รายการอ้างอิง

1. Noltingk,B.E. Jones'Instrument Technology : Electrical and radiation measurements (vol.3), (4rd edt). Butterworth & Co. England :1987.
2. Umbarger,C.J. Small computerize multipopose radiation monitors. IEEE Transaction on nuclear science Vol30 (February 1983): 528-530.
3. Hearn ,William E. A Wideband Precision Analog Telemetry Link Using Digital Techniques. IEEE Transaction on nuclear science Vol30 (February 1983): 570-574.
4. อำนวย สุดสาคร. การพัฒนาเครื่องส่งสัญญาณนิวเคลียร์ระบบไร้สาย. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชานิวเคลียร์เทคโนโลยี บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย , 2537.
5. Virul Mangclaviraj , et al. Laboratory Manual for nuclear instrumentation experiments. Department of Nuclear Technology, Faculty of Engineering, Chulalongkorn University ,1989.
6. ฉัตรชัย สุมามาลย์. การสื่อสารข้อมูลคอมพิวเตอร์และระบบเครือข่าย. กรุงเทพมหานคร: ด่านสุธาการพิมพ์ , 2539.
7. บัณฑิต โรจน์อารยานนท์. หลักการ ไฟฟ้าสื่อสาร. กรุงเทพมหานคร: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย , 2541.
8. Gage, Stan., Evans, Dave. , et al. Optoelectronics applications manual. Hewlett-Packard , New York , USA. ,1987.

ภาคผนวก

ภาคผนวก ก.1

โปรแกรมทดสอบการทำงานของวงจรแปลงสัญญาณอะนาลอกเป็นสัญญาณดิจิตอล

```
          ORG   8000H
IC_ADC    EQU   0E04AH      ; port input of ADC0804
D_DSP     EQU   0E001H      ; port control LED flag
P_DSP     EQU   0E002H      ; port enable LED flag
FLAG      EQU   06H        ; data to enable LED flag
DSP       EQU   9F00H       ; display buffer (7 segment)
BUF       EQU   9F07H       ; display buffer (BCD)


     ;******************
     ; MAIN PROGRAM  *
     ;******************

IN_ADC:   MOV   DPTR,#IC_ADC  ; point to input port
          MOVX  A,@DPTR       ; read input (from ADC0804)
          MOV   R0,A        ; save input
          LCALL  DSP_FLAG     ; display input to LED flag
          LCALL  DSP_SEG     ; display input to 7 segment
DELAY:     MOV   R6,#0FH      ; delay time
DELAY1:    DJNZ  R6,DELAY1
           AJMP  IN_ADC       ; loop for read input continue


      ;********************************
      ;  SUB. DISPLAY LED FLAG   *
      ;********************************

DSP_FLAG:  MOV   DPTR,#P_DSP   ; set flag active
           MOV   A,#FLAG
           MOVX  @DPTR,A      ; enable flag
           MOV   DPTR,#D_DSP
           MOV   A,R0        ; get input
          MOVX  @DPTR,A      ; out data to flag
          RET


      ;********************************
      ;  SUB. DISPLAY  7 SEGMENT  *
      ;********************************
DSP_SEG:  MOV   DPTR,#BUF+2
```

```
MOV   A,R0          ; get input
MOVX  @DPTR,A        ; store BUF+2
MOV   R2,A          ; store input HEX_BCDS
MOV   R3,#00H
MOV   A,#14H        ; call HEX_BCDS
LCALL 0030H         ; HEX -> DEC
MOV   DPTR,#BUF+1   ; store input UNPACK
MOV   A,R3
MOVX  @DPTR,A
MOV   DPTR,#BUF+0
MOV   A,R2
MOVX  @DPTR,A
MOV   A,#32H        ; call "UNPACK"
LCALL 0030H         ; BUF -> DSP
MOV   DPTR,#DSP+1   ; shift digit to left
MOVX  A,@DPTR
MOV   DPTR,#DSP+0
MOVX  @DPTR,A
MOV   DPTR,#DSP+2
MOVX  A,@DPTR
MOV   DPTR,#DSP+1
MOVX  @DPTR,A
MOV   DPTR,#DSP+3
MOVX  A,@DPTR
MOV   DPTR,#DSP+2
MOVX  @DPTR,A
MOV   DPTR,#DSP+3
MOV   A,#00H
MOVX  @DPTR,A
MOV   A,#0AH        ; call "DISPLAY"
LCALL 0030H
RET
END
```

ภาคผนวก ก.2

โปรแกรมทดสอบการทำงานของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอะนาลอก

```
ORG 8000H
IC_DAC    EQU 0E048H
DSP       EQU 9F00H        ; buffer display (7 seg.)
BUF       EQU 9F07H        ; buffer display (BCD)
HEXBUF    EQU 22H          ; data (HEX) buffer
DECBUF    EQU 23H          ; data (DEC) buffer
KEYBUF    EQU 25H          ; key buffer (7 seg.)


      ;********************
      ;*  MAIN PROGRAM  *
      ;********************

START:    MOV   HEXBUF,#00H   ; initial first display
          MOV   DECBUF,#00H
SETUP:    MOV   DPTR,#BUF+1
          MOV   A,DECBUF
          MOVX  @DPTR,A       ; input HEX
          MOV   DPTR,#BUF+2
          MOV   A,HEXBUF
          MOVX  @DPTR,A       ; input DEC
          MOV   A,#32H        ; call "UNPACK"
          LCALL 0030H         ; BUF -> DSP
          MOV   DPTR,#DSP+0
          MOVX  A,@DPTR
          ORL   A,#80H        ; set dot
          MOVX  @DPTR,A
          MOV   DPTR,#DSP+2   ; off DSP
          MOV   A,#00H
          MOVX  @DPTR,A
          MOV   DPTR,#DSP+3
          MOVX  @DPTR,A
SCANKEY:  MOV   A,#25H        ; call "SCAND"
          LCALL 0030H         ; scan DSP & KEY
          CJNE  A,#13H,CHECK1 ; check Enter key
          LCALL BEEP
          AJMP  ENDSCAN       ; if Enter -> ENDSCAND
CHECK1:   CJNE  A,#0AH,CHECK2 ; check 0-9 only
          AJMP  SCANKEY
CHECK2:   JNC   SCANKEY       ; over 9 -> SCANKEY
```

```
          LCALL  GET2        ; check 2 input
          LCALL  BEEP
          LCALL  DELAY
          AJMP   SCANKEY
ENDSCAN:  MOV    DPTR,#DSP+0
          MOVX   A,@DPTR
          ANL    A,#7FH      ; clear dot
          MOV    A,#28H      ; call "SCANPACK"
          LCALL  0030H       ; DSP -> BUF
          MOV    DPTR,#BUF+1
          MOVX   A,@DPTR
          CJNE   A,#51H,CHECK3
CHECK3:   JNC    SETUP
          MOV    R2,A
          MOV    DECBUF,A    ; save data display
          MOV    A,#05H      ; call "BCD_HEXS"
          LCALL  0030H       ; DEC -> HEX
          MOV    DPTR,#OUTTAB
          MOV    A,R2
          ADD    A,DPL
          MOV    DPL,A
          MOVX   A,@DPTR     ; data out
          MOV    HEXBUF,A    ; save data out
          MOV    DPTR,#IC_DAC ; out port DAC0832
          MOVX   @DPTR,A
          LCALL  DELAY
          LJMP   SETUP


       ;************************
       ;*** GET 2-DIGIT SUB. ***
       ;************************

GET2:     MOV    DPTR,#SEGTAB ; address table
          ADD    A,DPL       ; low byte + key code
          MOV    DPL,A       ; keep low byte address
          CLR    A
          ADDC   A,DPH
          MOV    DPH,A       ; keep high byte address
          MOVX   A,@DPTR     ; 7 code of key
          MOV    KEYBUF,A    ; save 7 code of key
DIGIT2:   MOV    DPTR,#DSP+1 ; point to digit 2
          MOVX   A,@DPTR
          CJNE   A,#00H,DIGIT1
          MOV    A,KEYBUF
          MOVX   @DPTR,A     ; write digit 2
          AJMP  ENDGET
DIGIT1:   MOV   A,#09H        ; DEC DPTR
          LCALL 0030H         ; point to digit 1
          MOV   A,KEYBUF
          ORL   A,#80H        ; set dot display
          MOVX  @DPTR,A       ; write digit 1
          INC   DPTR
          MOV   A,#00H
           MOVX  @DPTR,A
ENDGET:    RET

BEEP:     MOV   A,#18H        ; call "HI_BEEP"
          LCALL 0030H
           RET
DELAY:     MOV   R2,#00H
DELAY1:     MOV   R3,#00H
DELAY2:     DJNZ  R3,DELAY2
          DJNZ  R2,DELAY1
          RET

SEGTAB:    DB    3FH,06H,5BH,4FH,66H  ; 0 1 2 3 4
          DB    6DH,7DH,07H,7FH,6FH  ; 5 6 7 8 9

OUTTAB: DB  000H,005H,00AH,00FH,014H
          DB  019H,01FH,024H,029H,02EH
          DB  033H,038H,03DH,042H,047H
          DB  04DH,052H,057H,05CH.061H
          DB  066H,06BH,070H.075H.07AH
          DB  07FH,085H,08AH,08FH.094H
           DB  099H,09EH.0A3H.0A8H.0AEH
          DB  0B3H,0B8H,0BDH,0C2H.0C7H
           DB  0CCH,0D1H,0D6H,0DBH,0E1H
          DB  0E6H,0EBH,0F0H,0F5H.0FAH
          DB  0FFH
          END
```

ภาคผนวก ก.3

โปรแกรมทดสอบหาขีดจำกัดของระยะการทำงานของระบบรับส่งข้อมูลด้วยแสงอินฟราเรด

```
                org     8000h
                mov     SCON,#01010010b
                mov     TMOD,#00100000b
                mov     TH1,#11111101b
                setb    TR1
rx:             jnb     RI,$
                clr     RI
                mov     a,SBUF
                cjne    a,#0dh,tx0
                mov     dptr,#table
tx:             movx    a,@dptr
                lcall   tx1
                inc     dptr
                cjne    a,#0ah,tx
                sjmp    rx
tx0:            lcall   tx1
                mov     a,#'@'          ;Data = 20h
                lcall   tx1
                sjmp    rx
table:          db      'Program Test RS-232',0ah

                ;sub program send data 1byte to RS-232
tx1:            jnb     TI,$
                clr     TI
                mov     SBUF,a
                ret
```

