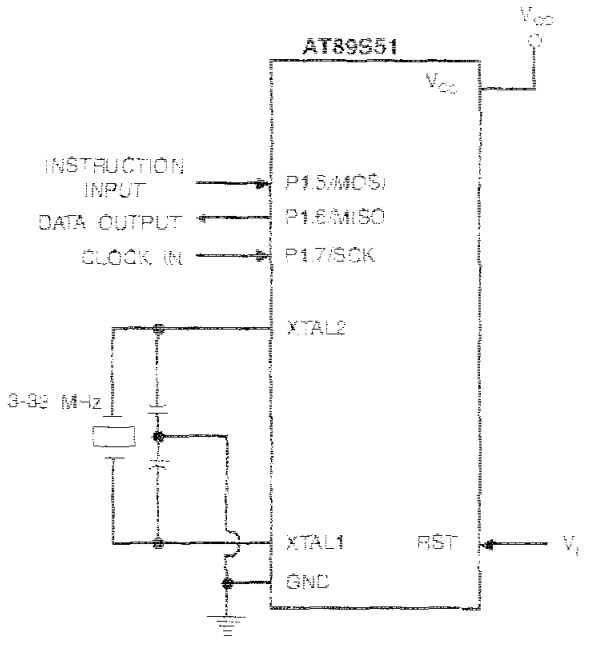


Figure 7. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 8. Serial Programming Waveforms

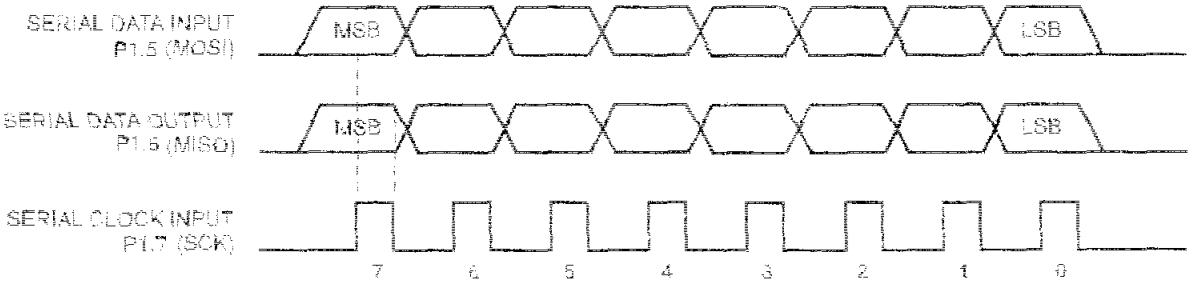


Table 8. Serial Programming Instruction Set

Instruction	Instruction Format Byte 1	Byte 2	Byte 3	Byte 4	Operation
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits[2]	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (2).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes[1]	0010 1000	xxx A5 A4 A3 A2 A1	A0 xxx xxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 *0000*	*xxxx* A11 A10 A9 A8	*Byte 0*	*Byte 1...* Byte 255	*Read data from Program memory in the Page Mode (256 bytes)*
Write Program Memory (Page Mode)	0101 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode *(256 bytes)*

Notes: 1. The signature bytes are not readable in Lock Bit Modes 3 and 4.

2. B1 = 0, B2 = 0 → Mode 1, no lock protection
   B1 = 0, B2 = 1 → Mode 2, lock bit 1 activated
   B1 = 1, B2 = 0 → Mode 3, lock bit 2 activated
   B1 = 1, B1 = 1 → Mode 4, lock bit 3 activated

   } Each of the lock bits needs to be activated sequentially before Mode 4 can be executed.

After Reset signal is high, SCK should be low for at least 64 system clocks before it goes high to clock in the enable data bytes. No pulsing of Reset signal is necessary. SCK should be no faster than 1/16 of the system clock at XTAL1.

For Page Read/Write, the data always starts from byte 0 to 255. After the command byte and upper address byte are latched, each byte thereafter is treated as data until all 256 bytes are shifted in/out. Then the next instruction will be ready to be decoded.

Serial Programming Characteristics

Figure 9. *Serial Programming Timing*

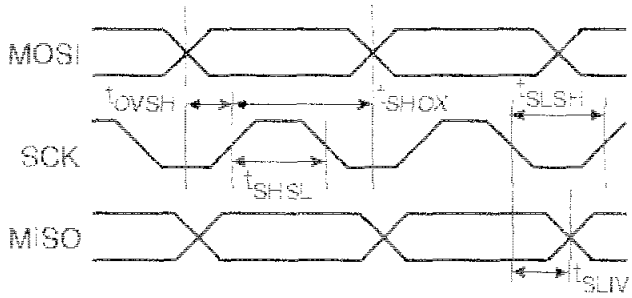


Table 9. Serial Programming Characteristics, T_A = -40°C to 85°C, V_{CC} = 4.0 - 5.5V (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0		33	MHz
t_{CLCL}	Oscillator Period	30			ns
t_{SHSL}	SCK Pulse Width High	8 t_{CLCL}			ns
t_{SLSH}	SCK Pulse Width Low	8 t_{CLCL}			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	2 t_{CLCL}			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t_{ERASE}	Chip Erase Instruction Cycle Time			500	ms
t_{SWC}	Serial Byte Write Cycle Time			64 t_{CLCL} + 400	µs

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.5V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 4.0V to 5.5V, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
R_{RST}	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode[2]	V_{CC} = 5.5V		50	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port:
   Port 0: 26 mA Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

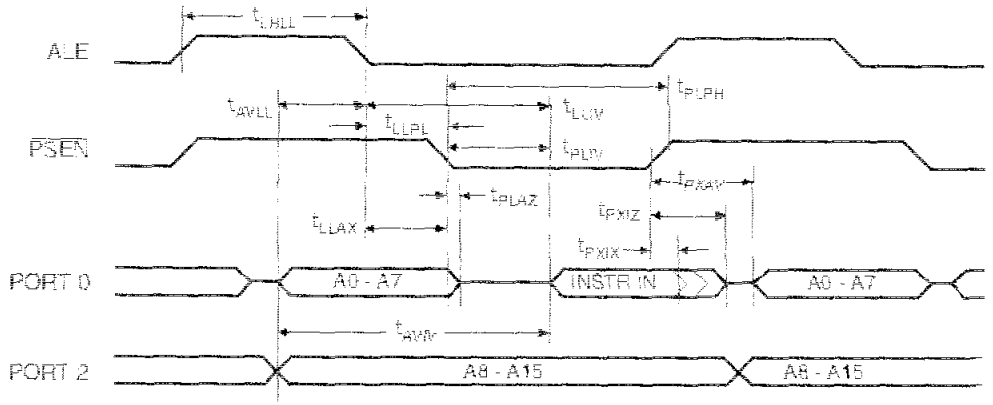
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

