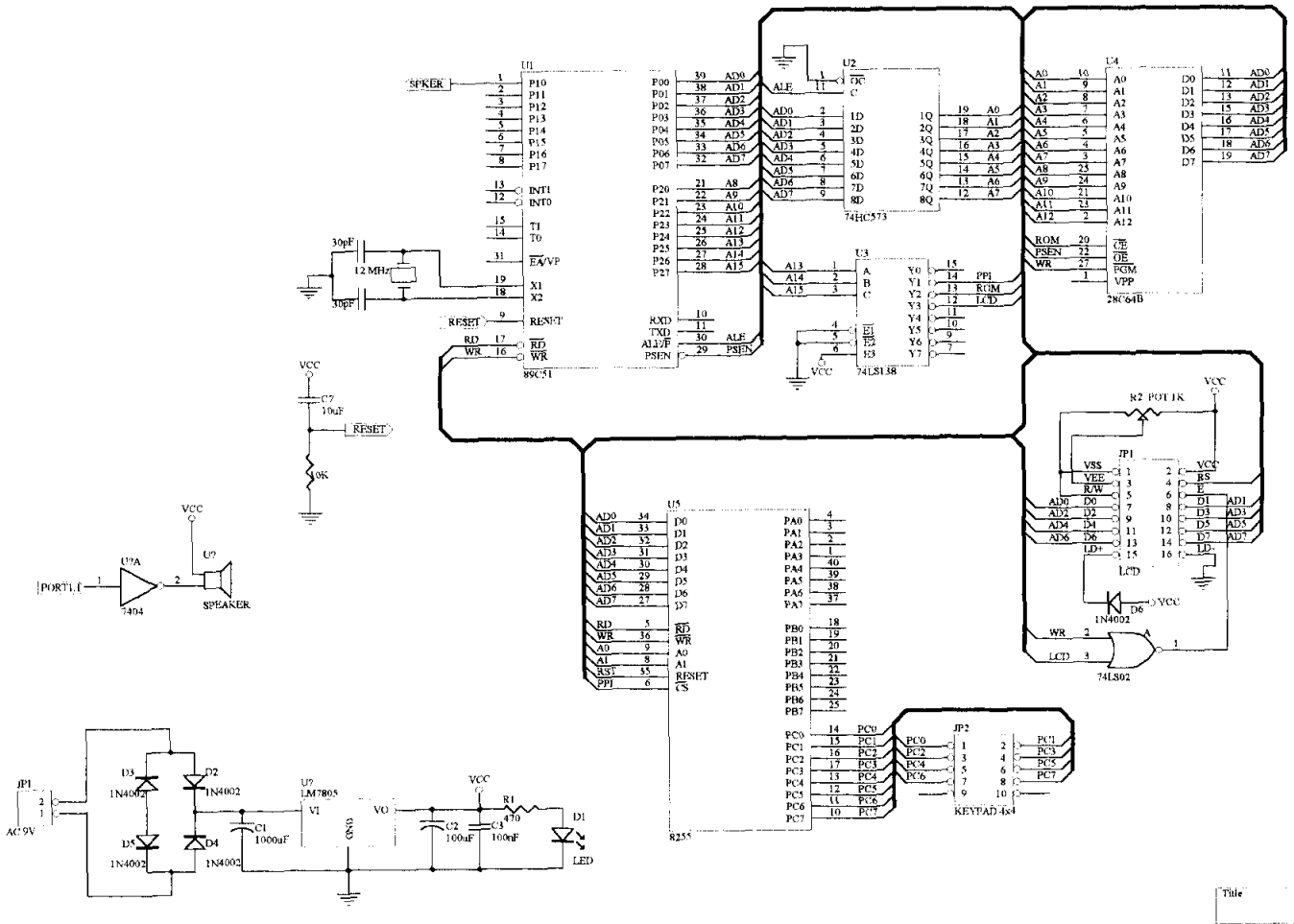
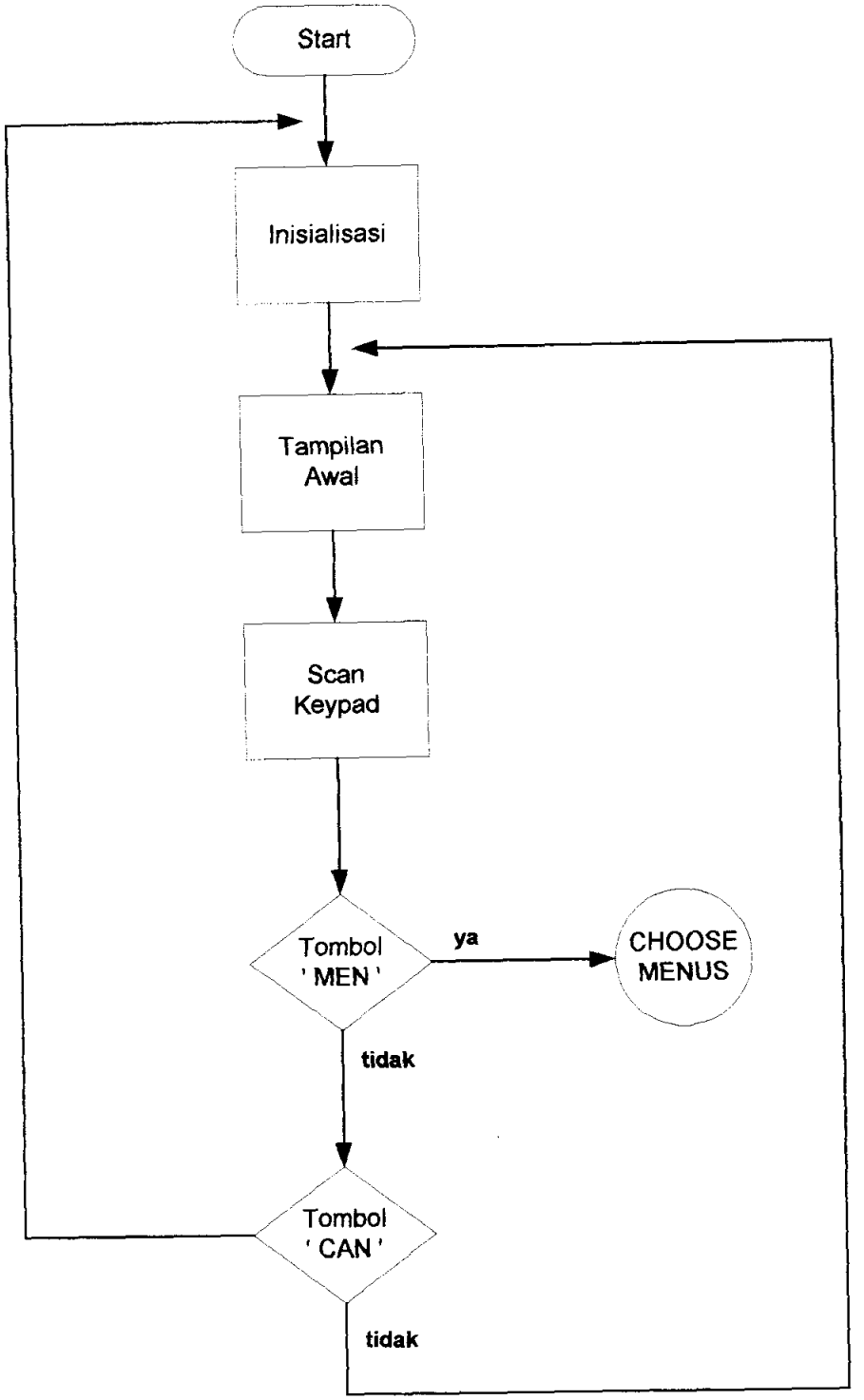
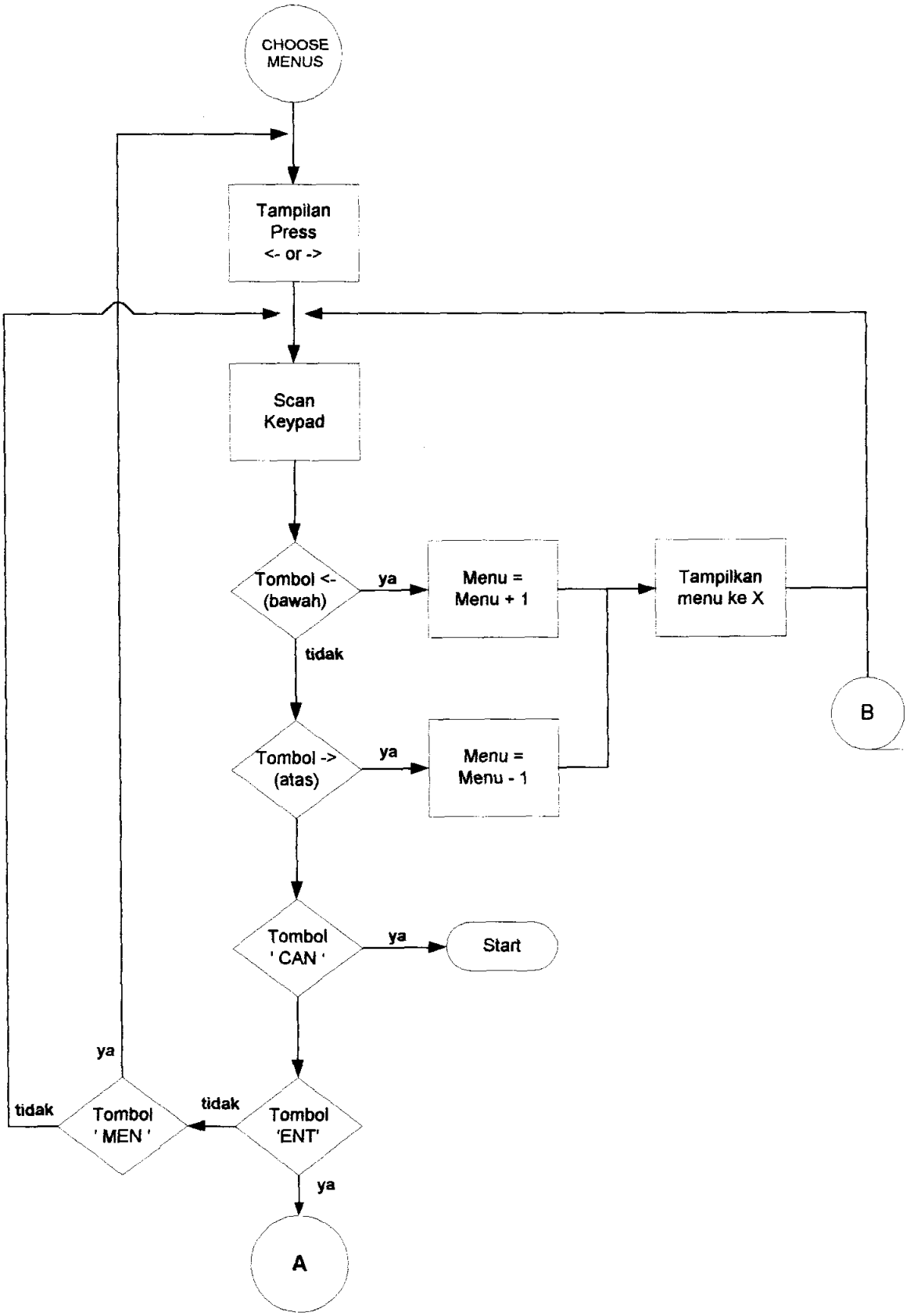
LAMPIRAN

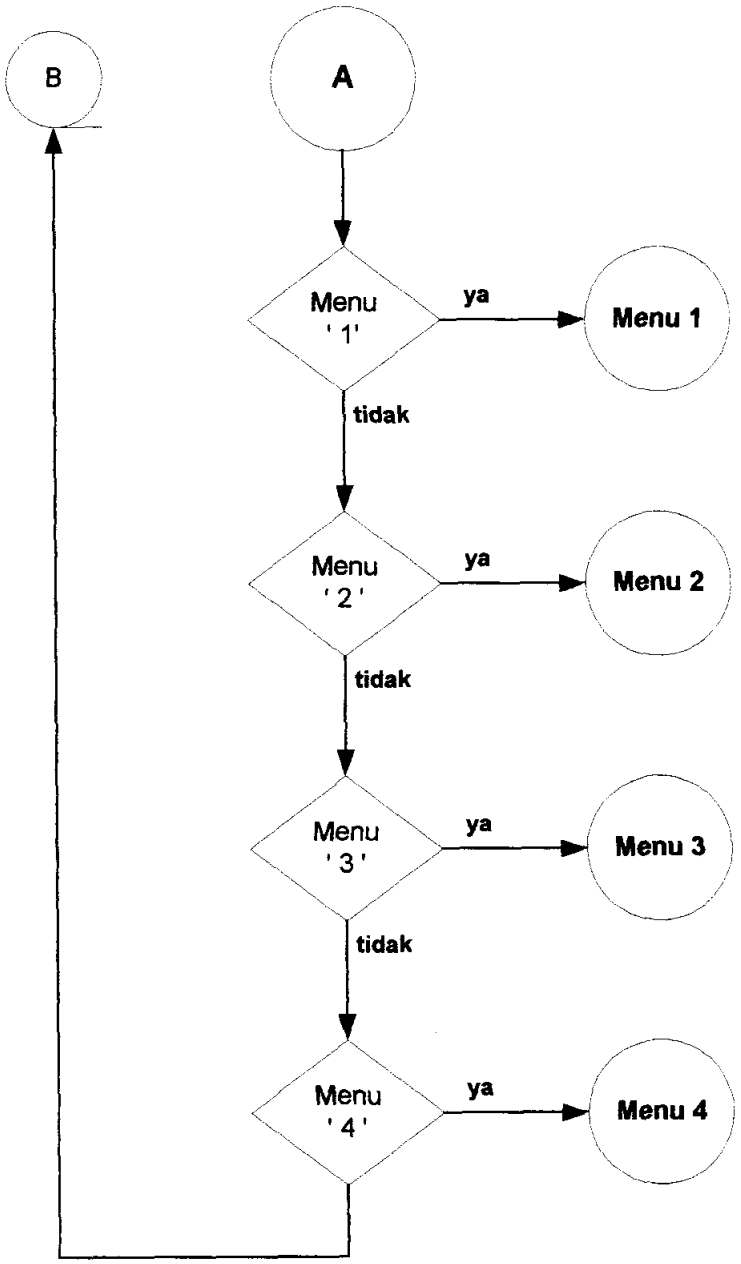


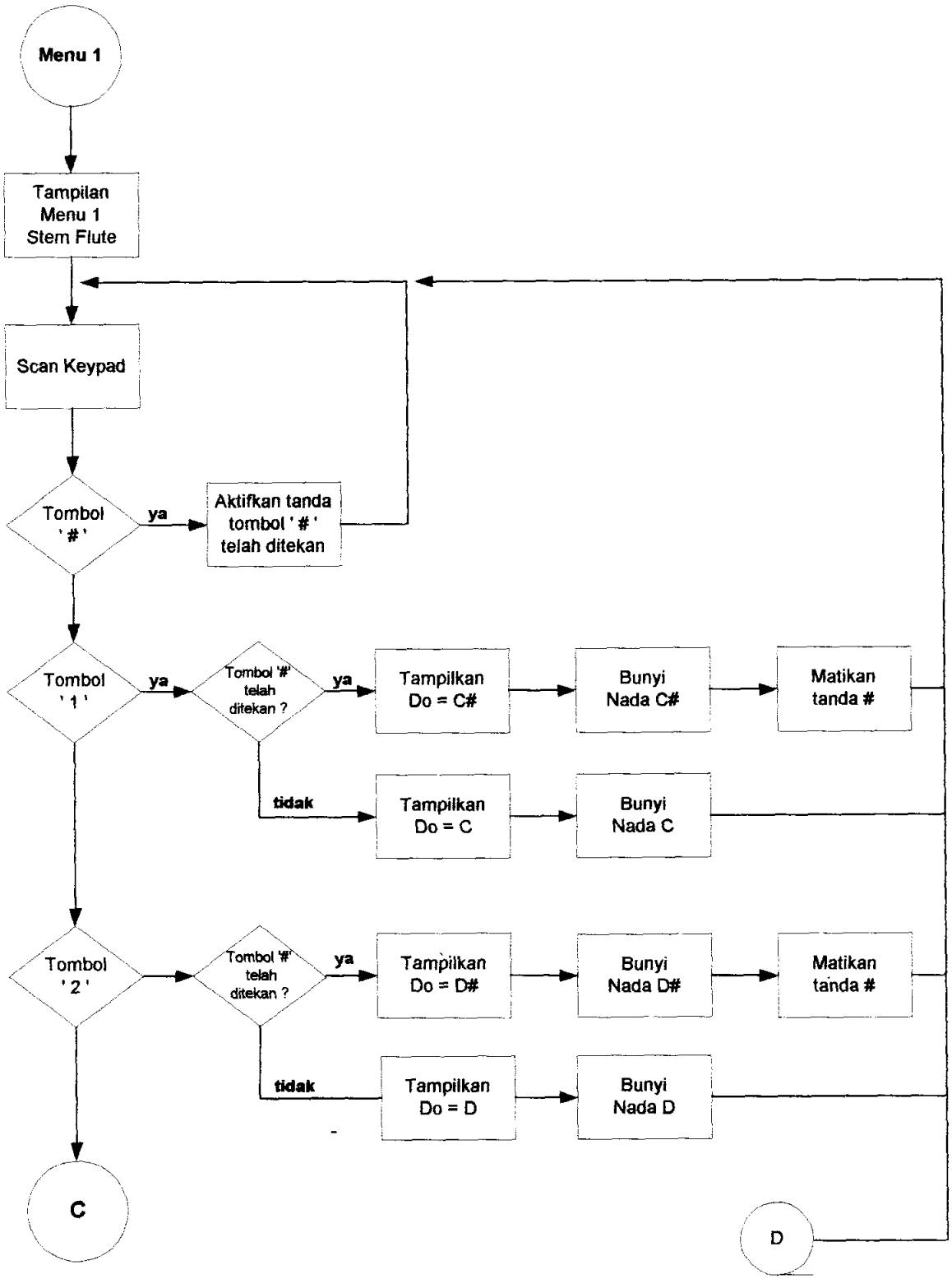
Title	METRONOME & STEM FLUTE DIGITAL		
Size	Number		Revision
B	SKRIPSI		
Date:	26-Jul-2002	Sheet of	1
File:	E:\W15NU~1\METRO.SCH	Drawn By:	WISNU YUWONO

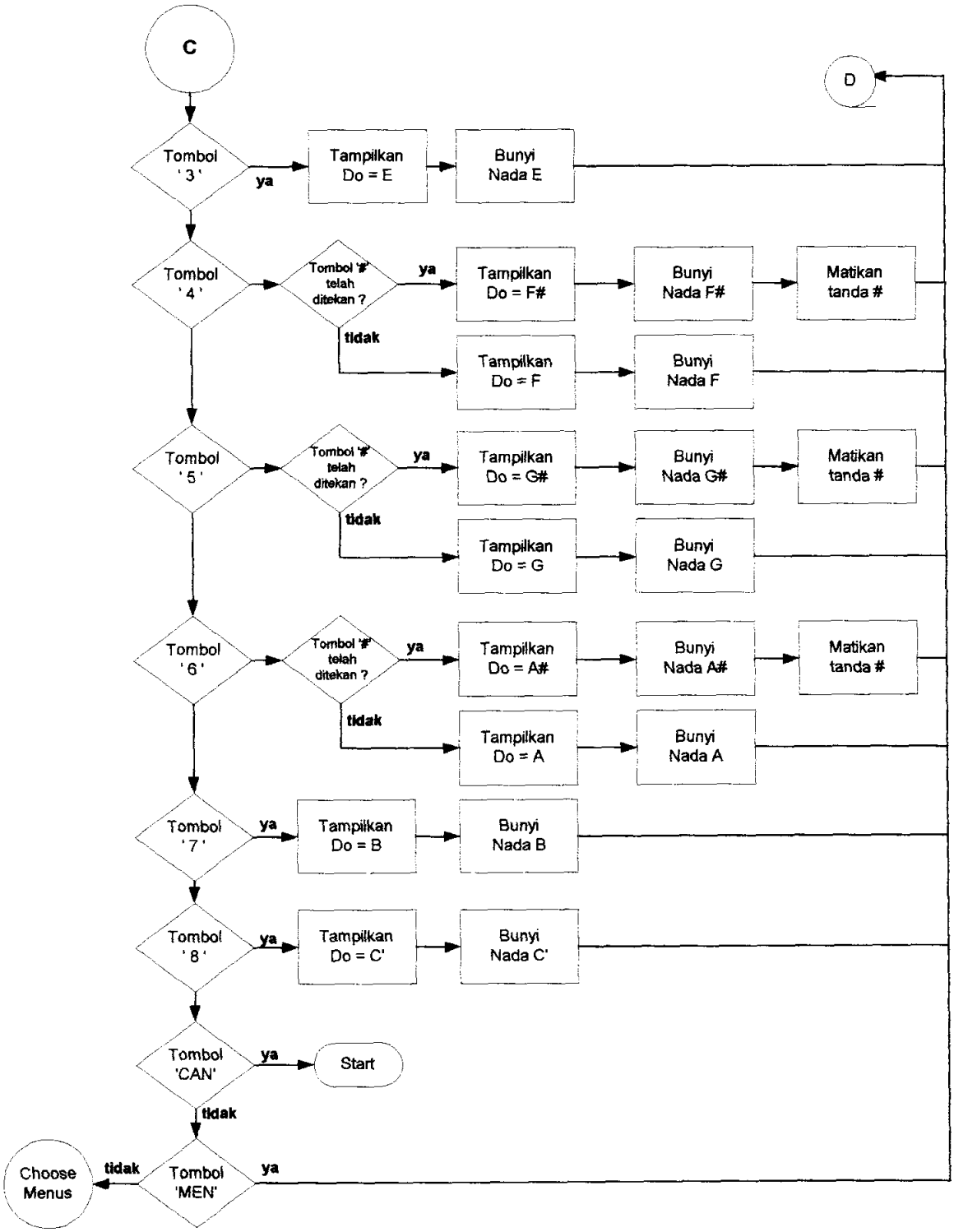
DIAGRAM ALUR SOFTWARE
METRONOME DAN STEM FLUTE DIGITAL

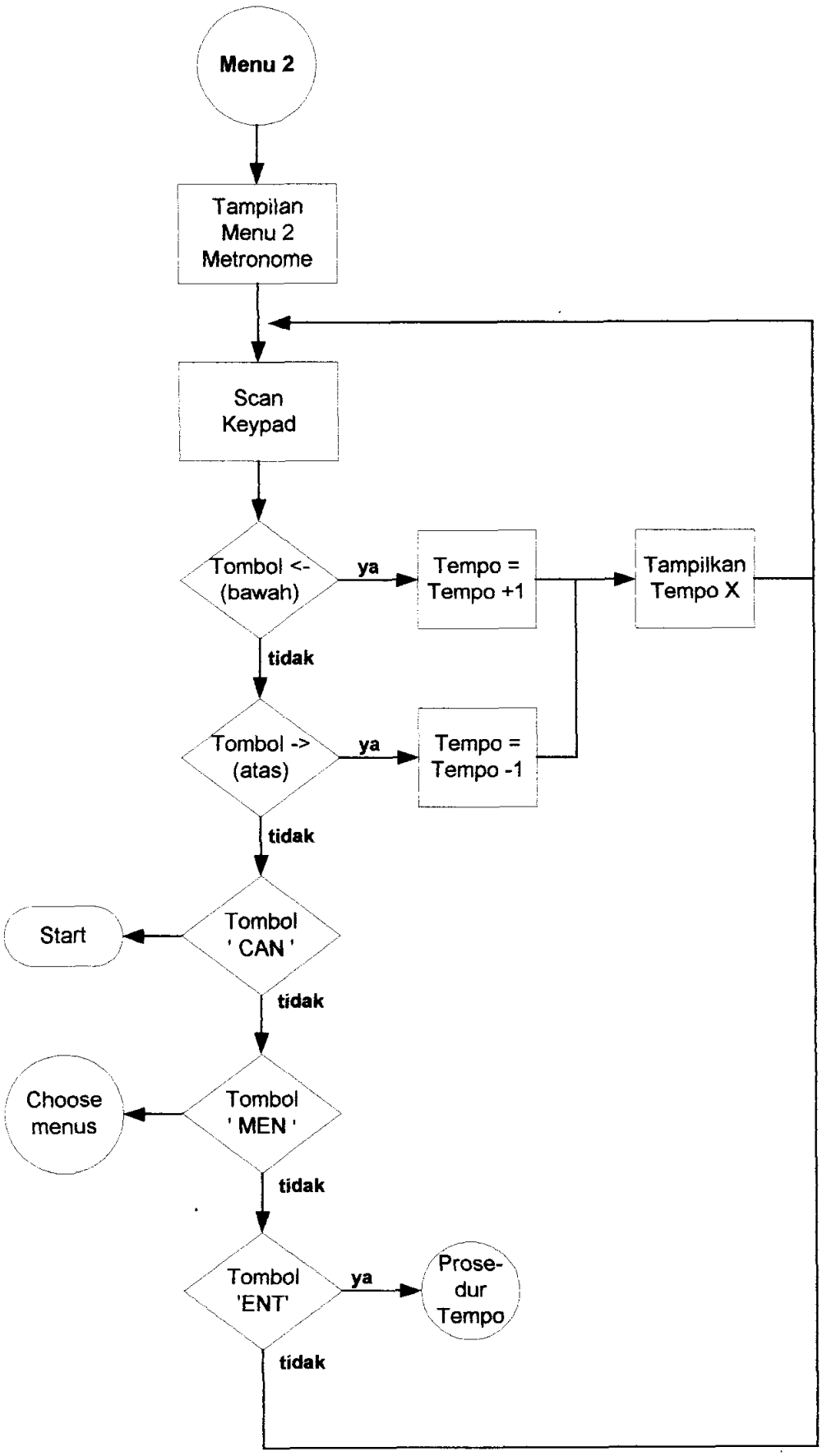


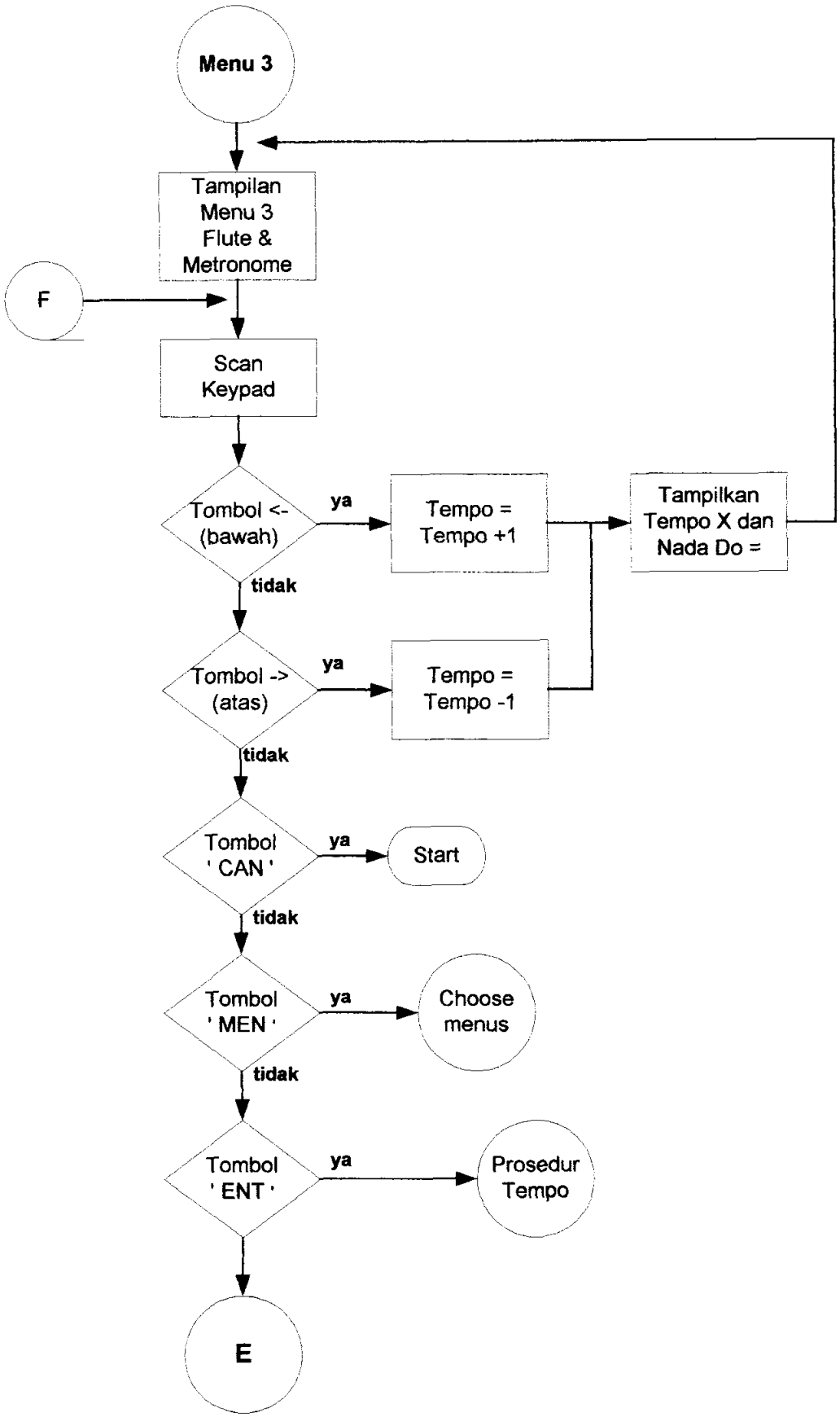

CHOOSE MENUS
Tampilan Press <- or ->
Scan Keypad
Tombol <- (bawah)
ya
Menu = Menu + 1
Tampilkan menu ke X
B
tidak
Tombol -> (atas)
ya
Menu = Menu - 1
Tombol ' CAN '
ya
Start
ya
tidak
Tombol ' MEN '
tidak
Tombol 'ENT'
ya
A

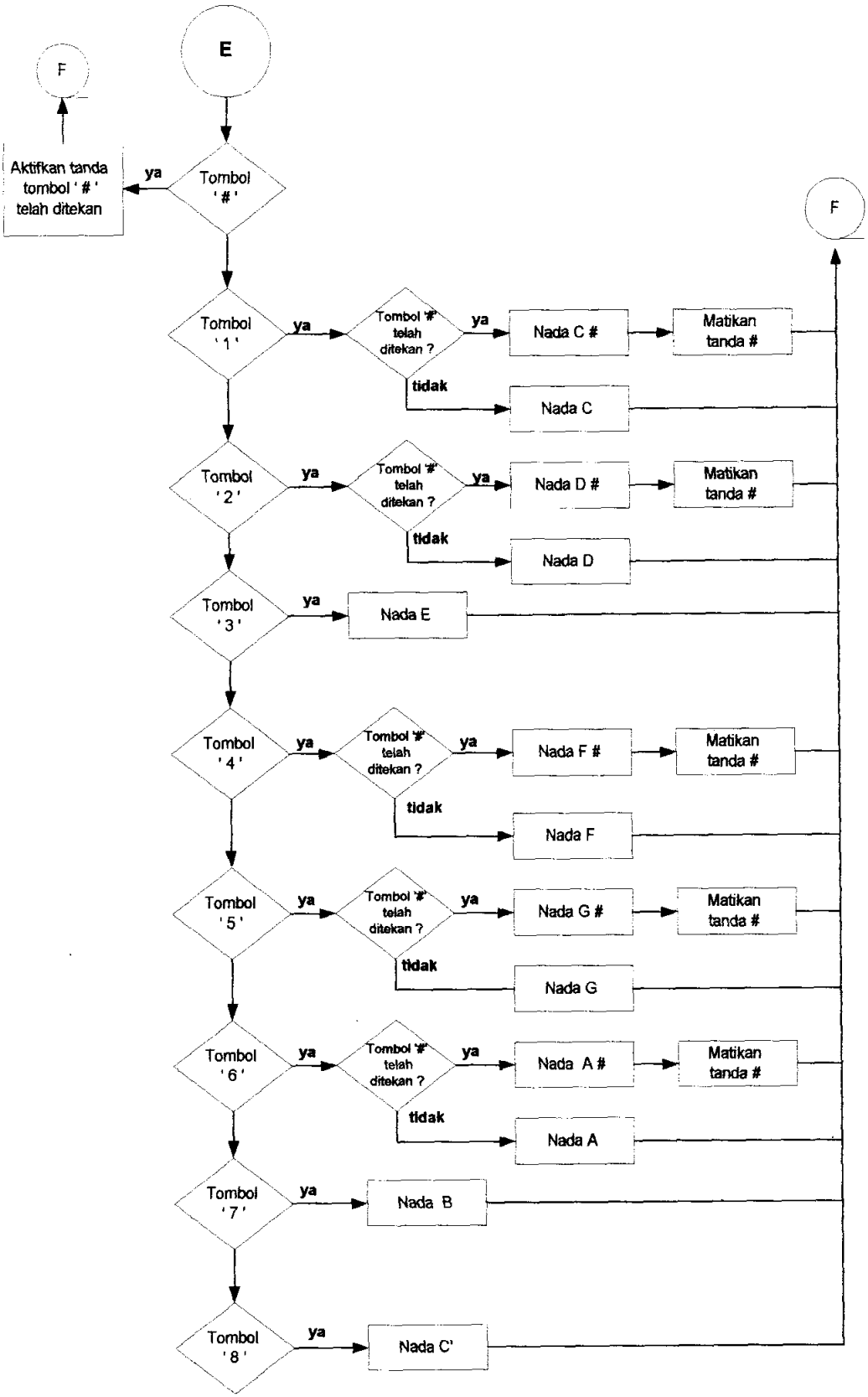

B
A
Menu
' 1'
ya
Menu 1
tidak
Menu
' 2 '
ya
Menu 2
tidak
Menu
' 3 '
ya
Menu 3
tidak
Menu
' 4 '
ya
Menu 4