ภาคผนวก ก.4

โปรแกรมทดสอบการใช้งานระบบรับ/ส่งข้อมูลในรูปแบบของเรตมิเตอร์

(1) โปรแกรมสำหรับอุปกรณ์รับ/ส่งข้อมูลต้นทาง

```
        ;*************************************
        ;* PROGRAM Ratemeter drive through     *
        ;* TO RS-232  out to com2 at 1200bps   *
        ;*************************************
               ;****REGISTER DECLARE****
               ORG    8000H        ; program start address
D_DSP          EQU    0E001H       ; port control LED flag
P_DSP          EQU    0E002H       ; port enable LED flag
IC_ADC         EQU    0E04AH       ; port input of ADC0804
FLAG           EQU    06H          ; data to enable LED flag
DSP            EQU    9F00H        ; display buffer (7 segment)
BUF            EQU    9F07H        ; display buffer (BCD)


         ;******************
         ;  MAIN PROGRAM  *
         ;******************
               ACALL  ST_RS232
STRT:          ACALL  RX_RS232
               CJNE   A,#12H,CHK_ENT   ;If R0:=#12H(ctrl-r) then reset
               MOV    A,#01H           ;call RESET sub routine
               LCALL  30H
CHK_ENT: CJNE  A,#13H,STRT
               ACALL  BEEP1
               MOV    DPTR,#IC_ADC  ;Point to input port
               MOVX   A,@DPTR       ;Read input form ADC0804
               MOV    R0,A          ;Save data for flag display
               ACALL  TX_RS232      ; send data to RS-232
               ACALL  DSP_FLAG      ; display input to LED flag
               ACALL  DSP_SEG       ; display input to 7 segment
               SJMP   STRT          ; loop for read command to send again
         ;=====================================
               ; 2 BEEP!! Gennerate
         ;=====================================
BEEP1:         PUSH   01H
               PUSH   02H
               PUSH   03H
               MOV    R1,#30H ;SET FREQUENCY 1
```

```
          MOV   R2,#FFH ;SET TIME
          MOV   A,#2CH  ;CALL SUB ROUTEEN
          LCALL  30H
          POP   03H
          POP   02H
          POP   01H
          RET
      ;*********************************
      ;   SUB.  DISPLAY LED FLAG    *
      ;*********************************

DSP_FLAG:  PUSH  ACC
          MOV   DPTR,#P_DSP   ; set flag active
          MOV   A,#FLAG
          MOVX  @DPTR,A       ; enable flag
          MOV   DPTR,#D_DSP
          MOV   A,R0          ; get input
          MOVX  @DPTR,A       ; out data to flag
          POP   ACC
          RET

      ;*********************************
      ;   SUB  DISPLAY  7 SEGMENT   *
      ;*********************************

DSP_SEG:  MOV   DPTR,#BUF+2
          MOV   A,R0          ; get input
          MOVX  @DPTR,A       ; store BUF+2
          MOV   R2,A          ; store input HEX_BCDS
          MOV   R3,#00H
          MOV   A,#14H        ; call HEX_BCDS
          LCALL 0030H         ; HEX -> DEC
          MOV   DPTR,#BUF+1   ; store input UNPACK
          MOV   A,R3
          MOVX  @DPTR,A
          MOV   DPTR,#BUF+0
          MOV   A,R2
          MOVX  @DPTR,A
          MOV   A,#32H        ; call "UNPACK"
          LCALL 0030H         ; BUF -> DSP
          MOV   DPTR,#DSP+1   ; shift digit to left
          MOVX  A,@DPTR
          MOV   DPTR,#DSP+0
          MOVX  @DPTR,A
          MOV   DPTR,#DSP+2
          MOVX  A,@DPTR
```

```
            MOV   DPTR,#DSP+1
            MOVX  @DPTR,A
            MOV   DPTR,#DSP+3
            MOVX  A,@DPTR
            MOV   DPTR,#DSP+2
            MOVX  @DPTR,A
            MOV   DPTR,#DSP+3
            MOV   A,#00H
            MOVX  @DPTR,A
            MOV   A,#0AH       ; call "DISPLAY"
            LCALL 0030H
            RET
      ;*************************************
      ;  Delay time = 256*256 uS.
      ;*************************************
DELAY:      PUSH  00H
            PUSH  01H
            MOV   R0,#FFH
NEXT1:      MOV   R1,#FFH
            DJNZ  R1,$
            DJNZ  R0,NEXT1
            POP   01H
            POP   00H
            RET
      ;*************************************
      ; sub. set port RS-232
      ;*************************************
ST_RS232:   MOV    SCON,#01010010B ;set SCON Mode 1
            MOV    TMOD,#00100000B ;8bit Auto reload
            MOV    TH1,#11101000B  ;SET BAUDRATE:= 1200 BPS
            SETB   TR1
            RET
TX_RS232:   JNB    TI,$  ;send data to RS-232 (R0:=input)
            CLR    TI
            MOV    SBUF,A
            RET
RX_RS232:   JNB    RI,$   ;recive data from RS-232 (A:=output)
            CLR    RI
            MOV    A,SBUF
            RET
END
```

(2) โปรแกรมสำหรับอุปกรณ์รับส่งข้อมูลปลายทาง

```
program sh;
Uses crt,dos;
const
  setport = $2f8;
Var
ch,n,I,T,cfac : integer;
inchr: char;
done : boolean;
procedure ShowTime;
 var
  h, m, s, hund : Word;
  function LeadingZero(w : Word) : String;
   var
    s : String;
   begin
    Str(w:0,s);
    if Length(s) = 1 then
    s := '0' + s;
    LeadingZero := s;
   end;
 begin
  GetTime(h,m,s,hund);
  GotoXY(25,10);
  Writeln('Start time ',LeadingZero(h),':',
        LeadingZero(m),':',LeadingZero(s){,
        ':',LeadingZero(hund)});
end;
procedure ShowTime2;
var
 h, m, s, hund : Word;
 function LeadingZero(w : Word) : String;
  var
   s : String;
  begin
   Str(w:0,s);
   if Length(s) = 1 then
   s := '0' + s;
   LeadingZero := s;
  end;
begin
 GetTime(h,m,s,hund);
 GotoXY(25,11);
 Writeln('It is now ',LeadingZero(h),':',
       LeadingZero(m),':',LeadingZero(s){,
      '.',LeadingZero(hund)});
 end;
procedure workindc;  {show status}
begin
   gotoxy(25,13);
   writeLn('GET DATA INPUT from SERIAL port..\..');
   delay(50);
   gotoxy(25,13);
   writeLn('GET DATA INPUT from SERIAL port..|..');
   delay(50);
   gotoxy(25,13);
   writeLn('GET DATA INPUT from SERIAL port../..');
   delay(50);
end;
Begin
  clrscr;
  GotoXY(20,2);
  writeLn('**** Program RATE METER ***** ');
  GotoXY(20,3);
  write('Enter a calibration factor...');
  readln(cfac);
  GotoXY(20,4);
  write('Press any key to start...');
  readln;
  {repeat }
  clrscr;
  GotoXY(20,4);
  writeLn('RATE METER running now !');
  GotoXY(20,5);
  writeLn('PRESS ANY KEY TO STOP !');
  ShowTime;
  repeat
  Showtime2;
  Port[setport]:=$13;  {ctrl-S}
  workindc;
  I:=(cfac)*Port[setport]; {multiply with calibration factor}
  gotoxy(25,14);
  write(I,'  ');
  gotoxy(32,14);
  write('Relaytive Counts.');
   {if i>=((cfac)*25) then
   begin
   write('worning! overflow count rate');
   read;
   end
   else}
  delay(100);
  until keypressed;
        gotoxy(25,15);
        Write('OK1..recived data Completed');
        Readln;
        clrscr;

End.
```

ภาคผนวก ก.5

โปรแกรมทดสอบการใช้งานระบบรับ/ส่งข้อมูลในรูปแบบของอุปกรณ์วิเคราะห์ชนิดช่องเดียว

(1) โปรแกรมสำหรับอุปกรณ์รับส่งข้อมูลต้นทาง

```
    ;****************************************************************
   ;*  PROGRAM SWEEP OUTPUT VOLTAGE    0-10 Vdc for LLD controll by send  *
   ;*  $14  (14H or CTRL-S) for each step and out data from rate meter to PC        *
   ;*  by send $13 (13H or CTRL-T) . TO Reset this program,                          *
   ;*  send $12 (12H or CTRL-R) from PC    out to com2 at 1200bps                    *
    ;****************************************************************
          ;****REGISTER DECLARE****
          ORG    8000H        ; program start address
D_DSP     EQU    0E001H        ; port control LED flag
P_DSP     EQU    0E002H        ; port enable LED flag
IC_ADC    EQU    0E04AH        ; port input of ADC0804
IC_DAC    EQU    0E048H        ; port output to DAC0832
FLAG      EQU    06H          ; data to enable LED flag
DSP       EQU    9F00H         ; display buffer (7 segment)
BUF       EQU    9F07H         ; display buffer (BCD)


     ;******************
     ;  MAIN PROGRAM   *
     ;******************
          ACALL  ST_RS232
          MOV    R5,#00H       ;clear ramp data
STRT:     ACALL  RX_RS232
          CJNE   A,#12H,POUT   ;If A:=#12H(ctrl-r) then reset
          MOV    A,#01H        ;call RESET sub routine
          LCALL  30H

POUT:     CJNE   A,#13H,SNDBCK ; if A:=#13H then out ramp
          ACALL  BEEP1
          PUSH   ACC           ;save A
          MOV    A,R5
          MOV    DPTR,#IC_DAC  ;Point to output port
          MOVX   @DPTR,A       ;send data to output port
          INC    R5
          POP    ACC
          NOP
          SJMP   STRT
```

```
SNDBCK: CJNE  A,#14H,STRT
        PUSH  ACC
        ACALL BEEP2
        MOV   DPTR,#IC_ADC
        MOVX  A,@DPTR       ;Read input form ADC0804
        MOV   R0,A        ;Save data for flag display
        ACALL TX_RS232     ; send data to RS-232
        ACALL DSP_FLAG     ; display input to LED flag
        ACALL DSP_SEG      ; display input to 7 segment
        POP   ACC
        CJNE  R5,#00H,GO
        MOV   A,#01H
        LCALL 30H
        NOP
GO:     SJMP  STRT        ; loop for read command to send again

;==========================================
. 2 BEEPII Gennerate type 1
;==========================================
BEEP1:  PUSH  01H
        PUSH  02H
        PUSH  03H
        MOV   R1,#45H ;SET FREQUENCY 1
        MOV   R2,#FFH ;SET TIME
        MOV   A,#2CH  ;CALL SUB ROUTINE
        LCALL  30H
        POP   03H
        POP   02H
        POP   01H
        RET
  ;==========================================
  ; 2 BEEPII Gennerate type2
  ;==========================================
BEEP2:  PUSH  01H
        PUSH  02H
        PUSH  03H
        MOV   R1,#4BH ;SET FREQUENCY 1
        MOV   R2,#FFH ;SET TIME
        MOV   A,#2CH  ;CALL SUB ROUTINE
        LCALL  30H
        POP   03H
        POP   02H
        POP   01H
        RET
        ;*********************************
        ;   SUB.  DISPLAY LED FLAG    *
        ;*********************************


DSP_FLAG: PUSH   ACC
          MOV   DPTR,#P_DSP   ; set flag active
          MOV   A,#FLAG
          MOVX  @DPTR,A       ; enable flag
          MOV   DPTR,#D_DSP
          MOV   A,R0          ; get input
          MOVX  @DPTR,A       ; out data to flag
          POP   ACC
          RET

        ;*********************************
        ;   SUB. DISPLAY  7 SEGMENT   *
        ;*********************************

DSP_SEG:  MOV   DPTR,#BUF+2
          MOV   A,R0          ; get input
          MOVX  @DPTR,A       ; store BUF+2
          MOV   R2,A          ; store input HEX_BCDS
          MOV   R3,#00H
          MOV   A,#14H        ; call HEX_BCDS
          LCALL 0030H         ; HEX -> DEC
          MOV   DPTR,#BUF+1   ; store input UNPACK
          MOV   A,R3
          MOVX  @DPTR,A
          MOV   DPTR,#BUF+0
          MOV   A,R2
          MOVX  @DPTR,A
          MOV   A,#32H        ; call "UNPACK"
          LCALL 0030H         ; BUF -> DSP
          MOV   DPTR,#DSP+1   ; shift digit to left
          MOVX  A,@DPTR
          MOV   DPTR,#DSP+0
          MOVX  @DPTR,A
          MOV   DPTR,#DSP+2
          MOVX  A,@DPTR
          MOV   DPTR,#DSP+1
          MOVX  @DPTR,A
          MOV   DPTR,#DSP+3
          MOVX  A,@DPTR
          MOV   DPTR,#DSP+2
          MOVX  @DPTR,A
          MOV   DPTR,#DSP+3
          MOV   A,#00H
          MOVX  @DPTR,A
          MOV   A,#0AH       ; call "DISPLAY"
          LCALL  0030H
          RET
     ;***************************************
     ;  Delay time = 256*256 uS.
     ;***************************************
DELAY:    PUSH  00H
          PUSH  01H
          MOV   R0,#FFH
NEXT1:    MOV   R1,#FFH
          DJNZ  R1,$
          DJNZ  R0,NEXT1
          POP   01H
          POP   00H
          RET
      ;***************************************
          ; sub. set port RS-232
      ;***************************************

ST_RS232: MOV    SCON,#01010010B ;set SCON Mode 1
          MOV    TMOD,#00100000B ;8bit Auto reload
          MOV    TH1,#11101000B  ;SET BAUDRATE:= 1200 BPS
          SETB   TR1
          RET

TX_RS232: JNB    TI,$   ;send data to RS-232 (R0:=input)
          CLR         TI
          MOV         SBUF,A
          RET

RX_RS232: JNB    RI,$   ;recive data from RS-232 (A:=output)
          CLR         RI
          MOV         A,SBUF
          RET
END
```