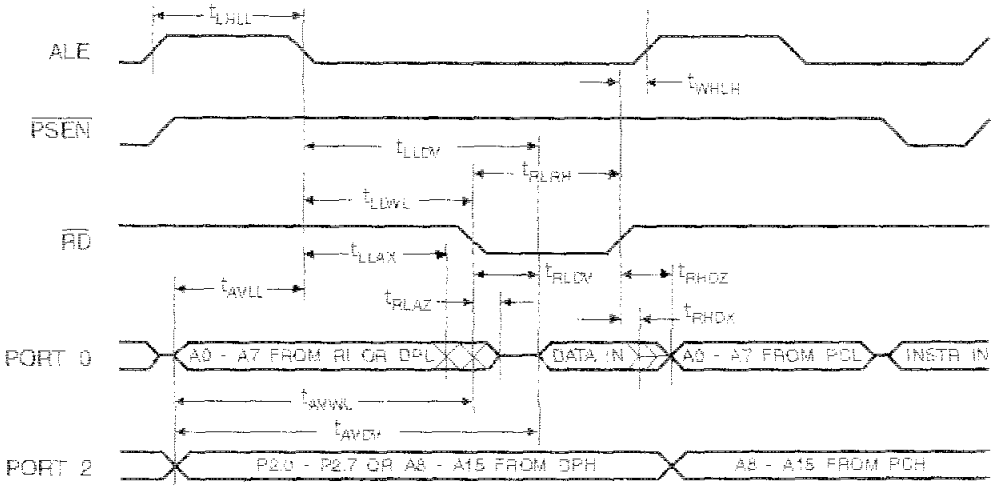
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	33	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-25		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-25		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-25		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-45		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-60	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-25	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-80	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-30		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-130		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-25		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-25	t_{CLCL}+25	ns

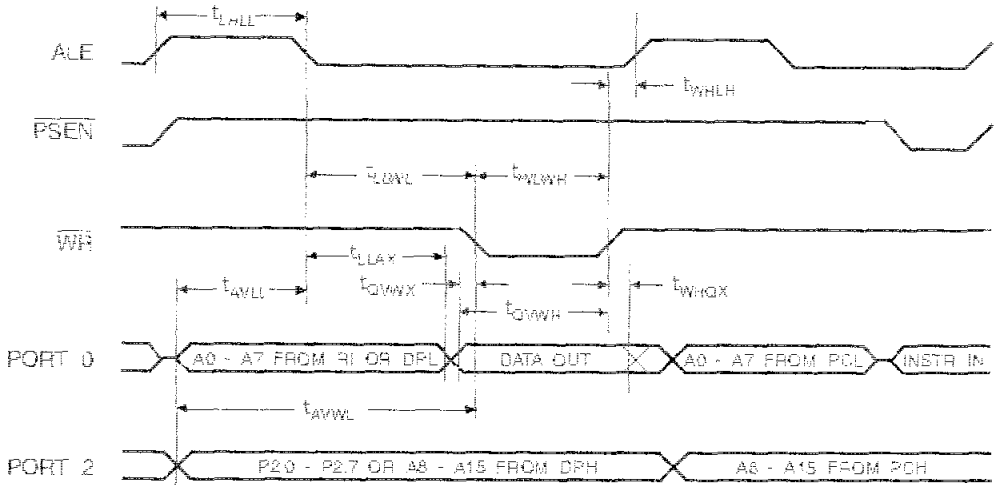
External Program Memory Read Cycle



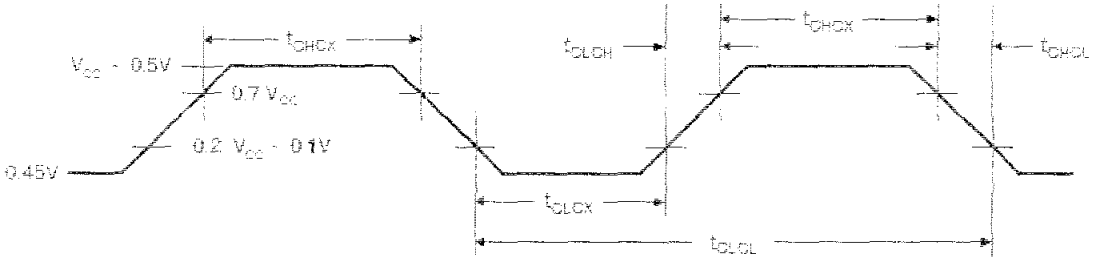
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

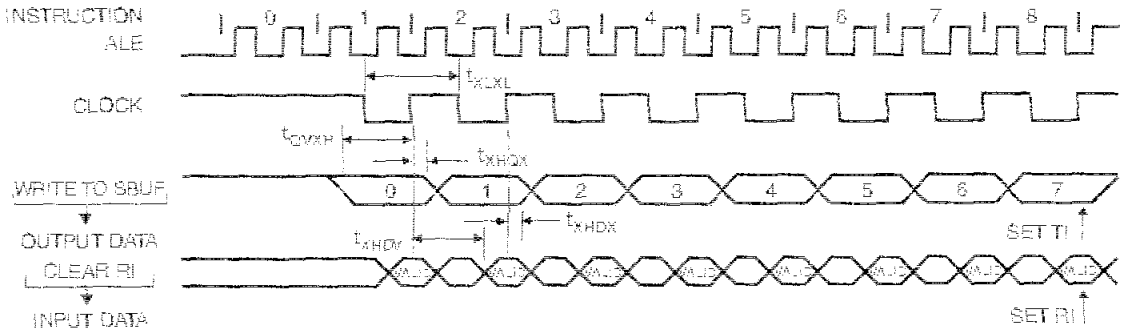
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	33	MHz
t_{CLCL}	Clock Period	30		ns
t_{CHCX}	High Time	12		ns
t_{CLCX}	Low Time	12		ns
t_{CLCH}	Rise Time		5	ns
t_{CHCL}	Fall Time		5	ns

Serial Port Timing: Shift Register Mode Test Conditions

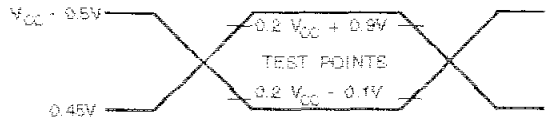
The values in this table are valid for V_{CC} = 4.0V to 5.5V and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-80		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms[1]



Note: 1. AC inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]

Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

Ordering Information

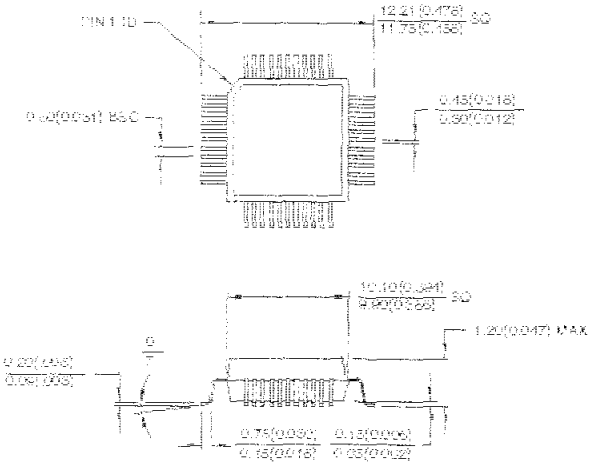
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S51-24AC AT89S51-24JC AT89S51-24PC	44A 44J 40P6	Commercial (0°C to 70°C)
		AT89S51-24AI AT89S51-24JI AT89S51-24PI	44A 44J 40P6	Industrial (-40°C to 85°C)
33	4.5V to 5.5V	AT89S51-33AC AT89S51-33JC AT89S51-33PC	44A 44J 40P6	Commercial (0°C to 70°C)

= Preliminary Availability

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)

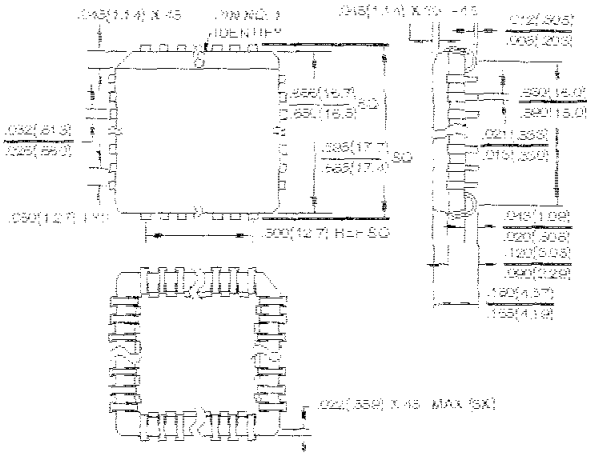
Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
Dimensions in Millimeters and (Inches)*

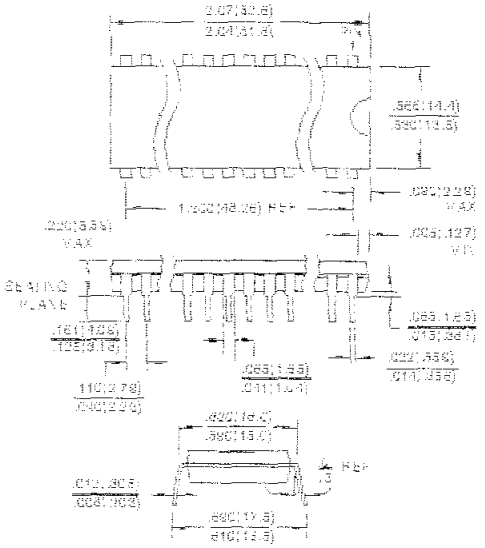


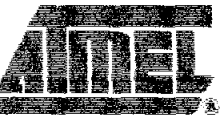
*Controlling dimension: millimeters

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)



40P6, 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-011 AC





Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Product Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Atmel Heilbronn
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

2487A 10/01/xM

Features

- Low-Voltage and Standard-Voltage Operation
  - 5.0 (V_{CC} = 4.5V to 5.5V)
  - 2.7 (V_{CC} = 2.7V to 5.5V)
  - 2.5 (V_{CC} = 2.5V to 5.5V)
  - 1.8 (V_{CC} = 1.8V to 5.5V)
- Internally Organized 128 x 8 (1K), 256 x 8 (2K), 512 x 8 (4K), 1024 x 8 (8K) or 2048 x 8 (16K)
- 2-Wire Serial Interface
- Schmitt Trigger, Filtered Inputs for Noise Suppression
- Bidirectional Data Transfer Protocol
- 100 kHz (1.8V, 2.5V, 2.7V) and 400 kHz (5V) Compatibility
- Write Protect Pin for Hardware Data Protection
- 8-Byte Page (1K, 2K), 16-Byte Page (4K, 8K, 16K) Write Modes
- Partial Page Writes Are Allowed
- Self-Timed Write Cycle (10 ms max)
- High Reliability
  - Endurance: 1 Million Write Cycles
  - Data Retention: 100 Years
  - ESD Protection: >3000V
- Automotive Grade and Extended Temperature Devices Available
- 8-Pin and 14-Pin JEDEC SOIC, 8-Pin PDIP, 8-Pin MSOP, and 8-Pin TSSOP Packages

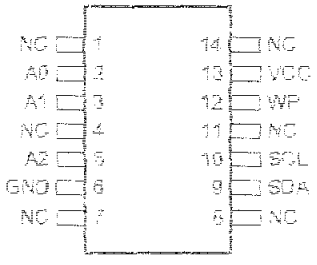
Description

The AT24C01A/02/04/08/16 provides 1024/2048/4096/8192/16384 bits of serial electrically erasable and programmable read only memory (EEPROM) organized as 128/256/512/1024/2048 words of 8 bits each. The device is optimized for use in many industrial and commercial applications where low power and low voltage operation are essential. The AT24C01A/02/04/08/16 is available in space saving 8-pin PDIP, (AT24C01A/02/04/08/16), 8-Pin MSOP (AT24C01A/02), 8-Pin TSSOP (AT24C01A/02/04/08/16), and 8-Pin and 14-Pin JEDEC SOIC (AT24C01A/02/04/08/16) packages and is accessed via a 2-wire serial interface. In addition, the entire family is available in 5.0V (4.5V to 5.5V), 2.7V (2.7V to 5.5V), 2.5V (2.5V to 5.5V) and 1.8V (1.8V to 5.5V) versions.

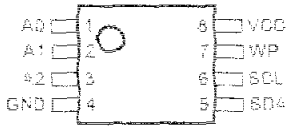
Pin Configurations

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect
NC	No Connect

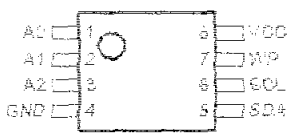
14-Pin SOIC



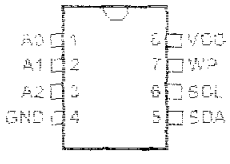
8-Pin TSSOP



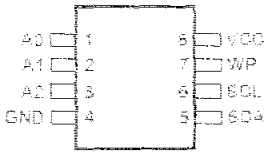
8-Pin MSOP



8-Pin PDIP



8-Pin SOIC



2-Wire Serial EEPROM

1K (128 x 8)

2K (256 x 8)

4K (512 x 8)

8K (1024 x 8)

16K (2048 x 8)