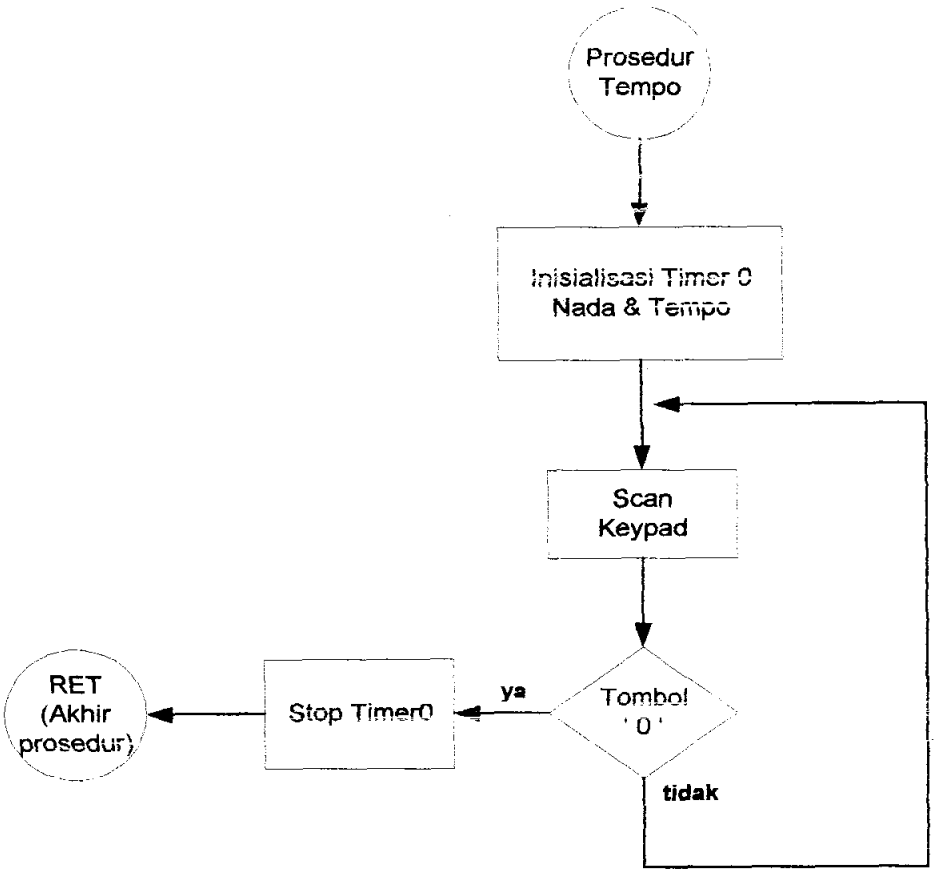

Menu 1
Tampilan Menu 1 Stem Flute
Scan Keypad
Tombol ' # '
ya
Aktifkan tanda tombol ' # ' telah ditekan
Tombol ' 1 '
ya
Tombol '#' telah ditekan ?
ya
Tampilkan Do = C#
Bunyi Nada C#
Matikan tanda #
tidak
Tampilkan Do = C
Bunyi Nada C
Tombol ' 2 '
Tombol '#' telah ditekan ?
ya
Tampilkan Do = D#
Bunyi Nada D#
Matikan tanda #
tidak
Tampilkan Do = D
Bunyi Nada D
C
D

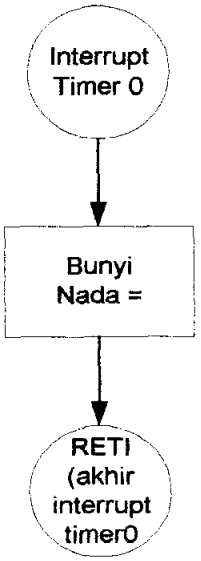

C
D
Tombol '3'
ya
Tampilkan Do = E
Bunyi Nada E
Tombol '4'
Tombol '#' telah ditekan ?
ya
Tampilkan Do = F#
Bunyi Nada F#
Matikan tanda #
tidak
Tampilkan Do = F
Bunyi Nada F
Tombol '5'
Tombol '#' telah ditekan ?
ya
Tampilkan Do = G#
Bunyi Nada G#
Matikan tanda #
tidak
Tampilkan Do = G
Bunyi Nada G
Tombol '6'
Tombol '#' telah ditekan ?
ya
Tampilkan Do = A#
Bunyi Nada A#
Matikan tanda #
tidak
Tampilkan Do = A
Bunyi Nada A
Tombol '7'
ya
Tampilkan Do = B
Bunyi Nada B
Tombol '8'
ya
Tampilkan Do = C'
Bunyi Nada C'
Tombol 'CAN'
ya
Start
tidak
Tombol 'MEN'
ya
tidak
Choose Menus


Menu 2
Tampilan Menu 2 Metronome
Scan Keypad
Tombol <- (bawah)
ya
Tempo = Tempo +1
Tampilkan Tempo X
tidak
Tombol -> (atas)
ya
Tempo = Tempo -1
tidak
Tombol ' CAN '
Start
tidak
Tombol ' MEN '
Choose menus
tidak
Tombol 'ENT'
ya
Prose-dur Tempo
tidak


Menu 3
Tampilan Menu 3 Flute & Metronome
F
Scan Keypad
Tombol <- (bawah)
ya
Tempo = Tempo +1
Tampilkan Tempo X dan Nada Do =
tidak
Tombol -> (atas)
ya
Tempo = Tempo -1
tidak
Tombol ' CAN '
ya
Start
tidak
Tombol ' MEN '
ya
Choose menus
tidak
Tombol ' ENT '
ya
Prosedur Tempo
E


E
F
Aktifkan tanda tombol '#' telah ditekan
ya
Tombol '#'
F
Tombol '1'
ya
Tombol '#' telah ditekan ?
ya
Nada C #
Matikan tanda #
tidak
Nada C
Tombol '2'
ya
Tombol '#' telah ditekan ?
ya
Nada D #
Matikan tanda #
tidak
Nada D
Tombol '3'
ya
Nada E
Tombol '4'
ya
Tombol '#' telah ditekan ?
ya
Nada F #
Matikan tanda #
tidak
Nada F
Tombol '5'
ya
Tombol '#' telah ditekan ?
ya
Nada G #
Matikan tanda #
tidak
Nada G
Tombol '6'
ya
Tombol '#' telah ditekan ?
ya
Nada A #
Matikan tanda #
tidak
Nada A
Tombol '7'
ya
Nada B
Tombol '8'
ya
Nada C'


Prosedur Tempo
Inisialisasi Timer 0 Nada & Tempo
Scan Keypad
Tombol ' 0 '
ya
tidak
Stop Timer0
RET (Akhir prosedur)


Interrupt Timer 0
Bunyi Nada =
RETI (akhir interrupt timer0

```
;skripsi_metronone_dan_stem_flute_digital
;wisnu yuwono_5103097039
;ADDRESS PPI

porta        equ   2000h
portb        equ   2001h
portc        equ   2002h
portcw       equ   2003h
nada         EQU   08H
simpan#      equ   09h
tombol       equ   0ah
simpanmenu   equ   0bh
simpantempo  equ   0ch
simpannada   equ   0dh
beat         equ   0eh

LCD          EQU   6000H ;LCD Address
LCD0         EQU   LCD+0 ;LCD CONTROL OPERATION
LCD1         EQU   LCD+1 ;LCD DATA OPERATION
;---------------------------------------------------------------
;                  INISIALISASI LCD
;---------------------------------------------------------------
DISPCLR EQU 00000001B       ;DISPLAY CLEAR
FUNCSET EQU 00111000B       ;INTERFACE DATA LENGTH :8 BITS
ENTRMOD EQU 00000110B       ;INCREMENT, NO DISPLAY SHIFT
DISPON  EQU 00001100B       ;DISPLAY ON, CURSOR OFF, BLINK OFF
CURSOR  EQU 00001110B       ;DISPLAY ON, CURSOR ON, BLINK OFF
BLINK   EQU 00001101B       ;DISPLAY ON, CURSOR OFF, BLINK ON

            ORG   00H
            LJMP  setawal
            ORG    36H
;---------------------------------------------------------------
;                     Procedure LCD Display
;---------------------------------------------------------------
POSISI2.1:          MOV   A,R7                  ;KOLOM
POSISI2:    ADD   A,#11000000B        ;POSISI DI BARIS 2
            SJMP  POSISI.SUB
POSISI1.1:          MOV   A,R6                  ;KOLOM
POSISI1:    ADD   A,#10000000B        ;POSISI DI BARIS 1
POSISI.SUB:         DEC   A                     ;AWALAN POSISI KOLOM DIMULAI DARI 0
            ACALL CONTROLOUT          ;KIRIM SEBAGAI OPERASI KONTROL
            RET

PRINTSTRING2:                               ;CETAK STRING DI BARIS 2 KOLOM 1
            ACALL POSISI2.1           ;BARIS 2 KOLOM 1
            SJMP PRINTSTRING          ;CETAK STRING
PRINTSTRING1:                               ;CETAK STRING DI BARIS 1 KOLOM 1
              ACALL POSISI1.1           ;BARIS 1 KOLOM 1
PRINTSTRING:                                ;CETAK STRING
            SJMP OUTSTRING            ;AMBIL DATA DULU
PRINTSTRINGLOOP:
            ACALL DATAOUT             ;KIRIM SEBAGAI OPERASI DATA
            INC DPTR                  ;POSISI DATA BERIKUTNYA
OUTSTRING:  CLR A                     ;POINTER=0
            MOVC A,@A+DPTR            ;AMBIL DATA BERDASARKAN DPTR
            JNZ PRINTSTRINGLOOP       ;APAKAH MASIH ADA DATA BERIKUTNYA
```

```
            RET
CONTROLOUT: PUSH DPH
            PUSH DPL
            MOV DPTR,#LCD0              ;ALAMAT OPERASI CONTROL LCD
            SJMP LCD.OUT                ;KIRIM KE LCD
DATAOUT:    PUSH DPH
            PUSH DPL
            MOV DPTR,#LCD1              ;ALAMAT OPERASI DATA LCD
LCD.OUT:                                    ;KIRIM KE LCD
            MOVX @DPTR,A
DELAY.LCD:
            MOV A,#250
            DJNZ A,$
            POP DPL
            POP DPH
            RET

DELAY.INIT.LCD:
            MOV R1,#020h            ;20h
DLY.LCD.LP:
            MOV R2,#0
            DJNZ R2,$
            DJNZ R1,DLY.LCD.LP
            RET
INIT.LCD:   MOV A,#DISPCLR              ;DISPLAY CLEAR
            ACALL CONTROLOUT
            ACALL DELAY.INIT.LCD
            MOV A,#FUNCSET              ;FUNCTION SET
            ACALL CONTROLOUT
            MOV A,#DISPON           ;DISPLAY ON
            ACALL CONTROLOUT
            MOV A,#ENTRMOD          ;ENTRY MODE
            ACALL CONTROLOUT
            RET
BUNYI:
masuk1:     SETB  P1.0
            ACALL DELAYnada
            CLR   P1.0
            ACALL DELAYnada
            RET

Delay3:     MOV   R2,#0aH
Del2:       MOV   r3,#0aH
            DJNZ  R3,$
            DJNZ  R2,Del2
            RET

DELAYnada   MOV   R5,nada
LAGI
            MOV   TMOD,#21h
            MOV   TH0,#0FFH
            MOV   TL0,#0F6H
            SETB  TR0
ULANG
            NOP
            JBC TF0,HITUNG
            SJMP ULANG
```

```
HITUNG
            djnz R5,LAGI
            RET

DELAY:      MOV   r1,#10
DEL:        MOV   r2,#30
            DJNZ  r2,$
            DJNZ  r1,DEL
            RET

okedeh:
            acall scankeypad
            mov   r0,tombol
            cjne  r0,#'0',lanjutx
            ajmp  selasai
lanjutx     mov   r3,#80
            acall oke
            ACALL delaybeat
            ajmp  okedeh
selasai
            RET

DELAYbeat
            MOV   R4,beat
LAGIx       mov   r2,#20
lagilagi
            MOV   TMOD,#21h
            MOV   TH1,#00H
            MOV   TL1,#00H
            SETB  TR1
ULANGe
            NOP
            JBC   TF1,HITUNGe
            SJMP  ULANGe
HITUNGe
            djnz  r2,lagilagi
            djnz  R4,LAGIx
            RET


siltek:     DB '   METRONOME     ',0
sil:  DB '  STEM  FLUTE    ',0
m12:  DB '     Menu 1      ',0
m22:  DB '     Menu 2      ',0
m32:  DB '     Menu 3      ',0
m42:  DB '     Menu 4      ',0
m1:   DB '   Stem Flute    ',0
m2:   DB '   Metronome     ',0
m3:   DB 'Flute-Metronome ',0
m4:    DB '  Song Example  ',0
menu:    db '  Choose Menus  ',0
pressb:  db ' Press <- or -> ',0
silcob1: db 'Press 1,2,3,... ',0
silcob2: db ' CAN for cancel ',0
silcob1x: db '1,2,... u/ Nada ',0
silcob2x: db '<- or ->u/ Tempo',0
doc:   db '     DO = C      ',0
```

```
dod:    db '      DO = D      ',0
doe:    db '      DO = E      ',0
dof:    db '      DO = F      ',0
dog:    db '      DO = G      ',0
doa:    db '      DO = A      ',0
dob:    db '      DO = B      ',0
doct:   db '      DO = C1     ',0
dock:   db '      DO = C#     ',0
dodk:   db '      DO = D#     ',0
dofk:   db '      DO = F#     ',0
dogk:   db '      DO = G#     ',0
doak:   db '      DO = A#     ',0
kosong:db '                  ',0
tem1    db '      LARGO       ',0
tem2    db '      LENTO       ',0
tem3    db '      ADAGIO      ',0
tem4    db '    LARGHETTO     ',0
tem5    db '    ADAGIETTO     ',0
tem6    db '    ANDANTE       ',0
tem7    db '    ANDANTINO     ',0
tem8    db '    MAESTOSO      ',0
kaspo3 db 'jangan ditulisi  ',0
tem10   db '    ALLEGRETO     ',0
tem11   db '    ANIMATO       ',0
tem12   db '    ALLEGRO       ',0
tem13   db '     ASSAI        ',0
tem14   db '     VIVACE       ',0
tem15   db '     PRESTO       ',0
tem16   db '   PRESTISSIMO    ',0
tem9    db '    MODERATO      ',0
tem1a   db '  ( 44 - 46 )     ',0
tem2a   db '  ( 52 - 54 )     ',0
tem3a   db '  ( 56 - 58 )     ',0
tem4a   db '  ( 60 - 63 )     ',0
tem5a   db '  ( 66 - 69 )     ',0
tem6a   db '  ( 72 - 76 )     ',0
tem7a   db '  ( 80 - 84 )     ',0
tem8a   db '  ( 88 - 92 )     ',0
tem9a   db '  ( 96 - 100 )    ',0
tem10a db '  ( 108 - 112 )   ',0
tem11a db '  ( 120 - 126 )   ',0
tem12a db '  ( 132 - 138 )   ',0
tem13a db '  ( 144 - 152 )   ',0
tem14a db '  ( 160 - 168 )   ',0
tem15a db '  ( 184 - 192 )   ',0
tem16a db '     ( 208 )      ',0
judul1 db 'JOYFUL, JOYFUL..',0
judul2 db '  -beethoven-    ',0
;-----------------------------------------
;               INITIALISASI PPI 8255
;-----------------------------------------
init.ppi    MOV   DPTR,#PORTcw
            MOV   A,#081h
            MOVX  @DPTR,A
            ret
;--------------------------------
;       scan  keypad
```

```
;--------------------------------
scankeypad:
            mov   tombol,#20h
            MOV   A,#01110000B      ; scan baris 1
            MOV   DPTR,#portc       ;ALAMAT CONTROL WORD
            MOVX  @DPTR,A
            CLR   A

            mov   dptr,#portc
            MOVx  A,@dptr
            mov   r1,a
            anl   a,#00001111b
            cjne  a,#0eh,cek_0

            mov   tombol,#'#'
            ajmp  ketemu

cek_0       mov   a,r1
            anl   a,#00001111b
            cjne  a,#0dh,cek_ENT

            mov   tombol,#'0'
            ajmp  ketemu

cek_ENT     mov   a,r1
            anl   a,#00001111b
            cjne  a,#0bh,cek_B

            mov   tombol,#'E'
            ajmp  ketemu

cek_B       mov   a,r1
            anl   a,#00001111b
            cjne  a,#07h,kosong2

            mov   tombol,#'b'
            ajmp  ketemu

kosong2:    MOV   A,#10110000B      ; scan baris 2
            MOV   DPTR,#portc       ;ALAMAT CONTROL WORD
            MOVX        @DPTR,A
            CLR   A

            mov   dptr,#portc
            MOVx  A,@dptr
            mov   r1,a
            anl   a,#00001111b
            cjne  a,#0eh,cek_8

            mov   tombol,#'7'
            ajmp  ketemu

cek_8:      mov   a,r1
            anl   a,#00001111b
            cjne  a,#0dh,cek_9

            mov   tombol,#'8'
```

```
            ajmp   ketemu

cek_9       mov    a,r1
            anl    a,#00001111b
            cjne   a,#0bh,cek_A
            mov    tombol,#'9'
            ajmp   ketemu

cek_A       mov    a,r1
            anl    a,#00001111b
            cjne   a,#07h,kosong3

            mov    tombol,#'A'
            ajmp   ketemu

kosong3:
            MOV    A,#11010000B      ; scan baris 3
            MOV    DPTR,#portc       ;ALAMAT CONTROL WORD
            MOVX   @DPTR,A
            CLR    A

            mov    dptr,#portc
            MOVx   A,@dptr
            mov    r1,a
            anl    a,#00001111b
            cjne   a,#0eh,cek_5

            mov    tombol,#'4'
            ajmp   ketemu

cek_5:      mov    a,r1
            anl    a,#00001111b
            cjne   a,#0dh,cek_6

            mov    tombol,#'5'
            ajmp   ketemu

cek_6       mov    a,r1
            anl    a,#00001111b
            cjne   a,#0bh,cek_C
            mov    tombol,#'6'
            ajmp   ketemu

cek_C       mov    a,r1
            anl    a,#00001111b
            cjne   a,#07h,kosong4

            mov    tombol,#'C'
            ajmp   ketemu

kosong4:    MOV    A,#11100000B      ; scan baris 4
            MOV    DPTR,#portc       ;ALAMAT CONTROL WORD
            MOVX   @DPTR,A
            CLR    A

            mov    dptr,#portc
            MOVx   A,@dptr
```

```
                mov   r1,a
                anl   a,#00001111b
                cjne  a,#0eh,cek_2

                mov   tombol,#'1'
                ajmp  ketemu

cek_2:          mov   a,r1
                anl   a,#00001111b
                cjne  a,#0dh,cek_3

                mov   tombol,#'2'
                ajmp  ketemu

cek_3           mov   a,r1
                anl   a,#00001111b
                cjne  a,#0bh,cek_MENU
                mov   tombol,#'3'
                ajmp  ketemu

cek_MENU        mov   45h,#03h     ;tanda belum dipencet
                mov   a,r1
                anl   a,#00001111b
                cjne  a,#07h,ketemu

                mov   tombol,#'M'
ketemu:         ret

oke:
ulul1:          acall bunyi
                djnz  r3,ulul1
                ret

tam_menu_ke     mov   r7,#05h
ulul3a:         mov   r6,#0ffh
ulul2a:         acall delay3
                djnz  r6,ulul2a
                djnz  r7,ulul3a

                mov   a,simpanmenu
                cjne  a,#01h,lancek0
                acall tam_menu1
                ajmp  lancek3
lancek0         cjne  a,#02h,lancek1
                acall tam_menu2
                ajmp  lancek3
lancek1         cjne  a,#03h,lancek2
                acall tam_menu3
                ajmp  lancek3
lancek2         cjne  a,#04h,lancek3
                acall tam_menu4
lancek3         ret

tam_menu1:      mov   r6,#1
                mov   r7,#1
                mov   dptr,#m12
                acall printstring1
```

```
                mov     dptr,#m1
                acall printstring2
                ret

tam_menu2:      mov     r6,#1
                mov     r7,#1
                mov     dptr,#m22
                acall printstring1
                mov     dptr,#m2
                acall printstring2
                ret

tam_menu3:      mov     r6,#1
                mov     r7,#1
                mov     dptr,#m32
                acall printstring1
                mov     dptr,#m3
                acall printstring2
                ret

tam_menu4:      mov     r6,#1
                mov     r7,#1
                mov     dptr,#m42
                acall printstring1
                mov     dptr,#m4
                acall printstring2
                ret

tam_menu5:      mov     r6,#1
                mov     r7,#1
                mov     dptr,#menu
                acall printstring1
                mov     dptr,#pressb
                acall printstring2
                ret

choosemenu:     acall tam_menu5
ulang2:         acall scankeypad
                mov     r0,tombol
                cjne    r0,#'b',cmenu1          ;cek bawah
                inc     simpanmenu
                mov     r0,simpanmenu
                cjne    r0,#05h,belum4          ;jika menu=5 -> menu=0 >5<
                mov     simpanmenu,#00h

belum4:         acall tam_menu_ke
                ajmp    ulang2
cmenu1
                                                ;cek atas
                mov     r0,tombol
                cjne    r0,#'A',cmenu2
                dec     simpanmenu
                mov     r1,simpanmenu
                cjne    r1,#0ffh,trusss1
                mov     simpanmenu,#00h
trusss1:        mov     r0,simpanmenu
                cjne    r0,#00h,belum0
                mov     simpanmenu,#04h
```

```
belum0:     acall tam_menu_ke
            ajmp  ulang2
cmenu2
            mov   r0,tombol
            cjne  r0,#'C',cmenu3
            Ljmp  setawal
cmenu3      mov   r0,tombol
            cjne  r0,#'E',cmenu4

;           mov   r0,simpanmenu          ;?????
            mov   a,simpanmenu           ;@r0
            cjne  a,#01h,lancek0x
            acall menu1
            ajmp  lancek3
lancek0x    cjne  a,#02h,lancek1x
            lcall menu2
            ajmp  lancek3x
lancek1x    cjne  a,#03h,lancek2x
            lcall menu3
            ajmp  lancek3
lancek2x    cjne  a,#04h,lancek3x
            lcall menu4
lancek3x    ljmp  setawal

cmenu4      mov   r0,tombol
            cjne  r0,#'M',cmenu5
            ajmp  choosemenu
cmenu5      ajmp  ulang2
            ret
tuliskosong:
            mov   r7,#1
            mov   dptr,#kosong
            acall printstring2
            ret

menu1:      acall tam_menu1          ;tampilan menu 1
            mov   r6,#1
            mov   r7,#1
            mov   dptr,#silcob1
            acall printstring1
            mov   dptr,#silcob2
            acall printstring2

menux1:     acall scankeypad         ;reset jika cancel
            mov   r0,tombol
            cjne  r0,#'C',menu1bal0
            ljmp  setawal
menu1bal0:
            mov   r0,tombol          ;tandai jika # ditekan
            cjne  r0,#'#',menu1bal1
            mov   simpan#,#0AAh
menu1bal1:
            mov   r0,tombol
            cjne  r0,#'1',menu1bal2
            mov   r1,simpan#
            cjne  r1,#0AAh,kres1          ;cek kres dipitek
            mov   r6,#1
```

```
            mov   dptr,#dock
            acall printstring1
            acall tuliskosong
            mov   r3,#200
            mov   nada,#74          ;set delay untuk nada kres
            mov   simpan#,#00h
            ajmp  okedo1
kres1:
            mov   r6,#1             ;nada biasa tanpa kres
            mov   dptr,#doc
            acall printstring1
            acall tuliskosong
            mov   r3,#200
            mov   nada,#78
okedo1:     acall oke
            mov   dptr,#portc
            MOVx  A,@dptr
            mov   r1,a
            anl   a,#00001111b
            cjne  a,#0eh,menulbal2

            ajmp  okedo1

menulbal2:
            mov   r0,tombol         ;idem menulbal1
            cjne  r0,#'2',menulbal3
            mov   r1,simpan#
            cjne  r1,#0AAh,kres2
            mov   r6,#1
            mov   dptr,#dodk
            acall printstring1
            acall tuliskosong
            mov   r3,#200
            mov   nada,#66
            mov   simpan#,#00h
            ajmp  okedo2

kres2       mov   r6,#1
            mov   dptr,#dod
            acall printstring1
            acall tuliskosong
            mov   r3,#200
            mov   nada,#71
okedo2      acall oke
            mov   dptr,#portc
            MOVx  A,@dptr
            mov   r1,a
            anl   a,#00001111b
            cjne  a,#0dh,menulbal3

            ajmp  okedo2

menulbal3:  mov   r0,tombol
            cjne  r0,#'3',menulbal4
            mov   r6,#1
            mov   dptr,#doe
            acall printstring1
```

```
                acall tuliskosong
                mov   r3,#200
                mov   nada,#62
okedox1         acall oke
                mov   dptr,#portc
                MOVx  A,@dptr
                mov   r1,a
                anl   a,#00001111b
                cjne  a,#0bh,menulbal4

                ajmp  okedox1

menulbal4:
                mov   r0,tombol
                cjne  r0,#'4',menulbal5
                mov   r1,simpan#
                cjne  r1,#0AAh,kres3
                mov   r6,#1
                mov   dptr,#dofk
                acall printstring1
                acall tuliskosong
                mov   r3,#250
                mov   nada,#55
                mov   simpan#,#00h
                ljmp  okedo3

kres3:          mov   r6,#1
                mov   dptr,#dof
                lcall       printstring1
                lcall tuliskosong
                mov   r3,#250
                mov   nada,#59
okedo3:               lcall oke
                mov   dptr,#portc
                MOVx  A,@dptr
                mov   r1,a
                anl   a,#00001111b
                cjne  a,#0eh,menulbal5

                ajmp  okedo3

menulbal5:
                mov   r0,tombol
                cjne  r0,#'5',menulbal6
                mov   r1,simpan#
                cjne  r1,#0AAh,kres4
                mov   r6,#1
                mov   dptr,#dogk
                lcall printstring1
                lcall tuliskosong
                mov   r3,#250
                mov   nada,#49
                mov   simpan#,#00h
                ajmp  okedo4

kres4           mov   r6,#1
                mov   dptr,#dog
```

```
            lcall printstring1
            lcall tuliskosong
            mov   r3,#250
            mov   nada,#52
okedo4      lcall oke
            mov   dptr,#portc
            MOVx  A,@dptr
            mov   r1,a
            anl   a,#00001111b
            cjne  a,#0dh,menu1bal6

            ajmp  okedo4

menu1bal6:
            mov   r0,tombol
            cjne  r0,#'6',menu1bal7
            mov   r1,simpan#
            cjne  r1,#0AAh,kres5
            mov   r6,#1
            mov   dptr,#doak
            lcall printstring1
            lcall tuliskosong
            mov   r3,#250
            mov   nada,#44
            mov   simpan#,#00h
            ajmp  okedo5

kres5:      mov   r6,#1
            mov   dptr,#doa
            lcall printstring1
            lcall tuliskosong
            mov   r3,#250
            mov   nada,#46
okedo5      lcall oke
            mov   dptr,#portc
            MOVx  A,@dptr
            mov   r1,a
            anl   a,#00001111b
            cjne  a,#0bh,menu1bal7
            ajmp  okedo5

menu1bal7:
            mov   r0,tombol
            cjne  r0,#'7',menu1bal8
            mov   r6,#1
            mov   dptr,#dob
            lcall printstring1
            lcall tuliskosong
            mov   r3,#230
            mov   nada,#41
okedox2     lcall oke
            mov   dptr,#portc
            MOVx  A,@dptr
            mov   r1,a
            anl   a,#00001111b
            cjne  a,#0eh,menu1bal8
```

```
                ajmp    okedox2

menu1bal8:
                mov     r0,tombol
                cjne    r0,#'8',menu1bal9
                mov     r6,#1
                mov     dptr,#doct
                lcall   printstring1
                lcall   tuliskosong
                mov     r3,#230
                mov     nada,#39
okedox3         lcall   oke
                mov     dptr,#portc
                MOVx    A,@dptr
                mov     r1,a
                anl     a,#00001111b
                cjne    a,#0dh,menu1bal9

                ajmp    okedox3

menu1bal9:      ljmp    menux1
                ret

ta_menu_2_donk
                mov     r7,#05h
ulul3axx:       mov     r6,#0ffh
ulul2axx:       lcall   delay3
                djnz    r6,ulul2axx
                djnz    r7,ulul3axx

                mov     a,simpantempo
                cjne    a,#01h,lancek0xx
                mov     r6,#1
                mov     dptr,#tem1
                lcall   printstring1
                mov     r7,#1
                mov     dptr,#tem1a
                lcall   printstring2
                mov     beat,#223
                ljmp    lancek16x
lancek0xx       cjne    a,#02h,lancek1xx
                mov     r6,#1
                mov     dptr,#tem2
                lcall   printstring1
                mov     r7,#1
                mov     dptr,#tem2a
                lcall   printstring2
                mov     beat,#188
                ljmp    lancek16x
lancek1xx       cjne    a,#03h,lancek2xx
                mov     r6,#1
                mov     dptr,#tem3
                lcall   printstring1
                mov     r7,#1
                mov     dptr,#tem3a
                lcall   printstring2
                mov     beat,#168
```

```
            ljmp  lancek16x
lancek2xx   cjne  a,#04h,lancek3xx
            mov   r6,#1
            mov   dptr,#tem4
            lcall printstring1
            mov   r7,#1
            mov   dptr,#tem4a
            lcall printstring2
            mov   beat,#153
            ljmp  lancek16x
lancek3xx   cjne  a,#05h,lancek4x
            mov   r6,#1
            mov   dptr,#tem5
            lcall printstring1
            mov   r7,#1
            mov   dptr,#tem5a
            lcall printstring2
            mov   beat,#138
            ljmp  lancek16x
lancek4x    cjne  a,#06h,lancek5x
            mov   r6,#1
            mov   dptr,#tem6
            lcall printstring1
            mov   r7,#1
            mov   dptr,#tem6a
            lcall printstring2
            mov   beat,#127
            ljmp  lancek16x
lancek5x    cjne  a,#07h,lancek6x
            mov   r6,#1
            mov   dptr,#tem7
            lcall printstring1
            mov   r7,#1
            mov   dptr,#tem7a
            lcall printstring2
            mov   beat,#110
            ljmp  lancek16x
lancek6x    cjne  a,#08h,lancek7x
            mov   r6,#1
            mov   dptr,#tem8
            lcall printstring1
            mov   r7,#1
            mov   dptr,#tem8a
            lcall printstring2
            mov   beat,#99
            ljmp  lancek16x
lancek7x    cjne  a,#09h,lancek8x
            mov   r6,#1
            mov   dptr,#tem9
            lcall printstring1
            mov   r7,#1
            mov   dptr,#tem9a
            lcall printstring2
            mov   beat,#87
            ljmp  lancek16x
lancek8x    cjne  a,#0ah,lancek9x
            mov   r6,#1
```

```
                mov   dptr,#tem10
                lcall printstring1
                mov   r7,#1
                mov   dptr,#tem10a
                lcall printstring2
                mov   beat,#75
                ajmp  lancek16x
lancek9x        cjne  a,#0bh,lancek10x
                mov   r6,#1
                mov   dptr,#tem11
                lcall printstring1
                mov   r7,#1
                mov   dptr,#tem11a
                lcall printstring2
                mov   beat,#65
                ajmp  lancek16x
lancek10x       cjne  a,#0ch,lancek11x
                mov   r6,#1
                mov   dptr,#tem12
                lcall printstring1
                mov   r7,#1
                mov   dptr,#tem12a
                lcall printstring2
                mov   beat,#55
                ajmp  lancek16x
lancek11x       cjne  a,#0dh,lancek12x
                mov   r6,#1
                mov   dptr,#tem13
                lcall printstring1
                mov   r7,#1
                mov   dptr,#tem13a
                lcall printstring2
                mov   beat,#49
                ajmp  lancek16x
lancek12x       cjne  a,#0eh,lancek13x
                mov   r6,#1
                mov   dptr,#tem14
                lcall printstring1
                mov   r7,#1
                mov   dptr,#tem14a
                lcall printstring2
                mov   beat,#44
                ajmp  lancek16x
lancek13x       cjne  a,#0fh,lancek14x
                mov   r6,#1
                mov   dptr,#tem15
                lcall printstring1
                mov   r7,#1
                mov   dptr,#tem15a
                lcall printstring2
                mov   beat,#32
                ajmp  lancek16x
lancek14x       cjne  a,#10h,lancek16x
                mov   r6,#1
                mov   dptr,#tem16
                lcall printstring1
                mov   r7,#1
```

```
            mov   dptr,#teml6a
            lcall printstring2
            mov   beat,#24
lancek16x
            ret

mulai_beat
            mov   nada,#78
            lcall okedeh
            ret

menu2
            lcall tam_menu2
            lcall ta_menu_2_donk

            mov   r7,#05h
ulul3axxx:  mov   r6,#0ffh
ulul2axxx:  lcall delay3
            djnz  r6,ulul2axxx
            djnz  r7,ulul3axxx

ulang2x:
            lcall scankeypad
            mov   r0,tombol
            cjne  r0,#'b',cmenu1x          ;cek bawah
            inc   simpantempo
            mov   r0,simpantempo
            cjne  r0,#17,belum14    ;jika tempo=14 -> tempo=0 >5<
            mov   simpantempo,#00h
belum14:    lcall ta_menu_2_donk
            ajmp  ulang2x
cmenu1x                                    ;cek atas
            mov   r0,tombol
            cjne  r0,#'A',cmenu2x
            dec   simpantempo
            mov   r1,simpantempo
            cjne  r1,#0ffh,trusss1x
            mov   simpantempo,#00h
trusss1x:   mov   r0,simpantempo
            cjne  r0,#00h,belum00
            mov   simpantempo,#16
belum00:    lcall ta_menu_2_donk
            ajmp  ulang2x
cmenu2x
            mov   r0,tombol
            cjne  r0,#'C',cmenu3x
            ajmp  setawal
cmenu3x     mov   r0,tombol
            cjne  r0,#'E',cmenu4x
            lcall mulai_beat
            ajmp  ulang2x

cmenu4x     mov   r0,tombol
            cjne  r0,#'M',cmenu5x
            ljmp  choosemenu
cmenu5x
            ajmp  ulang2x
```

```
            ret

delayku     mov   r7,#05h
ulul3axxxx: mov   r6,#0ffh
ulul2axxxx: lcall delay3
            djnz  r6,ulul2axxxx
            djnz  r7,ulul3axxxx
            ret
menu3
            lcall tam_menu3
            lcall ta_menu_2_donk
            mov   r7,#1             ;nada biasa tanpa kres
            mov   dptr,#doc
            lcall printstring2
            acall delayku

ulang3x:
            lcall scankeypad
            mov   r0,tombol
            cjne  r0,#'b',cmenu1xx        ;cek bawah
            inc   simpantempo
            mov   r0,simpantempo
            cjne  r0,#17,belum14x         ;jika tempo=14 -> tempo=0 >5<
            mov   simpantempo,#00h
belum14x:   lcall ta_menu_2_donk
            ajmp  ulang3x
cmenu1xx                                  ;cek atas
            mov   r0,tombol
            cjne  r0,#'A',cmenu2xx
            dec   simpantempo
            mov   r1,simpantempo
            cjne  r1,#0ffh,trusss1xx
            mov   simpantempo,#00h
trusss1xx:  mov   r0,simpantempo
            cjne  r0,#00h,belum000
            mov   simpantempo,#16
belum000:   lcall ta_menu_2_donk
            ajmp  ulang3x
cmenu2xx
            mov   r0,tombol
            cjne  r0,#'C',cmenu3xx
            ajmp  setawal
cmenu3xx    mov   r0,tombol
            cjne  r0,#'E',cmenu4xx

            mov   r7,#10
bunyi5detik lcall oke
            djnz  r7,bunyi5detik

            lcall okedeh
            ajmp  ulang3x

cmenu4xx    mov   r0,tombol
            cjne  r0,#'M',cmenu5xx
            ljmp  choosemenu
cmenu5xx
            mov   r0,tombol           ;tandai jika # ditekan
```

```
            cjne  r0,#'#',menu1ball1x
            mov   simpan#,#0AAh
            mov   r7,#1
            lcall posisi2.1
            mov   a,#'#'
            lcall dataout
            ljmp  ulang3x
menu1ball1x:
            mov   r0,tombol
            cjne  r0,#'1',menu1bal2x
            mov   r1,simpan#
            cjne  r1,#0AAh,kres1x          ;cek kres dipitek
            mov   r7,#1
            mov   dptr,#dock
            lcall printstring2
            mov   simpan#,#00h
            mov   nada,#74
            acall delayku
            ajmp  ulang3x
kres1x:
            mov   r7,#1              ;nada biasa tanpa kres
            mov   dptr,#doc
            lcall printstring2
            mov   nada,#78
            acall delayku
            ajmp  ulang3x

menu1bal2x:
            mov   r0,tombol          ;idem menu1ball
            cjne  r0,#'2',menu1bal3x
            mov   r1,simpan#
            cjne  r1,#0AAh,kres2x
            mov   r7,#1
            mov   dptr,#dodk
            lcall printstring2
            mov   simpan#,#00h
            mov   nada,#66
            acall delayku
            ajmp  ulang3x

kres2x      mov   r7,#1
            mov   dptr,#dod
            lcall printstring2
            mov   nada,#71
            acall delayku
            ajmp  ulang3x

menu1bal3x: mov   r0,tombol
            cjne  r0,#'3',menu1bal4x
            mov   r7,#1
            mov   dptr,#doe
            lcall printstring2
            mov   nada,#62
            acall delayku
            ajmp  ulang3x

menu1bal4x:
```

```
            mov   r0,tombol
            cjne  r0,#'4',menulbal5x
            mov   r1,simpan#
            cjne  r1,#0AAh,kres3x
            mov   r7,#1
            mov   dptr,#dofk
            lcall printstring2
            mov   simpan#,#00h
            mov   nada,#55
            acall delayku
            ajmp  ulang3x

kres3x:     mov   r7,#1
            mov   dptr,#dof
            lcall printstring2
            mov   nada,#59
            acall delayku
            ajmp  ulang3x

menulbal5x:
            mov   r0,tombol
            cjne  r0,#'5',menulbal6x
            mov   r1,simpan#
            cjne  r1,#0AAh,kres4x
            mov   r7,#1
            mov   dptr,#dogk
            lcall printstring2
            mov   simpan#,#00h
            mov   nada,#49
            acall delayku
            ajmp  ulang3x

kres4x      mov   r7,#1
            mov   dptr,#dog
            lcall printstring2
            mov   nada,#52
            acall delayku
            ajmp  ulang3x

menulbal6x:
            mov   r0,tombol
            cjne  r0,#'6',menulbal7x
            mov   r1,simpan#
            cjne  r1,#0AAh,kres5x
            mov   r7,#1
            mov   dptr,#doak
            lcall printstring2
            mov   simpan#,#00h
            mov   nada,#44
            acall delayku
            ajmp  ulang3x

kres5x:     mov   r7,#1
            mov   dptr,#doa
            lcall printstring2
            mov   nada,#46
            acall delayku
```

```
            ajmp  ulang3x

menu1bal7x:
            mov   r0,tombol
            cjne  r0,#'7',menu1bal8x
            mov   r7,#1
            mov   dptr,#dob
            lcall printstring2
            mov   nada,#41
            acall delayku
            ajmp  ulang3x

menu1bal8x:
            mov   r0,tombol
            cjne  r0,#'8',menu1bal9x
            mov   r7,#1
            mov   dptr,#doct
            lcall printstring2
            mov   nada,#39
            acall delayku
menu1bal9x: ajmp  ulang3x
            ret
;-----------------------------------------
menu4
;JOYFUL, JOYFUL WE ADORE THEE -beethoven-
;MM = 116
;-----------------------------------------
            mov   r6,#1
            mov   r7,#1
            mov   dptr,#judul1
            lcall printstring1
            mov   dptr,#judul2
            lcall printstring2
            mov   beat,#75
            mov   r3,#80
            mov   nada,#41          ;mi
            lcall oke
            lcall delaybeat
            mov   r3,#80
            mov   nada,#41    ;mi
            lcall oke
            lcall delaybeat
            mov   r3,#80
            mov   nada,#39    ;fa
            lcall oke
            lcall delaybeat
            mov   r3,#80
            mov   nada,#35    ;sol
            lcall oke
            lcall delaybeat
            mov   r3,#80
            mov   nada,#35    ;sol
            lcall oke
            lcall delaybeat
            mov   r3,#80
            mov   nada,#39    ;fa
            lcall oke
```

```
lcall delaybeat
mov   r3,#80
mov   nada,#41     ;mi
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46     ;re
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#52     ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#52     ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46     ;re
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#41     ;mi
lcall oke
lcall delaybeat

mov   r3,#200
mov   nada,#41     ;mi 1 1/2
lcall oke
mov   r3,#100
mov   nada,#41     ;mi 1 1/2
lcall oke

mov   beat,#37
lcall delaybeat
mov   beat,#75

mov   r3,#40
mov   nada,#46     ;re 1/2
lcall oke
mov   r3,#80
mov   nada,#46     ;re
lcall oke

MOV   beat,#37
lcall delaybeat
MOV   beat,#75

mov   r3,#80
mov   nada,#46     ;re
lcall oke

lcall delaybeat
lcall delaybeat

mov   r3,#80
mov   nada,#41          ;mi
```

```
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#41    ;mi
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#39    ;fa
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#35    ;sol
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#35    ;sol
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#39    ;fa
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#41    ;mi
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46    ;re
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#52    ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#52    ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46    ;re
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#41    ;mi
lcall oke
lcall delaybeat

mov   r3,#200
mov   nada,#46    ;re 1 1/2
lcall oke
mov   r3,#200
mov   nada,#46    ;re 1 1/2
lcall oke

mov   beat,#37
lcall delaybeat
mov   beat,#75
```

```
mov   r3,#40
mov   nada,#52    ;do 1/2
lcall oke
mov   r3,#80
mov   nada,#52    ;do
lcall oke

MOV   beat,#37
lcall delaybeat
MOV   beat,#75

mov   r3,#80
mov   nada,#52    ;do
lcall oke
lcall delaybeat
lcall delaybeat

mov   r3,#80
mov   nada,#46    ;re
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46    ;re
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#41    ;mi
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#52    ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46    ;re
lcall oke
lcall delaybeat

mov   r3,#40
mov   nada,#41    ;mi 1/2
lcall oke

MOV   beat,#37
lcall delaybeat
MOV   beat,#75

mov   r3,#40
mov   nada,#39    ;FA 1/2
lcall oke

MOV   beat,#37
lcall delaybeat
MOV   beat,#75

mov   r3,#80
mov   nada,#41    ;mi
```

```
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#52     ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46     ;re
lcall oke
lcall delaybeat

mov   r3,#40
mov   nada,#41     ;mi 1/2
lcall oke

MOV   beat,#37
lcall delaybeat
MOV   beat,#75

mov   r3,#30
mov   nada,#39     ;FA  1/2
lcall oke

MOV   beat,#37
lcall delaybeat
MOV   beat,#75

mov   r3,#80
mov   nada,#41     ;mi
lcall oke

lcall delaybeat

mov   r3,#80
mov   nada,#46     ;re
lcall oke
lcall delaybeat

mov   r3,#80
mov   nada,#52     ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46     ;re
lcall oke
lcall delaybeat

mov   r3,#200
mov   nada,#71     ;sol r 2x
lcall oke
mov   r3,#200
mov   nada,#71     ;sol r 2x
lcall oke

lcall delaybeat
mov   r3,#80
mov   nada,#41     ;mi
```

```
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#41    ;mi
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#39    ;FA
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#35    ;SOL
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#35    ;SOL
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#39    ;FA
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#41    ;mi
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46    ;re
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#52    ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#52    ;do
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#46    ;re
lcall oke
lcall delaybeat
mov   r3,#80
mov   nada,#41    ;mi
lcall oke
lcall delaybeat

mov   r3,#200
mov   nada,#46    ;re  1 1/2
lcall oke
mov   r3,#200
mov   nada,#46    ;re  1 1/2
lcall oke

mov   beat,#37
lcall delaybeat
mov   beat,#75
```

```
        mov   r3,#40
        mov   nada,#52     ;DO
        lcall oke

        MOV   beat,#37
        lcall delaybeat
        MOV   beat,#75

        mov   r3,#120
        mov   nada,#52     ;do
        lcall oke
        ret
setawal:
        mov   simpanmenu,#00h   ;data menu yg dipilih
        mov   simpantempo,#01h  ;xxxx
        mov   simpannada,#01h   ;xxxx
        mov   simpan#,#00h      ;data tombol kres ditekan
        mov   ie,#00h
        mov   sp,#60h           ;push-pop max 15X !!! 60-7f
        lcall init.lcd
        lcall init.ppi
        MOV   A,#0
        MOV   DPTR,#portc
        MOVX  @DPTR,A
        mov   r6,#1             ;kolom1
        mov   r7,#1             ;idem
        mov   dptr,#siltek
        lcall printstring1            ;baris1
        mov   dptr,#sil
        lcall printstring2            ;baris2
ulang1
        lcall scankeypad
        mov   r0,tombol         ;move tombol -> R0
        cjne  r0,#'M',menut1    ;bandingkan isi r0 dengan 'm'
        lcall choosemenu        ;jika ya panggil choosemenu
        ajmp  ulang1            ;jika tidak lompat ulang1
menut1:
        mov   r0,tombol
        cjne  r0,#'C',ulang1
        ajmp  setawal
```