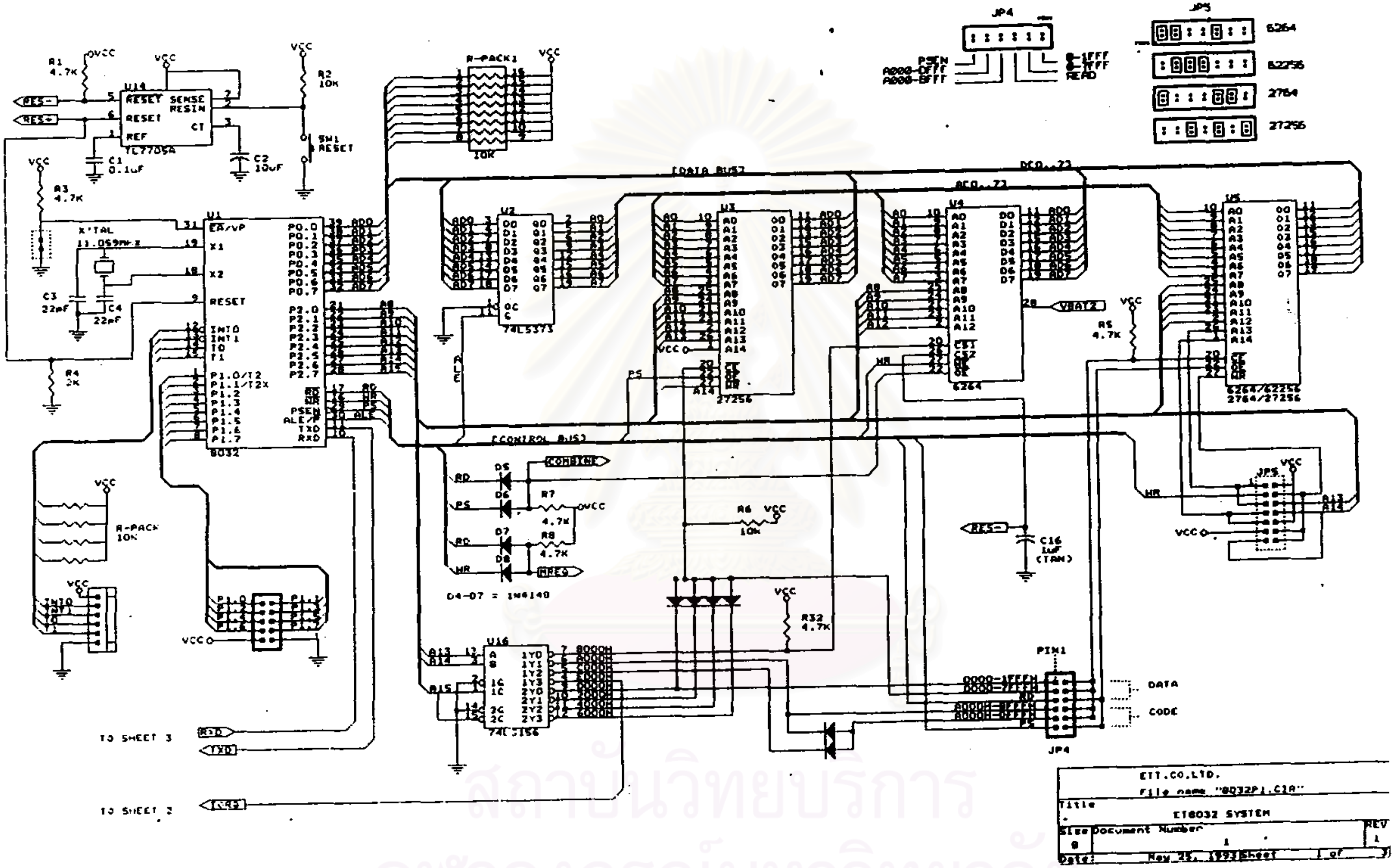
(2) โปรแกรมสำหรับอุปกรณ์รับส่ง/ข้อมูลปลายทาง

```
Program XY_SCAN;
Uses Crt,Graph;

Const
   Pattern  :  FillPatternType = ($FF,$FF,$FF,$FF,$FF,$FF,$FF,$FF);
   Setport  =  $2f8;(*com2*)
   Null     =  chr($00);
Var
    Max     : integer;
    MaxX    : integer;
    MaxY    : integer;
    Count   : integer;
    Loop    : integer;
    WhereX  : integer;
    WhereY  : integer;
    Free    : integer;
    StepY   : array[0..257] of integer; {304}
    Channal : integer;
    Ch      : char;
    strloop : string[4];
    Logo    : string[33];

Procedure Beep;
Begin
  Write(chr(7));
End;

Procedure Boom;
Begin
  for count:=500 to 3000 do
    Begin
      Sound(count);
      Delay(1);
    End;
    Nosound;
End;

Procedure Tick;
Begin
  count:=1500;
  while count<=2500 do
    Begin
      sound(count);
      delay(1);
      count:=count+100;
     End;
   nosound;
 end;

 Procedure Tock;
 Begin
   count:=500;
   while count<=1500 do
    Begin
      sound(count);
      delay(1);
      count:=count+100;
    End;
   nosound;
 end;

 Procedure GraphInit;
 Var     Gr,Gm : Integer;
 Begin
    Gr:=Detect;
    InitGraph (Gr, Gm, 'c:\dosappli\tp\bgi');
    MaxX:=GetMaxX;
    MaxY:=GetMaxY;
 End;

 Procedure SetGrp (X1,Y1,X2,Y2,Color : integer);
 Begin
    SetFillPattern (Pattern, Color);
    Bar ((X1*2)+15, 380-Y1, (X2*2)+15, 380-Y2);
 End;

 Procedure Setline (X1,Y1,X2,Y2,Color : integer);
 Begin
    Setcolor(Color);
    Line ((X1*2)+15, 380-Y1, (X2*2)+15, 380-Y2);
 End;


 Procedure SetBar (X1,Y1,X2,Y2,Color : integer);
 Begin
    SetFillPattern (Pattern, Color);
    Bar (X1, Y1, X2, Y2);
 End;

 Procedure SetScale(Color:integer);
```