AT24C01A
AT24C02
AT24C04
AT24C08
AT24C16

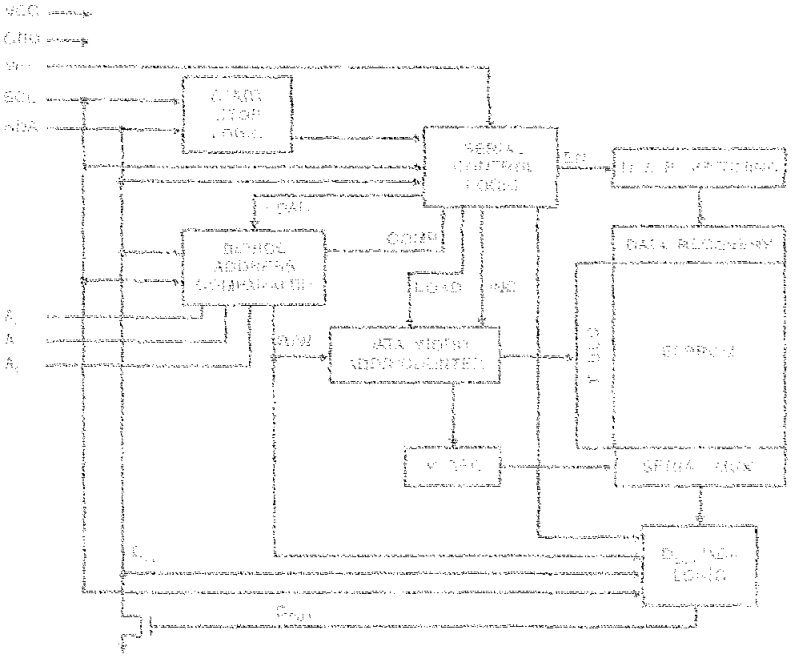
Rev. 0150D–10/98

Absolute Maximum Ratings

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.25V
DC Output Current	5.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Block Diagram



Pin Description

SERIAL CLOCK (SCL): The SCL input is used to positive edge clock data into each EEPROM device and negative edge clock data out of each device.

SERIAL DATA (SDA): The SDA pin is bidirectional for serial data transfer. This pin is open-drain driven and may be wire-ORed with any number of other open-drain or open collector devices.

DEVICE/PAGE ADDRESSES (A2, A1, A0): The A2, A1 and A0 pins are device address inputs that are hard wired for the AT24C01A and the AT24C02. As many as eight 1K/2K devices may be addressed on a single bus system (device addressing is discussed in detail under the Device Addressing section).

The AT24C04 uses the A2 and A1 inputs for hard wire addressing and a total of four 4K devices may be addressed on a single bus system. The A0 pin is a no connect.

The AT24C08 only uses the A2 input for hardwire addressing and a total of two 8K devices may be addressed on a single bus system. The A0 and A1 pins are no connects.

The AT24C16 does not use the device address pins which limits the number of devices on a single bus to one. The A0, A1 and A2 pins are no connects.

WRITE PROTECT (WP): The AT24C01A/02/04/16 has a Write Protect pin that provides hardware data protection. The Write Protect pin allows normal read/write operations when connected to ground (GND). When the Write Protect pin is connected to V_{CC}, the write protection feature is enabled and operates as shown in the following table.

WP Pin Status	Part of the Array Protected				
	24C01A	24C02	24C04	24C08	24C16
At V_{CC}	Full (1K) Array	Full (2K) Array	Full (4K) Array	Normal Read/ Write Operation	Upper Half (8K) Array
At GND	Normal Read/Write Operations				

Memory Organization

AT24C01A, 1K SERIAL EEPROM: Internally organized with 128 pages of 1-byte each, the 1K requires a 7-bit data word address for random word addressing.

AT24C02, 2K SERIAL EEPROM: Internally organized with 256 pages of 1-byte each, the 2K requires an 8-bit data word address for random word addressing.

AT24C04, 4K SERIAL EEPROM: The 4K is internally organized with 256 pages of 2 bytes each. Random word addressing Chip Number requires a 9-bit data word address.

AT24C08, 8K SERIAL EEPROM: The 8K is internally organized with 4 blocks of 256 pages of 4 bytes each. Random word addressing requires a 10-bit data word address.

AT24C16, 16K SERIAL EEPROM: The 16K is internally organized with 8 blocks of 256 pages of 8 bytes each. Random word addressing requires an 11-bit data word address.

Pin Capacitance[1]

Applicable over recommended operating range from T_A = 25°C, f = 1.0 MHz, V_{CC} = +1.8V.

Symbol	Test Condition	Max	Units	Conditions
$C_{I/O}$	Input/Output Capacitance (SDA)	8	pF	$V_{I/O}$ = 0V
C_{IN}	Input Capacitance (A_0, A_1, A_2, SCL)	6	pF	V_{IN} = 0V

Note: 1. This parameter is characterized and is not 100% tested.

DC Characteristics

Applicable over recommended operating range from: T_{AI} = -40°C to +85°C, V_{CC} = +1.8V to +5.5V, T_{AC} = 0°C to +70°C, V_{CC} = +1.8V to +5.5V (unless otherwise noted).

Symbol	Parameter	Test Condition	Min	Typ	Max	Units
V_{CC1}	Supply Voltage		1.8		5.5	V
V_{CC2}	Supply Voltage		2.5		5.5	V
V_{CC3}	Supply Voltage		2.7		5.5	V
V_{CC4}	Supply Voltage		4.5		5.5	V
I_{CC}	Supply Current V_{CC} = 5.0V	READ at 100 kHz		0.4	1.0	mA
I_{CC}	Supply Current V_{CC} = 5.0V	WRITE at 100 kHz		2.0	3.0	mA
I_{SB1}	Standby Current V_{CC} = 1.8V	V_{IN} = V_{CC} or V_{SS}		0.6	3.0	µA
I_{SB2}	Standby Current V_{CC} = 2.5V	V_{IN} = V_{CC} or V_{SS}		1.4	4.0	µA
I_{SB3}	Standby Current V_{CC} = 2.7V	V_{IN} = V_{CC} or V_{SS}		1.6	4.0	µA
I_{SB4}	Standby Current V_{CC} = 5.0V	V_{IN} = V_{CC} or V_{SS}		8.0	18.0	µA
I_{LI}	Input Leakage Current	V_{IN} = V_{CC} or V_{SS}		0.10	3.0	µA
I_{LO}	Output Leakage Current	V_{OUT} = V_{CC} or V_{SS}		0.05	3.0	µA
V_{IL}	Input Low Level[1]		-0.6		V_{CC} x 0.3	V
V_{IH}	Input High Level[1]		V_{CC} x 0.7		V_{CC} + 0.5	V
V_{OL2}	Output Low Level V_{CC} = 3.0V	I_{OL} = 2.1 mA			0.4	V
V_{OL1}	Output Low Level V_{CC} = 1.8V	I_{OL} = 0.15 mA			0.2	V

Note: 1. V_{IL} min and V_{IH} max are reference only and are not tested.

AC Characteristics

Applicable over recommended operating range from T_A = -40°C to +85°C, V_{CC} = +1.8V to +5.5V, CL = 1 TTL Gate and 100pF (unless otherwise noted).