eatures

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 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
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

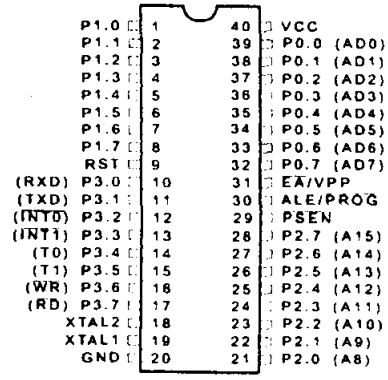
escription

e AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K tes of Flash Programmable and Erasable Read Only Memory (PEROM). The vice is manufactured using Atmel's high density nonvolatile memory technology d is compatible with the industry standard MCS-51™ instruction set and pinout. The -chip Flash allows the program memory to be reprogrammed in-system or by a conntional nonvolatile memory programmer. By combining a versatile 8-bit CPU with ash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which ovides a highly flexible and cost effective solution to many embedded control applitions.

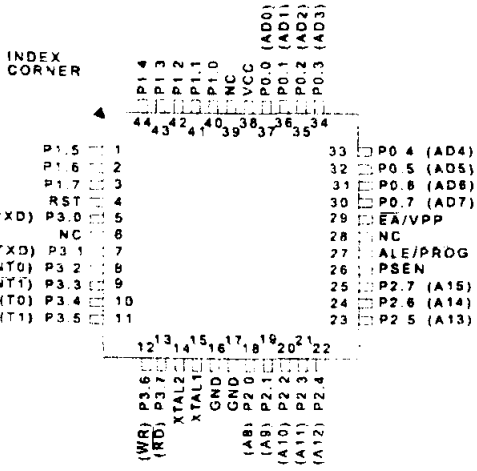
(continued)

in Configurations

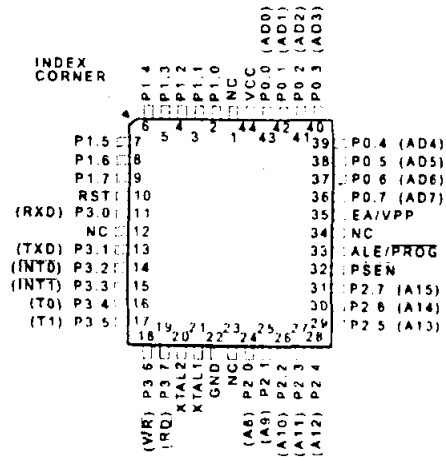
PDIP



PQFP/TQFP



PLCC

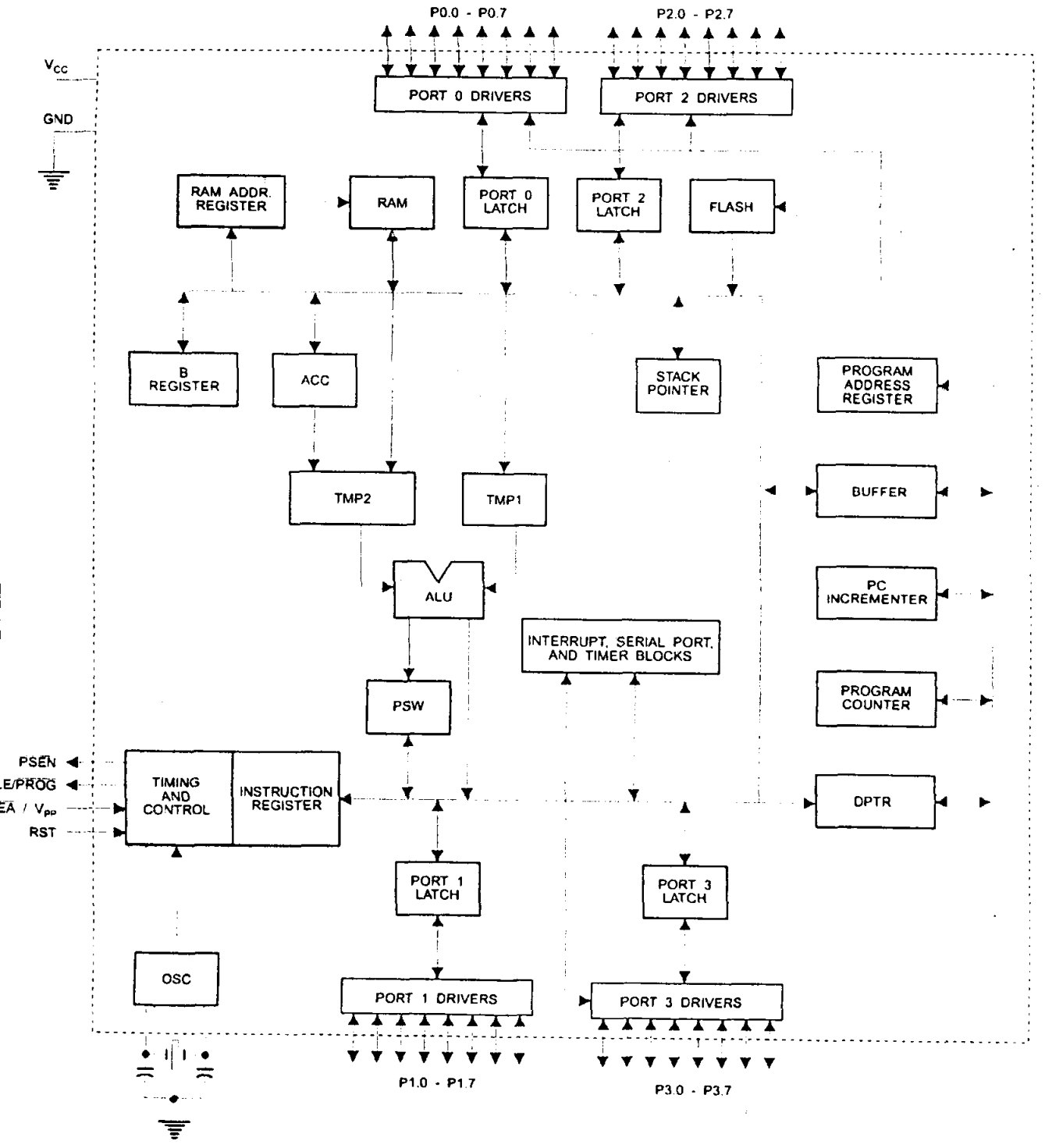


8-Bit Microcontroller with 4K Bytes Flash

AT89C51

0265F-A-12/97

ock Diagram


P0.0 - P0.7
P2.0 - P2.7
V_{CC}
GND
PORT 0 DRIVERS
PORT 2 DRIVERS
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
TMP2
TMP1
BUFFER
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT, AND TIMER BLOCKS
PSW
PROGRAM COUNTER
PSEN
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
RST
PORT 1 LATCH
PORT 3 LATCH
OSC
PORT 1 DRIVERS
PORT 3 DRIVERS
P1.0 - P1.7
P3.0 - P3.7

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, 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 AT89C51 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 hardware reset.

Pin Description

V_{CC}
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 may 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 pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1
Port 1 is an 8-bit bidirectional I/O port with internal pullups. 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 pullups 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 pullups.

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

Port 2
Port 2 is an 8-bit bidirectional I/O port with internal pullups. 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 pullups 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 pullups.

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 it uses strong internal pullups 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 pullups. 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 pullups 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 pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

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)

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

RST
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$
Address Latch Enable 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 is the read strobe to external program memory.

When the AT89C51 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.

$\overline{EA}$/V_{PP}

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, for parts that require 12-volt V_{PP}.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

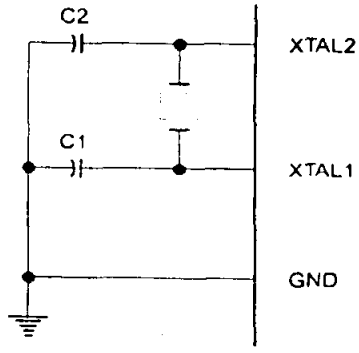
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. 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.

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 functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

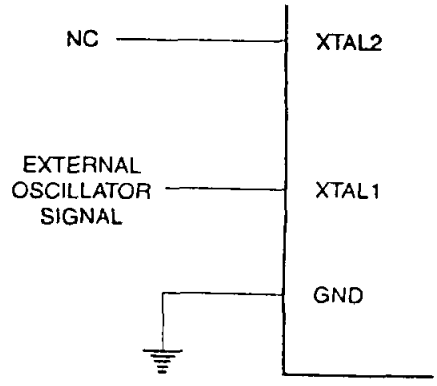
It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port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

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	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

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. The only exit from power down is a hardware reset. 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.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

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. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
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, EA is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, 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 for the high-voltage programming mode.
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 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 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 lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased 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.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10-ms PROG pulse.

Figure 3. Programming the Flash

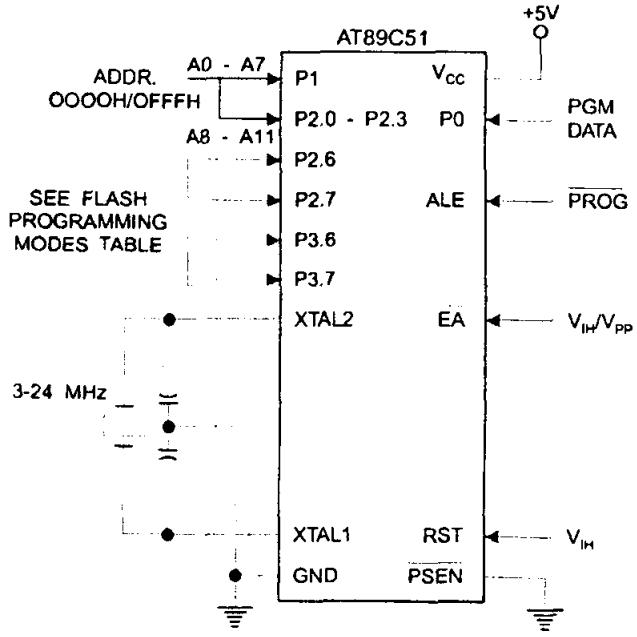
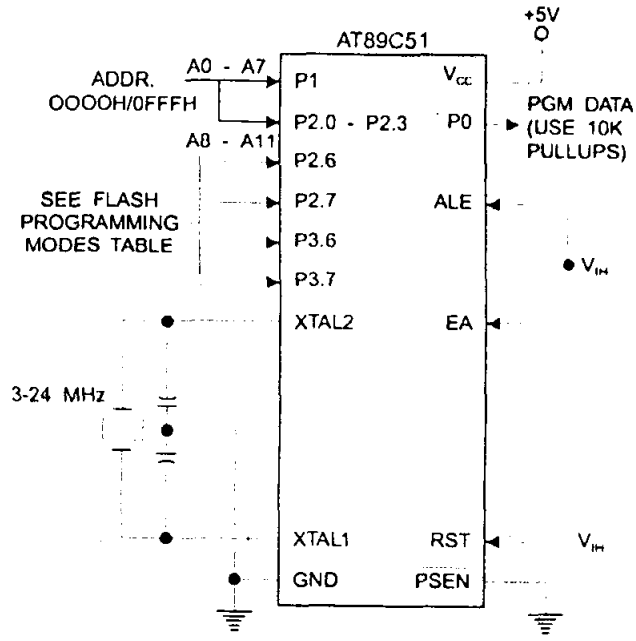


Figure 4. Verifying the Flash



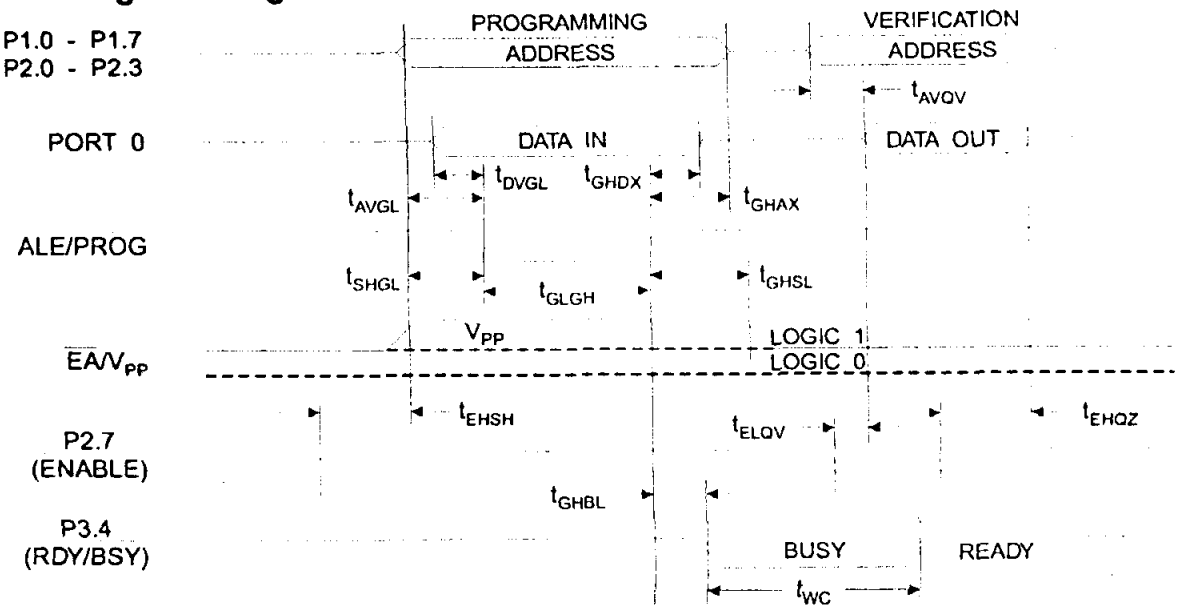
Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

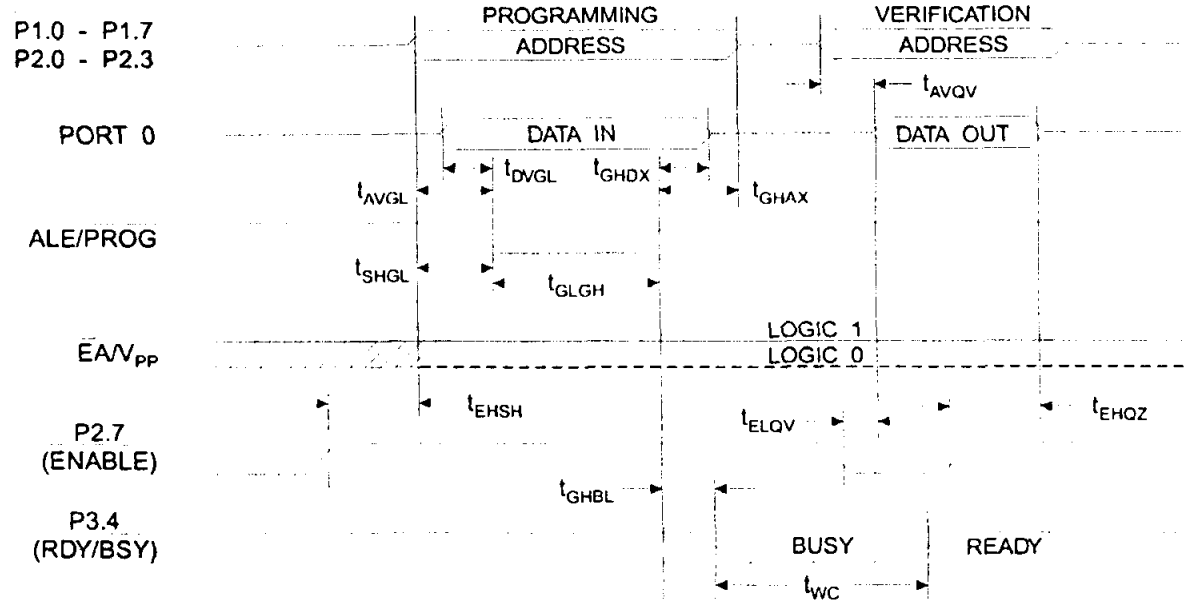
Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	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}(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	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		2.0	ms

Note: 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms - High Voltage Mode (V_{PP} = 12V)



Flash Programming and Verification Waveforms - Low Voltage Mode (V_{PP} = 5V)



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	6.6V
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

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 20% (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, VCC = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	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		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode[2]	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	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.