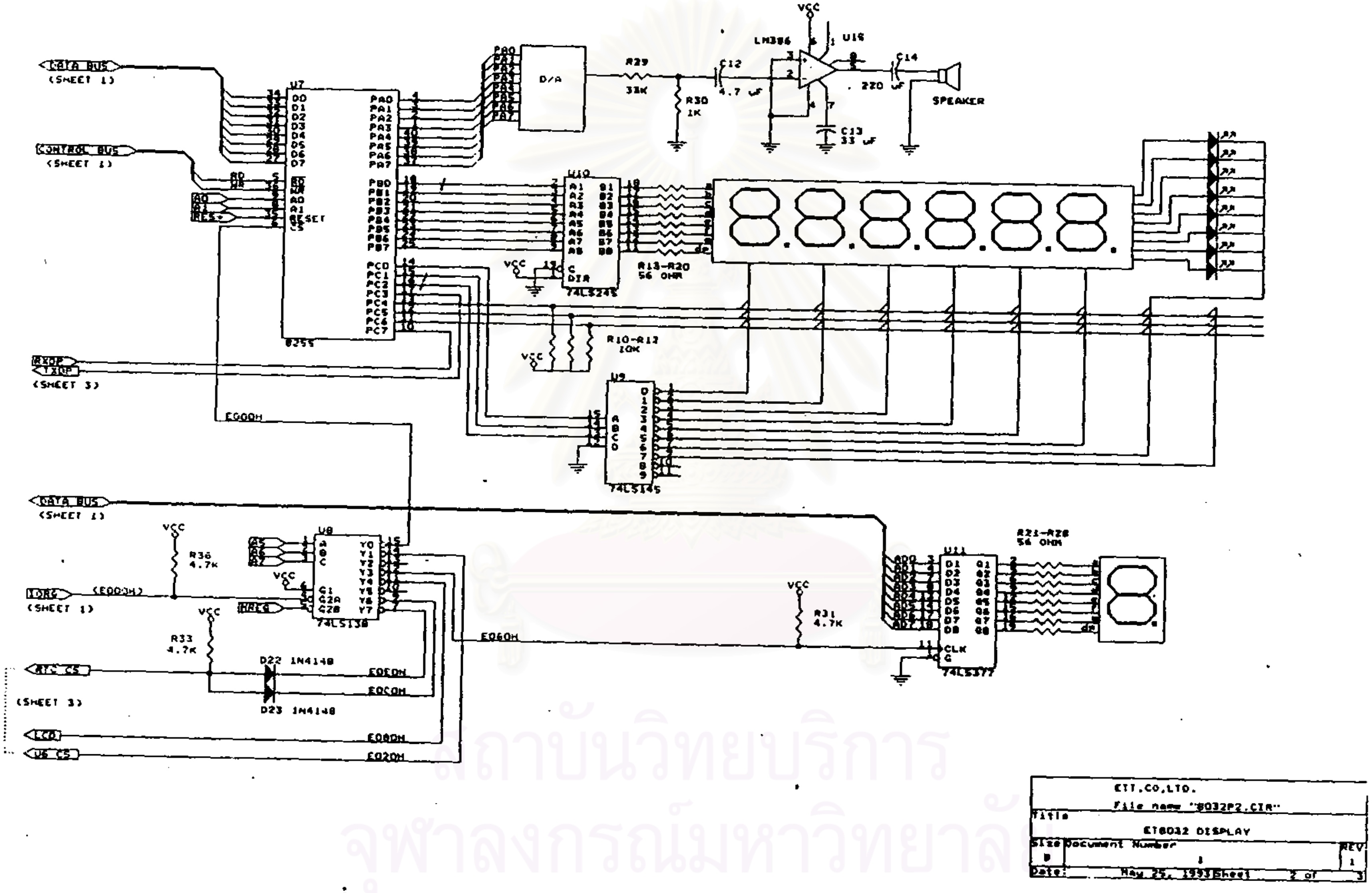
```
Begin
   SetColor(Color);
   Line((MaxX+1) div 8 * 7, 15, (MaxX+1) div 8 * 7, MaxY-61);  {}
   Line((MaxX+1) div 8 * 6, 15, (MaxX+1) div 8 * 6, MaxY-61);  {}
   Line((MaxX+1) div 8 * 5, 15, (MaxX+1) div 8 * 5, MaxY-61);  {}
   Line((MaxX+1) div 8 * 4, 15, (MaxX+1) div 8 * 4, MaxY-61);  {Set Line X}
   Line((MaxX+1) div 8 * 3, 15, (MaxX+1) div 8 * 3, MaxY-61);  {}
   Line((MaxX+1) div 8 * 2, 15, (MaxX+1) div 8 * 2, MaxY-61);  {}
   Line((MaxX+1) div 8,    15, (MaxX+1) div 8,    MaxY-61);  {}
   Line(15, 381, MaxX-15, 381);          {}
   Line(15, 381-50, MaxX-15, 381-50);      {}
   Line(15, 381-100, MaxX-15, 381-100);    {}
   Line(15, 381-150, MaxX-15, 381-150);    { Set Line Y }
   Line(15, 381-200, MaxX-15, 381-200);    {}
   Line(15, 381-250, MaxX-15, 381-250);    {}
   Line(15, 381-300, MaxX-15, 381-300);    {}
   Line(15, 381-350, MaxX-15, 381-350);    {}
End;

Procedure SetBarControl (Color : integer);
Begin
   SetBar(2, MaxY-59, MaxX-2, MaxY-2, Color);
   SetBar(12, Maxy-59, MaxX-12, MaxY-57, White);
   SetBar(2, Maxy-59, 4, MaxY-2, White);
   SetBar(2, Maxy-2, MaxX-2, MaxY-4, DarkGray);
   SetBar(MaxX-2, Maxy-59, MaxX-5, MaxY-2, DarkGray);
   SetBar(4, 4, MaxX-4, MaxY-60, LightGray);
   SetBar(2, 2, MaxX-2, 4, White);
   SetBar(2, 2, 4, MaxY-60, White);
   SetBar(2, MaxY-4, MaxX-2, MaxY-2, DarkGray);
   SetBar(MaxX-1, 2, MaxX-5, MaxY-60, DarkGray);
   SetBar(MaxX-3, 2, MaxX-5, 2, White);  {           }
   SetBar(MaxX-4, 3, MaxX-5, 3, White);  { Conner Up Out }
   SetBar(MaxX-5, 4, MaxX-5, 4, White);  {           }
   SetBar(12, 12, MaxX-12, 14, DarkGray);
   SetBar(12, 12, 14, MaxY-58, DarkGray);  {           }
   SetBar(13, MaxY-58, 14, MaxY-58, White); { Conner In Down }
   SetBar(14, MaxY-59, 14, MaxY-59, White); {              }
   SetBar(MaxX-12, 12, MaxX-14, MaxY-60, White);
   SetBar(MaxX-14, 12, MaxX-12, 12, DarkGray);  {           }
   SetBar(MaxX-14, 13, MaxX-13, 13, DarkGray);  { Conner In Up }
   SetBar(MaxX-14, 14, MaxX-14, 14, DarkGray);  {           }
   SetBar(15, 15, MaxX-15, MaxY-61, Blue);
   Setcolor(Red);
   Line(10, MaxY-54, MaxX-10, MaxY-54);
   Line(10, MaxY-7, MaxX-10, MaxY-7);
   Line(10, MaxY-7, 10, MaxY-54);
   Line(MaxX-10, MaxY-7, MaxX-10, MaxY-54);

End;

Procedure SetSpeed;
Begin
   Setbar(MaxX-150,MaxY-30,MaxX-50,MaxY-10,Darkgray);
   Setbar(MaxX-135,MaxY-30,MaxX-65,MaxY-10,Black);
   Setcolor(Blue);
   Outtextxy(MaxX-125,MaxY-40,'Channel');
   Setcolor(LightGreen);
   Outtextxy(MaxX-145,MaxY-25,'-');
   Outtextxy(MaxX-60,MaxY-25,'+');
  (* Outtextxy(MaxX-110,MaxY-25,'0500'); *)
End;

Procedure SetX_YRecorder;
var i:integer;
Begin
   SetBar (0, 0, GetMaxX, GetMaxY, Blue);
    SetBarControl(LightGray);
   SetSpeed;
    Setcolor(Blue);
    Outtextxy(maxX div 12 *5 ,MaxY-50,'X-Y Recorder');
    Setcolor(Magenta);
    Outtextxy(15,MaxY-35,'* Reset');
    Outtextxy(15,MaxY-20,'+ Start');
{    Outtextxy(15,MaxY-35,'/ Range');  }
   WhereX:=MaxX div 16 *5;
   WhereY:=MaxY-20;
    Logo := 'Department of Nuclear Technology ';
   For count :=1 to 33 do
     Begin
        Setcolor(count mod 7);
        Outtextxy(whereX+count*8,whereY,Logo[count]);
     End;
End;

Procedure Screen_end;
var str1:string[32];
   str2:string[33];
   str3:string[23];
   str4:string[25];
Begin
  Textbackground(black);
  Clrscr;
  loop:=2;
  str1:='By. Mr Tanate Siritriwattanaporn';
  str2:='Department of Nuclear Technology ';
  str3:='Faculty of Engineering.';
  str4:='Chulalongkorn University.';
  Gotoxy(15,6);
  for count:=1 to 32 do
  begin
    if loop>15 then loop:=2;
    textcolor(loop);
    write(str1[count]);
    loop:=loop+1;
  end;
  Gotoxy(25,7);
  for count:=1 to 33 do
  begin
    if loop>15 then loop:=2;
    textcolor(loop);
    write(str2[count]);
    loop:=loop+1;
  end;
  Gotoxy(15,9);
  for count:=1 to 23 do
  begin
    if loop>15 then loop:=2;
    textcolor(loop);
    write(str3[count]);
    loop:=loop+1;
  end;
  for count:=1 to 25 do
  begin
    if loop>15 then loop:=2;
    textcolor(loop);
    write(str4[count]);
    loop:=loop+1;
  end;
  Gotoxy(17,25);
End;

Procedure RunX_YRecorder;
Var  InOK, ScaleOK : boolean;
     Init        : integer;
     st          : string;
Begin
   GraphInit;
  SetX_Yrecorder;



  Channal:=0;
  loop:=500;{Old value = 500  (01/06/97 Tanate)}
  InOK:=True;
  ScaleOK:=False;
  ch:=readkey;
   case ch of
    '+':    Begin
           repeat
           Channal:=channal+1;
           Port[setport]:=$13;
           delay(200);
           Port[setport]:=$14;
           StepY[channal]:=port[setport];
           SetLine(channal-1,stepY[channal-1],channal,stepY[channal],yellow);
           Setbar(MaxX-135,MaxY-30,MaxX-65,MaxY-10,Black);
           str(channal:3,st);    {convert channel no. to string for display}
           Outtextxy(MaxX-125,MaxY-25,(st)); {display channel no.}
           delay(500);
           until channal=256;
           Setcolor(Red);
           Outtextxy(maxX div 12 *5 ,MaxY-30,'Complete!');
           repeat until keypressed;
           closegraph;
           End;
     '*':  begin
          closegraph;
          port[setport]:=$12;
          end;
     end(*end case*);
     Screen_end;
End;

Begin
  RunX_YRecorder;
End.
```

ภาคผนวก ข.

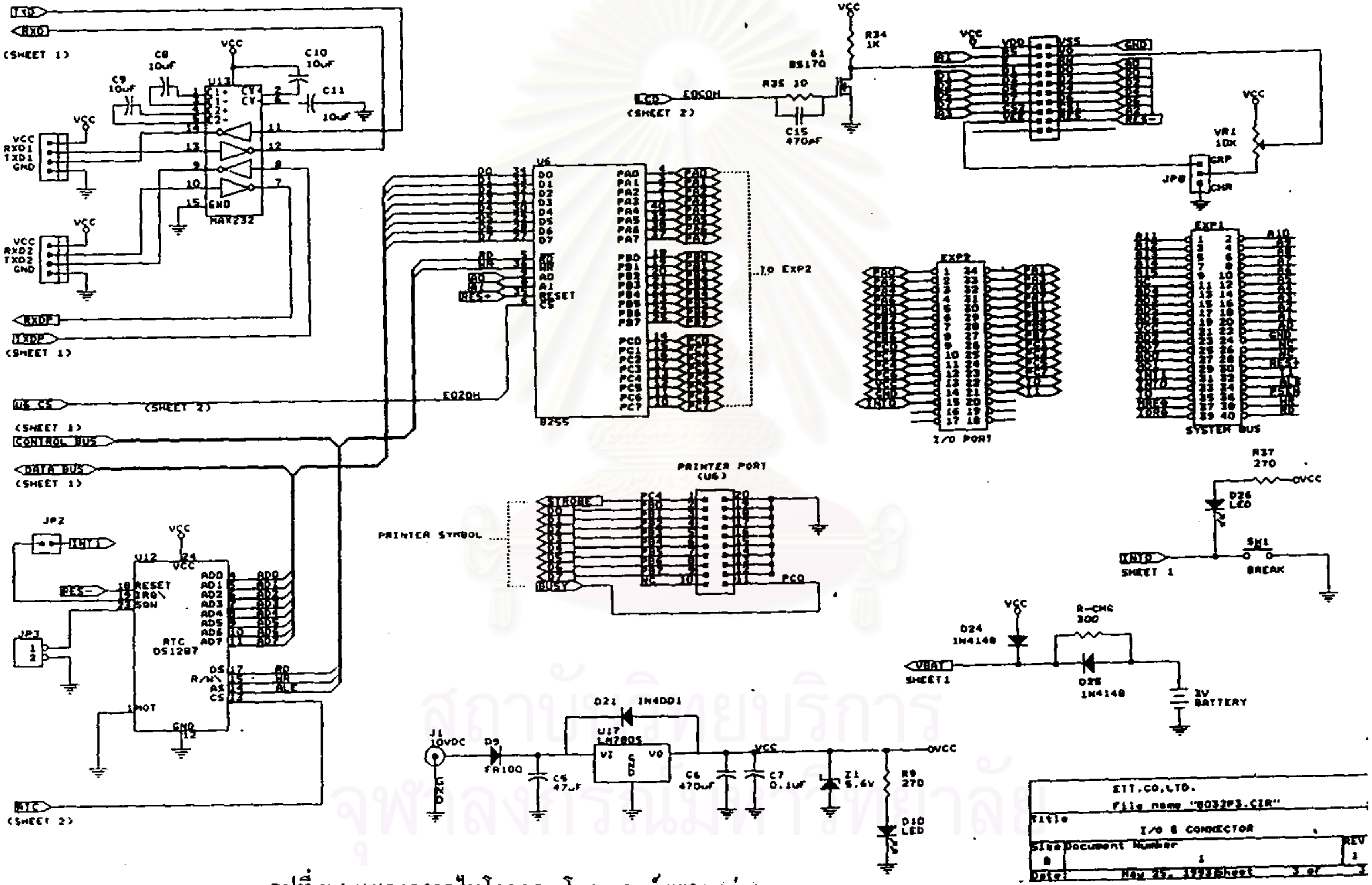
- วงจรไมโครคอนโทรลเลอร์ 8031
- วงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกและแปลงสัญญาณอะนาลอก เป็นดิจิตอล
- วงจรภาคส่งของ อินฟราเรดโมเด็ม
- วงจรภาครับของ อินฟราเรดโมเด็ม
- วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณ FSK และวงจรแปลงสัญญาณ FSK เป็น สัญญาณดิจิตอล