Symbol	Parameter	2.7-, 2.5-, 1.8-volt		5.0-volt		Units
		Min	Max	Min	Max	
f_{SCL}	Clock Frequency, SCL		100		400	kHz
t_{LOW}	Clock Pulse Width Low	4.7		1.2		µs
t_{HIGH}	Clock Pulse Width High	4.0		0.6		µs
t_I	Noise Suppression Time[1]		100		50	ns
t_{AA}	Clock Low to Data Out Valid	0.1	4.5	0.1	0.9	µs
t_{BUF}	Time the bus must be free before a new transmission can start[1]	4.7		1.2		µs
$t_{HD.STA}$	Start Hold Time	4.0		0.6		µs
$t_{SU.STA}$	Start Set-up Time	4.7		0.6		µs
$t_{HD.DAT}$	Data In Hold Time	0		0		µs
$t_{SU.DAT}$	Data In Set-up Time	200		100		ns
t_R	Inputs Rise Time[1]		1.0		0.3	µs
t_F	Inputs Fall Time[1]		300		300	ns
$t_{SU.STO}$	Stop Set-up Time	4.7		0.6		µs
t_{DH}	Data Out Hold Time	100		50		ns
t_{WR}	Write Cycle Time		10		10	ms
Endurance[1]	5.0V, 25°C, Page Mode	1M		1M		Write Cycles

Note: 1. This parameter is characterized and is not 100% tested.

Device Operation

CLOCK and DATA TRANSITIONS: The SDA pin is normally pulled high with an external device. Data on the SDA pin may change only during SCL low time periods (refer to Data Validity timing diagram). Data changes during SCL high periods will indicate a start or stop condition as defined below.

START CONDITION: A high-to-low transition of SDA with SCL high is a start condition which must precede any other command (refer to Start and Stop Definition timing diagram).

STOP CONDITION: A low-to-high transition of SDA with SCL high is a stop condition. After a read sequence, the stop command will place the EEPROM in a standby power mode (refer to Start and Stop Definition timing diagram).

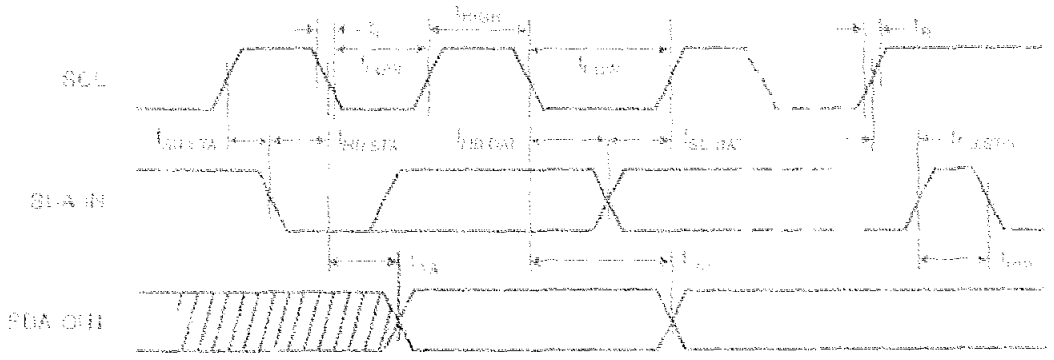
ACKNOWLEDGE: All addresses and data words are serially transmitted to and from the EEPROM in 8-bit words. The EEPROM sends a zero to acknowledge that it has received each word. This happens during the ninth clock cycle.

STANDBY MODE: The AT24C01A/02/04/08/16 features a low power standby mode which is enabled: (a) upon power-up and (b) after the receipt of the STOP bit and the completion of any internal operations.

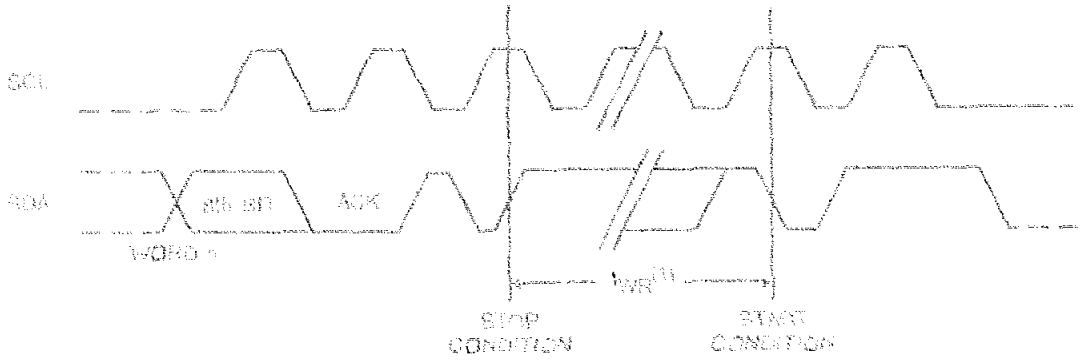
MEMORY RESET: After an interruption in protocol, power loss or system reset, any 2-wire part can be reset by following these steps:

1. Clock up to 9 cycles.
2. Look for SDA high in each cycle while SCL is high.
3. Create a start condition as SDA is high.

us Timing
CL: Serial Clock, SDA: Serial Data I/O

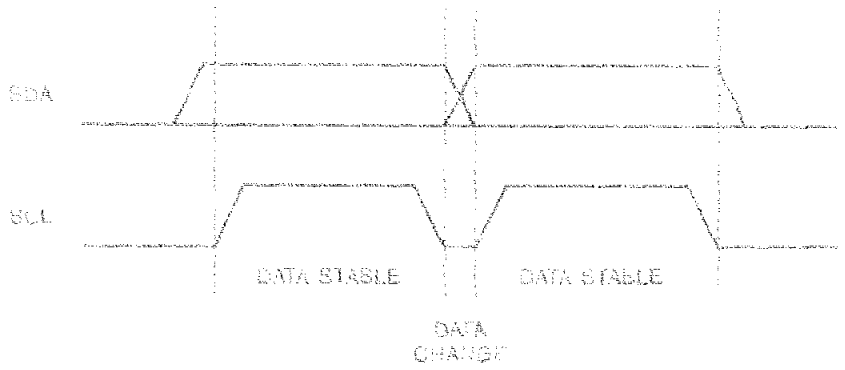


Write Cycle Timing
SCL: Serial Clock, SDA: Serial Data I/O

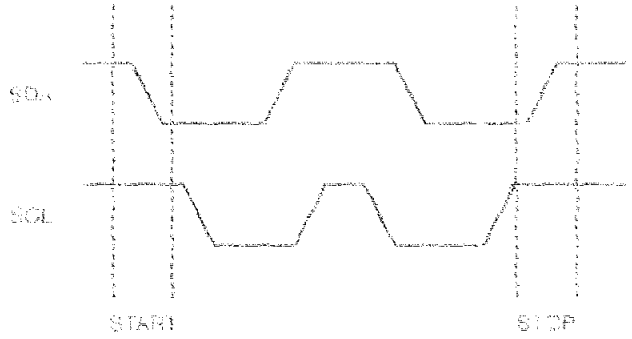


ote: 1. The write cycle time t_{WR} is the time from a valid stop condition of a write sequence to the end of the internal clear/write cycle.