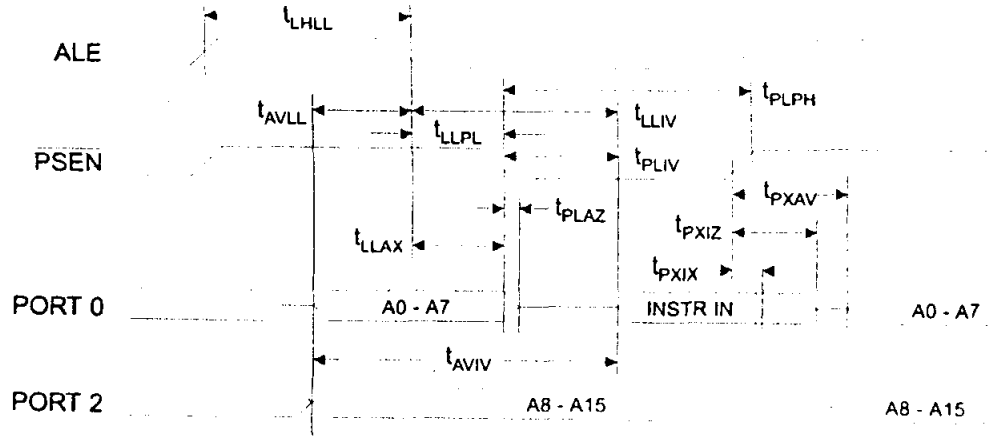
AC Characteristics

(Under Operating Conditions; Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for all other outputs = 80 pF)

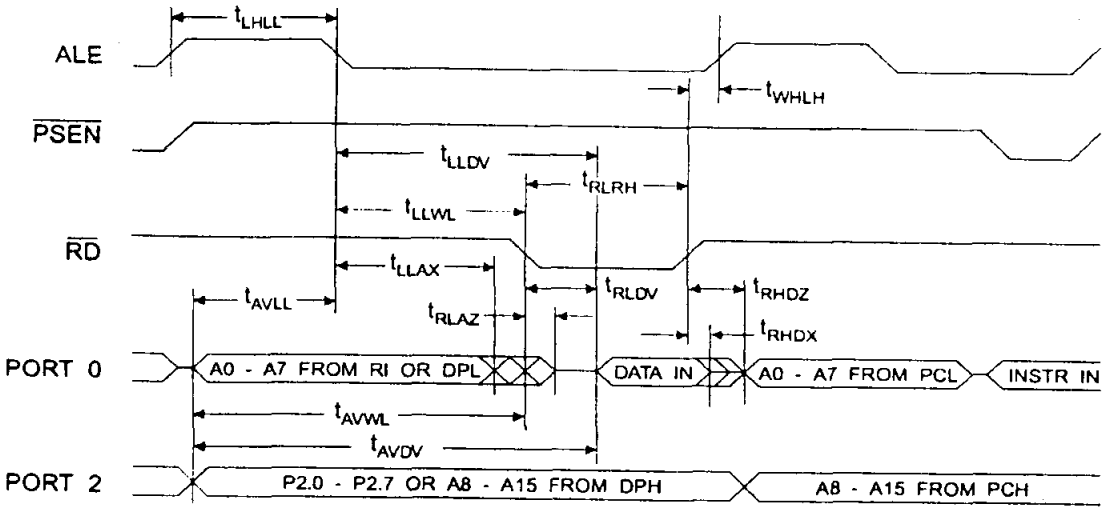
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{CLCL}-13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{CLCL}-20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}-65$	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		$t_{CLCL}-13$		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}-20$		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}-45$	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		$t_{CLCL}-10$	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		$t_{CLCL}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}-55$	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		$t_{CLCL}-20$		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}-120$		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		$t_{CLCL}-20$		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	$t_{CLCL}-20$	$t_{CLCL}+25$	ns

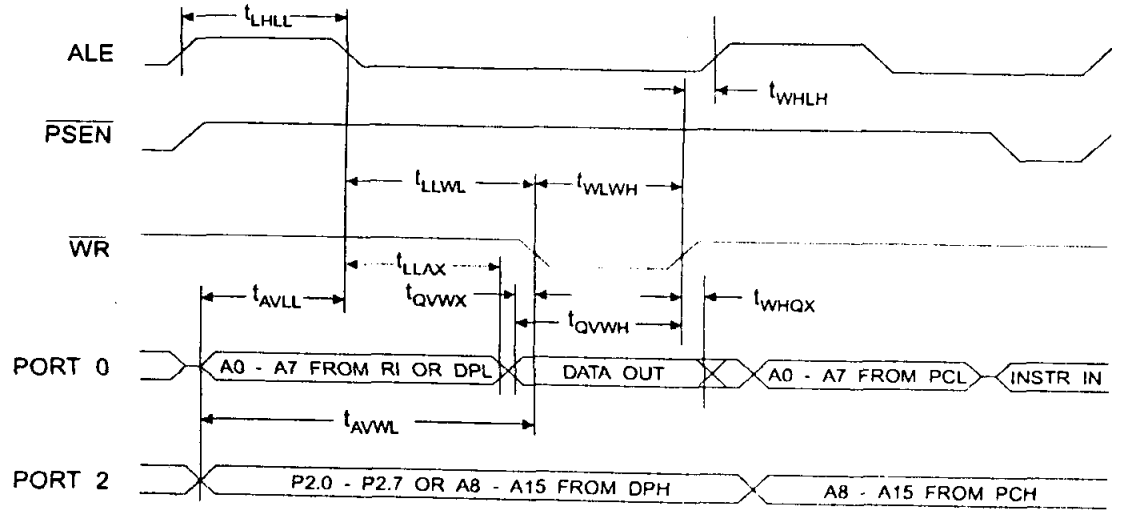
External Program Memory Read Cycle



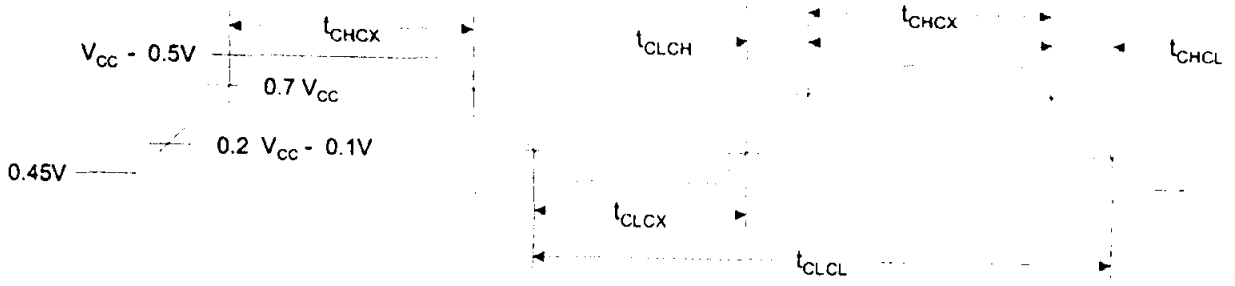
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

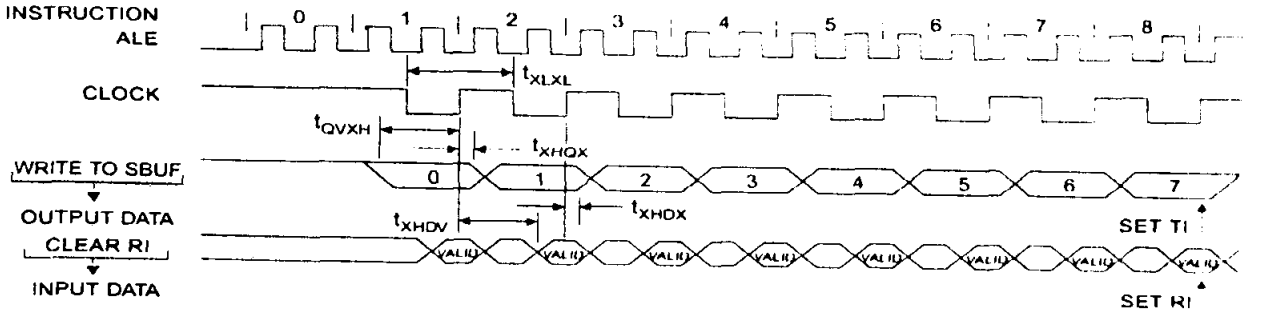
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

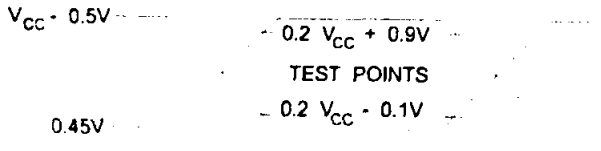
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

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		$2t_{CLCL}$-117		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

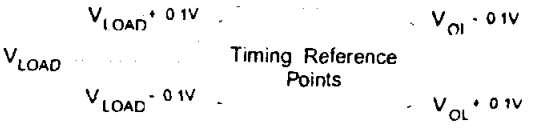


AC Testing Input/Output Waveforms[1]



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 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)
		AT89C51-12JC	44J	
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial (-40°C to 85°C)
		AT89C51-12JI	44J	
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
		AT89C51-12AA	44A	Automotive (-40°C to 105°C)
		AT89C51-12JA	44J	
		AT89C51-12PA	40P6	
		AT89C51-12QA	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial (0°C to 70°C)
		AT89C51-16JC	44J	
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial (-40°C to 85°C)
		AT89C51-16JI	44J	
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
		AT89C51-16AA	44A	Automotive (-40°C to 105°C)
		AT89C51-16JA	44J	
		AT89C51-16PA	40P6	
		AT89C51-16QA	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)
		AT89C51-20JC	44J	
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial (-40°C to 85°C)
		AT89C51-20JI	44J	
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	5V ± 20%	AT89C51-24AC	44A	Commercial
		AT89C51-24JC	44J	(0°C to 70°C)
		AT89C51-24PC	44P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial
		AT89C51-24JI	44J	(-40°C to 85°C)
		AT89C51-24PI	44P6	
		AT89C51-24QI	44Q	

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

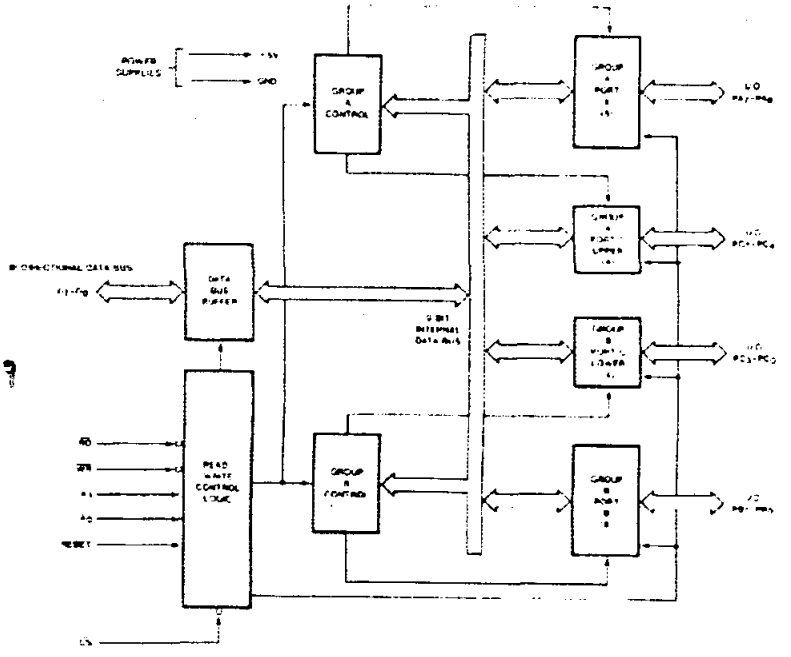
82C55A
CHMOS PROGRAMMABLE PERIPHERAL INTERFACE

- Compatible with all Intel and Most Other Microprocessors
- High Speed, "Zero Wait State" Operation with 8 MHz 8086/88 and 80186/188
- 24 Programmable I/O Pins
- Low Power CHMOS
- Completely TTL Compatible
- Control Word Read-Back Capability
- Direct Bit Set/Reset Capability
- 2.5 mA DC Drive Capability on all I/O Port Outputs
- Available in 40-Pin DIP and 44-Pin PLCC
- Available in EXPRESS
  - Standard Temperature Range
  - Extended Temperature Range

The Intel 82C55A is a high-performance, CHMOS version of the industry standard 8255A general purpose programmable I/O device which is designed for use with all Intel and most other microprocessors. It provides 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The 82C55A is pin compatible with the NMOS 8255A and 8255A-5.

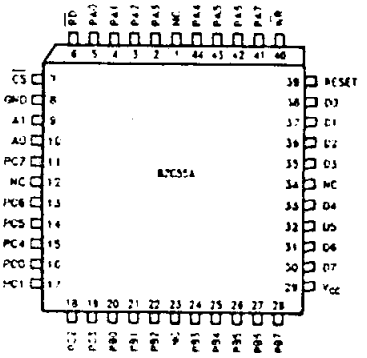
In MODE 0, each group of 12 I/O pins may be programmed in sets of 4 and 8 to be inputs or outputs. In MODE 1, each group may be programmed to have 8 lines of input or output. 3 of the remaining 4 pins are used for handshaking and interrupt control signals. MODE 2 is a strobed bi-directional bus configuration.

The 82C55A is fabricated on Intel's advanced CHMOS III technology which provides low power consumption with performance equal to or greater than the equivalent NMOS product. The 82C55A is available in 40-pin DIP and 44-pin plastic leaded chip carrier (PLCC) packages.

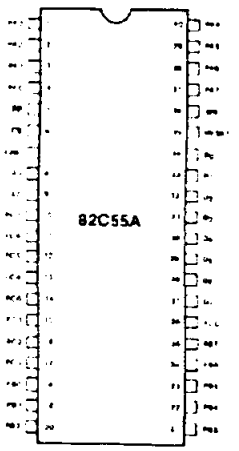


231256-1

Figure 1. 82C55A Block Diagram



231256-31



231256-2

Figure 2. 82C55A Pinout
Diagrams are for pin reference only. Package sizes are not to scale.

Table 1. Pin Description

Symbol	Pin Number Dip	Pin Number PLCC	Type	Name and Function
PA_{3-0}	1-4	2-5	I/O	**PORT A, PINS 0-3:** Lower nibble of an 8-bit data output latch/buffer and an 8-bit data input latch.
$\overline{RD}$	5	6	I	**READ CONTROL:** This input is low during CPU read operations.
$\overline{CS}$	6	7	I	**CHIP SELECT:** A low on this input enables the 82C55A to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and WR are ignored otherwise.
GND	7	8		**System Ground**
A_{1-0}	8-9	9-10	I	**ADDRESS:** These input signals, in conjunction $\overline{RD}$ and $\overline{WR}$, control the selection of one of the three ports or the control word registers.

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (Read)
0	0	0	1	0	Port A - Data Bus
0	1	0	1	0	Port B - Data Bus
1	0	0	1	0	Port C - Data Bus
1	1	0	1	0	Control Word - Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus - Port A
0	1	1	0	0	Data Bus - Port B
1	0	1	0	0	Data Bus - Port C
1	1	1	0	0	Data Bus - Control
					Disable Function
X	X	X	X	1	Data Bus - 3 - State
X	X	1	1	0	Data Bus - 3 - State

PC_{7-4}	10-13	11,13-15	I/O	**PORT C, PINS 4-7:** Upper nibble of an 8-bit data output latch/buffer and an 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.
PC_{0-3}	14-17	16-19	I/O	**PORT C, PINS 0-3:** Lower nibble of Port C.
PB_{0-7}	18-25	20-22, 24-28	I/O	**PORT B, PINS 0-7:** An 8-bit data output latch/buffer and an 8-bit data input buffer.
V_{CC}	26	29		**SYSTEM POWER:** + 5V Power Supply.
D_{7-0}	27-34	30-33, 35-38	I/O	**DATA BUS:** Bi-directional, tri-state data bus lines, connected to system data bus.
RESET	35	39	I	**RESET:** A high on this input clears the control register and all ports are set to the input mode.
$\overline{WR}$	36	40	I	**WRITE CONTROL:** This input is low during CPU write operations.
PA_{7-4}	37-40	41-44	I/O	**PORT A, PINS 4-7:** Upper nibble of an 8-bit data output latch/buffer and an 8-bit data input latch.
NC		1, 12, 23, 34		No Connect

82C55A FUNCTIONAL DESCRIPTION

General

The 82C55A is a programmable peripheral interface device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 82C55A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7–C4)
Control Group B - Port B and Port C lower (C3–C0)

The control word register can be both written and read as shown in the address decode table in the pin descriptions. Figure 6 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

Ports A, B, and C

The 82C55A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 82C55A.

Port A. One 8-bit data output latch/buffer and one 8-bit input latch buffer. Both "pull-up" and "pull-down" bus hold devices are present on Port A.

Port B. One 8-bit data input/output latch/buffer. Only "pull-up" bus hold devices are present on Port B.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B. Only "pull-up" bus hold devices are present on Port C.

See Figure 4 for the bus-hold circuit configuration for Port A, B, and C.

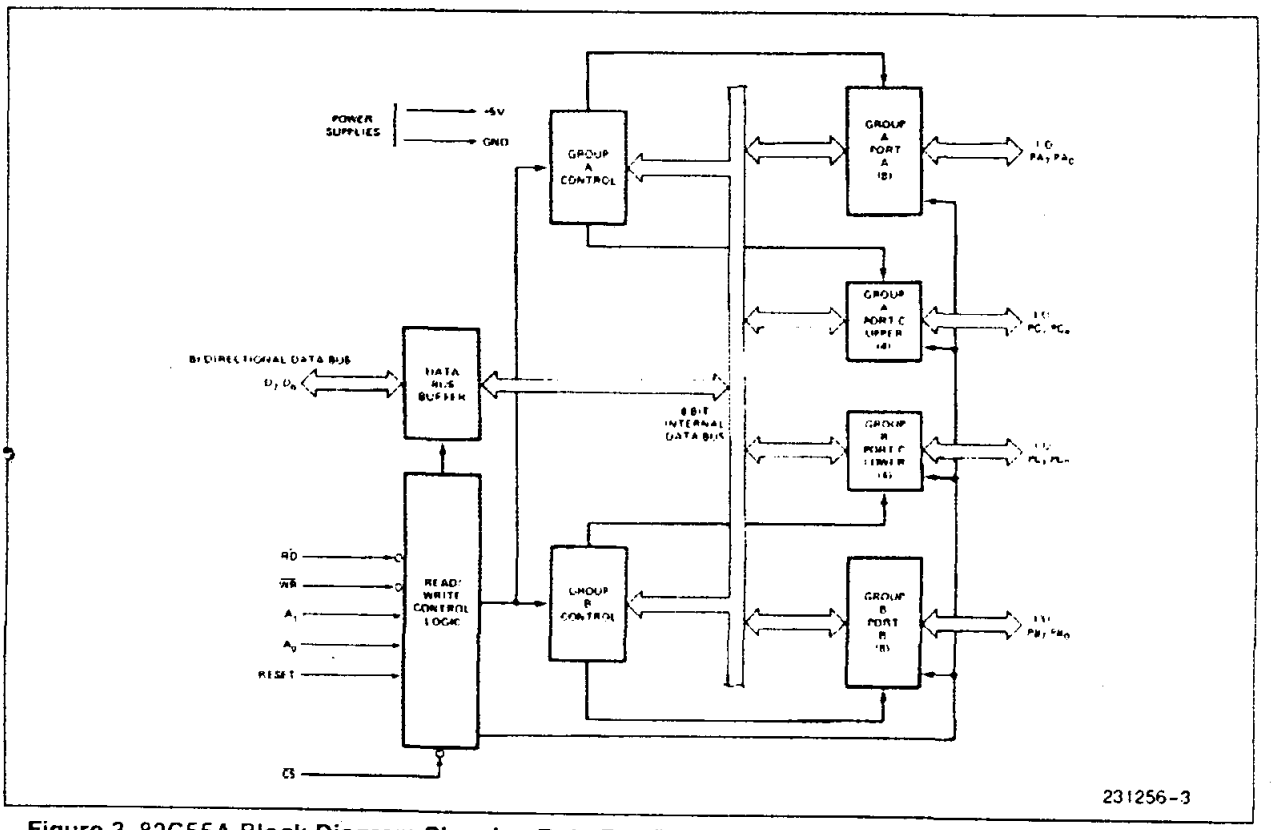
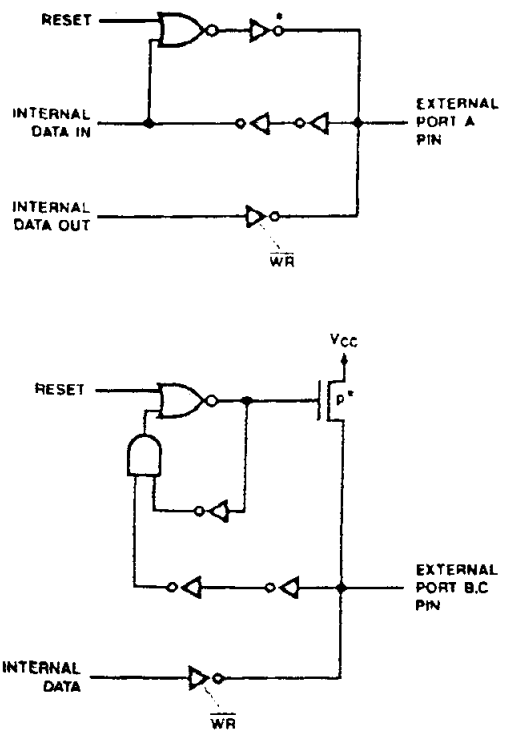


Figure 3. 82C55A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions



***NOTE:**
Port pins loaded with more than 20 pF capacitance may not have their logic level guaranteed following a hardware reset.

Figure 4. Port A, B, C, Bus-hold Configuration

82C55A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 — Basic input/output
Mode 1 — Strobed Input/output
Mode 2 — Bi-directional Bus

When the reset input goes "high" all ports will be set to the input mode with all 24 port lines held at a logic "one" level by the internal bus hold devices (see Figure 4 Note). After the reset is removed the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need for pullup or pulldown devices in "all CMOS" designs. During the execution of the system program, any of the other modes may be selected by using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

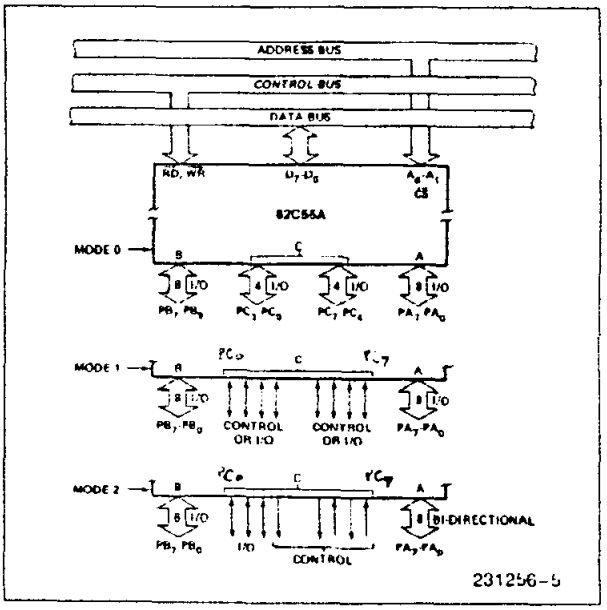


Figure 5. Basic Mode Definitions and Bus Interface

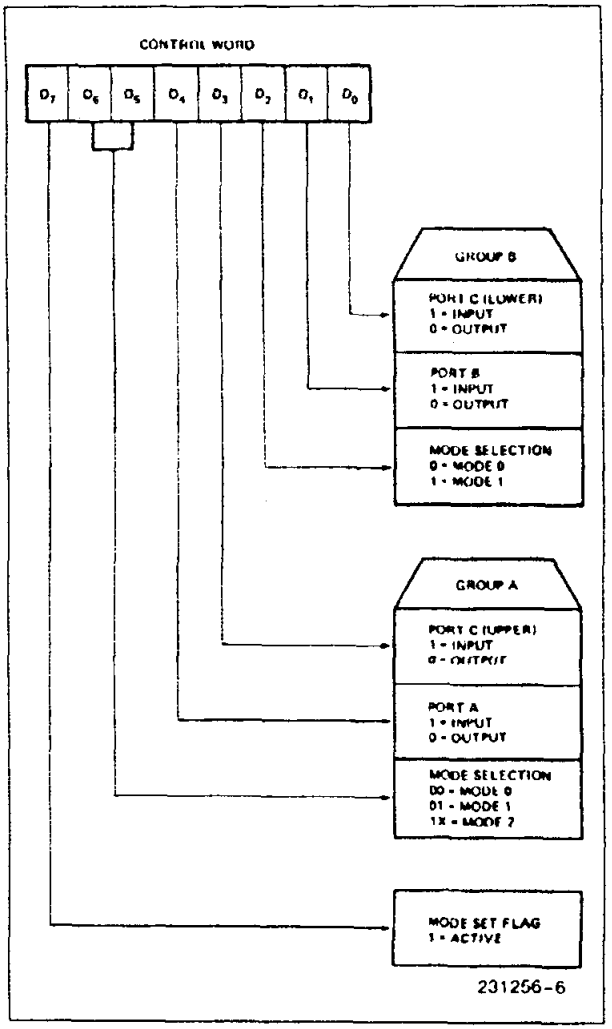


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

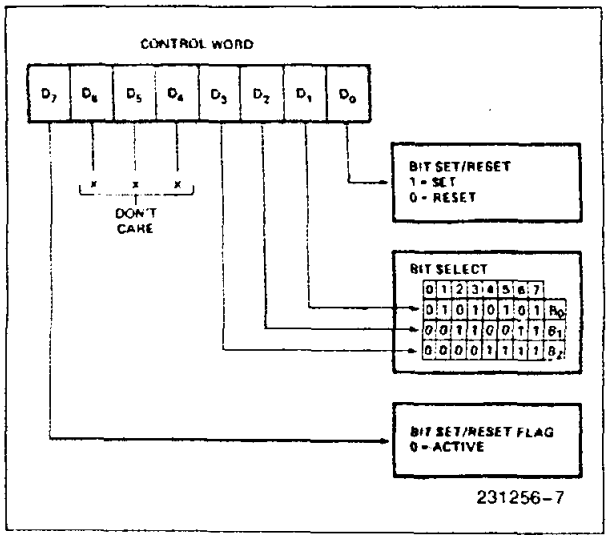


Figure 7. Bit Set/Reset Format

Interrupt Control Functions

When the 82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is SET—Interrupt enable
(BIT-RESET)—INTE is RESET—Interrupt disable

Note:
All Mask flip-flops are automatically reset during mode selection and device Reset.

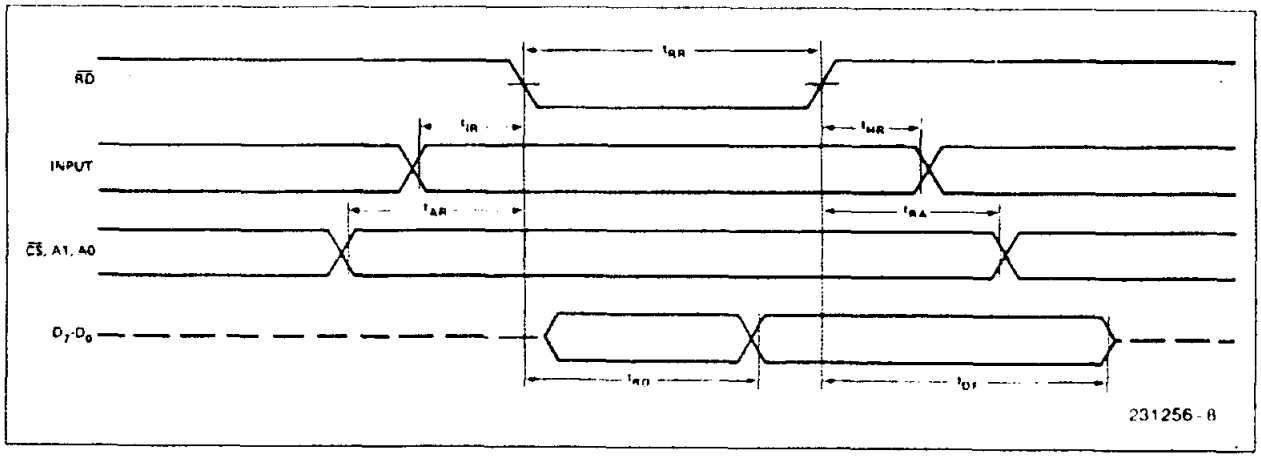
Operating Modes

Mode 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

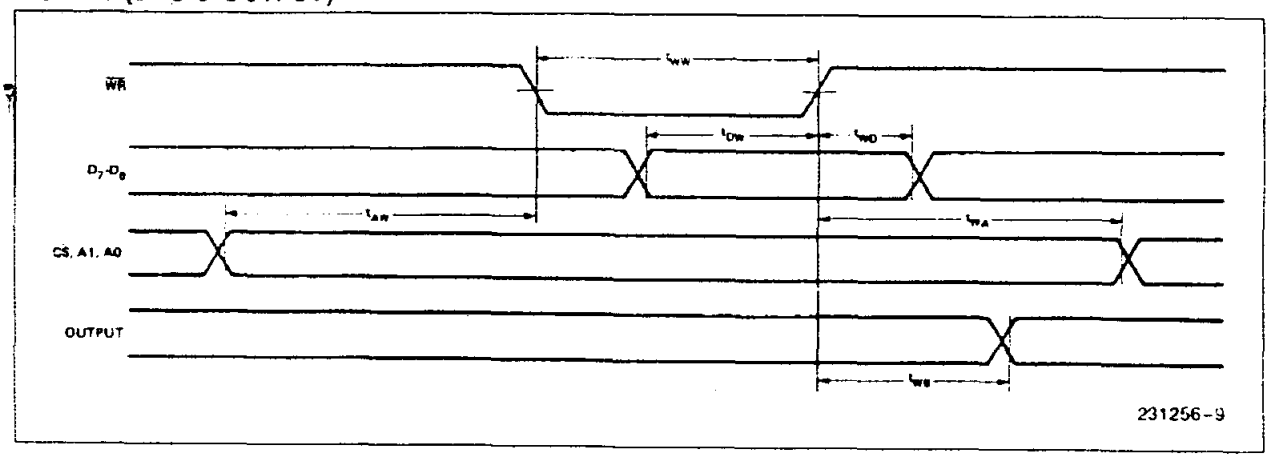
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)



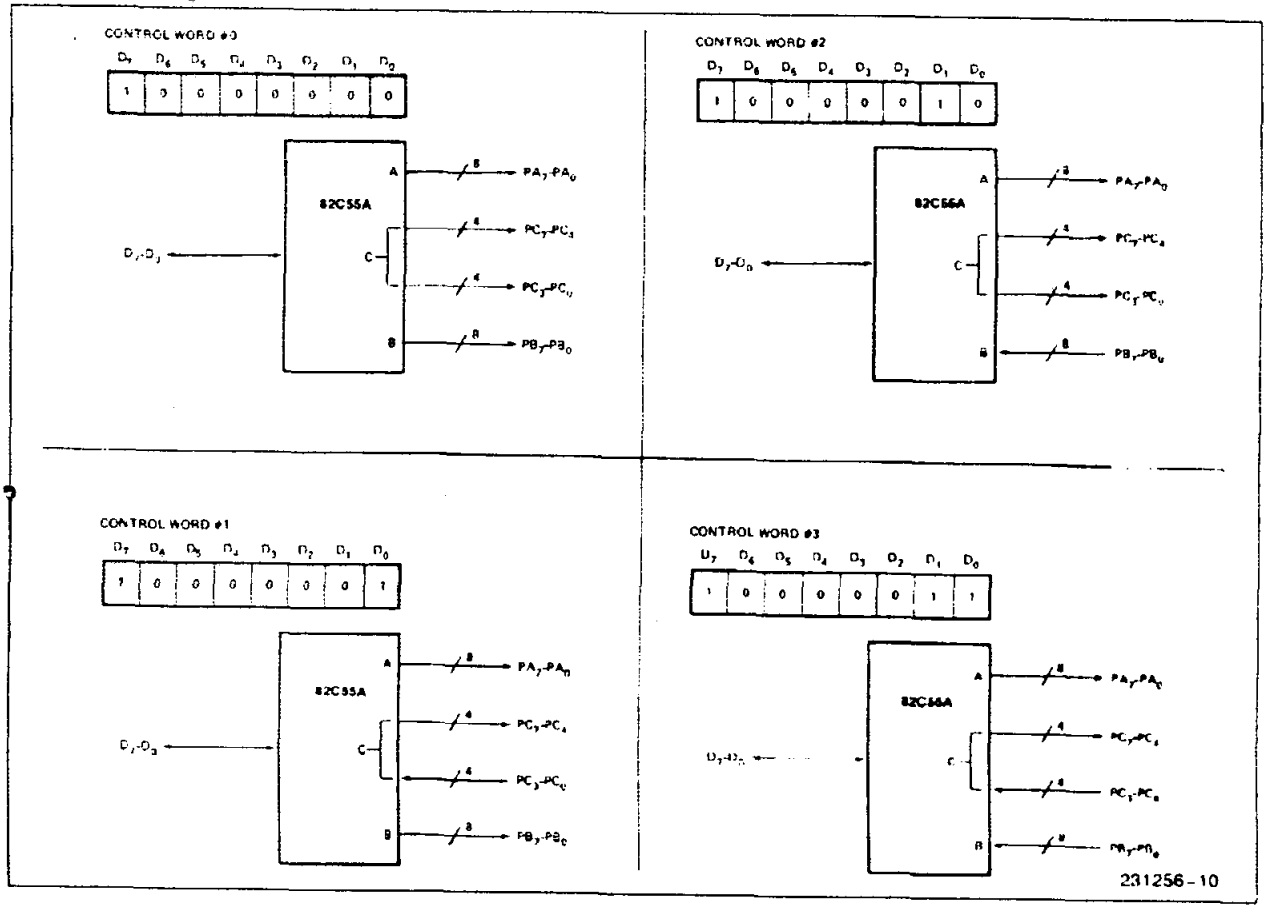
MODE 0 (BASIC OUTPUT)



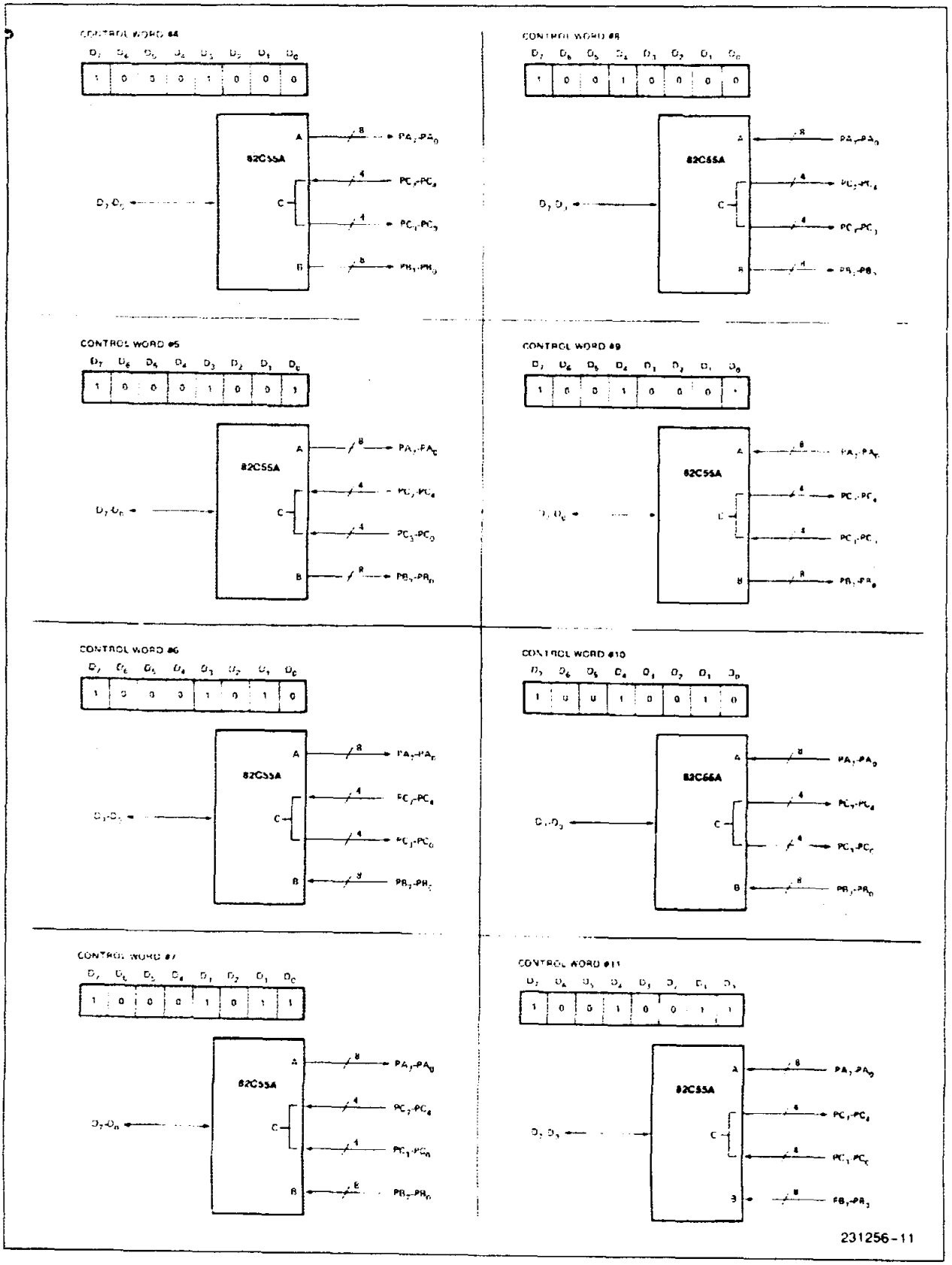
MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations