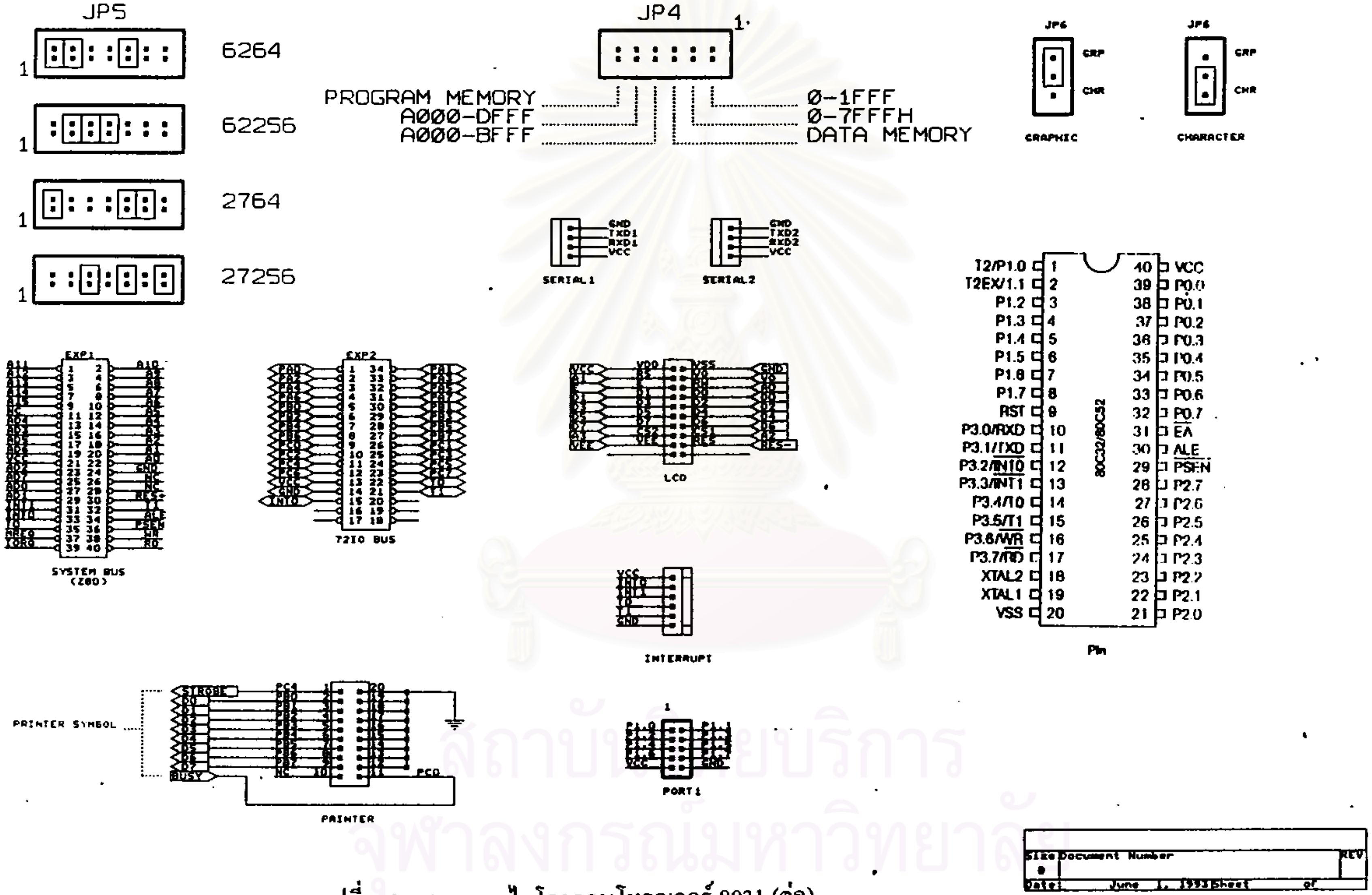
รูปที่ ข.1 แสดงวงจรไมโครคอนโทรลเลอร์ 8031



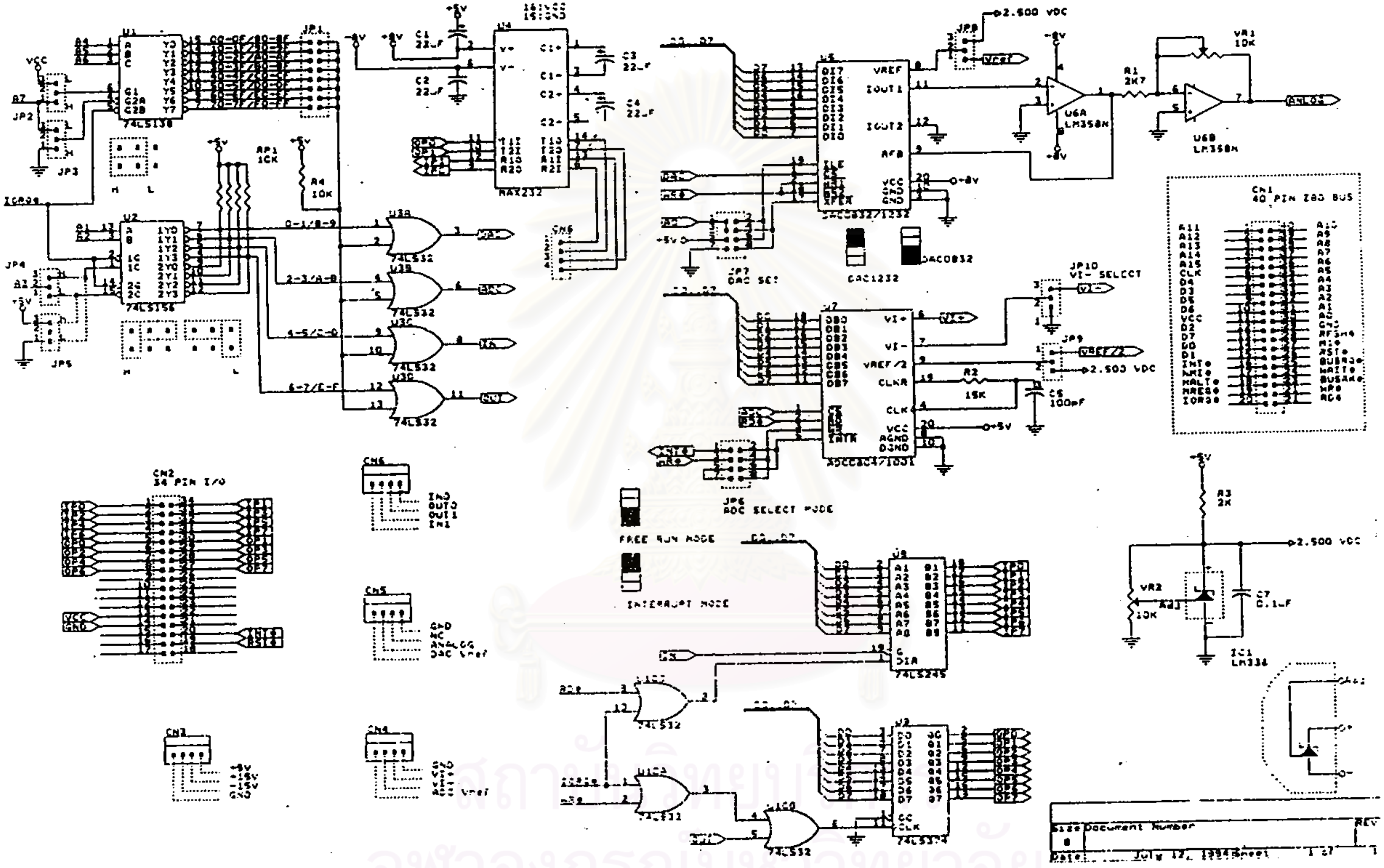
รูปที่ ข.1 แสดงวงจรไมโครคอนโทรลเลอร์ 8031 (ต่อ)



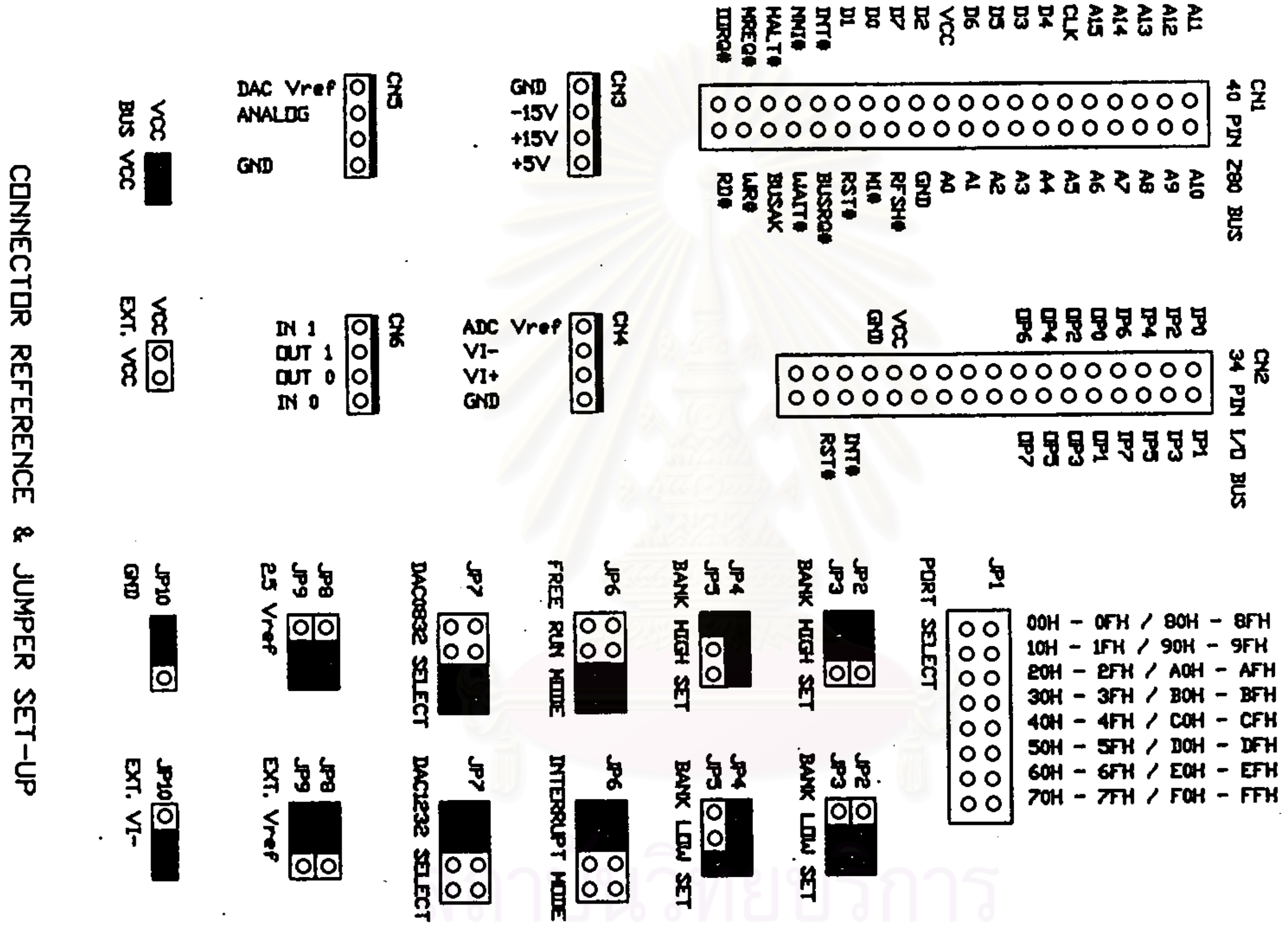
รูปที่ ข.1 แสดงวงจรไมโครคอนโทรลเลอร์ 8031 (ต่อ)



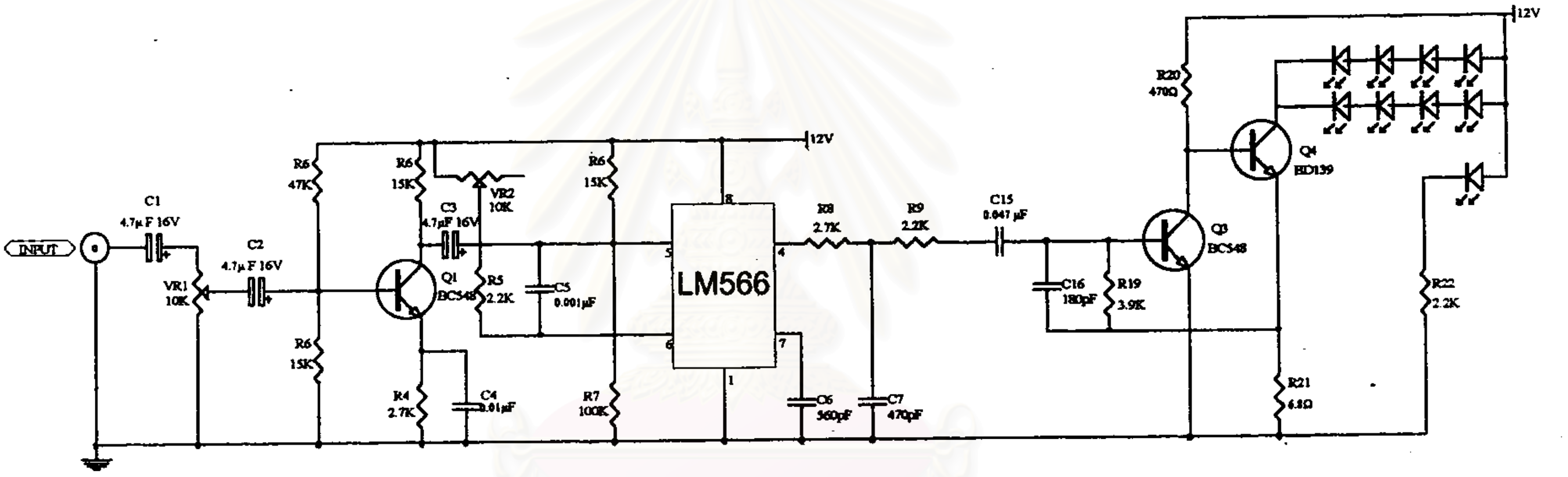
รูปที่ ข.1 แสดงวงจรไมโครคอนโทรลเลอร์ 8031 (ต่อ)