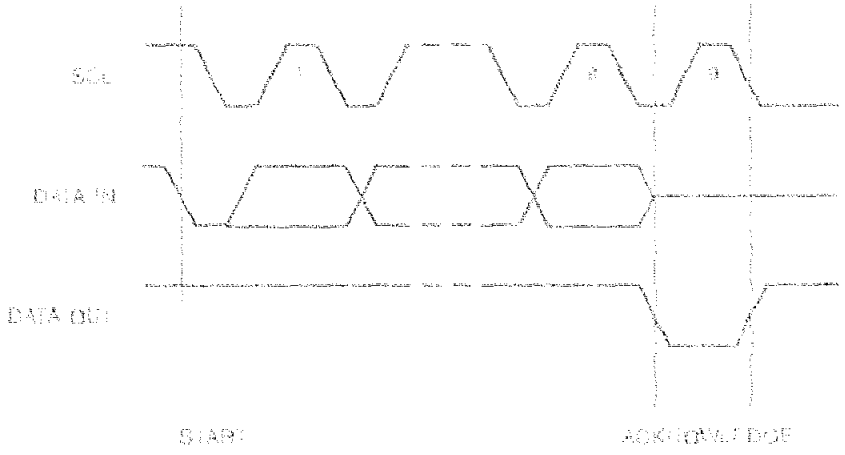
Data Validity



Start and Stop Definition



Output Acknowledge



Device Addressing

The 1K, 2K, 4K, 8K and 16K EEPROM devices all require an 8-bit device address word following a start condition to enable the chip for a read or write operation (refer to Figure 5).

The device address word consists of a mandatory ones, zero sequence for the first four most significant bits as shown. This is common to all the EEPROM devices.

The next 3 bits are the A2, A1 and A0 device address bits for the 1K/2K EEPROM. These 3 bits must compare to their corresponding hard-wired input pins.

The 4K EEPROM only uses the A2 and A1 device address bits with the third bit being a memory page address bit. The two device address bits must compare to their corresponding hard-wired input pins. The A0 pin is no connect.

The 8K EEPROM only uses the A2 device address bit with the next 2 bits being for memory page addressing. The A2 bit must compare to its corresponding hard-wired input pin. The A1 and A0 pins are no connect.

The 16K does not use any device address bits but instead the 3 bits are used for memory page addressing. These page addressing bits on the 4K, 8K, and 16K devices should be considered the most significant bits of the data word address which follows. The A0, A1 and A2 pins are no connect.

The eighth bit of the device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.

Upon a compare of the device address, the EEPROM will output a zero. If a compare is not made, the chip will return to a standby state.

Write Operations

BYTE WRITE: A write operation requires an 8-bit data word address following the device address word and acknowledgment. Upon receipt of this address, the EEPROM will again respond with a zero and then clock in the first 8-bit data word. Following receipt of the 8-bit data word, the EEPROM will output a zero and the addressing device, such as a microcontroller, must terminate the write sequence with a stop condition. At this time the EEPROM enters an internally-timed write cycle, t_{WR}, to the nonvolatile memory. All inputs are disabled during this write cycle and the EEPROM will not respond until the write is complete (refer to Figure 2).

PAGE WRITE: The 1K/2K EEPROM is capable of an 8-byte page write, and the 4K, 8K and 16K devices are capable of 16-byte page writes.

A page write is initiated the same as a byte write, but the microcontroller does not send a stop condition after the first data word is clocked in. Instead, after the EEPROM acknowledges receipt of the first data word, the microcontroller can transmit up to seven (1K/2K) or fifteen (4K, 8K, 16K) more data words. The EEPROM will respond with a zero after each data word received. The microcontroller must terminate the page write sequence with a stop condition (refer to Figure 3).

The data word address lower three (1K/2K) or four (4K, 8K, 16K) bits are internally incremented following the receipt of each data word. The higher data word address bits are not incremented, retaining the memory page row location. When the word address, internally generated, reaches the page boundary, the following byte is placed at the beginning of the same page. If more than eight (1K/2K) or sixteen (4K, 8K, 16K) data words are transmitted to the EEPROM, the data word address will "roll over" and previous data will be overwritten.

ACKNOWLEDGE POLLING: Once the internally-timed write cycle has started and the EEPROM inputs are disabled, acknowledge polling can be initiated. This involves sending a start condition followed by the device address word. The read/write bit is representative of the operation desired. Only if the internal write cycle has completed will the EEPROM respond with a zero allowing the read or write sequence to continue.

Read Operations

Read operations are initiated the same way as write operations with the exception that the read/write select bit in the device address word is set to one. There are three read operations: current address read, random address read and sequential read.

CURRENT ADDRESS READ: The internal data word address counter maintains the last address accessed during the last read or write operation, incremented by one. This address stays valid between operations as long as the chip power is maintained. The address "roll over" during read is from the last byte of the last memory page to the first byte of the first page. The address "roll over" during write is from the last byte of the current page to the first byte of the same page.

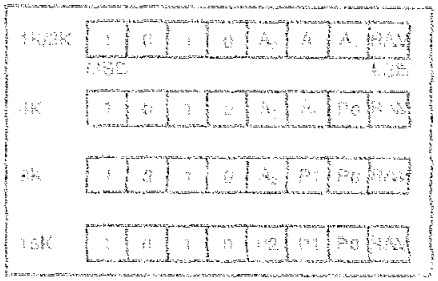
Once the device address with the read/write select bit set to one is clocked in and acknowledged by the EEPROM, the current address data word is serially clocked out. The microcontroller does not respond with an input zero but does generate a following stop condition (refer to Figure 4).

RANDOM READ: A random read requires a "dummy" byte write sequence to load in the data word address. Once the device address word and data word address are clocked in and acknowledged by the EEPROM, the microcontroller must generate another start condition. The microcontroller now initiates a current address read by sending a device address with the read/write select bit high. The EEPROM acknowledges the device address and serially clocks out

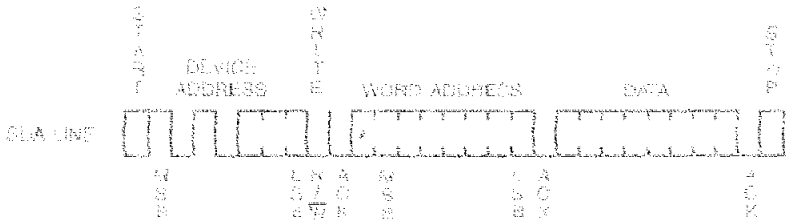
he data word. The microcontroller does not respond with a ero but does generate a following stop condition (refer to igure 5).