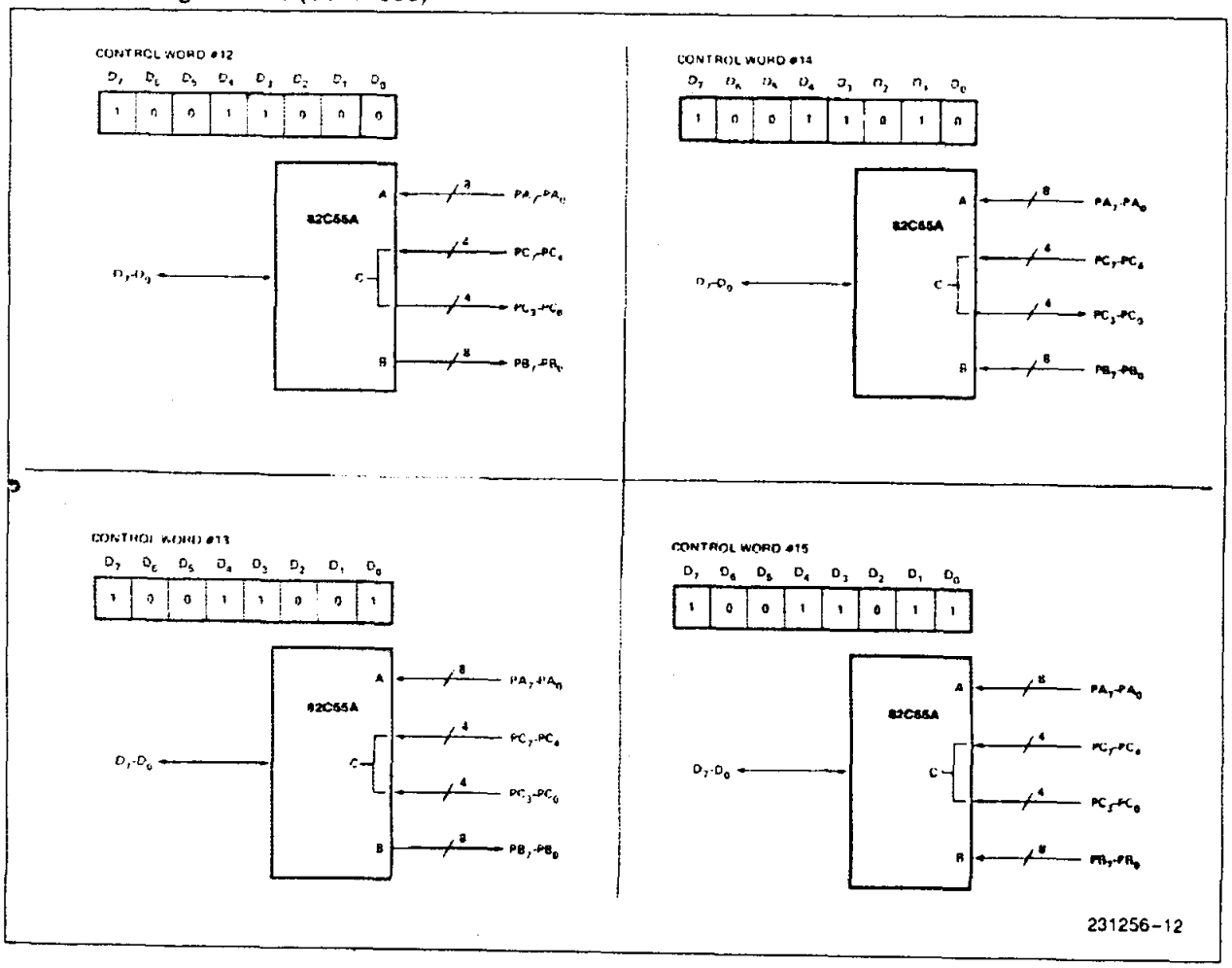


MODE 0 Configurations (Continued)



231256-11

MODE 0 Configurations (Continued)



Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic functional Definitions:

- Two Groups (Group A and Group B).
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by $\overline{STB}$ input being low and is reset by the rising edge of the $\overline{RD}$ input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

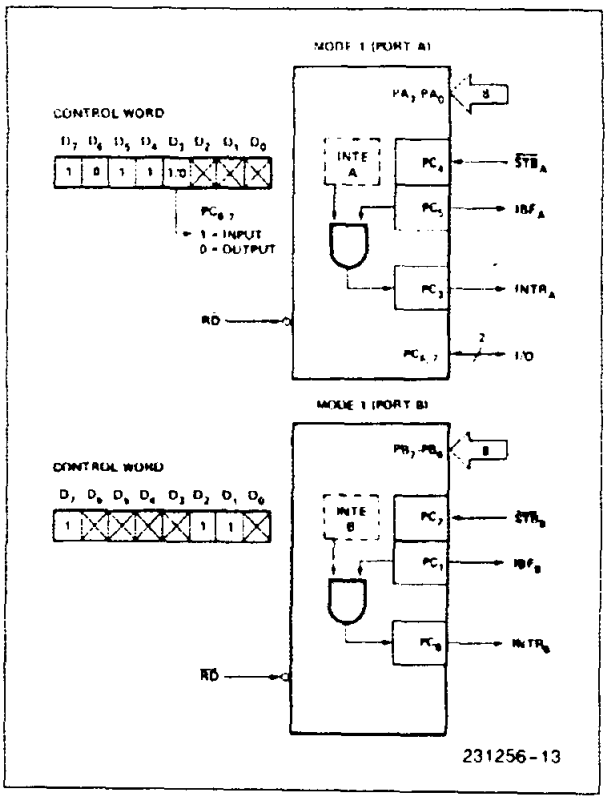


Figure 8. MODE 1 Input

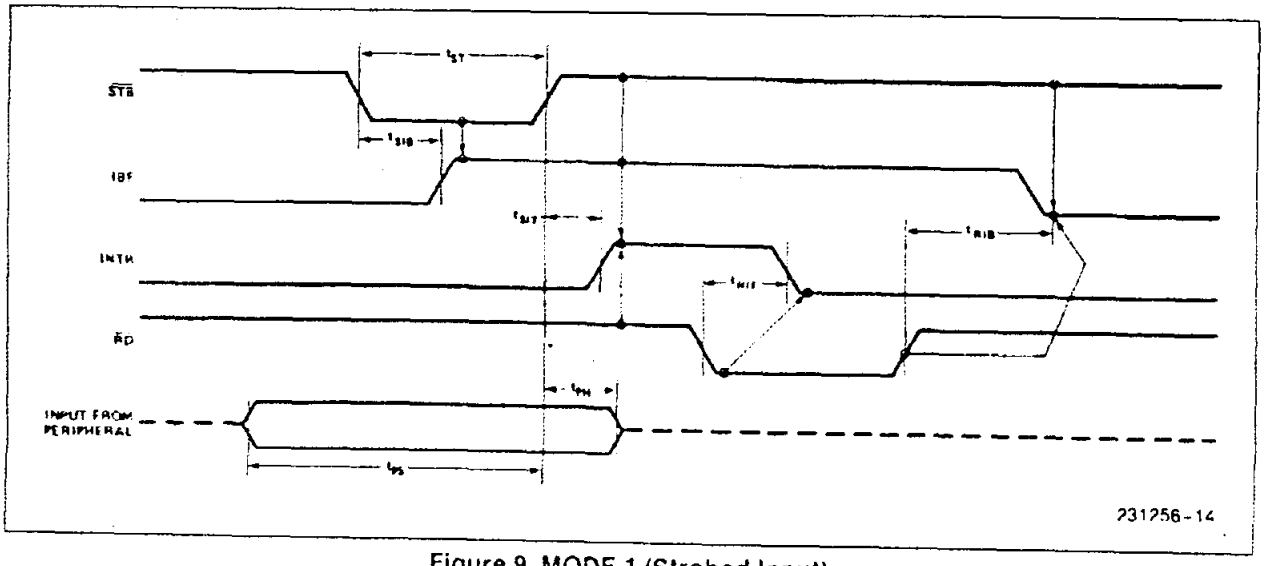


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ Input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 82C55A that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

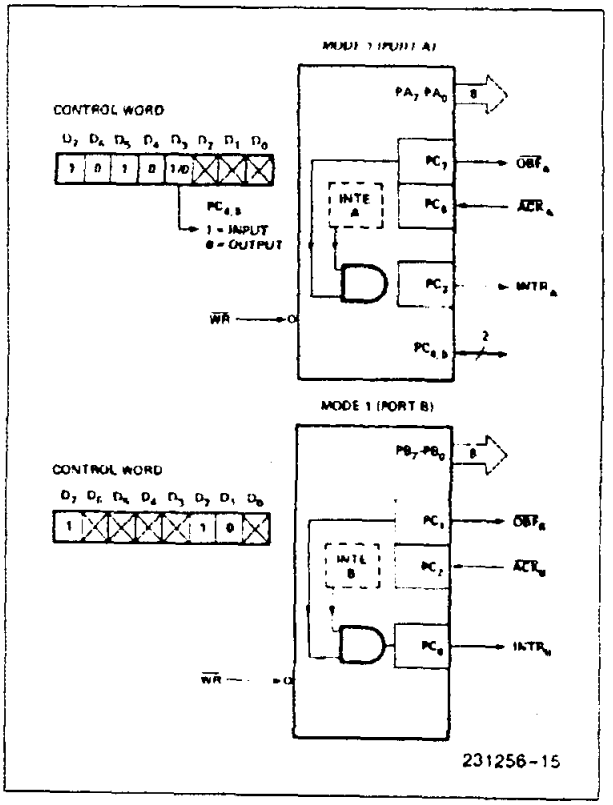


Figure 10. MODE 1 Output

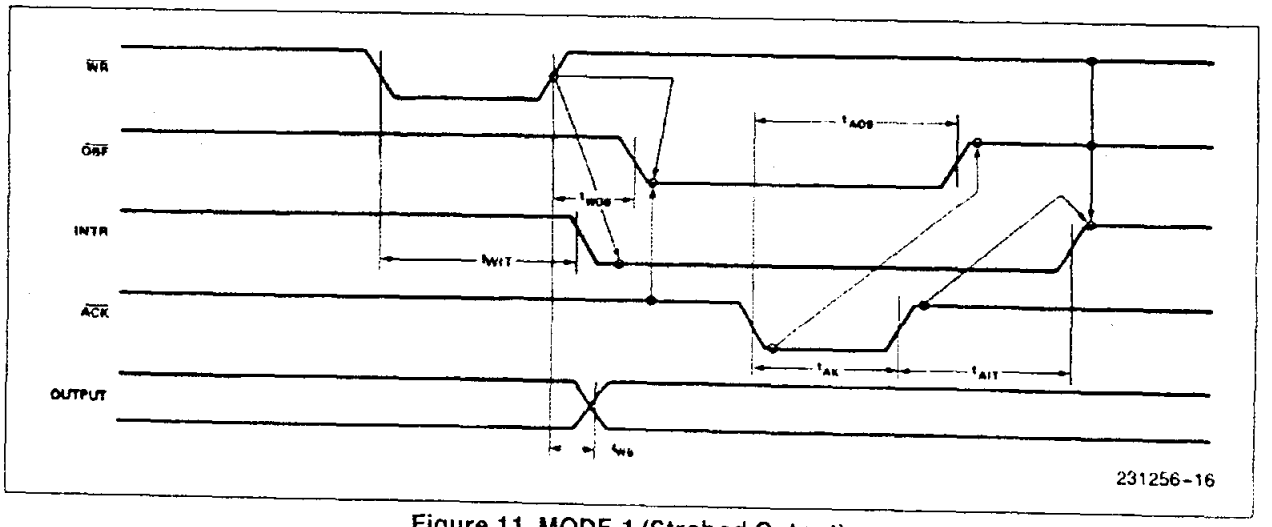


Figure 11. MODE 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

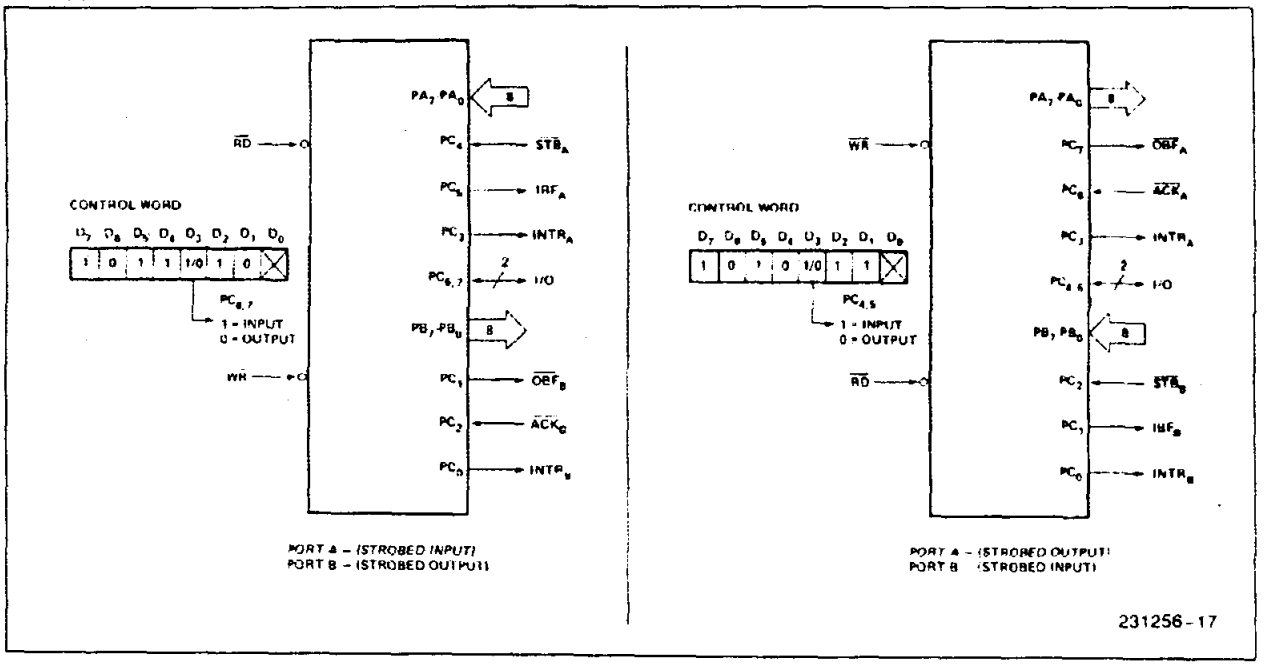


Figure 12. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus port (Port A) and a 5-bit control port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with $\overline{OBF}$). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

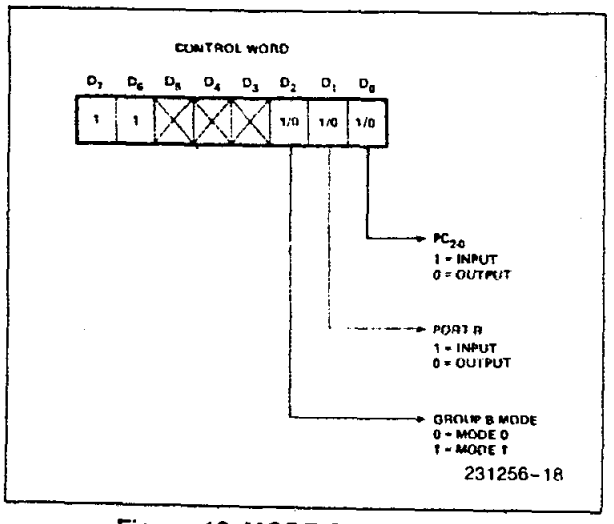


Figure 13. MODE Control Word

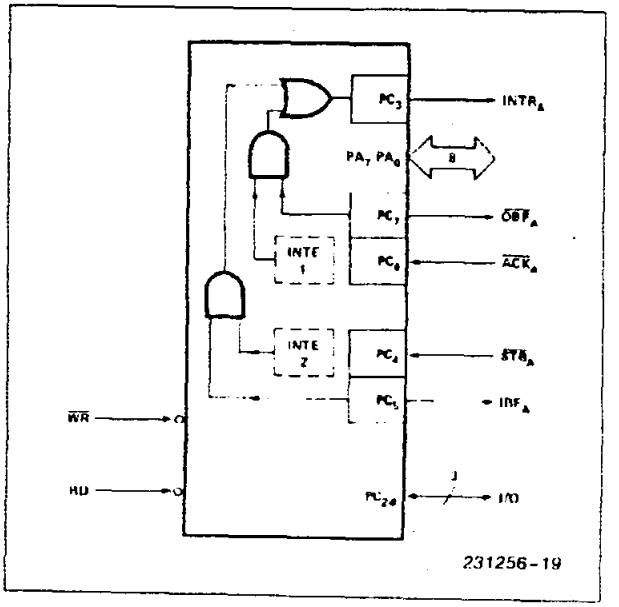


Figure 14. MODE 2

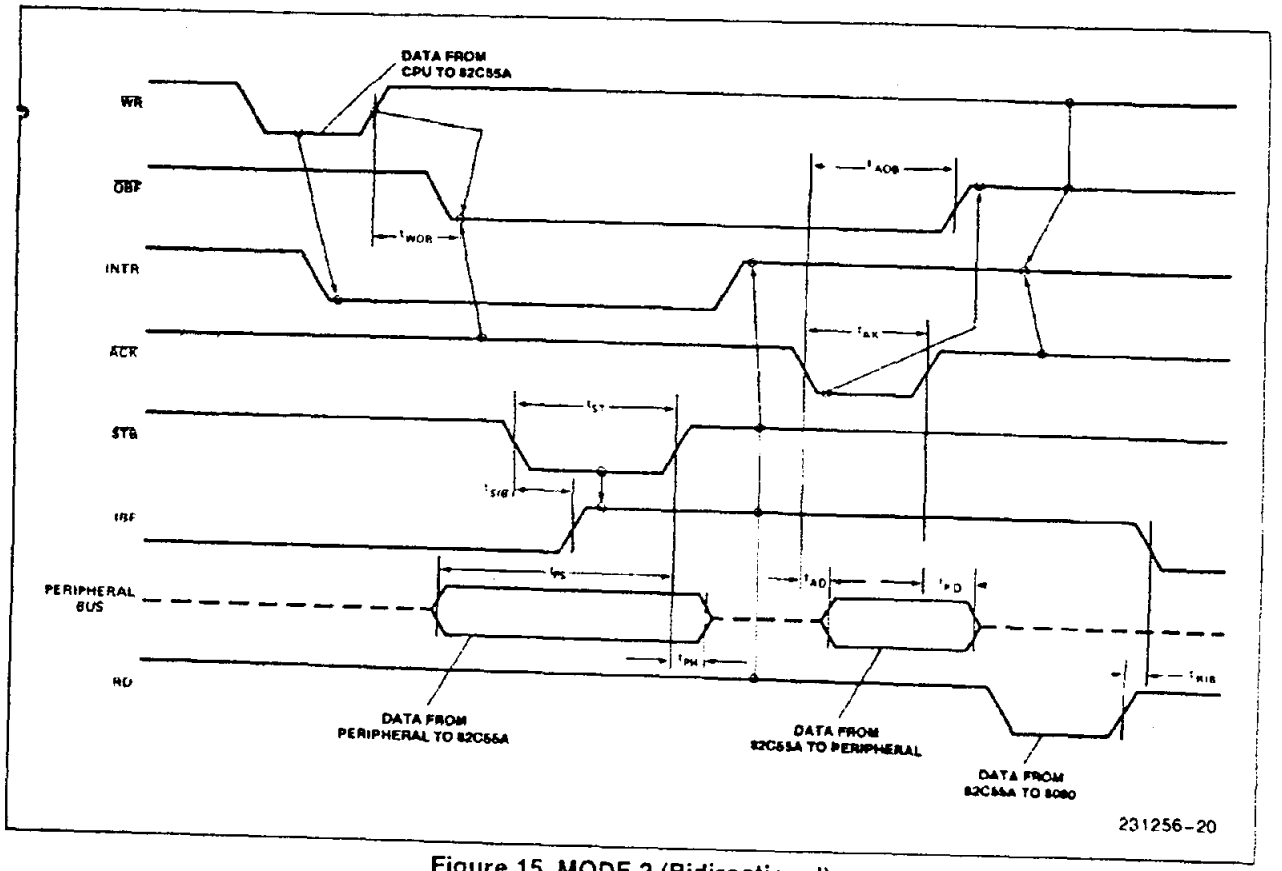


Figure 15. MODE 2 (Bidirectional)

NOTE:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$, and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \bullet \overline{MASK} \bullet \overline{STB} \bullet \overline{RD} + \overline{OBF} \bullet \overline{MASK} \bullet \overline{ACK} \bullet \overline{WR}$)

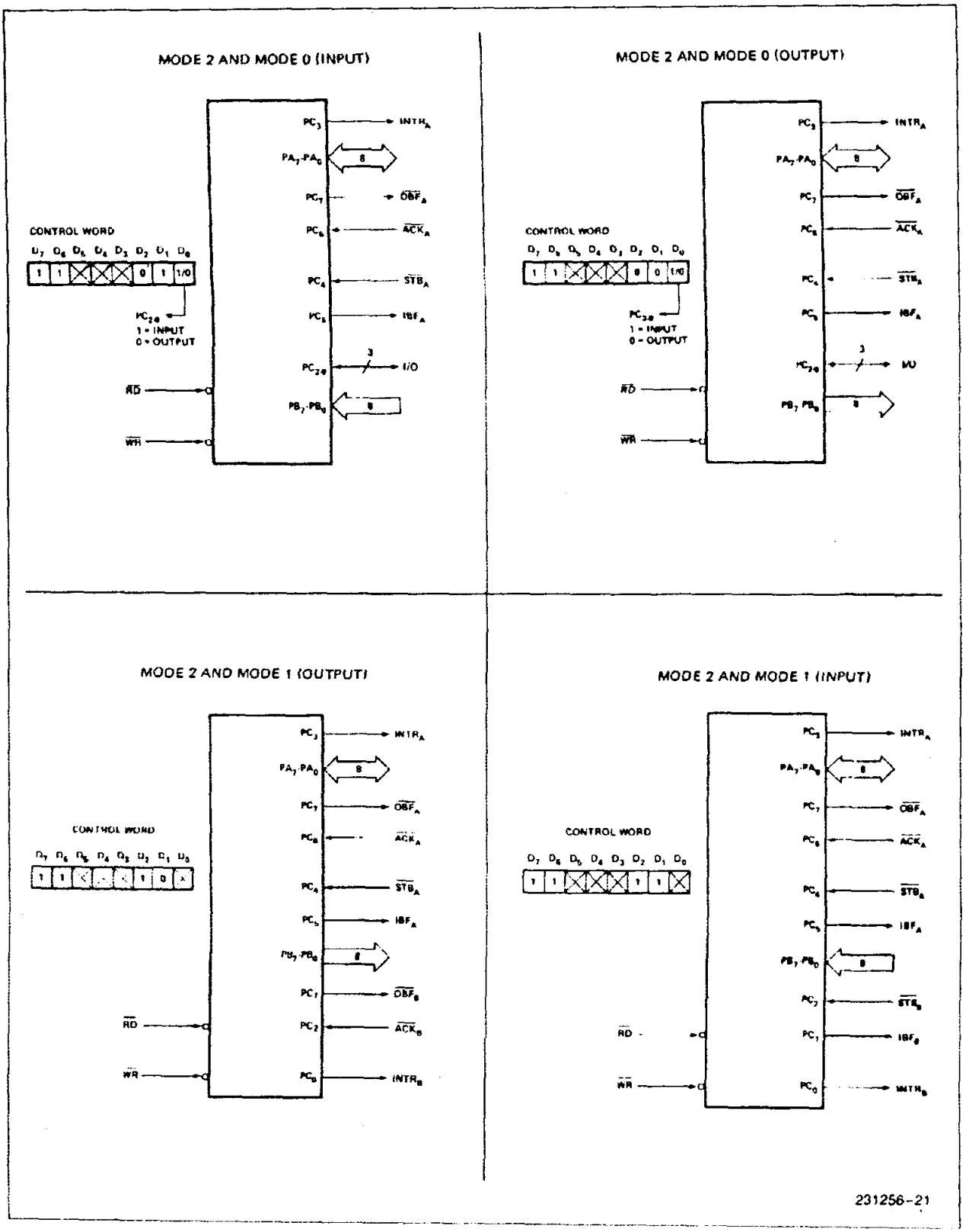


Figure 16. MODE 1/4 Combinations

Mode Definition Summary

	MODE 0	
	IN	OUT
PA_0	IN	OUT
PA_1	IN	OUT
PA_2	IN	OUT
PA_3	IN	OUT
PA_4	IN	OUT
PA_5	IN	OUT
PA_6	IN	OUT
PA_7	IN	OUT
PB_0	IN	OUT
PB_1	IN	OUT
PB_2	IN	OUT
PB_3	IN	OUT
PB_4	IN	OUT
PB_5	IN	OUT
PB_6	IN	OUT
PB_7	IN	OUT
PC_0	IN	OUT
PC_1	IN	OUT
PC_2	IN	OUT
PC_3	IN	OUT
PC_4	IN	OUT
PC_5	IN	OUT
PC_6	IN	OUT
PC_7	IN	OUT

MODE 1	
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
IN	OUT
$INTR_B$	$INTR_B$
IBF_B	$\overline{OBF}_B$
$\overline{STB}_B$	$\overline{ACK}_B$
$INTR_A$	$INTR_A$
$\overline{STB}_A$	I/O
IBF_A	I/O
I/O	$\overline{ACK}_A$
I/O	$\overline{OBF}_A$

MODE 2
GROUP A ONLY
⟷
⟷
⟷
⟷
⟷
⟷
⟷
⟷
—
—
—
—
—
—
—
—
I/O
I/O
I/O
$INTR_A$
$\overline{STB}_A$
IBF_A
$\overline{ACK}_A$
$\overline{OBF}_A$

MODE 0 OR MODE 1 ONLY

Special Mode Combination Considerations

There are several combinations of modes possible. For any combination, some or all of the Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the $\overline{ACK}$ and $\overline{STB}$ lines, will be placed on the data bus. In place of the $\overline{ACK}$ and $\overline{STB}$ line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 18.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including INTR, IBF and $\overline{OBF}$) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including $\overline{ACK}$ and $\overline{STB}$ lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the $\overline{ACK}$ and $\overline{STB}$ lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 18.

Current Drive Capability

Any output on Port A, B or C can sink or source 2.5 mA. This feature allows the 82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

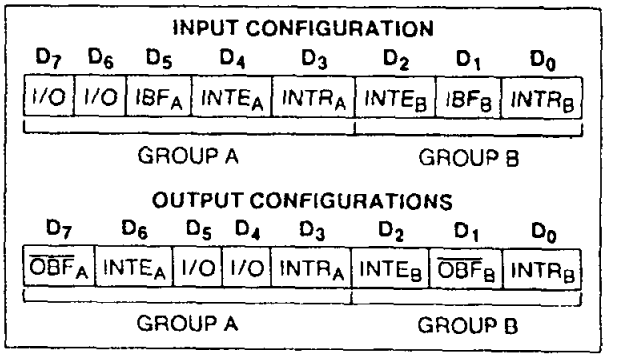


Figure 17a. MODE 1 Status Word Format

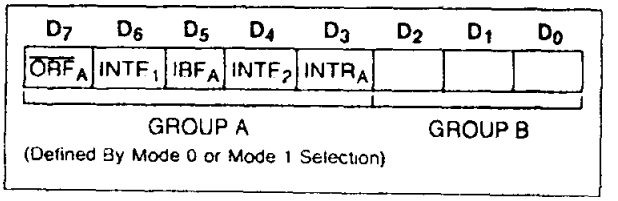


Figure 17b. MODE 2 Status Word Format

Interrupt Enable Flag	Position	Alternate Port C Pin Signal (Mode)
INTE B	PC2	$\overline{ACK}_B$ (Output Mode 1) or $\overline{STB}_B$ (Input Mode 1)
INTE A2	PC4	$\overline{STB}_A$ (Input Mode 1 or Mode 2)
INTE A1	PC6	$\overline{ACK}_A$ (Output Mode 1 or Mode 2

Figure 18. Interrupt Enable Flags in Modes 1 and 2

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . 0°C to + 70°C
Storage Temperature . . . . . − 65°C to + 150°C
Supply Voltage . . . . . − 0.5 to + 8.0V
Operating Voltage . . . . . + 4V to + 7V
Voltage on any Input . . . . . GND − 2V to + 6.5V
Voltage on any Output . . GND − 0.5V to V_{CC} + 0.5V
Power Dissipation . . . . . 1 Watt

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V (T_A = −40°C to +85°C for Extended Temperture)

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.5 mA
V_{OH}	Output High Voltage	3.0 V_{CC} − 0.4		V V	I_{OH} = −2.5 mA I_{OH} = −100 µA
I_{IL}	Input Leakage Current		±1	µA	V_{IN} = V_{CC} to 0V (Note 1)
I_{OFL}	Output Float Leakage Current		±10	µA	V_{IN} = V_{CC} to 0V (Note 2)
I_{DAR}	Darlington Drive Current	±2.5	(Note 4)	mA	Ports A, B, C R_{ext} = 500Ω V_{ext} = 1.7V
I_{PHL}	Port Hold Low Leakage Current	+50	+300	µA	V_{OUT} = 1.0V Port A only
I_{PHH}	Port Hold High Leakage Current	−50	−300	µA	V_{OUT} = 3.0V Ports A, B, C
I_{PHLO}	Port Hold Low Overdrive Current	−350		µA	V_{OUT} = 0.8V
I_{PHHO}	Port Hold High Overdrive Current	+350		µA	V_{OUT} = 3.0V
I_{CC}	V_{CC} Supply Current		10	mA	(Note 3)
I_{CCSB}	V_{CC} Supply Current-Standby		10	µA	V_{CC} = 5.5V V_{IN} = V_{CC} or GND Port Conditions If I/P = Open/High O/P = Open Only With Data Bus = High/Low $\overline{CS}$ = High Reset = Low Pure Inputs = Low/High

NOTES:
1. Pins A_1, A_0, $\overline{CS}$, $\overline{WR}$, $\overline{RD}$, Reset.
2. Data Bus; Ports B, C.
3. Outputs open.
4. Limit output current to 4.0 mA.

CAPACITANCE

T_A = 25°C, V_{CC} = GND = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
C_{IN}	Input Capacitance		10	pF	Unmeasured pins returned to GND f_c = 1 MHz(5)
$C_{I/O}$	I/O Capacitance		20	pF	

NOTE:
5. Sampled not 100% tested.