รูปที่ ข.2 แสดงวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกและแปลงสัญญาณอะนาลอกเป็นดิจิตอล

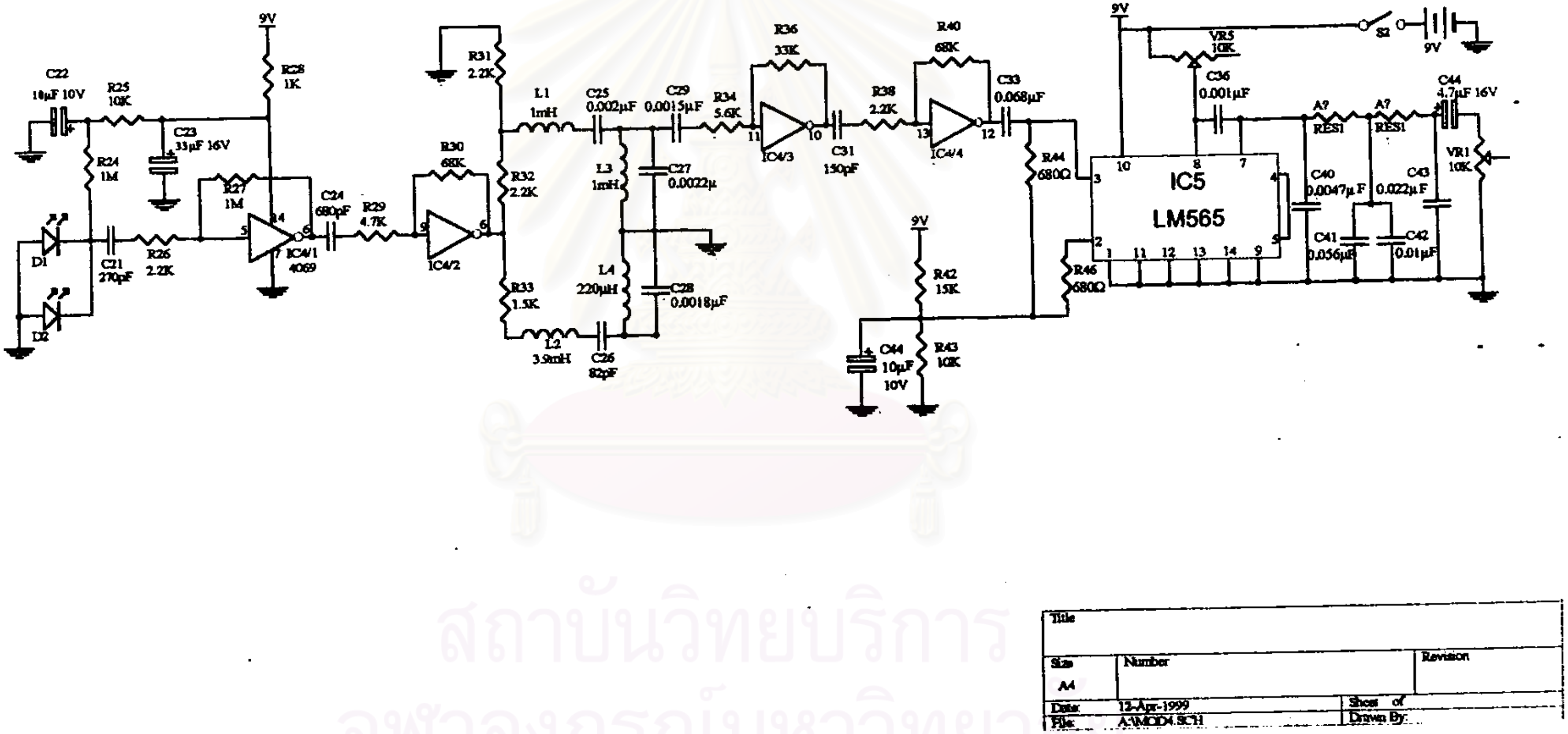


รูปที่ ข.2 แสดงวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกและแปลงสัญญาณอะนาลอกเป็นดิจิตอล (ต่อ)

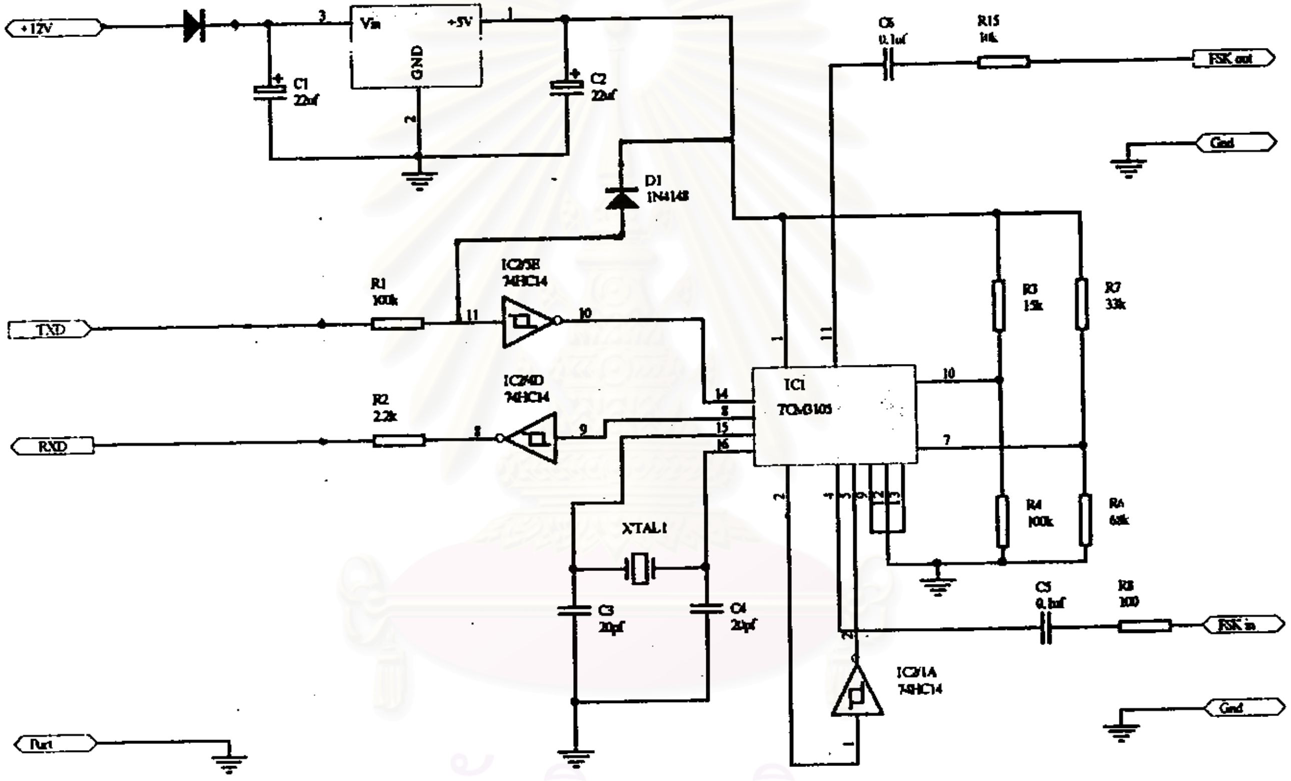


Title			
Size A4	Number		Revision
Date:	12-Apr-1999	Sheet of	
File:	A:\MOD2.SCH	Drawn By:	

รูปที่ ข.3 แสดงวงจรภาคส่งของอินฟราเรดโมเด็ม



รูปที่ ข.4 แสดงวงจรภาครับของอินฟราเรดโมเด็ม



รูปที่ ข.5 แสดงวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณ FSK และ วงจรแปลงสัญญาณ FSK เป็นสัญญาณดิจิตอล

ภาคผนวก ค.

DATA SHEETS

February 1995

National Semiconductor

DAC0830/DAC0831/DAC0832 8-Bit µP Compatible, Double-Buffered D to A Converters

General Description

The DAC0830 is an advanced CMOS/Si-Cr 8-bit multiplying DAC designed to interface directly with the 8080, 8048, 8085, Z80®, and other popular microprocessors. A deposited silicon-chromium R-2R resistor ladder network divides the reference current and provides the circuit with excellent temperature tracking characteristics (0.05% of Full Scale Range maximum linearity error over temperature). The circuit uses CMOS current switches and control logic to achieve low power consumption and low output leakage current errors. Special circuitry provides TTL logic input voltage level compatibility.

Double buffering allows these DACs to output a voltage corresponding to one digital word while holding the next digital word. This permits the simultaneous updating of any number of DACs.

The DAC0830 series are the 8-bit members of a family of microprocessor-compatible DACs (MICRO-DAC™). For applications demanding higher resolution, the DAC1000 series (10-bits) and the DAC1208 and DAC1230 (12-bits) are available alternatives.

BI-FET™ and MICRO-DAC™ are trademarks of National Semiconductor Corporation.
Z80® is a registered trademark of Zilog Corporation.

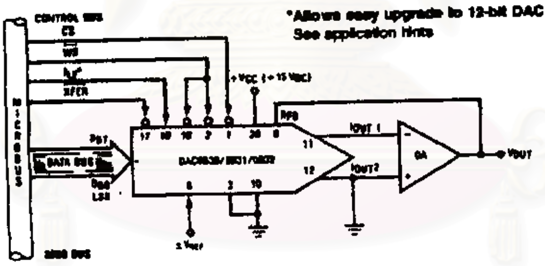
Features

- Double-buffered, single-buffered or flow-through digital data inputs
- Easy interchange and pin-compatible with 12-bit DAC1230 series
- Direct interface to all popular microprocessors
- Linearity specified with zero and full scale adjust only—NOT BEST STRAIGHT LINE FIT.
- Works with ±10V reference-full 4-quadrant multiplication
- Can be used in the voltage switching mode
- Logic inputs which meet TTL voltage level specs (1.4V logic threshold)
- Operates "STAND ALONE" (without µP) if desired
- Available in 20-pin small-outline or molded chip carrier package

Key Specifications

■ Current settling time	1 µs
■ Resolution	8 bits
■ Linearity (guaranteed over temp.)	8, 9, or 10 bits
■ Gain Tempco	0.0002% FS/°C
■ Low power dissipation	20 mW
■ Single power supply	5 to 15 V_{DC}

Typical Application

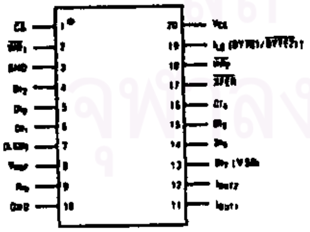


*Allows easy upgrade to 12-bit DAC1230, See application hints

TL/H/5608–1

Connection Diagrams (Top Views)

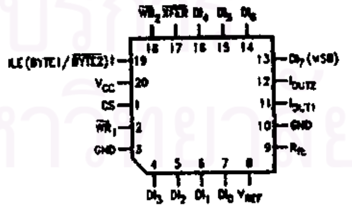
Dual-In-Line and Small-Outline Packages



†This is necessary for the 12-bit DAC1230 series to permit interchanging from an 8-bit to a 12-bit DAC with No PC board changes and no software changes. See applications section.