EQUENTIAL READ: Sequential reads are initiated by *ither a current address read or a random address read.* fter the microcontroller receives a data word, it responds ith an acknowledge. As long as the EEPROM receives an acknowledge, it will continue to increment the data word address and serially clock out sequential data words. When the memory address limit is reached, the data word address will "roll over" and the sequential read will continue. The sequential read operation is terminated when the microcontroller does not respond with a zero but does generate a following stop condition (refer to Figure 6).

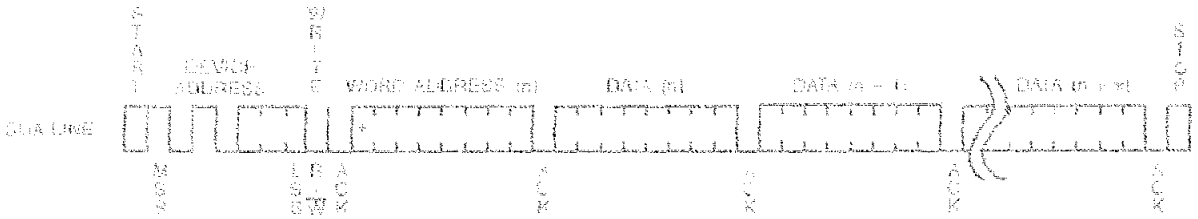
igure 1. Device Address



igure 2. Byte Write

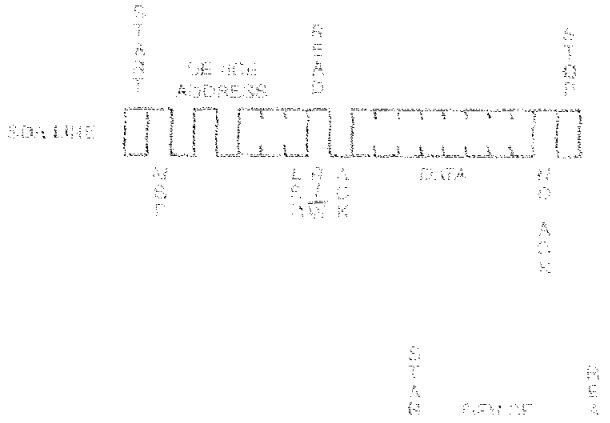


igure 3. Page Write

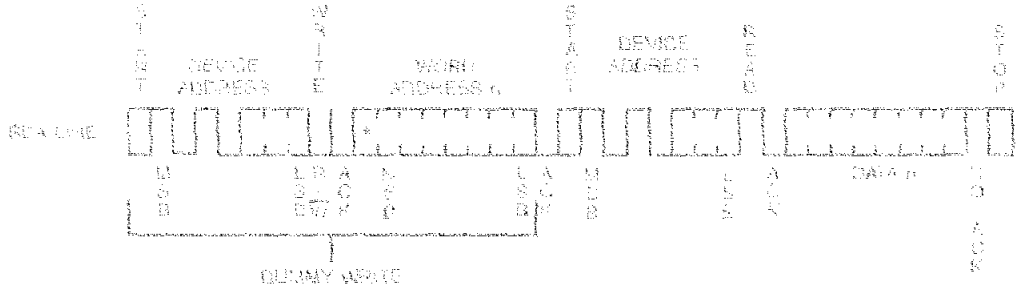


(* = DON'T CARE bit for 1K)

igure 4. Current Address Read

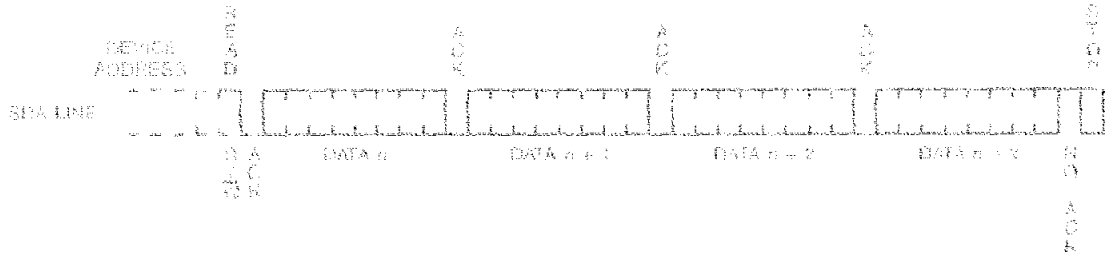


igure 5. Random Read



(* = DON'T CARE bit for 1K)

igure 6. Sequential Read



T24C01A Ordering Information

t_{WR} (max) (ms)	I_{CC} (max) (µA)	I_{SB} (max) (µA)	f_{MAX} (kHz)	Ordering Code	Package	Operation Range
10	3000	18	400	AT24C01A-10PC	8P3	Commercial (0°C to 70°C)
				AT24C01A-10SC	8S1	
				AT24C01A-10MC	8M	
				AT24C01A-10TC	8T	
	3000	18	400	AT24C01A-10PI	8P3	Industrial (-40°C to 85°C)
				AT24C01A-10SI	8S1	
				AT24C01A-10MI	8M	
				AT24C01A-10TI	8T	
10	1500	4	100	AT24C01A-10PC-2.7	8P3	Commercial (0°C to 70°C)
				AT24C01A-10SC-2.7	8S1	
				AT24C01A-10MC-2.7	8M	
				AT24C01A-10TC-2.7	8T	
	1500	4	100	AT24C01A-10PI-2.7	8P3	Industrial (-40°C to 85°C)
				AT24C01A-10SI-2.7	8S1	
				AT24C01A-10MI-2.7	8M	
				AT24C01A-10TI-2.7	8T	

Package Type	
8M	8-Lead, 0.118" Wide, Miniature Small Outline Package (MSOP)
8P3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8T	8-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
Options	
Blank	Standard Operation (4.5V to 5.5V)
-2.7	Low Voltage (2.7V to 5.5V)
-2.5	Low Voltage (2.5V to 5.5V)
-1.8	Low Voltage (1.8V to 5.5V)

AT24C01A Ordering Information (Continued)

t_{WR} (max) (ms)	I_{CC} (max) (µA)	I_{SB} (max) (µA)	f_{MAX} (kHz)	Ordering Code	Package	Operation Range
10	1000	4	100	AT24C01A-10PC-2.5 AT24C01A-10SC-2.5 AT24C01A-10MC-2.5 AT24C01A-10TC-2.5	8P3 8S1 8M 8T	Commercial (0°C to 70°C)
	1000	4	100	AT24C01A-10PI-2.5 AT24C01A-10SI-2.5 AT24C01A-10MI-2.5 AT24C01A-10TI-2.5	8P3 8S1 8M 8T	Industrial (-40°C to 85°C)
10	800	3	100	AT24C01A-10PC-1.8 AT24C01A-10SC-1.8 AT24C01A-10MC-1.8 AT24C01A-10TC-1.8	8P3 8S1 8M 8T	Commercial (0°C to 70°C)
	800	3	100	AT24C01A-10PI-1.8 AT24C01A-10SI-1.8 AT24C01A-10MI-1.8 AT24C01A-10TI-1.8	8P3 8S1 8M 8T	Industrial (-40°C to 85°C)