A.C. CHARACTERISTICS

T_A = 0° to 70°C, V_{CC} = +5V ±10%, GND = 0V

T_A = −40°C to +85°C for Extended Temperature

BUS PARAMETERS

READ CYCLE

Symbol	Parameter	82C55A-2		Units	Test Conditions
		Min	Max		
t_{AR}	Address Stable Before $\overline{RD}$ ↓	0		ns	
t_{RA}	Address Hold Time After $\overline{RD}$ ↑	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	150		ns	
t_{RD}	Data Delay from $\overline{RD}$ ↓		120	ns	
t_{DF}	$\overline{RD}$ ↑ to Data Floating	10	75	ns	
t_{RV}	Recovery Time between $\overline{RD}$/$\overline{WR}$	200		ns	

WRITE CYCLE

Symbol	Parameter	82C55A-2		Units	Test Conditions
		Min	Max		
t_{AW}	Address Stable Before $\overline{WR}$ ↓	0		ns	
t_{WA}	Address Hold Time After $\overline{WR}$ ↑	20		ns	Ports A & B
		20		ns	Port C
t_{WW}	$\overline{WR}$ Pulse Width	100		ns	
t_{DW}	Data Setup Time Before $\overline{WR}$ ↑	100		ns	
t_{WD}	Data Hold Time After $\overline{WR}$ ↑	30		ns	Ports A & B
		30		ns	Port C

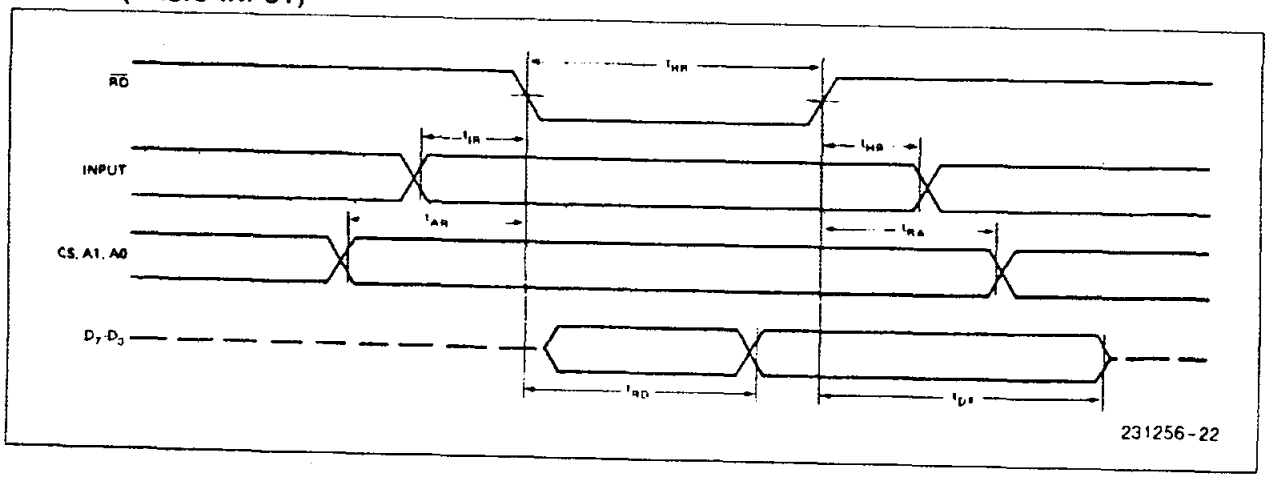
OTHER TIMINGS

Symbol	Parameter	82C55A-2		Units Conditions	Test
		Min	Max		
t_{WB}	$\overline{WR}$ = 1 to Output		350	ns	
t_{IR}	Peripheral Data Before $\overline{RD}$	0		ns	
t_{HR}	Peripheral Data After $\overline{RD}$	0		ns	
t_{AK}	$\overline{ACK}$ Pulse Width	200		ns	
t_{ST}	$\overline{STB}$ Pulse Width	100		ns	
t_{PS}	Per. Data Before $\overline{STB}$ High	20		ns	
t_{PH}	Per. Data After $\overline{STB}$ High	50		ns	
t_{AD}	$\overline{ACK}$ = 0 to Output		175	ns	
t_{KD}	$\overline{ACK}$ = 1 to Output Float	20	250	ns	
t_{WOB}	$\overline{WR}$ = 1 to $\overline{OBF}$ = 0		150	ns	
t_{AOB}	$\overline{ACK}$ = 0 to $\overline{OBF}$ = 1		150	ns	
t_{SIB}	$\overline{STB}$ = 0 to IBF = 1		150	ns	
t_{RIB}	$\overline{RD}$ = 1 to IBF = 0		150	ns	
t_{RIT}	$\overline{RD}$ = 0 to INTR = 0		200	ns	
t_{SIT}	$\overline{STB}$ = 1 to INTR = 1		150	ns	
t_{AIT}	$\overline{ACK}$ = 1 to INTR = 1		150	ns	
t_{WIT}	$\overline{WR}$ = 0 to INTR = 0		200	ns	see note 1
t_{RES}	Reset Pulse Width	500		ns	see note 2

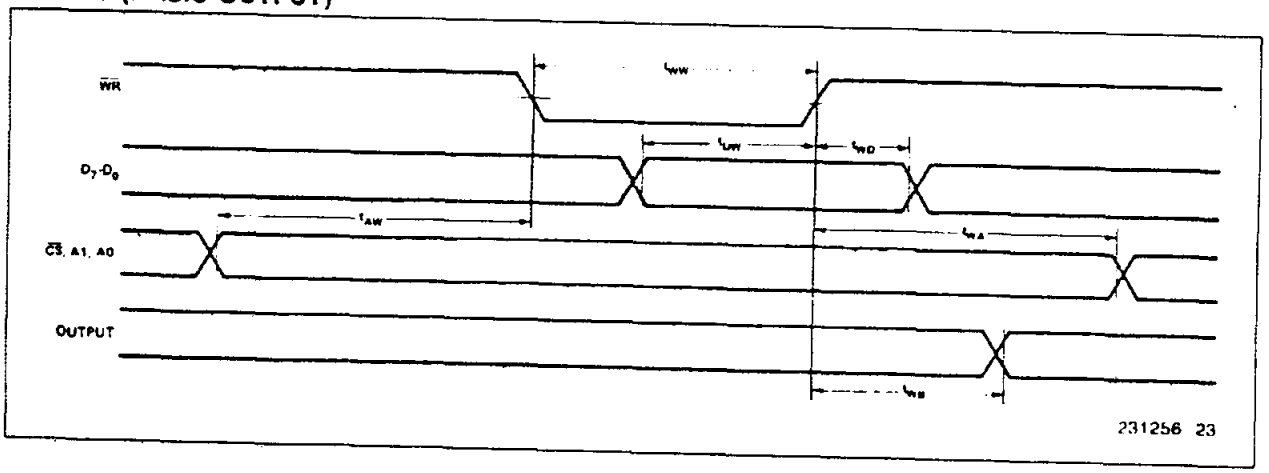
NOTE:
1. INTR ↑ may occur as early as $\overline{WR}$ ↓.
2. Pulse width of initial Reset pulse after power on must be at least 50 μSec. Subsequent Reset pulses may be 500 ns minimum. The output Ports A, B, or C may glitch low during the reset pulse but all port pins will be held at a logic "one" level after the reset pulse.

WAVEFORMS

MODE 0 (BASIC INPUT)

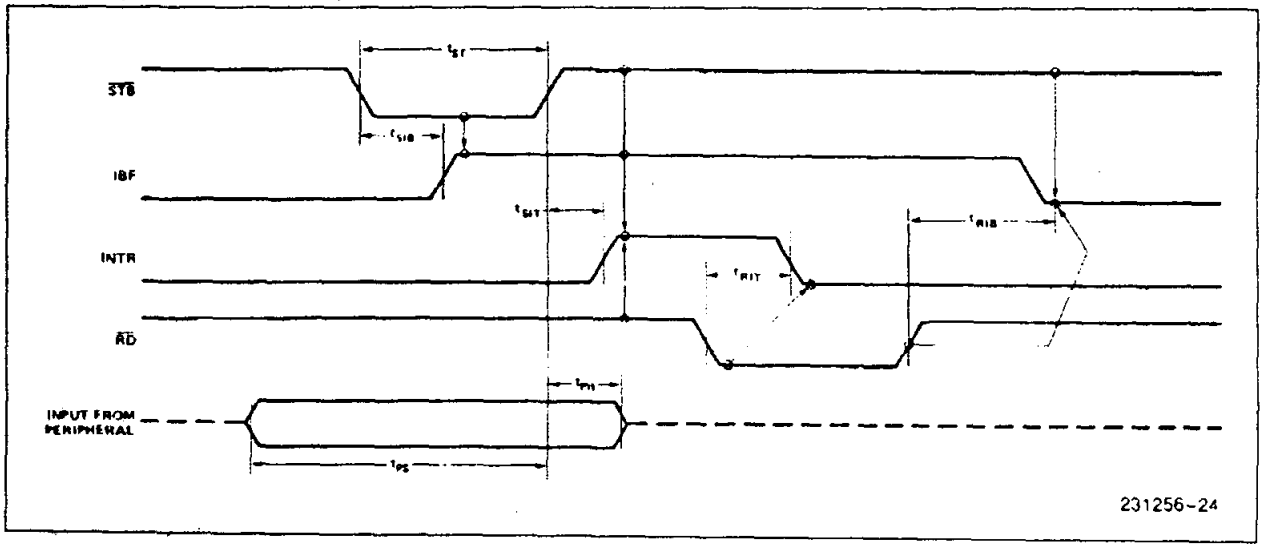


MODE 0 (BASIC OUTPUT)

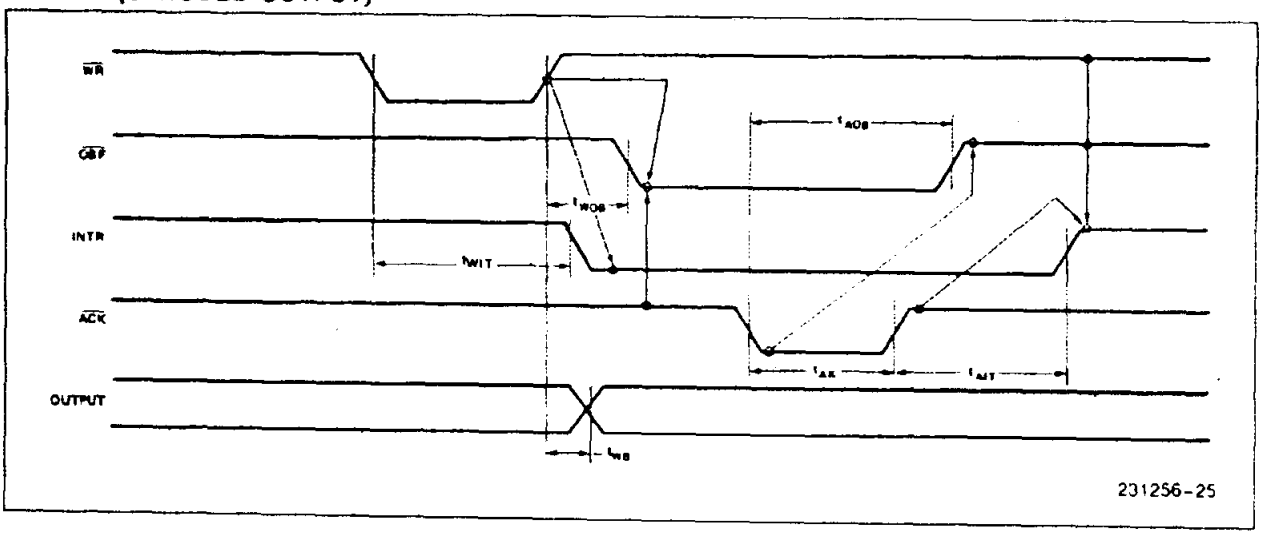


WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)

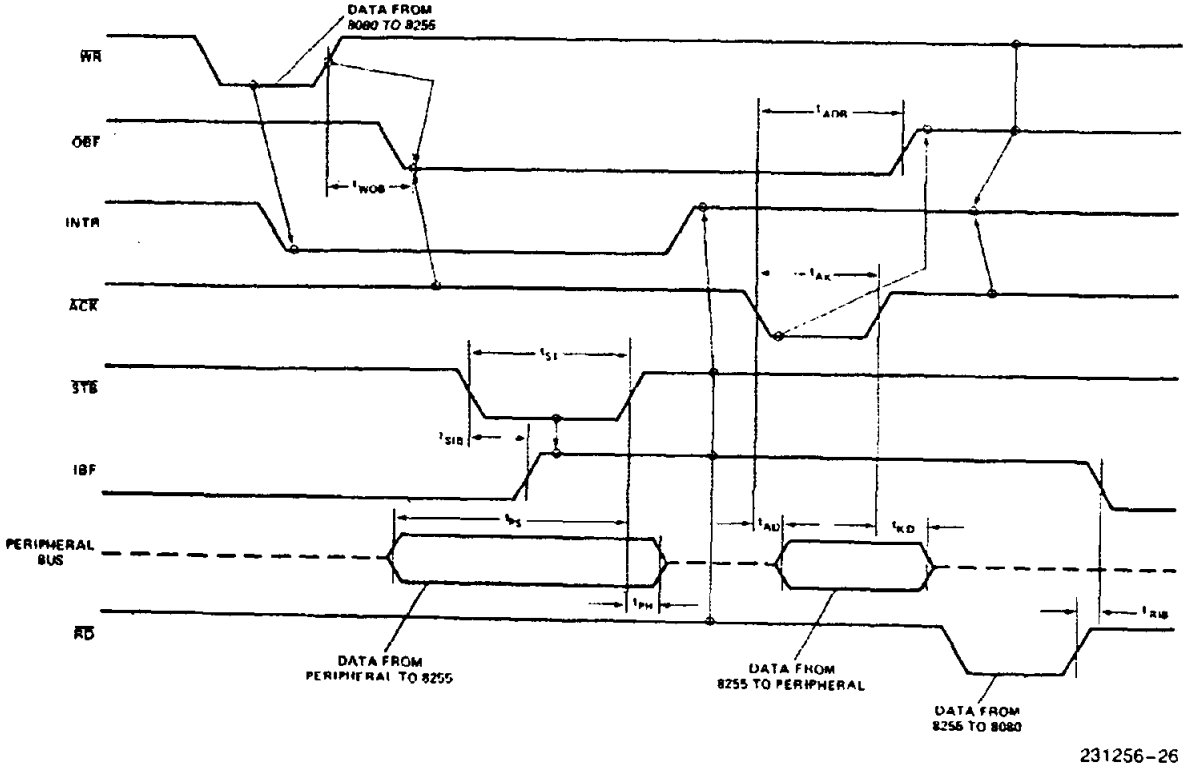


MODE 1 (STROBED OUTPUT)



WAVEFORMS (Continued)

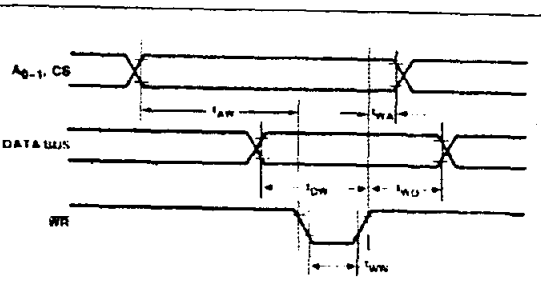
MODE 2 (BIDIRECTIONAL)



231256-26

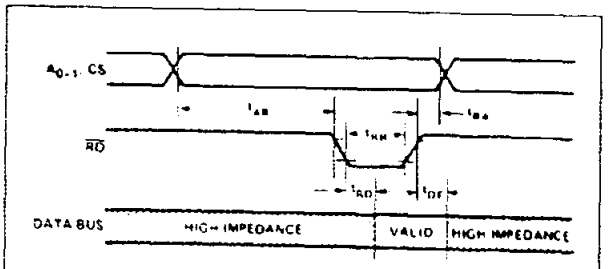
Note:
Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ AND $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \bullet \overline{MASK} \bullet \overline{STB} \bullet \overline{RD} + \overline{OBF} \bullet \overline{MASK} \bullet \overline{ACK} \bullet \overline{WR}$)

WRITE TIMING



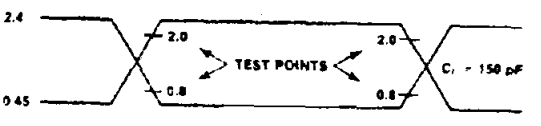
231256-27

READ TIMING



231256-28

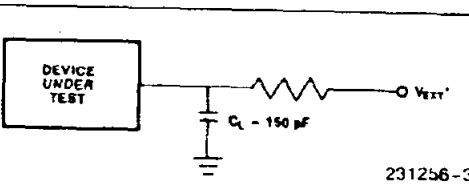
A.C. TESTING INPUT, OUTPUT WAVEFORM



231256-29

A.C. Testing Inputs Are Driven At 2.4V For A Logic 1 And 0.45V For A Logic 0 Timing Measurements Are Made At 2.0V For A Logic 1 And 0.8 For A Logic 0.

A.C. TESTING LOAD CIRCUIT



231256-30

*V_{EXT} Is Set At Various Voltages During Testing To Guarantee The Specification. C_L Includes Jig Capacitance.

DATA SHEET

74LV573
Octal D-type transparent latch (3-State)

Product specification
Supersedes data of 1997 Jun 06
IC24 Data Handbook

1998 Jun 10

Philips
Semiconductors

PHILIPS

Octal D-type transparent latch (3-State) 74LV573

FEATURES

- Wide operating voltage: 1.0 to 5.5V
- Optimized for Low Voltage applications: 1.0V to 3.6V
- Accepts TTL input levels between V_{CC} = 2.7V and V_{CC} = 3.6V
- Typical V_{OLP} (output ground bounce) < 0.8V at V_{CC} = 3.3V, T_{amb} = 25°C
- Typical V_{OHV} (output V_{OH} undershoot) > 2V at V_{CC} = 3.3V, T_{amb} = 25°C
- Inputs and outputs on opposite sides of package allowing easy interface with microprocessors
- Useful as input or output port for microprocessors/microcomputer
- Common 3-State output enable input
- Output capability: bus driver
- I_{CC} category: MSI

DESCRIPTION

The 74LV573 is a low-voltage Si-gate CMOS device that is pin and function compatible with 74HC/HCT573.

The 74LV573 is an octal D-type transparent latch featuring separate D-type inputs for each latch and 3-State outputs for bus oriented applications. A latch enable (LE) input and an output enable (OE) input are common to all internal latches.

The '573' consists of eight D-type transparent latches with 3-State true outputs. When LE is HIGH, data at the D_n inputs enters the latches. In this condition the latches are transparent, i.e., a latch output will change each time its corresponding D-input changes.

When LE is LOW the latches store the information that was present at the D-inputs a set-up time preceding the HIGH-to-LOW transition of LE. When OE is LOW, the contents of the eight latches are available at the outputs. When OE is HIGH, the outputs go to the high impedance OFF-state. Operation of the OE input does not affect the state of the latches.

The '573' is functionally identical to the '563' and the '373', but the '563' has inverted outputs and the '373' has a different pin arrangement.

QUICK REFERENCE DATA

GND = 0V; T_{amb} = 25°C; $t_r = t_f \leq 2.5$ ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL	UNIT
t_{PHL}/t_{PLH}	Propagation delay Dn to Qn LE to Qn	C_L = 15pF V_{CC} = 3.3V	 12 13	ns
C_I	Input capacitance		3.5	pF
C_{PD}	Power dissipation capacitance per latch	Notes 1, 2	26	pF

NOTES:

1. C_{PD} is used to determine the dynamic power dissipation (P_D in µW)
   $P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o)$ where:
   f_i = input frequency in MHz; C_L = output load capacity in pF;
   f_o = output frequency in MHz; V_{CC} = supply voltage in V;
   $\sum (C_L \times V_{CC}^2 \times f_o)$ = sum of the outputs.
2. The condition is V_I = GND to V_{CC}.

ORDERING AND PACKAGE INFORMATION

PACKAGES	TEMPERATURE RANGE	OUTSIDE NORTH AMERICA	NORTH AMERICA	PKG. DWG. #
20-Pin Plastic DIL	–40°C to +125°C	74LV573 N	74LV573 N	SOT146-1
20-Pin Plastic SO	–40°C to +125°C	74LV573 D	74LV573 D	SOT163-1
20-Pin Plastic SSOP Type II	–40°C to +125°C	74LV573 DB	74LV573 DB	SOT339-1
20-Pin Plastic TSSOP Type I	–40°C to +125°C	74LV573 PW	74LV573PW DH	SOT360-1

PIN DESCRIPTION

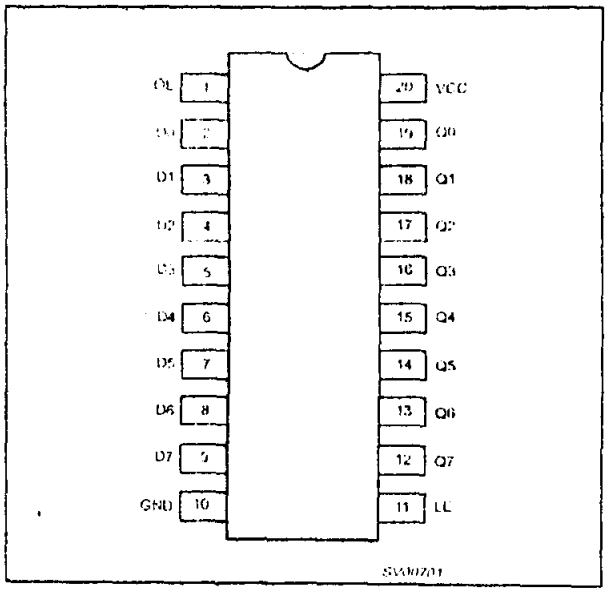
PIN NUMBER	SYMBOL	FUNCTION
1	$\overline{OE}$	Output enabled input (active LOW)
2, 3, 4, 5, 6, 7, 8, 9	D0–D7	Data inputs
19, 18, 17, 16, 15, 14, 13, 12	Q0–Q7	Data outputs
10	GND	Ground (0V)
11	LE	Latch enable input (active HIGH)
20	VCC	Positive supply voltage

FUNCTION TABLE

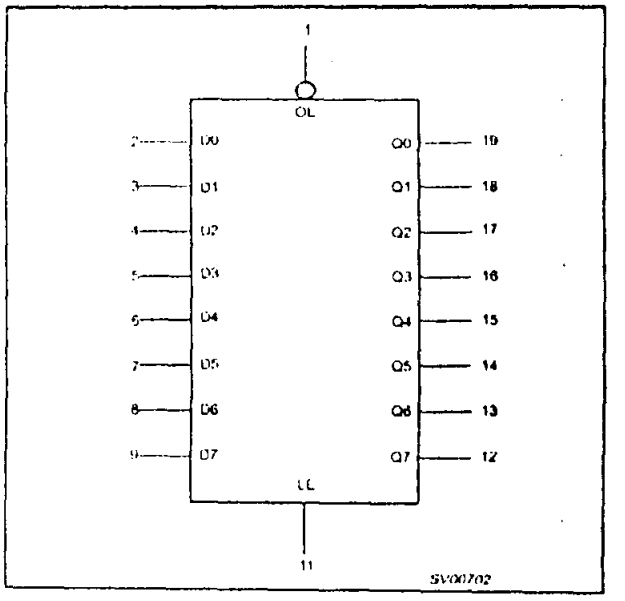
OPERATING MODES	INPUTS			INTERNAL LATCHES	OUTPUTS
	OE	LE	Dn		Q0 to Q7
Enable and read register (transparent mode)	L L	H H	L H	L H	L H
Latch and read register	L L	L L	l h	L H	L H
Latch register and disable outputs	H H	L L	l h	L H	Z Z

H = HIGH voltage level
h = HIGH voltage level one set-up time prior to the HIGH-to-LOW LE transition
L = LOW voltage level
l = LOW voltage level one set-up time prior to the HIGH-to-LOW LE transition
Z = *High impedance OFF-state*

PIN CONFIGURATION

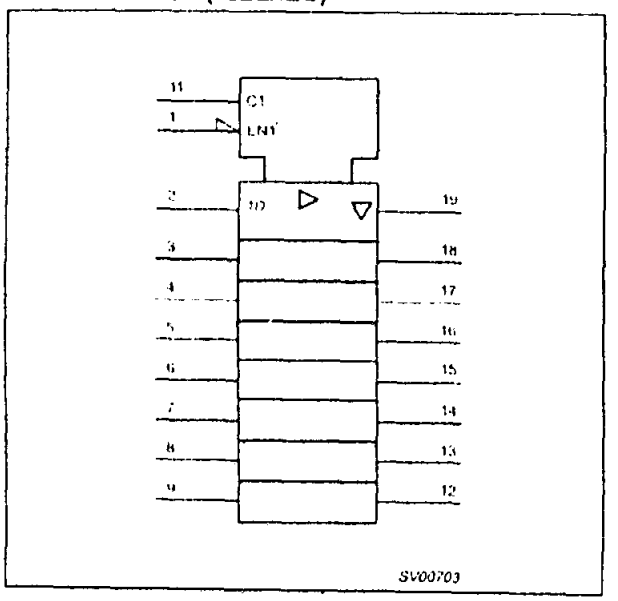


LOGIC SYMBOL

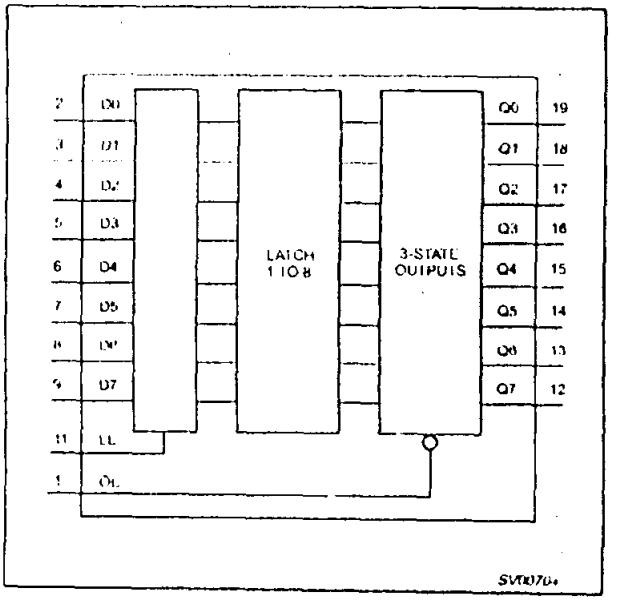


Octal D-type transparent latch (3-State) 74LV573

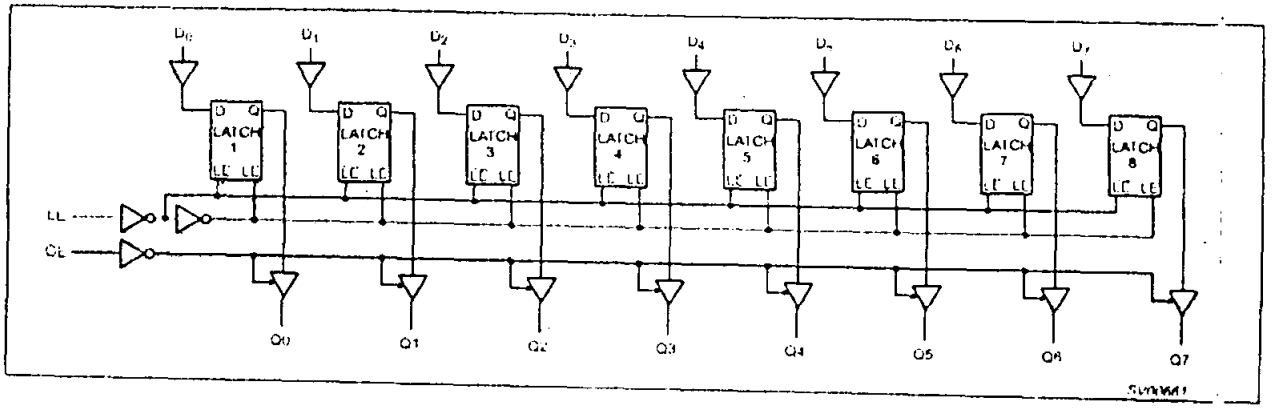
LOGIC SYMBOL (IEEE/IEC)



FUNCTIONAL DIAGRAM



LOGIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS[1, 2]

In accordance with the Absolute Maximum Rating System (IEC 134)
Voltages are referenced to GND (ground = 0V)

SYMBOL	PARAMETER	CONDITIONS	RATING	UNIT
V_{CC}	DC supply voltage		-0.5 to +7.0	V
$\pm I_{IK}$	DC input diode current	$V_I < -0.5$ or $V_I > V_{CC} + 0.5V$	20	mA
$\pm I_{OK}$	DC output diode current	$V_O < -0.5$ or $V_O > V_{CC} + 0.5V$	50	mA
$\pm I_O$	DC output source or sink current – bus driver outputs	$-0.5V < V_O < V_{CC} + 0.5V$	35	mA
$\pm I_{GND}$, $+I_{CC}$	DC V_{CC} or GND current for types with –bus driver outputs		70	mA
T_{stg}	Storage temperature range		–65 to +150	°C
P_{tot}	Power dissipation per package –plastic DIL –plastic mini-pack (SO) –plastic shrink mini-pack (SSOP and TSSOP)	for temperature range: –40 to +125°C above +70°C derate linearly with 12mW/K above +70 °C derate linearly with 8 mW/K above +60 °C derate linearly with 5.5 mW/K	 750 500 400	mW

NOTES:

1. Stresses beyond those listed may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
2. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP.	MAX	UNIT
V_{CC}	DC supply voltage	See Note 1	1.0	3.3	5.5	V
V_I	Input voltage		0	–	V_{CC}	V
V_O	Output voltage		0	–	V_{CC}	V
T_{amb}	Operating ambient temperature range in free air	See DC and AC characteristics	–40 –40		+85 +125	°C
t_r, t_f	Input rise and fall times	V_{CC} = 1.0V to 2.0V V_{CC} = 2.0V to 2.7V V_{CC} = 2.7V to 3.6V V_{CC} = 3.6V to 5.5V	– – – –	– – – –	500 200 100 50	ns/V

NOTE:

1. The LV is guaranteed to function down to V_{CC} = 1.0V (input levels GND or V_{CC}); DC characteristics are guaranteed from V_{CC} = 1.2V to V_{CC} = 5.5V.

DC CHARACTERISTICS FOR THE LV FAMILY

Over recommended operating conditions voltages are referenced to GND (ground = 0V)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS					UNIT
			-40 C to +85 C			-40 C to +125°C		
			MIN	TYP[1]	MAX	MIN	MAX	
V_{IH}	HIGH level input voltage	V_{CC} = 1.2V	0.9			0.9		V
		V_{CC} = 2.0V	1.4			1.4		
		V_{CC} = 2.7 to 3.6V	2.0			2.0		
		V_{CC} = 4.5 to 5.5V	$0.7*V_{CC}$			$0.7*V_{CC}$		
V_{IL}	LOW level input voltage	V_{CC} = 1.2V			0.3		0.3	V
		V_{CC} = 2.0V			0.6		0.6	
		V_{CC} = 2.7 to 3.6V			0.8		0.8	
		V_{CC} = 4.5 to 5.5			$0.3*V_{CC}$		$0.3*V_{CC}$	
V_{OH}	HIGH level output voltage; all outputs	V_{CC} = 1.2V; $V_I = V_{IH}$ or V_{IL}; $-I_O$ = 100µA		1.2				V
		V_{CC} = 2.0V; $V_I = V_{IH}$ or V_{IL}; $-I_O$ = 100µA	1.8	2.0		1.8		
		V_{CC} = 2.7V; $V_I = V_{IH}$ or V_{IL}; $-I_O$ = 100µA	2.5	2.7		2.5		
		V_{CC} = 3.0V; $V_I = V_{IH}$ or V_{IL}; $-I_O$ = 100µA	2.8	3.0		2.8		
		V_{CC} = 4.5V; $V_I = V_{IH}$ or V_{IL}; $-I_O$ = 100µA	4.3	4.5		4.3		
	HIGH level output voltage; BUS driver outputs	V_{CC} = 3.0V; $V_I = V_{IH}$ or V_{IL}; $-I_O$ = 8mA	2.40	2.82		2.20		
		V_{CC} = 4.5V; $V_I = V_{IH}$ or V_{IL}; $-I_O$ = 16mA	3.60	4.20		3.50		
V_{OL}	LOW level output voltage; all outputs	V_{CC} = 1.2V; $V_I = V_{IH}$ or V_{IL}; I_O = 100µA		0				V
		V_{CC} = 2.0V; $V_I = V_{IH}$ or V_{IL}; I_O = 100µA		0	0.2		0.2	
		V_{CC} = 2.7V; $V_I = V_{IH}$ or V_{IL}; I_O = 100µA		0	0.2		0.2	
		V_{CC} = 3.0V; $V_I = V_{IH}$ or V_{IL}; I_O = 100µA		0	0.2		0.2	
		V_{CC} = 4.5V; $V_I = V_{IH}$ or V_{IL}; I_O = 100µA		0	0.2		0.2	
	LOW level output voltage; BUS driver outputs	V_{CC} = 3.0V; $V_I = V_{IH}$ or V_{IL}; I_O = 8mA		0.20	0.40		0.50	
		V_{CC} = 4.5V; $V_I = V_{IH}$ or V_{IL}; I_O = 16mA		0.35	0.55		0.65	
I_I	Input leakage current	V_{CC} = 5.5V; $V_I = V_{CC}$ or GND			1.0		1.0	µA
I_{OZ}	3-State output OFF-state current	V_{CC} = 5.5V; $V_I = V_{IH}$ or V_{IL}; $V_O = V_{CC}$ or GND			5		10	µA
I_{CC}	Quiescent supply current; MSI	V_{CC} = 5.5V; $V_I = V_{CC}$ or GND; I_O = 0			20.0		160	µA
ΔI_{CC}	Additional quiescent supply current per input	V_{CC} = 2.7V to 3.6V; $V_I = V_{CC} - 0.6V$			500		850	µA

NOTE:

1. All typical values are measured at T_{amb} = 25°C.

AC CHARACTERISTICS

GND = 0V; $t_r = t_f \leq 2.5$ns; C_L = 50pF; R_L = 1KΩ

SYMBOL	PARAMETER	WAVEFORM	CONDITION	LIMITS −40 to +85 °C			LIMITS −40 to +125 °C		UNIT
			V_{CC}(V)	MIN	TYP	MAX	MIN	MAX	
t_{PHL}/t_{PLH}	Propagation delay Dn to Qn	Figures 1, 5	1.2	–	75	–	–	–	ns
			2.0	–	28	39	–	49	
			2.7	–	19	29	–	36	
			3.0 to 3.6	–	14[2]	23	–	29	
			4.5 to 5.5	–	–	19	–	24	
t_{PHL}/t_{PLH}	Propagation delay LE to Qn	Figures 2, 5	1.2	–	80	–	–	–	ns
			2.0	–	27	43	–	53	
			2.7	–	20	31	–	34	
			3.0 to 3.6	–	15[2]	25	–	31	
			4.5 to 5.5	–	–	21	–	26	
t_{PZH}/t_{PZL}	3-State output enable time OE to Qn	Figures 3, 5	1.2	–	70	–	–	–	ns
			2.0	–	24	37	–	48	
			2.7	–	18	28	–	35	
			3.0 to 3.6	–	13[2]	22	–	28	
			4.5 to 5.5	–	–	18	–	23	
t_{PHZ}/t_{PLZ}	3-State output disable time OE to Qn	Figures 3, 5	1.2	–	80	–	–	–	ns
			2.0	–	29	39	–	48	
			2.7	–	22	29	–	36	
			3.0 to 3.6	–	17[2]	24	–	29	
			4.5 to 5.5	–	–	20	–	24	
t_W	LE pulse width HIGH	Figure 2	2.0	34	9	–	41	–	ns
			2.7	25	6	–	30	–	
			3.0 to 3.6	20	5[2]	–	24	–	
t_{su}	Setup time Dn to LE	Figure 4	1.2	–	25	–	–	–	ns
			2.0	17	9	–	20	–	
			2.7	13	6	–	15	–	
			3.0 to 3.6	10	5[2]	–	12	–	
t_h	Hold time Dn to LE	Figure 4	1.2	–	5	–	–	–	ns
			2.0	8	2	–	8	–	
			2.7	8	2	–	8	–	
			3.0 to 3.6	8	1[2]	–	8	–	

NOTES:

All typical values are measured at T_{amb} = 25°C

1. Typical values are measured at V_{CC} = 3.3V

AC WAVEFORMS

V_M = 1.5V at $V_{CC} \geq$ 2.7V and $\leq$ 3.6V
V_M = 0.5 * V_{CC} at V_{CC} < 2.7V and $\geq$ 4.5V
V_{OL} and V_{OH} are the typical output voltage drop that occur with the output load
V_X = V_{OL} + 0.3V at $V_{CC} \geq$ 2.7V and $\leq$ 3.6V
V_X = V_{OL} + 0.1V_{CC} at V_{CC} < 2.7V and $\geq$ 4.5V
V_Y = V_{OH} - 0.3V at $V_{CC} \geq$ 2.7V and $\leq$ 3.6V
V_Y = V_{OH} - 0.1V_{CC} at V_{CC} < 2.7V and $\geq$ 4.5V

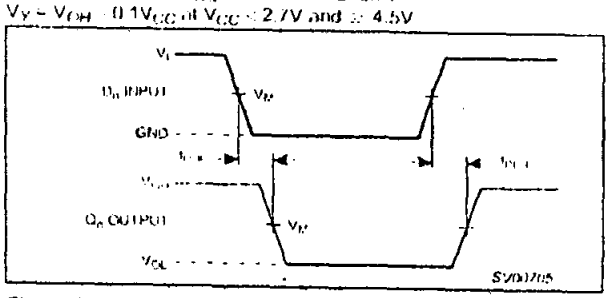


Figure 1. Data input (D_n) to output (Q_n) propagation delays and the output transition times

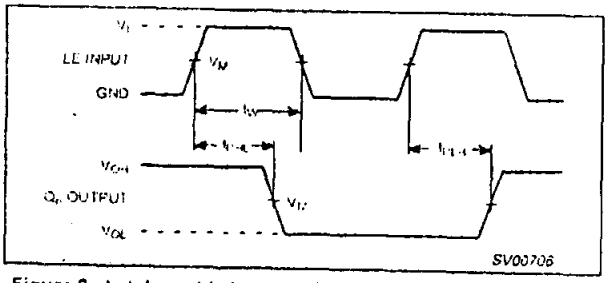


Figure 2. Latch enable input (LE) pulse width, the latch enable input to output (Q_n) propagation delays and the output transition times.