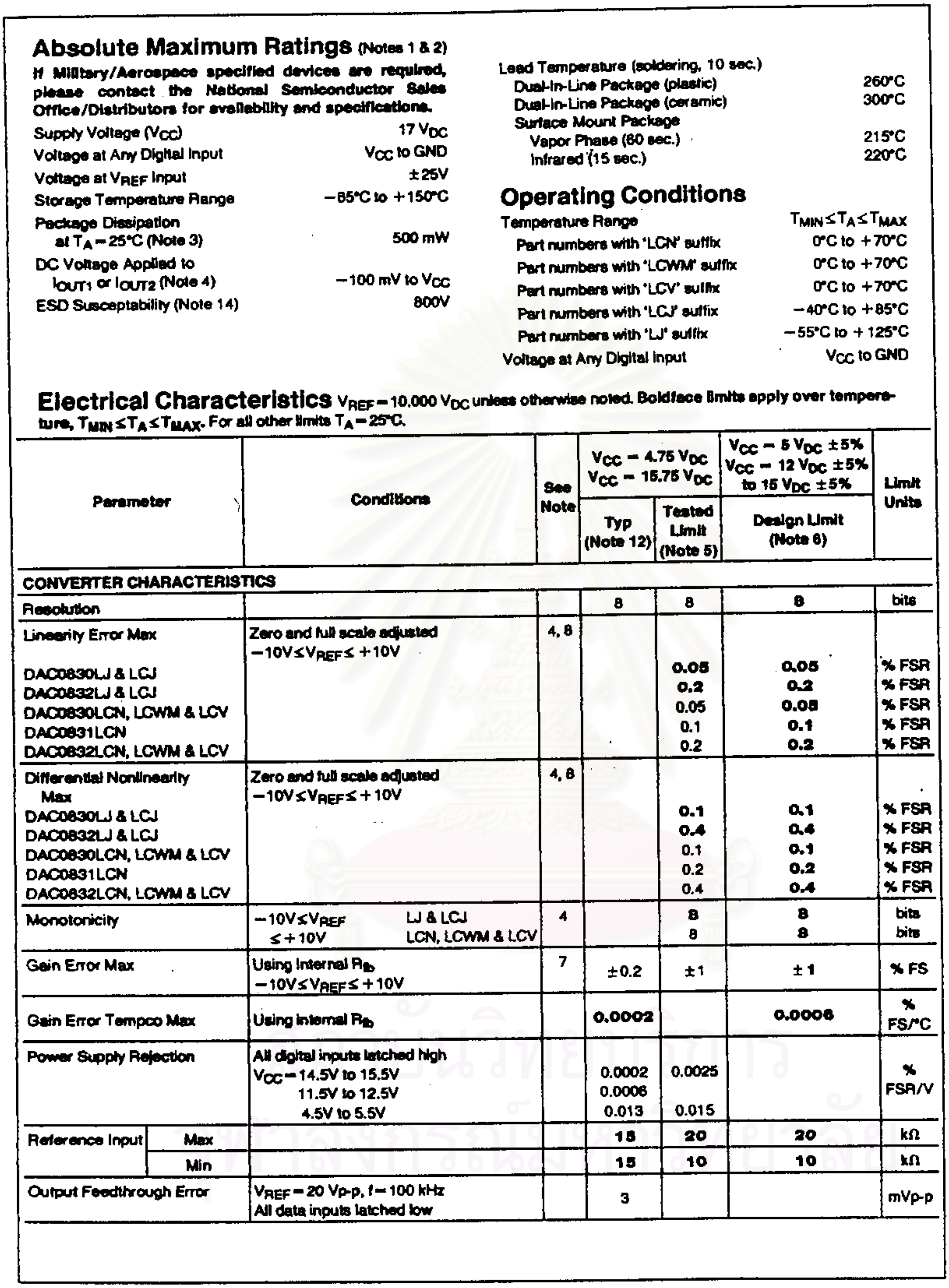
TL/H/5608–21

Molded Chip Carrier Package



TL/H/5608–22

©1995 National Semiconductor Corporation TL/H/5608



Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC})	17 V_{DC}
Voltage at Any Digital Input	V_{CC} to GND
Voltage at V_{REF} Input	±25V
Storage Temperature Range	−65°C to +150°C
Package Dissipation at $T_A = 25°C$ (Note 3)	500 mW
DC Voltage Applied to I_{OUT1} or I_{OUT2} (Note 4)	−100 mV to V_{CC}
ESD Susceptability (Note 14)	800V
Lead Temperature (soldering, 10 sec.)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 sec.)	215°C
Infrared (15 sec.)	220°C

Operating Conditions

Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
Part numbers with 'LCN' suffix	0°C to +70°C
Part numbers with 'LCWM' suffix	0°C to +70°C
Part numbers with 'LCV' suffix	0°C to +70°C
Part numbers with 'LCJ' suffix	−40°C to +85°C
Part numbers with 'LJ' suffix	−55°C to +125°C
Voltage at Any Digital Input	V_{CC} to GND

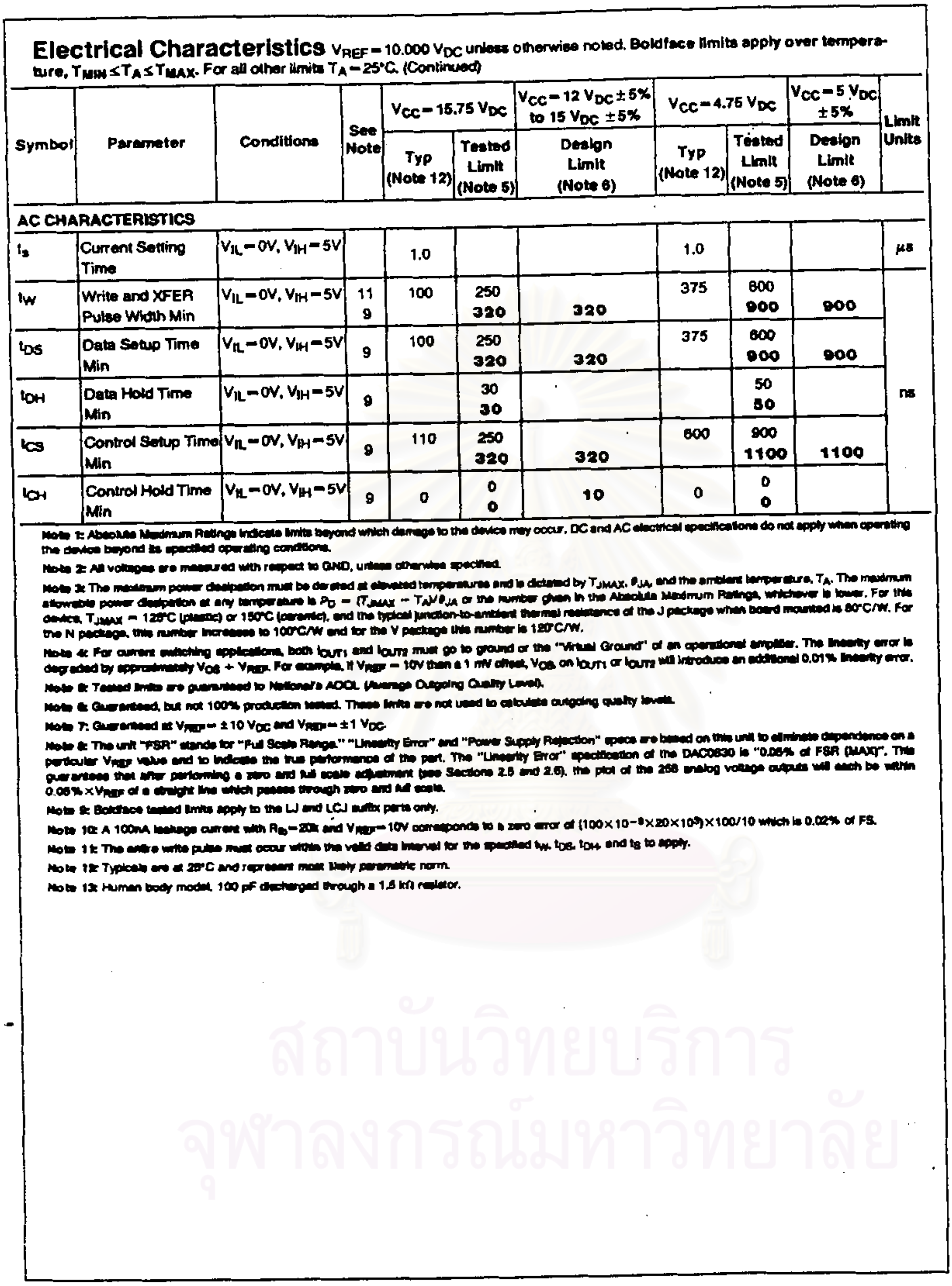
Electrical Characteristics

$V_{REF} = 10.000\ V_{DC}$ unless otherwise noted. **Boldface limits apply over temperature, $T_{MIN} \le T_A \le T_{MAX}$.** For all other limits $T_A = 25°C$.

Parameter	Conditions	See Note	$V_{CC} = 4.75\ V_{DC}$ $V_{CC} = 15.75\ V_{DC}$		$V_{CC} = 5\ V_{DC} \pm 5\%$ $V_{CC} = 12\ V_{DC} \pm 5\%$ to 15 $V_{DC} \pm 5\%$	Limit Units
			Typ (Note 12)	Tested Limit (Note 5)	Design Limit (Note 6)	
CONVERTER CHARACTERISTICS						
Resolution			8	8	8	bits
Linearity Error Max	Zero and full scale adjusted $-10V \le V_{REF} \le +10V$	4, 8				
DAC0830LJ & LCJ				**0.05**	**0.05**	% FSR
DAC0832LJ & LCJ				**0.2**	**0.2**	% FSR
DAC0830LCN, LCWM & LCV				0.05	**0.05**	% FSR
DAC0831LCN				0.1	**0.1**	% FSR
DAC0832LCN, LCWM & LCV				0.2	**0.2**	% FSR
Differential Nonlinearity Max	Zero and full scale adjusted $-10V \le V_{REF} \le +10V$	4, 8				
DAC0830LJ & LCJ				**0.1**	**0.1**	% FSR
DAC0832LJ & LCJ				**0.4**	**0.4**	% FSR
DAC0830LCN, LCWM & LCV				0.1	**0.1**	% FSR
DAC0831LCN				0.2	**0.2**	% FSR
DAC0832LCN, LCWM & LCV				0.4	**0.4**	% FSR
Monotonicity	$-10V \le V_{REF} \le +10V$ LJ & LCJ	4		**8**	**8**	bits
	LCN, LCWM & LCV			8	**8**	bits
Gain Error Max	Using Internal R_{fb} $-10V \le V_{REF} \le +10V$	7	±0.2	±1	**±1**	% FS
Gain Error Tempco Max	Using internal R_{fb}		**0.0002**		**0.0006**	% FS/°C
Power Supply Rejection	All digital inputs latched high $V_{CC} = 14.5V$ to 15.5V 11.5V to 12.5V 4.5V to 5.5V		0.0002 0.0006 0.013	0.0025 0.015		% FSR/V
Reference Input Max			**15**	**20**	**20**	kΩ
Min			**15**	**10**	**10**	kΩ
Output Feedthrough Error	$V_{REF} = 20$ Vp-p, f = 100 kHz All data inputs latched low		3			mVp-p

Electrical Characteristics V_{REF} = 10.000 V_{DC} unless otherwise noted. Boldface limits apply over temperature, $T_{MIN} \le T_A \le T_{MAX}$. For all other limits T_A = 25°C. (Continued)