Package Type	
8M	8-Lead, 0.118" Wide, Miniature Small Outline Package (MSOP)
8P3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8T	8-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
Options	
Blank	Standard Operation (4.5V to 5.5V)
2.7	Low Voltage (2.7V to 5.5V)
2.5	Low Voltage (2.5V to 5.5V)
1.8	Low Voltage (1.8V to 5.5V)

AT24C02 Ordering Information

t_{WR} (max) (ms)	I_{CC} (max) (µA)	I_{SB} (max) (µA)	f_{MAX} (kHz)	Ordering Code	Package	Operation Range
10	3000	18	400	AT24C02-10PC AT24C02N-10SC AT24C02-10SC AT24C02-10MC AT24C02-10TC	8P3 8S1 14S 8M 8T	Commercial (0°C to 70°C)
	3000	18	400	AT24C02-10PI AT24C02N-10SI AT24C02-10SI AT24C02-10MI AT24C02-10TI	8P3 8S1 14S 8M 8T	Industrial (-40°C to 85°C)
10	1500	4	100	AT24C02-10PC-2.7 AT24C02N-10SC-2.7 AT24C02-10SC-2.7 AT24C02-10MC-2.7 AT24C02-10TC-2.7	8P3 8S1 14S 8M 8T	Commercial (0°C to 70°C)
	1500	4	100	AT24C02-10PI-2.7 AT24C02N-10SI-2.7 AT24C02-10SI-2.7 AT24C02-10MI-2.7 AT24C02-10TI-2.7	8P3 8S1 14S 8M 8T	Industrial (-40°C to 85°C)

	Package Type
8M	8-Lead, 0.118" Wide, Miniature Small Outline Package (MSOP)
8P3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8T	8-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
14S	14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
	Options
Blank	Standard Operation (4.5V to 5.5V)
-2.7	Low Voltage (2.7V to 5.5V)
-2.5	Low Voltage (2.5V to 5.5V)
-1.8	Low Voltage (1.8V to 5.5V)

AT24C02 Ordering Information (Continued)

t_{WR} (max) (ms)	I_{CC} (max) (µA)	I_{SS} (max) (µA)	f_{MAX} (kHz)	Ordering Code	Package	Operation Range
10	1000	4	100	AT24C02-10PC-2.5 AT24C02N-10SC-2.5 AT24C02-10SC-2.5 AT24C02-10MC-2.5 AT24C02-10TC-2.5	8P3 8S1 14S 8M 8T	Commercial (0°C to 70°C)
	1000	4	100	AT24C02-10PI-2.5 AT24C02N-10SI-2.5 AT24C02-10SI-2.5 AT24C02-10MI-2.5 AT24C02-10TI-2.5	8P3 8S1 14S 8M 8T	Industrial (-40°C to 85°C)
10	800	3	100	AT24C02-10PC-1.8 AT24C02N-10SC-1.8 AT24C02-10SC-1.8 AT24C02-10MC-1.8 AT24C02-10TC-1.8	8P3 8S1 14S 8M 8T	Commercial (0°C to 70°C)
	800	3	100	AT24C02-10PI-1.8 AT24C02N-10SI-1.8 AT24C02-10SI-1.8 AT24C02-10MI-1.8 AT24C02-10TI-1.8	8P3 8S1 14S 8M 8T	Industrial (-40°C to 85°C)

Package Type	
8M	8-Lead, 0.118" Wide, Miniature Small Outline Package (MSOP)
8P3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8T	8-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
14S	14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
Options	
Blank	Standard Operation (4.5V to 5.5V)
-2.7	Low Voltage (2.7V to 5.5V)
-2.5	Low Voltage (2.5V to 5.5V)
-1.8	Low Voltage (1.8V to 5.5V)

AT24C04 Ordering Information

t_{WR} (max) (ms)	I_{CC} (max) (µA)	I_{SB} (max) (µA)	f_{MAX} (kHz)	Ordering Code	Package	Operation Range
10	3000	18	400	AT24C04-10PC AT24C04N-10SC AT24C04-10SC AT24C04-10TC	8P3 8S1 14S 8T	Commercial (0°C to 70°C)
	3000	18	400	AT24C04-10PI AT24C04N-10SI AT24C04-10SI AT24C04-10TI	8P3 8S1 14S 8T	Industrial (-40°C to 85°C)
10	1500	4	100	AT24C04-10PC-2.7 AT24C04N-10SC-2.7 AT24C04-10SC-2.7 AT24C04-10TC-2.7	8P3 8S1 14S 8T	Commercial (0°C to 70°C)
	1500	4	100	AT24C04-10PI-2.7 AT24C04N-10SI-2.7 AT24C04-10SI-2.7 AT24C04-10TI-2.7	8P3 8S1 14S 8T	Industrial (-40°C to 85°C)

Package Type	
8P3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8T	8-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
14S	14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
Options	
Blank	Standard Operation (4.5V to 5.5V)
-2.7	Low Voltage (2.7V to 5.5V)
-2.5	Low Voltage (2.5V to 5.5V)
-1.8	Low Voltage (1.8V to 5.5V)

AT24C04 Ordering Information (Continued)

t_{WR} (max) (ms)	I_{CC} (max) (µA)	I_{SB} (max) (µA)	f_{MAX} (kHz)	Ordering Code	Package	Operation Range
10	1000	4	100	AT24C04-10PC-2.5	8P3	Commercial (0°C to 70°C)
				AT24C04N-10SC-2.5	8S1	
				AT24C04-10SC-2.5	14S	
				AT24C04-10TC-2.5	8T	
	1000	4	100	AT24C04-10PI-2.5	8P3	Industrial (-40°C to 85°C)
				AT24C04N-10SI-2.5	8S1	
				AT24C04-10SI-2.5	14S	
				AT24C04-10TI-2.5	8T	
10	800	3	100	AT24C04-10PC-1.8	8P3	Commercial (0°C to 70°C)
				AT24C04N-10SC-1.8	8S1	
				AT24C04-10SC-1.8	14S	
				AT24C04-10TC-1.8	8T	
	800	3	100	AT24C04-10PI-1.8	8P3	Industrial (-40°C to 85°C)
				AT24C04N-10SI-1.8	8S1	
				AT24C04-10SI-1.8	14S	
				AT24C04-10TI-1.8	8T	

Package Type	
8P3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8T	8-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
14S	14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
Options	
Blank	Standard Operation (4.5V to 5.5V)
-2.7	Low Voltage (2.7V to 5.5V)
-2.5	Low Voltage (2.5V to 5.5V)
-1.8	Low Voltage (1.8V to 5.5V)