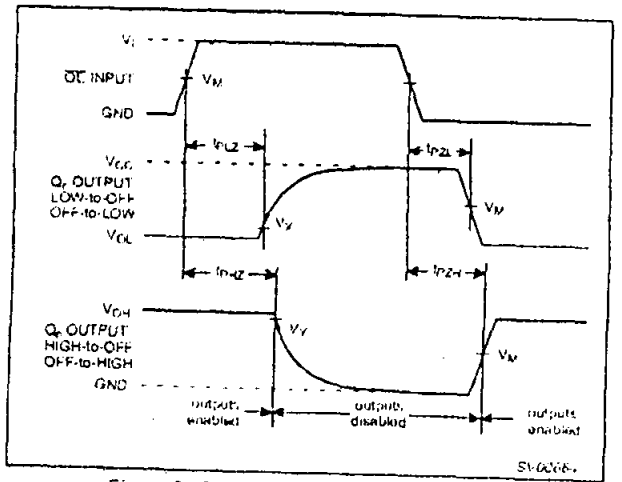


Figure 3. 3-State enable and disable times

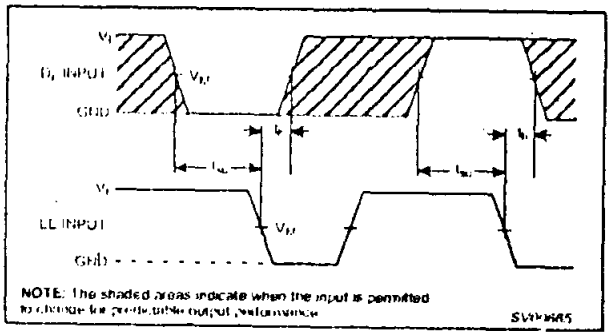


Figure 4. Data set-up and hold times for the D_n input to the LE input

NOTE:
The shaded areas indicate when the input is permitted to change for predictable output performance

TEST CIRCUIT

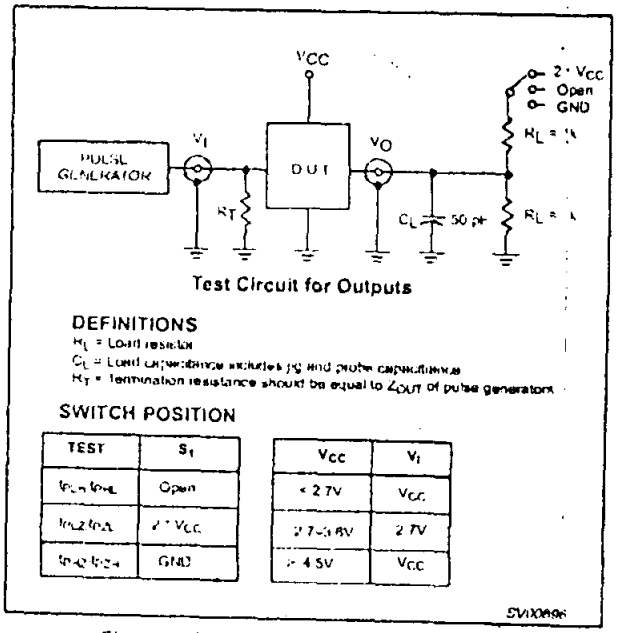
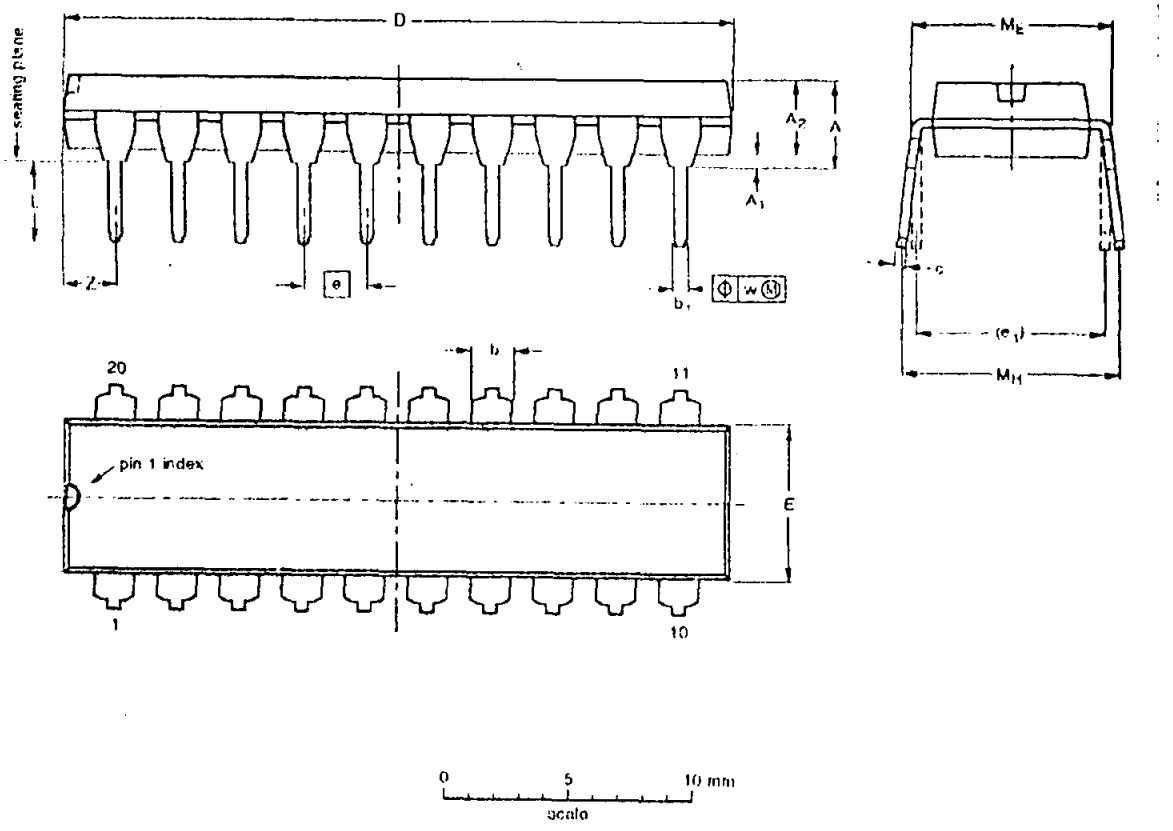


Figure 5. Load circuitry for switching times

DIP20: plastic dual in-line package; 20 leads (300 mil) SOT146-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

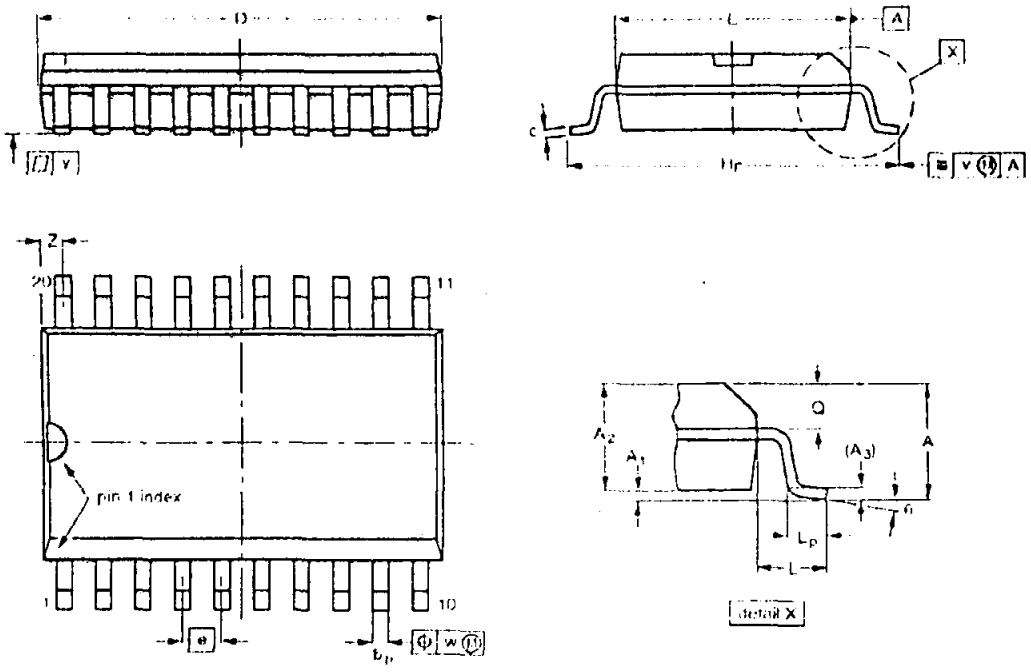
UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	4.2	0.51	3.2	1.73 1.30	0.53 0.38	0.36 0.23	26.92 26.54	6.40 6.22	2.54	7.62	3.60 3.05	8.25 7.80	10.0 8.3	0.254	2.0
inches	0.17	0.020	0.13	0.068 0.051	0.021 0.015	0.014 0.009	1.060 1.045	0.25 0.24	0.10	0.30	0.14 0.12	0.32 0.31	0.39 0.33	0.01	0.078

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT146-1			SC603			~~92-11-17~~ 95-05-24

SO20: plastic small outline package; 20 leads; body width 7.5 mm **SOT163-1**



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

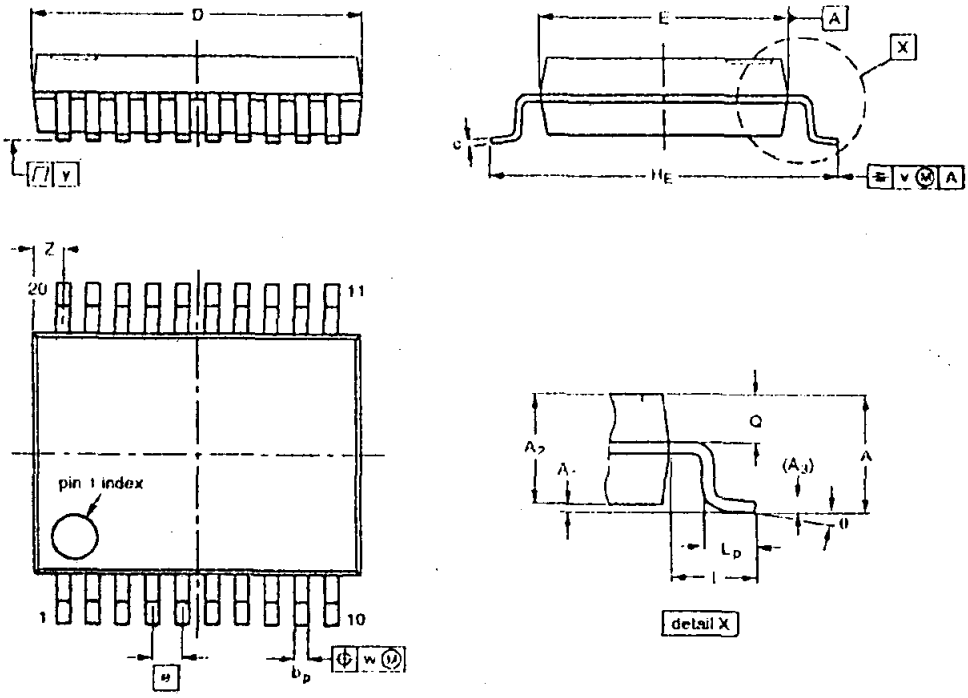
UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2.65	0.30 0.10	2.45 2.25	0.25	0.49 0.36	0.32 0.23	13.0 12.6	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.10	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.51 0.49	0.30 0.29	0.050	0.42 0.39	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

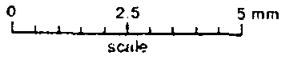
Note

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included

OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ		
SOT163-1	075E04	MS-013AC			~~92-11-17~~ 95-01-24

SSOP20: plastic shrink small outline package; 20 leads; body width 5.3 mm **SOT339-1**





DIMENSIONS (mm are the original dimensions)

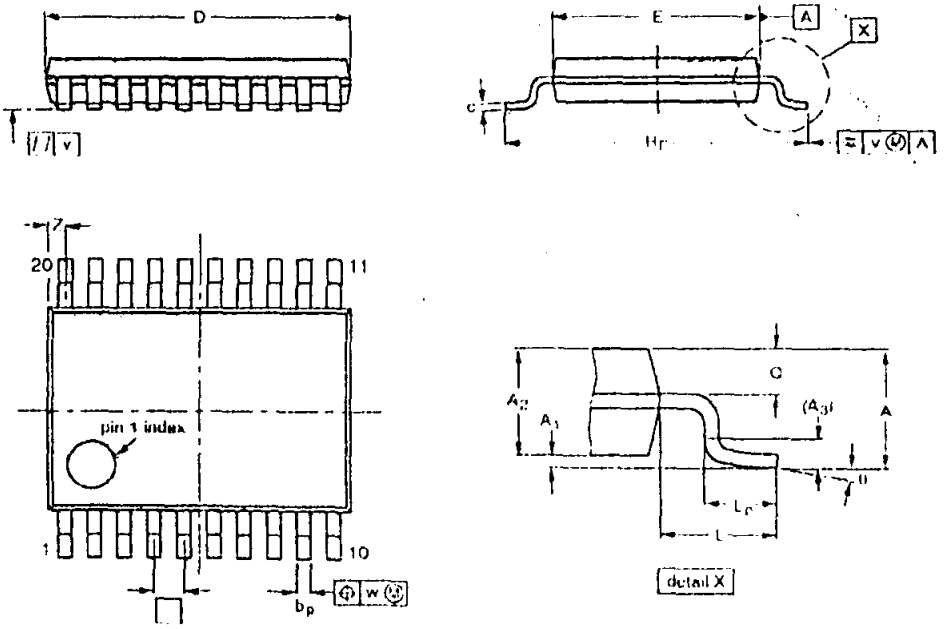
UNIT	A max.	A_1	A_2	A_3	b_p	c	D(1)	E(1)	e	H_E	L	L_p	Q	v	w	y	Z(1)	θ
mm	2.0	0.21 0.05	1.80 1.65	0.25	0.38 0.25	0.20 0.09	7.4 7.0	5.4 5.2	0.65	7.9 7.6	1.25	1.03 0.63	0.9 0.7	0.2	0.13	0.1	0.9 0.5	8° 0°

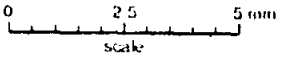
Note

1. Plastic or metal protrusions of 0.20 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT339-1		MO-150AE				~~93-09-08~~ 95-02-04

TSSOP20: plastic thin shrink small outline package; 20 leads; body width 4.4 mm **SOT360-1**





DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (2)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	1.10	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	6.6 6.4	4.5 4.3	0.65	6.6 6.2	1.0	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.5 0.2	8° 0°

Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT360-1		MO-153AC				~~93-06-16~~ 95-02-04

DEFINITIONS

Data Sheet Identification	Product Status	Definition
Objective Specification	Formative or in Design	This data sheet contains the design target or goal specifications for product development. Specifications may change in any manner without notice.
Preliminary Specification	Preproduction Product	This data sheet contains preliminary data, and supplementary data will be published at a later date. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product Specification	Full Production	This data sheet contains Final Specifications. Philips Semiconductors reserves the right to make changes at any time without notice, in order to improve design and supply the best possible product.

Philips Semiconductors and Philips Electronics North America Corporation reserve the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

LIFE SUPPORT APPLICATIONS

Philips Semiconductors and Philips Electronics North America Corporation Products are not designed for use in life support appliances, devices, or systems where malfunction of a Philips Semiconductors and Philips Electronics North America Corporation Product can reasonably be expected to result in a personal injury. Philips Semiconductors and Philips Electronics North America Corporation customers using or selling Philips Semiconductors and Philips Electronics North America Corporation Products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors and Philips Electronics North America Corporation for any damages resulting from such improper use or sale.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088–3409
Telephone 800-234-7381

print code Date of release: 05-96

Document order number: 9397-750-04453

Let's make things better.

Philips Semiconductors

PHILIPS

DATA SHEET

For a complete data sheet, please also download:

- The IC06 74HC/HCT/HCU/HCMOS Logic Family Specifications
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Information
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Outlines

74HC/HCT138
3-to-8 line decoder/demultiplexer; inverting

Product specification
File under Integrated Circuits, IC06

September 1993

Philips
Semiconductors

PHILIPS

3-to-8 line decoder/demultiplexer; inverting | 74HC/HCT138

FEATURES

- Demultiplexing capability
- Multiple input enable for easy expansion
- Ideal for memory chip select decoding
- Active LOW mutually exclusive outputs
- Output capability: standard
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT138 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT138 decoders accept three binary weighted address inputs (A_0, A_1, A_2) and when enabled, provide 8 mutually exclusive active LOW outputs ($\overline{Y}_0$ to $\overline{Y}_7$).

The "138" features three enable inputs: two active LOW ($\overline{E}_1$ and $\overline{E}_2$) and one active HIGH (E_3). Every output will be HIGH unless $\overline{E}_1$ and $\overline{E}_2$ are LOW and E_3 is HIGH.

This multiple enable function allows easy parallel expansion of the "138" to a 1-of-32 (5 lines to 32 lines) decoder with just four "138" ICs and one inverter.

The "138" can be used as an eight output demultiplexer by using one of the active LOW enable inputs as the data input and the remaining enable inputs as strobes. Unused enable inputs must be permanently tied to their appropriate active HIGH or LOW state.

The "138" is identical to the "238" but has inverting outputs.

QUICK REFERENCE DATA

GND = 0 V; T_{amb} = 25 °C; $t_r = t_f$ = 6 ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
	propagation delay	C_L = 15 pF; V_{CC} = 5 V			
t_{PHL}/ t_{PLH}	A_n to $\overline{Y}_n$		12	17	ns
t_{PHL}/ t_{PLH}	E_3 to $\overline{Y}_n$ $\overline{E}_n$ to $\overline{Y}_n$		14	19	ns
C_I	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per package	notes 1 and 2	67	67	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in µW):

   $P_D = C_{PD} \times V_{CC}^2 \times f_i + \Sigma (C_L \times V_{CC}^2 \times f_o)$ where:

   f_i = input frequency in MHz

   f_o = output frequency in MHz

   $\Sigma (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs

   C_L = output load capacitance in pF

   V_{CC} = supply voltage in V

2. For HC the condition is V_I = GND to V_{CC}
   For HCT the condition is V_I = GND to V_{CC} − 1.5 V

ORDERING INFORMATION

See *"74HC/HCT/HCU/HCMOS Logic Package Information"*.

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 2, 3	A_0 to A_2	address inputs
4, 5	$\overline{E}_1$, $\overline{E}_2$	enable inputs (active LOW)
6	E_3	enable input (active HIGH)
8	GND	ground (0 V)
15, 14, 13, 12, 11, 10, 9, 7	$\overline{Y}_0$ to $\overline{Y}_7$	outputs (active LOW)
16	V_{CC}	positive supply voltage

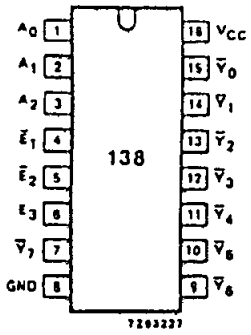


Fig.1 Pin configuration.

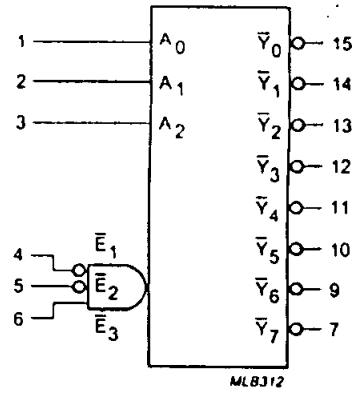


Fig.2 Logic symbol.

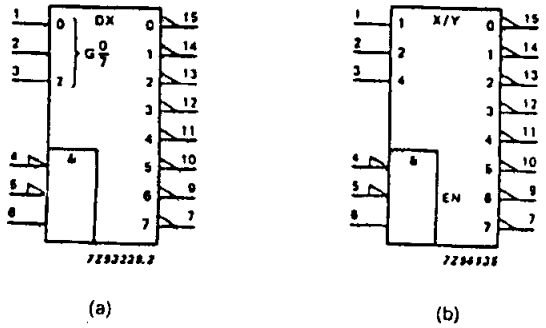


Fig.3 IEC logic symbol.

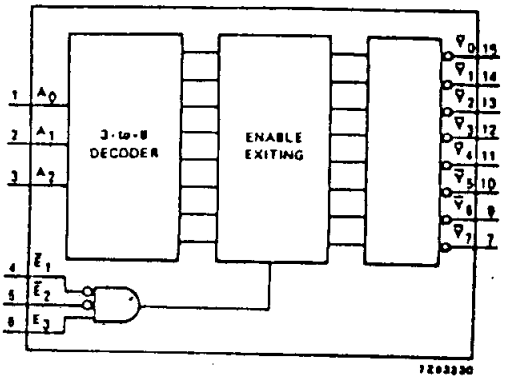


Fig.4 Functional diagram.

FUNCTION TABLE

INPUTS						OUTPUTS							
$\overline{E}_1$	$\overline{E}_2$	E_3	A_0	A_1	A_2	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

Notes

1. H = HIGH voltage level
   L = LOW voltage level
   X = don't care

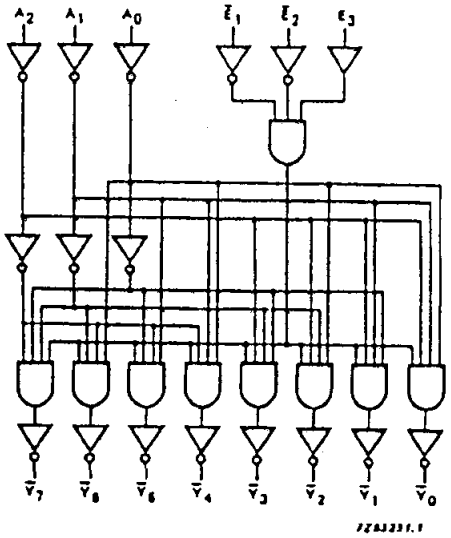


Fig.5 Logic diagram.

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see *"74HC/HCT/HCU/HCMOS Logic Family Specifications"*.

Output capability: standard
I_{CC} category: MSI

AC CHARACTERISTICS FOR 74HC

GND = 0 V; $t_r = t_f$ = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T_{amb} (°C)							UNIT	TEST CONDITIONS	
		74HC								V_{CC} (V)	WAVEFORMS
		+25			−40 to +85		−40 to +125				
		min.	typ.	max.	min.	max.	min.	max.			
t_{PHL}/ t_{PLH}	propagation delay A_n to $\overline{Y}_n$		41 15 12	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.6
t_{PHL}/ t_{PLH}	propagation delay E_3 to $\overline{Y}_n$		47 17 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.6
t_{PHL}/ t_{PLH}	propagation delay $\overline{E}_n$ to $\overline{Y}_n$		47 17 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.7
t_{THL}/ t_{TLH}	output transition time		19 7 6	75 15 13		95 19 16		110 22 19	ns	2.0 4.5 6.0	Figs 6 and 7

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see *"74HC/HCT/HCU/HCMOS Logic Family Specifications"*.

Output capability: standard
I_{CC} category: MSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications. To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

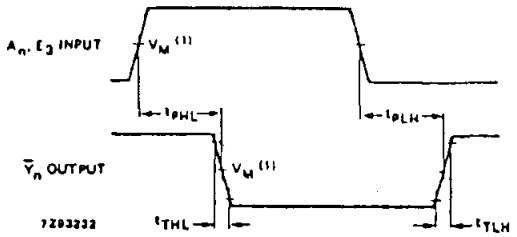
INPUT	UNIT LOAD COEFFICIENT
A_n	1.50
$\overline{E}_n$	1.25
E_3	1.00

AC CHARACTERISTICS FOR 74HCT

GND = 0 V; $t_r = t_f$ = 6 ns; C_L = 50 pF

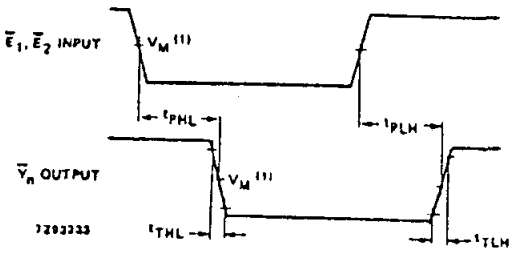
SYMBOL	PARAMETER	T_{amb} (°C)							UNIT	TEST CONDITIONS	
		74HCT								V_{CC} (V)	WAVEFORMS
		+25			−40 to +85		−40 to +125				
		min.	typ.	max.	min.	max.	min.	max.			
t_{PHL}/ t_{PLH}	propagation delay A_n to $\overline{Y}_n$		20	35		44		53	ns	4.5	Fig.6
t_{PHL}/ t_{PLH}	propagation delay E_3 to $\overline{Y}_n$		18	40		50		60	ns	4.5	Fig.6
t_{PHL}/ t_{PLH}	propagation delay $\overline{E}_n$ to $\overline{Y}_n$		19	40		50		60	ns	4.5	Fig.7
t_{THL}/ t_{TLH}	output transition time		7	15		19		22	ns	4.5	Figs 6 and 7

AC WAVEFORMS



(1) HC : V_M = 50%; V_I = GND to V_{CC}.
HCT: V_M = 1.3 V; V_I = GND to 3 V.

Fig.6 Waveforms showing the address input (A_n) and enable input (E_3) to output ($\overline{Y}_n$) propagation delays and the output transition times.



(1) HC : V_M = 50%; V_I = GND to V_{CC}.
HCT: V_M = 1.3 V; V_I = GND to 3 V.

Fig.7 Waveforms showing the enable input ($\overline{E}_n$) to output ($\overline{Y}_n$) propagation delays and the output transition times.

PACKAGE OUTLINES

See *"74HC/HCT/HCU/HCMOS Logic Package Outlines"*.

Features

- Fast Read Access Time - 150 ns
- Automatic Page Write Operation
  - Internal Address and Data Latches for 64 Bytes
- Fast Write Cycle Times
  - Page Write Cycle Time: 10 ms Maximum
  - 1 to 64-byte Page Write Operation
- Low Power Dissipation
  - 40 mA Active Current
  - 100 µA CMOS Standby Current
- Hardware and Software Data Protection
- DATA Polling and Toggle Bit for End of Write Detection
- High Reliability CMOS Technology
  - Endurance: 100,000 Cycles
  - Data Retention: 10 Years
- Single 5V ± 10% Supply
- CMOS and TTL Compatible Inputs and Outputs
- JEDEC Approved Byte-wide Pinout
- Commercial and Industrial Temperature Ranges

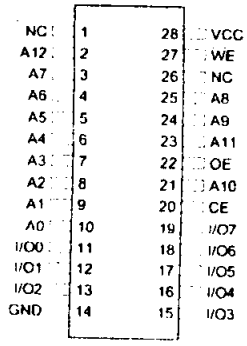
Description

The AT28C64B is a high-performance electrically-erasable and programmable read only memory (EEPROM). Its 64K of memory is organized as 8,192 words by 8 bits. Manufactured with Atmel's advanced nonvolatile CMOS technology, the device offers access times to 150 ns with power dissipation of just 220 mW. When the device is deselected, the CMOS standby current is less than 100 µA. *(continued)*

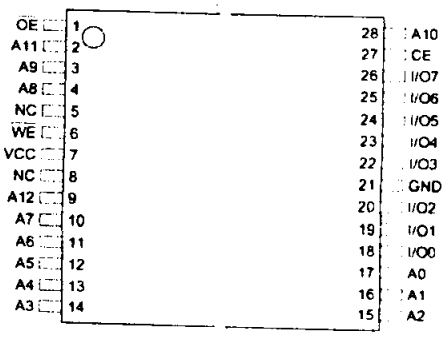
Pin Configurations

Pin Name	Function
A0 - A12	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
I/O0 - I/O7	Data Inputs/Outputs
NC	No Connect
DC	Don't Connect

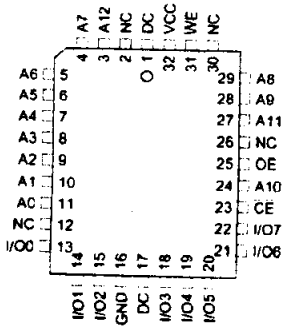
PDIP, SOIC
Top View



TSOP
Top View



PLCC
Top View



Note: PLCC package pins 1 and 17 are DON'T CONNECT.

64K (8K x 8) Parallel EEPROM with Page Write and Software Data Protection

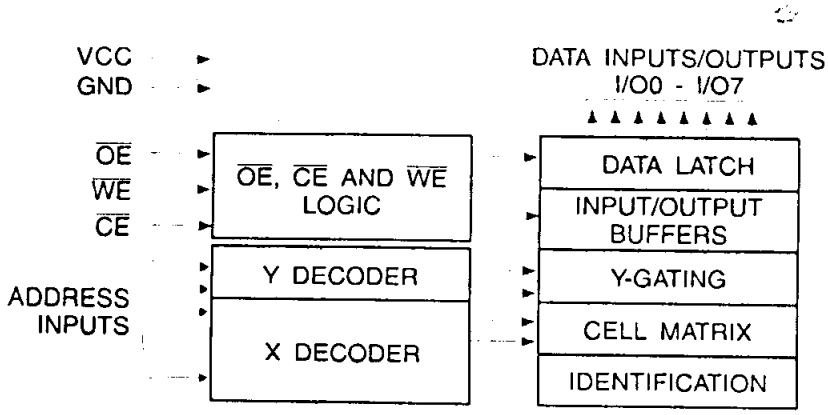
AT28C64B

Rev. 0270G–06/99

The AT28C64B is accessed like a Static RAM for the read or write cycle without the need for external components. The device contains a 64-byte page register to allow writing of up to 64 bytes simultaneously. During a write cycle, the addresses and 1 to 64 bytes of data are internally latched, freeing the address and data bus for other operations. Following the initiation of a write cycle, the device will automatically write the latched data using an internal control timer. The end of a write cycle can be detected by DATA POLLING of I/O_7. Once the end of a write cycle has been detected, a new access for a read or write can begin.

Atmel's AT28C64B has additional features to ensure high quality and manufacturability. The device utilizes internal error correction for extended endurance and improved data retention characteristics. An optional software data protection mechanism is available to guard against inadvertent writes. The device also includes an extra 64 bytes of EEPROM for device identification or tracking.

Block Diagram



Absolute Maximum Ratings*

Temperature Under Bias	-55°C to +125°C
Storage Temperature	-65°C to +150°C
All Input Voltages (including NC Pins) with Respect to Ground	-0.6V to +6.25V
All Output Voltages with Respect to Ground	-0.6V to V_{CC} + 0.6V
Voltage on $\overline{OE}$ and A9 with Respect to Ground	-0.6V to +13.5V

*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

Device Operation

READ: The AT28C64B is accessed like a Static RAM. When $\overline{CE}$ and $\overline{OE}$ are low and $\overline{WE}$ is high, the data stored at the memory location determined by the address pins is asserted on the outputs. The outputs are put in the high-impedance state when either $\overline{CE}$ or $\overline{OE}$ is high. This dual line control gives designers flexibility in preventing bus contention in their systems.

BYTE WRITE: A low pulse on the $\overline{WE}$ or $\overline{CE}$ input with $\overline{CE}$ or $\overline{WE}$ low (respectively) and $\overline{OE}$ high initiates a write cycle. The address is latched on the falling edge of $\overline{CE}$ or $\overline{WE}$, whichever occurs last. The data is latched by the first rising edge of $\overline{CE}$ or $\overline{WE}$. Once a byte write has been started, it will automatically time itself to completion. Once a programming operation has been initiated and for the duration of t_{WC}, a read operation will effectively be a polling operation.

PAGE WRITE: The page write operation of the AT28C64B allows 1 to 64 bytes of data to be written into the device during a single internal programming period. A page write operation is initiated in the same manner as a byte write; after the first byte is written, it can then be followed by 1 to 63 additional bytes. Each successive byte must be loaded within 150 µs (t_{BLC}) of the previous byte. If the t_{BLC} limit is exceeded, the AT28C64B will cease accepting data and commence the internal programming operation. All bytes during a page write operation must reside on the same page as defined by the state of the A6 to A12 inputs. For each $\overline{WE}$ high to low transition during the page write operation, A6 to A12 must be the same.

The A0 to A5 inputs specify which bytes within the page are to be written. The bytes may be loaded in any order and may be altered within the same load period. Only bytes which are specified for writing will be written; unnecessary cycling of other bytes within the page does not occur.

DATA POLLING: The AT28C64B features $\overline{DATA}$ Polling to indicate the end of a write cycle. During a byte or page write cycle an attempted read of the last byte written will result in the complement of the written data to be presented on I/O_7. Once the write cycle has been completed, true data is valid on all outputs, and the next write cycle may begin. $\overline{DATA}$ Polling may begin at any time during the write cycle.

TOGGLE BIT: In addition to $\overline{DATA}$ Polling, the AT28C64B provides another method for determining the end of a write cycle. During the write operation, successive attempts to read data from the device will result in I/O_6 toggling between one and zero. Once the write has completed, I/O_6 will stop toggling, and valid data will be read. Toggle bit reading may begin at any time during the write cycle.

DATA PROTECTION: If precautions are not taken, inadvertent writes may occur during transitions of the host system power supply. Atmel has incorporated both hardware and software features that will protect the memory against inadvertent writes.

HARDWARE DATA PROTECTION: Hardware features protect against inadvertent writes to the AT28C64B in the following ways: (a) V_{CC} sense—if V_{CC} is below 3.8V (typical), the write function is inhibited; (b) V_{CC} power-on delay—once V_{CC} has reached 3.8V, the device will automatically time out 5 ms (typical) before allowing a write; (c) write inhibit—holding any one of $\overline{OE}$ low, $\overline{CE}$ high, or $\overline{WE}$ high inhibits write cycles; and (d) noise filter—pulses of less than 15 ns (typical) on the $\overline{WE}$ or $\overline{CE}$ inputs will not initiate a write cycle.

SOFTWARE DATA PROTECTION: A software controlled data protection feature has been implemented on the AT28C64B. When enabled, the software data protection (SDP), will prevent inadvertent writes. The SDP feature may be enabled or disabled by the user; the AT28C64B is shipped from Atmel with SDP disabled.

SDP is enabled by the user issuing a series of three write commands in which three specific bytes of data are written to three specific addresses (refer to the *Software Data Protection Algorithm* diagram in this data sheet). After writing the 3-byte command sequence and waiting t_{WC}, the entire AT28C64B will be protected against inadvertent writes. It should be noted that even after SDP is enabled, the user may still perform a byte or page write to the AT28C64B by preceding the data to be written by the same 3-byte command sequence used to enable SDP.

Once set, SDP remains active unless the disable command sequence is issued. Power transitions do not disable SDP, and SDP protects the AT28C64B during power-up and power-down conditions. All command sequences must conform to the page write timing specifications. The data in the enable and disable command sequences is not actually written into the device; their addresses may still be written with user data in either a byte or page write operation.

After setting SDP, any attempt to write to the device without the 3-byte command sequence will start the internal write timers. No data will be written to the device. However, for the duration of t_{WC}, read operations will effectively be polling operations.

DEVICE IDENTIFICATION: An extra 64 bytes of EEPROM memory are available to the user for device identification. By raising A9 to 12V ± 0.5V and using address locations 1FC0H to 1FFFH, the additional bytes may be written to or read from in the same manner as the regular memory array.

DC and AC Operating Range

		AT28C64B-15	AT28C64B-20	AT28C64B-25
Operating Temperature (Case)	Com.	0°C - 70°C	0°C - 70°C	0°C - 70°C
	Ind.	-40°C - 85°C	-40°C - 85°C	-40°C - 85°C
V_{CC} Power Supply		5V ± 10%	5V ± 10%	5V ± 10%

Operating Modes

Mode	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	I/O
Read	V_{IL}	V_{IL}	V_{IH}	D_{OUT}
Write[2]	V_{IL}	V_{IH}	V_{IL}	D_{IN}
Standby/Write Inhibit	V_{IH}	X[1]	X	High Z
Write Inhibit	X	X	V_{IH}	
Write Inhibit	X	V_{IL}	X	
Output Disable	X	V_{IH}	X	High Z
Chip Erase	V_{IL}	V_H[3]	V_{IL}	High Z

Notes: 1. X can be V_{IL} or V_{IH}.
2. Refer to the *AC Write Waveforms* diagrams in this data sheet.
3. $V_H = 12.0V \pm 0.5V$.

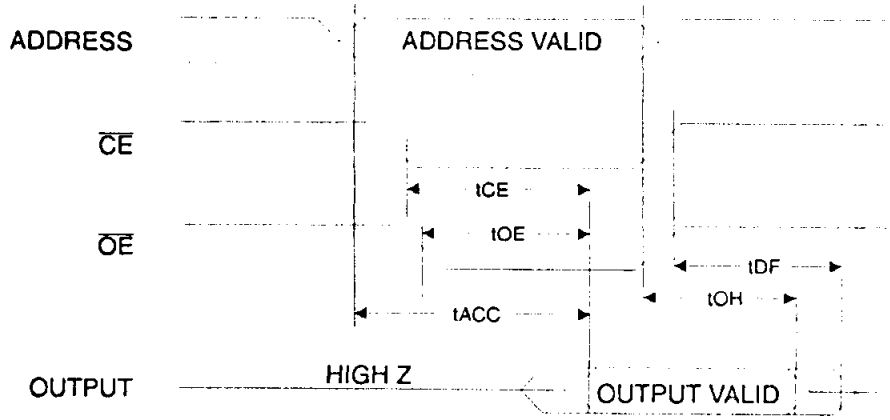
DC Characteristics

Symbol	Parameter	Condition	Min	Max	Units
I_{LI}	Input Load Current	$V_{IN} = 0V$ to $V_{CC} + 1V$		10	µA
I_{LO}	Output Leakage Current	$V_{I/O} = 0V$ to V_{CC}		10	µA
I_{SB1}	V_{CC} Standby Current CMOS	$\overline{CE} = V_{CC} - 0.3V$ to $V_{CC} + 1V$ Com., Ind.		100	µA
I_{SB2}	V_{CC} Standby Current TTL	$\overline{CE} = 2.0V$ to $V_{CC} + 1V$		2	mA
I_{CC}	V_{CC} Active Current	f = 5 MHz; $I_{OUT} = 0$ mA		40	mA
V_{IL}	Input Low Voltage			0.8	V
V_{IH}	Input High Voltage		2.0		V
V_{OL}	Output Low Voltage	$I_{OL} = 2.1$ mA		0.40	V
V_{OH}	Output High Voltage	$I_{OH} = -400$ µA	2.4		V

AC Read Characteristics

Symbol	Parameter	AT28C64B-15 Min	AT28C64B-15 Max	AT28C64B-20 Min	AT28C64B-20 Max	AT28C64B-25 Min	AT28C64B-25 Max	Units
t_{ACC}	Address to Output Delay		150		200		250	ns
t_{CE}[1]	$\overline{CE}$ to Output Delay		150		200		250	ns
t_{OE}[2]	$\overline{OE}$ to Output Delay	0	70	0	80	0	100	ns
t_{DF}[3][4]	$\overline{CE}$ or $\overline{OE}$ to Output Float	0	50	0	55	0	60	ns
t_{OH}	Output Hold from $\overline{OE}$, $\overline{CE}$ or Address, whichever occurred first	0		0		0		ns

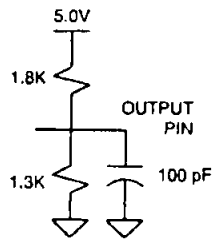
AC Read Waveforms[1][2][3][4]



Notes:
1. $\overline{CE}$ may be delayed up to $t_{ACC} - t_{CE}$ after the address transition without impact on t_{ACC}.
2. $\overline{OE}$ may be delayed up to $t_{CE} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE} or by $t_{ACC} - t_{OE}$ after an address change without impact on t_{ACC}.
3. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$ whichever occurs first (C_L = 5 pF).
4. This parameter is characterized and is not 100% tested.

Input Test Waveforms and Measurement Level

Output Test Load



Pin Capacitance

f = 1 MHz, T = 25°C[1]

Symbol	Typ	Max	Units	Conditions
C_{IN}	4	6	pF	V_{IN} = 0V
C_{OUT}	8	12	pF	V_{OUT} = 0V

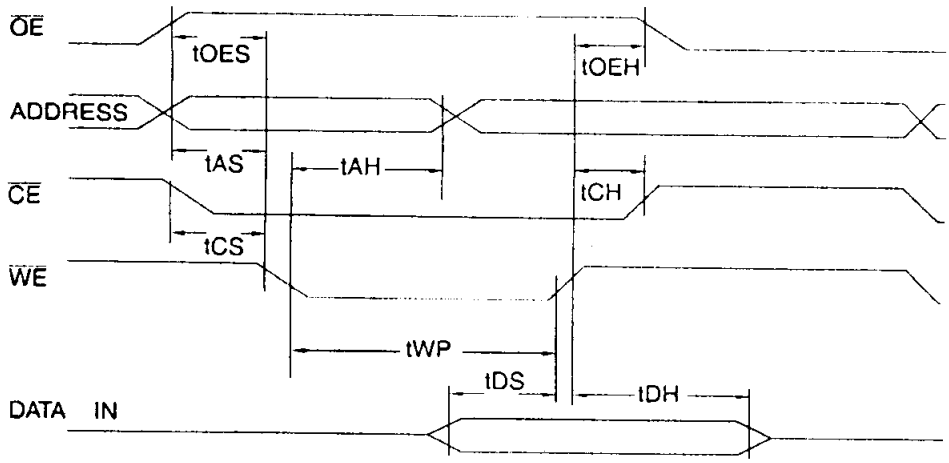
Note: 1. This parameter is characterized and is not 100% tested.

AC Write Characteristics

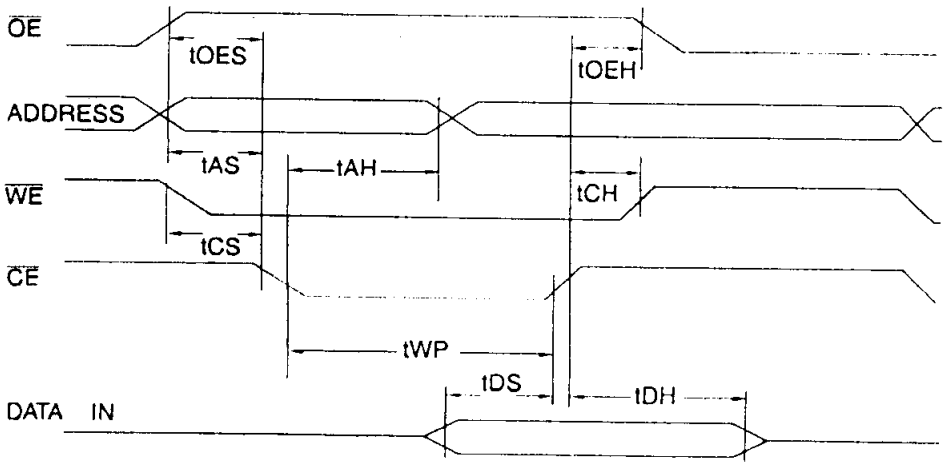
Symbol	Parameter	Min	Max	Units
t_{AS}, t_{OES}	Address, $\overline{OE}$ Set-up Time	0		ns
t_{AH}	Address Hold Time	50		ns
t_{CS}	Chip Select Set-up Time	0		ns
t_{CH}	Chip Select Hold Time	0		ns
t_{WP}	Write Pulse Width ($\overline{WE}$ or $\overline{CE}$)	100		ns
t_{DS}	Data Set-up Time	50		ns
t_{DH}, t_{OEH}	Data, $\overline{OE}$ Hold Time	0		ns

AC Write Waveforms

WE Controlled



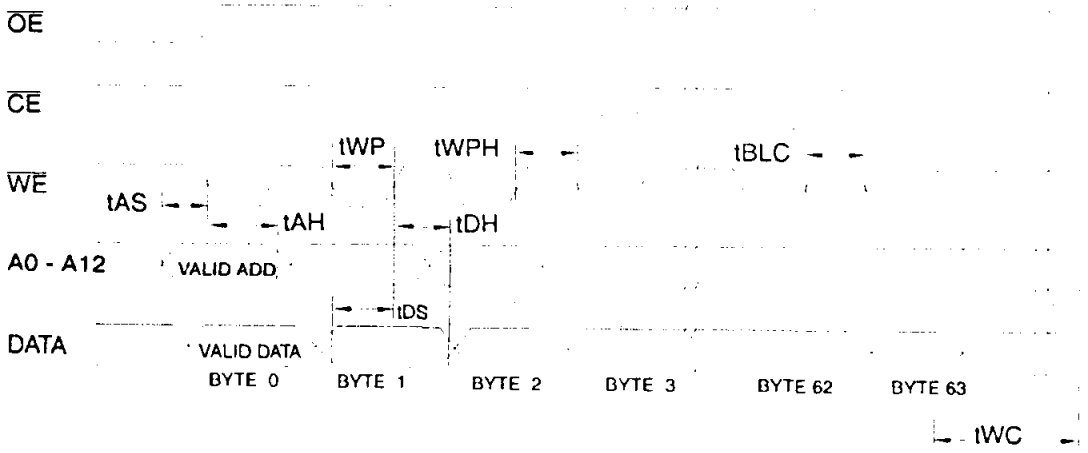
CE Controlled



Page Mode Characteristics

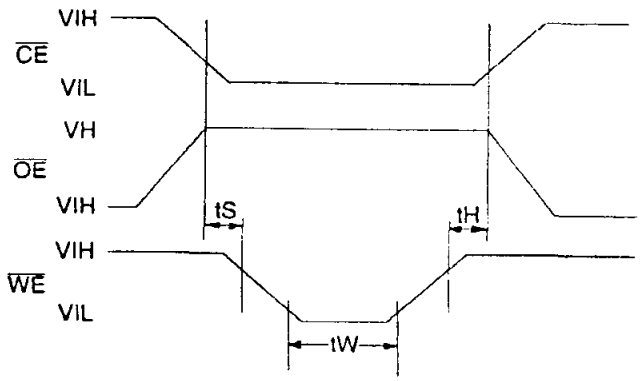
Symbol	Parameter	Min	Max	Units
t_{WC}	Write Cycle Time		10	ms
t_{WC}	Write Cycle Time (28C64B SL184)	0	2	ms
t_{AS}	Address Set-up Time	0		ns
t_{AH}	Address Hold Time	50		ns
t_{DS}	Data Set-up Time	50		ns
t_{DH}	Data Hold Time	0		ns
t_{WP}	Write Pulse Width	100		ns
t_{BLC}	Byte Load Cycle Time		150	µs
t_{WPH}	Write Pulse Width High	50		ns

Page Mode Write Waveforms[1][2]



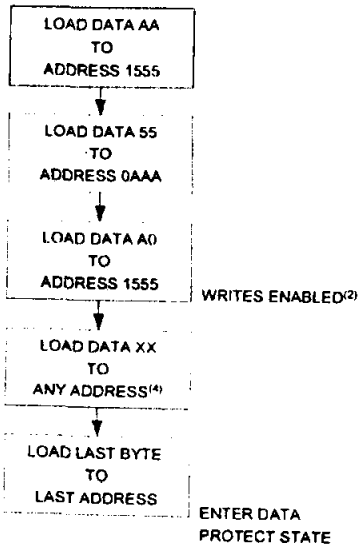
Notes:
1. A6 through A12 must specify the same page address during each high to low transition of $\overline{WE}$ (or $\overline{CE}$).
2. $\overline{OE}$ must be high only when $\overline{WE}$ and $\overline{CE}$ are both low.

Chip Erase Waveforms

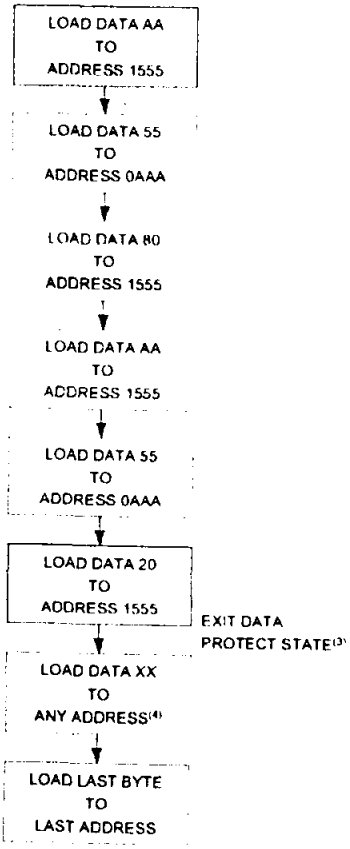


$t_S = t_H$ = 1 µsec (min.)
t_W = 10 msec (min.)
V_H = 12.0 ± 0.5V

Software Data Protection Enable Algorithm[1]



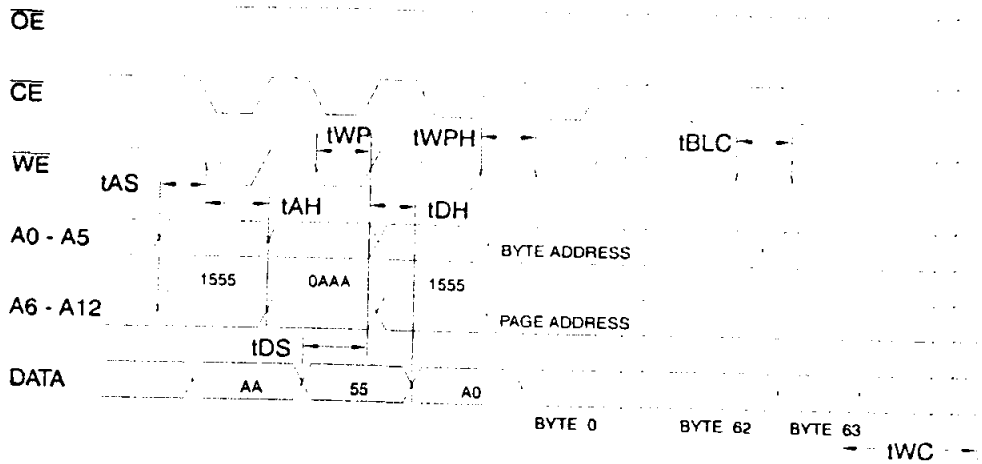
Software Data Protection Disable Algorithm[1]



Notes for software program code:

1. Data Format: I/O7 - I/O0 (Hex);
   Address Format: A12 - A0 (Hex).
2. Write Protect state will be activated at end of write even if no other data is loaded.
3. Write Protect state will be deactivated at end of write period even if no other data is loaded.
4. 1 to 64 bytes of data are loaded.

Software Protected Write Cycle Waveforms[1][2]



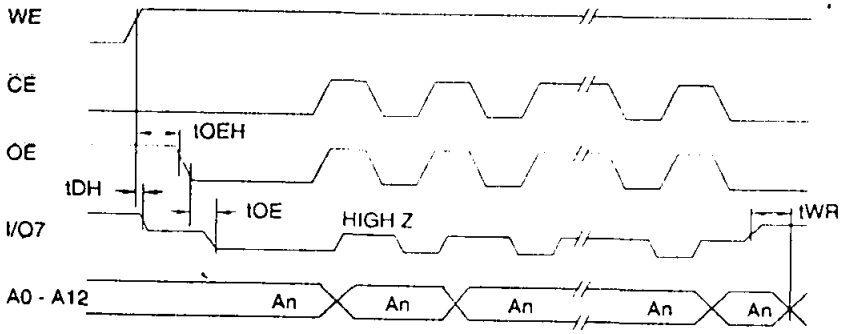
Notes:
1. A6 through A12 must specify the same page address during each high to low transition of $\overline{WE}$ (or $\overline{CE}$) after the software code has been entered.
2. $\overline{OE}$ must be high only when $\overline{WE}$ and $\overline{CE}$ are both low.

Data Polling Characteristics[1]

Symbol	Parameter	Min	Typ	Max	Units
t_{DH}	Data Hold Time	0			ns
t_{OEH}	$\overline{OE}$ Hold Time	0			ns
t_{OE}	$\overline{OE}$ to Output Delay[2]				ns
t_{WR}	Write Recovery Time	0			ns

Notes: 1. These parameters are characterized and not 100% tested.
2. See AC Read Characteristics.

Data Polling Waveforms

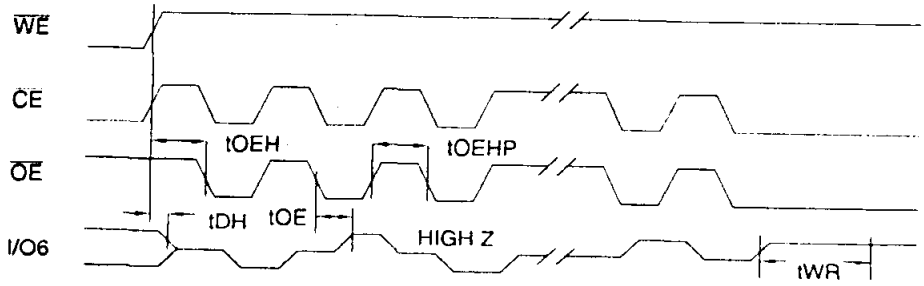


Toggle Bit Characteristics[1]

Symbol	Parameter	Min	Typ	Max	Units
t_{DH}	Data Hold Time	10			ns
t_{OEH}	$\overline{OE}$ Hold Time	10			ns
t_{OE}	$\overline{OE}$ to Output Delay[2]				ns
t_{OEHP}	$\overline{OE}$ High Pulse	150			ns
t_{WR}	Write Recovery Time	0			ns

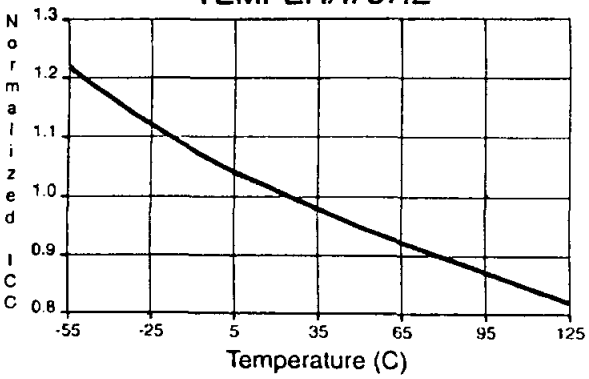
Notes: 1. These parameters are characterized and not 100% tested.
2. See AC Read Characteristics.

Toggle Bit Waveforms[1][2][3]

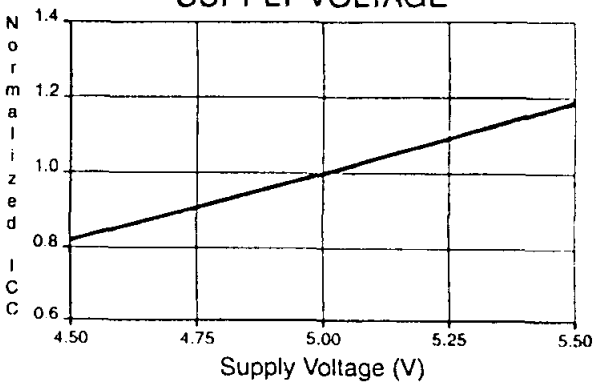


Notes: 1. Toggling either $\overline{OE}$ or $\overline{CE}$ or both $\overline{OE}$ and $\overline{CE}$ will operate toggle bit.
2. Beginning and ending state of I/O6 will vary.
3. Any address location may be used but the address should not vary.

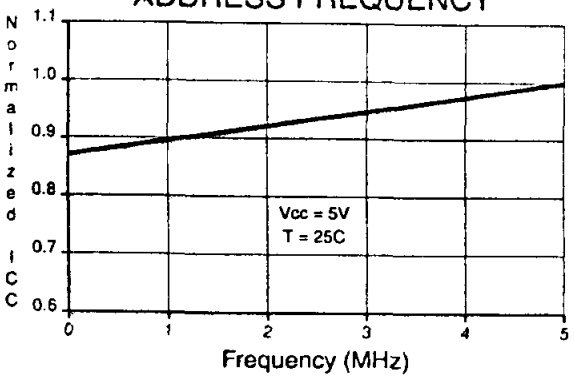
NORMALIZED SUPPLY CURRENT vs. TEMPERATURE



NORMALIZED SUPPLY CURRENT vs. SUPPLY VOLTAGE



NORMALIZED SUPPLY CURRENT vs. ADDRESS FREQUENCY



Ordering Information[1]

t_{ACC} (ns)	I_{CC} (mA)		Ordering Code	Package	Operation Range
	Active	Standby			
150	40	0.1	AT28C64B-15JC	32J	Commercial
			AT28C64B-15PC	28P6	(0°C to 70°C)
			AT28C64B-15SC	28S	
			AT28C64B-15TC	28T	
			AT28C64B-15JI	32J	Industrial
			AT28C64B-15PI	28P6	(-40°C to 85°C)
			AT28C64B-15SI	28S	
			AT28C64B-15TI	28T	
200	40	0.1	AT28C64B-20JC	32J	Commercial
			AT28C64B-20PC	28P6	(0°C to 70°C)
			AT28C64B-20SC	28S	
			AT28C64B-20TC	28T	
			AT28C64B-20JI	32J	Industrial
			AT28C64B-20PI	28P6	(-40°C to 85°C)
			AT28C64B-20SI	28S	
			AT28C64B-20TI	28T	
250	40	0.1	AT28C64B-25JC	32J	Commercial
			AT28C64B-25PC	28P6	(0°C to 70°C)
			AT28C64B-25SC	28S	
			AT28C64B-25TC	28T	
			AT28C64B-25JI	32J	Industrial
			AT28C64B-25PI	28P6	(-40°C to 85°C)
			AT28C64B-25SI	28S	
			AT28C64B-25TI	28T	

Note: 1. See Valid Part Numbers table below.

Valid Part Numbers

The following table lists standard Atmel products that can be ordered.

Device Numbers	Speed	Package and Temperature Combinations
AT28C64B	15	JC, JI, PC, PI, SC, SI, TC, TI
AT28C64B	20	JC, JI, PC, PI, SC, SI, TC, TI
AT28C64B	25	JC, JI, PC, PI, SC, SI, TC, TI
AT28C64B	-	W

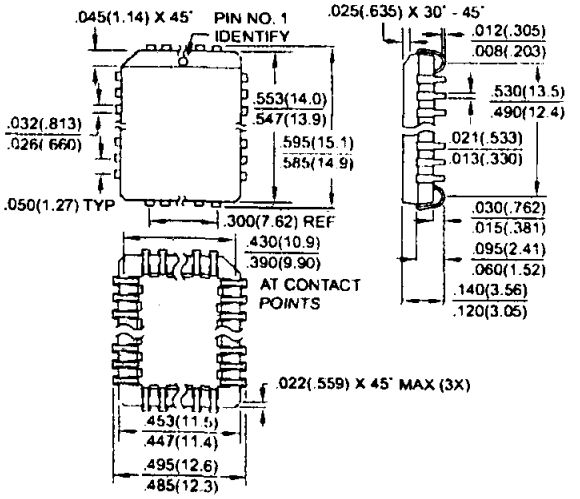
Die Products

Reference Section: Parallel EEPROM Die Products

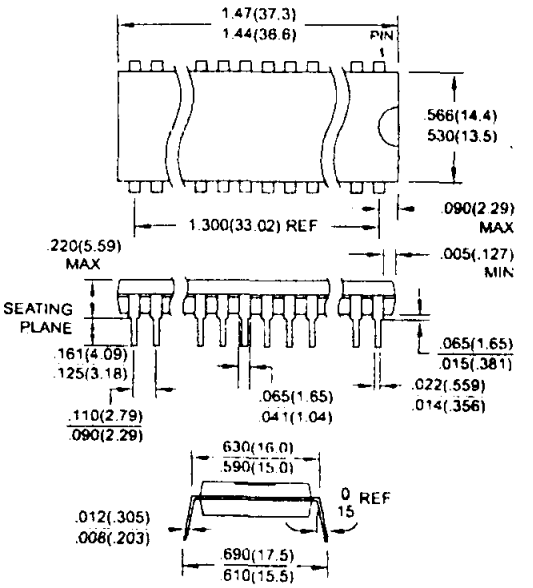
Package Type	
32J	32-lead, Plastic J-leaded Chip Carrier (PLCC)
28P6	28-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
28S	28-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)
28T	28-lead, Plastic Thin Small Outline Package (TSOP)
W	Die

Packaging Information

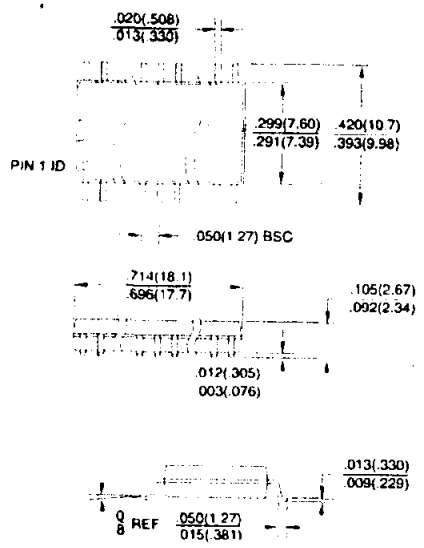
32J, 32-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-016 AE



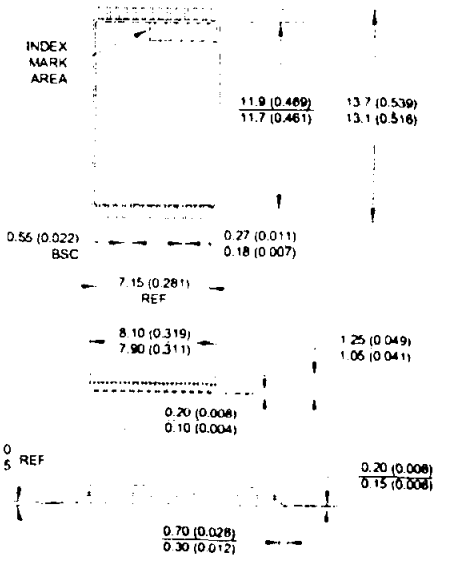
28P6, 28-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-011 AB



28S, 28-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)
Dimensions in Inches and (Millimeters)



28T, 28-lead, Plastic Thin Small Outline Package (TSOP)
Dimensions in Millimeters and (Inches)*



BIODATA

Nama	: WISNU YUWONO
NRP	: 5103097039
NIRM	: 97.7.003.31073.38715
Alamat	: JL. BROMO 76 KAUMAN
	KALANGBRET
	TULUNGAGUNG
Agama	: KATOLIK

Riwayat Pendidikan :

- SD Negeri Kauman I, Tulungagung, Tahun 1984-1990
- SMP Negeri I Kauman, Tulungagung, Tahun 1990-1993
- SMA Katolik Santo Thomas Aquino Tulungagung, Tahun 1993-1996
- Mahasiswa Universitas Katolik Widya Mandala Surabaya, Fakultas Teknik Jurusan Teknik Elektro, Angkatan 1997

Selama kuliah aktif sebagai :

- Anggota Himpunan Mahasiswa Jurusan Teknik Elektro periode 1997-1998 divisi seminar
- Wakil Ketua I PSM Cantate Domino Universitas Katolik Widya Mandala Surabaya periode 1998-1999
- Sekretaris I PSM Cantate Domino Universitas Katolik Widya Mandala Surabaya periode 1999-2000
- Asisten Praktikum Dasar Telekomunikasi