Parameter		Conditions		See Note	V_{CC} = 4.75 V_{DC} V_{CC} = 15.75 V_{DC}		V_{CC} = 5 V_{DC} ±5% V_{CC} = 12 V_{DC} ±5% to 15 V_{DC} ±5%	Limit Units
					Typ (Note 12)	Tested Limit (Note 5)	Design Limit (Note 6)	
CONVERTER CHARACTERISTICS (Continued)								
Output Leakage Current Max	I_{OUT1}	All data inputs latched low	LJ & LCJ	10		**100**	**100**	nA
			LCN, LCWM & LCV			50	**100**	
	I_{OUT2}	All data inputs latched high	LJ & LCJ			**100**	**100**	nA
			LCN, LCWM & LCV			50	**100**	
Output Capacitance	I_{OUT1} I_{OUT2}	All data inputs latched low			45 115			pF
	I_{OUT1} I_{OUT2}	All data inputs latched high			130 30			pF
DIGITAL AND DC CHARACTERISTICS								
Digital Input Voltages	Max	Logic Low	LJ 4.75V			**0.6**		V_{DC}
			LJ 15.75V			**0.8**		
			LCJ 4.75V			**0.7**		
			LCJ 15.75V			**0.8**		
			LCN, LCWM, LCV			0.95	**0.8**	
	Min	Logic High	LJ & LCJ			**2.0**	**2.0**	V_{DC}
			LCN, LCWM, LCV			1.9	**2.0**	
Digital Input Currents	Max	Digital inputs <0.8V	LJ & LCJ		−50	**−200**	**−200**	µA
			LCN, LCWM, LCV			−160	**−200**	µA
		Digital inputs >2.0V	LJ & LCJ		0.1	**+10**	**+10**	µA
			LCN, LCWM, LCV			+8	**+10**	
Supply Current Drain	Max		LJ & LCJ		1.2	**3.5**	**3.5**	mA
			LCN, LCWM, LCV			1.7	**2.0**	



Electrical Characteristics V_{REF} = 10.000 V_{DC} unless otherwise noted. Boldface limits apply over temperature, $T_{MIN} \le T_A \le T_{MAX}$. For all other limits T_A = 25°C. (Continued)

Symbol	Parameter	Conditions	See Note	V_{CC} = 15.75 V_{DC}		V_{CC} = 12 V_{DC} ±5% to 15 V_{DC} ±5%	V_{CC} = 4.75 V_{DC}		V_{CC} = 5 V_{DC} ±5%	Limit Units
				Typ (Note 12)	Tested Limit (Note 5)	Design Limit (Note 6)	Typ (Note 12)	Tested Limit (Note 5)	Design Limit (Note 6)	
AC CHARACTERISTICS										
t_s	Current Setting Time	V_{IL} = 0V, V_{IH} = 5V		1.0			1.0			μs
t_W	Write and XFER Pulse Width Min	V_{IL} = 0V, V_{IH} = 5V	11 9	100	250 **320**	**320**	375	600 **900**	**900**	ns
t_{DS}	Data Setup Time Min	V_{IL} = 0V, V_{IH} = 5V	9	100	250 **320**	**320**	375	600 **900**	**900**	
t_{DH}	Data Hold Time Min	V_{IL} = 0V, V_{IH} = 5V	9		30 **30**			50 **50**		
t_{CS}	Control Setup Time Min	V_{IL} = 0V, V_{IH} = 5V	9	110	250 **320**	**320**	600	900 **1100**	**1100**	
t_{CH}	Control Hold Time Min	V_{IL} = 0V, V_{IH} = 5V	9	0	0 **0**	**10**	0	0 **0**		

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: The maximum power dissipation must be derated at elevated temperatures and is dictated by T_{JMAX}, θ_{JA}, and the ambient temperature, T_A. The maximum allowable power dissipation at any temperature is $P_D = (T_{JMAX} - T_A)/\theta_{JA}$ or the number given in the Absolute Maximum Ratings, whichever is lower. For this device, T_{JMAX} = 125°C (plastic) or 150°C (ceramic), and the typical junction-to-ambient thermal resistance of the J package when board mounted is 80°C/W. For the N package, this number increases to 100°C/W and for the V package this number is 120°C/W.

Note 4: For current switching applications, both I_{OUT1} and I_{OUT2} must go to ground or the "Virtual Ground" of an operational amplifier. The linearity error is degraded by approximately $V_{OS} \div V_{REF}$. For example, if V_{REF} = 10V then a 1 mV offset, V_{OS}, on I_{OUT1} or I_{OUT2} will introduce an additional 0.01% linearity error.

Note 5: Tested limits are guaranteed to National's AOQL (Average Outgoing Quality Level).

Note 6: Guaranteed, but not 100% production tested. These limits are not used to calculate outgoing quality levels.

Note 7: Guaranteed at V_{REF} = ±10 V_{DC} and V_{REF} = ±1 V_{DC}.

Note 8: The unit "FSR" stands for "Full Scale Range." "Linearity Error" and "Power Supply Rejection" specs are based on this unit to eliminate dependence on a particular V_{REF} value and to indicate the true performance of the part. The "Linearity Error" specification of the DAC0830 is "0.05% of FSR (MAX)". This guarantees that after performing a zero and full scale adjustment (see Sections 2.5 and 2.6), the plot of the 256 analog voltage outputs will each be within 0.05% × V_{REF} of a straight line which passes through zero and full scale.

Note 9: Boldface tested limits apply to the LJ and LCJ suffix parts only.

Note 10: A 100nA leakage current with R_{fb} = 20k and V_{REF} = 10V corresponds to a zero error of $(100\times10^{-9}\times20\times10^{3})\times100/10$ which is 0.02% of FS.

Note 11: The entire write pulse must occur within the valid data interval for the specified t_W, t_{DS}, t_{DH}, and t_S to apply.

Note 12: Typicals are at 25°C and represent most likely parametric norm.

Note 13: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

intel.

80C31BH/80C51BH/87C51
MCS® 51
CHMOS SINGLE-CHIP 8-BIT MICROCONTROLLER

Automotive

- **Extended Automotive Temperature Range (−40°C to +125°C Ambient)**
- **High Performance CHMOS Process**
- **Power Control Modes**
- **4 Kbyte On-Chip ROM/EPROM**
- **128 x 8-bit RAM**
- **32 Programmable I/O Lines**
- **Two 16-Bit Timer/Counters**
- **5 Interrupt Sources**
- **Quick-Pulse EPROM Programming**
- **2-Level Program Memory Lock EPROM**
- **Boolean Processor**
- **Programmable Serial Port**
- **TTL- and CMOS-Compatible Logic Levels**
- **64K External Program Memory Space**
- **64K External Data Memory Space**
- **IDLE and POWER DOWN Modes**
- **ONCE Mode Facilitates System Testing**
- **Available in 12 MHz and 16 MHz Versions**
- **Available in PLCC and DIP Packages**
  (See Packaging Specification, Order #231369)

The MCS® 51 CHMOS microcontroller products are fabricated on Intel's reliable CHMOS process and are functionally compatible with the standard MCS 51 HMOS microcontroller products. This technology combines the high speed and density characteristics of HMOS with the low power attributes of CHMOS. This combination expands the effectiveness of the powerful MCS 51 microcontroller architecture and instruction set.

Like the MCS 51 HMOS microcontroller versions, the MCS 51 CHMOS microcontroller products have the following features: 4 Kbytes of EPROM/ROM (87C51/80C51BH respectively); 128 bytes of RAM; 32 I/O lines; two 16-bit timer/counters; a five-source two-level interrupt structure; a full duplex serial port; and on-chip oscillator and clock circuitry. In addition, the MCS 51 CHMOS microcontroller products exhibit low operating power, along with two software selectable modes of reduced activity for further power reduction—Idle and Power Down.

The Idle mode freezea 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, causing all other chip functions to be inoperative.

The 87C51 is the EPROM version of the 80C51BH. It contains 4 Kbytes of on-chip program memory that can be electrically programmed, and can be erased by exposure to ultraviolet light. The 87C51 EPROM array uses a modified Quick-Pulse Programming algorithm, by which the entire 4 Kbyte array can be programmed in about 12 seconds.

NOTICE:
This datasheet contains information on products in full production. Specifications within this datasheet are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

*Other brands and names are the property of their respective owners.
Information in this document is provided in connection with Intel products. Intel assumes no liability whatsoever, including infringement of any patent or
copyright, for sale and use of Intel products except as provided in Intel's Terms and Conditions of Sale for such products. Intel retains the right to make
changes to these specifications at any time, without notice. Microcomputer Products may have minor variations to this specification known as errata.

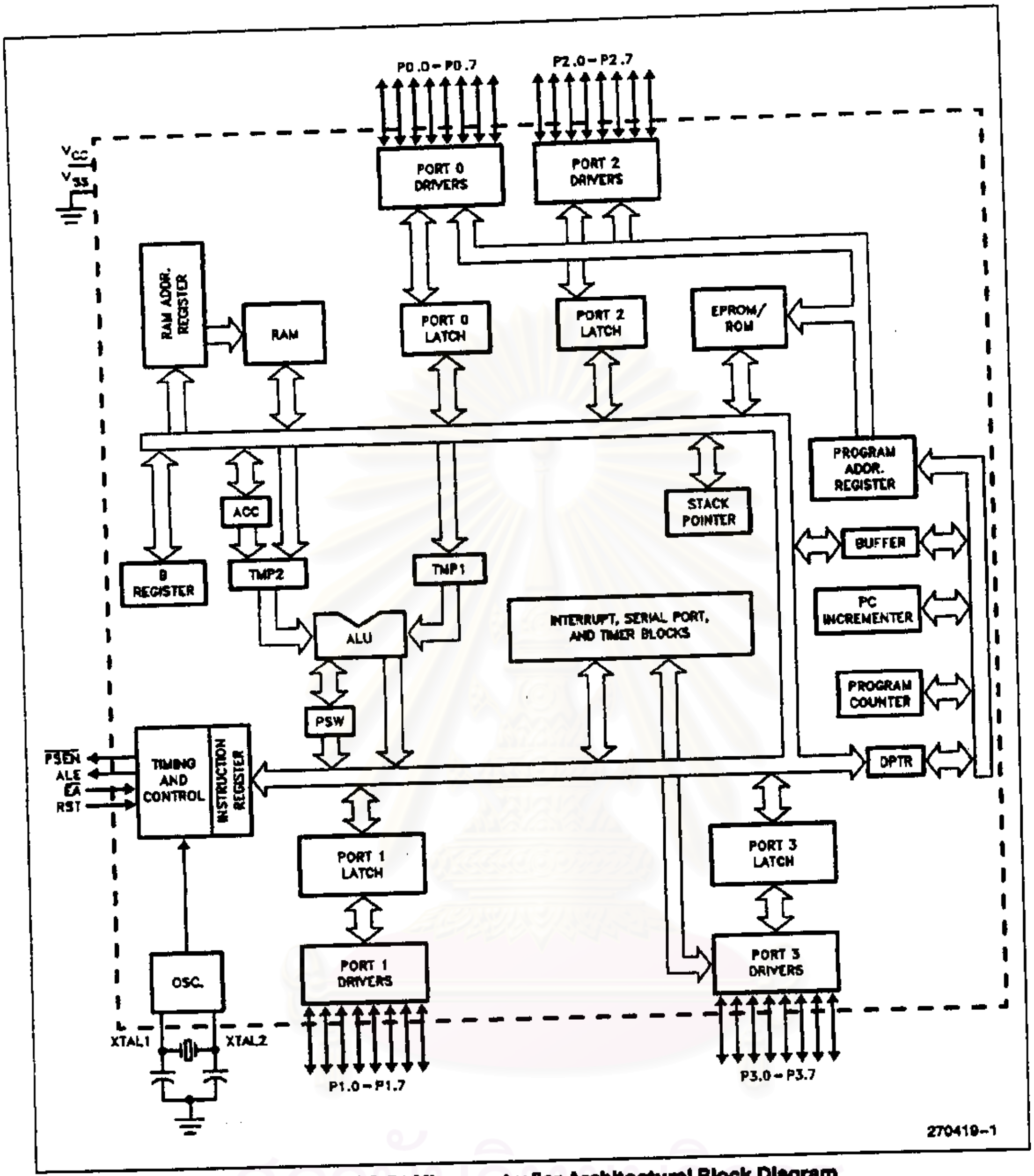


Figure 1. MCS® 51 Microcontroller Architectural Block Diagram

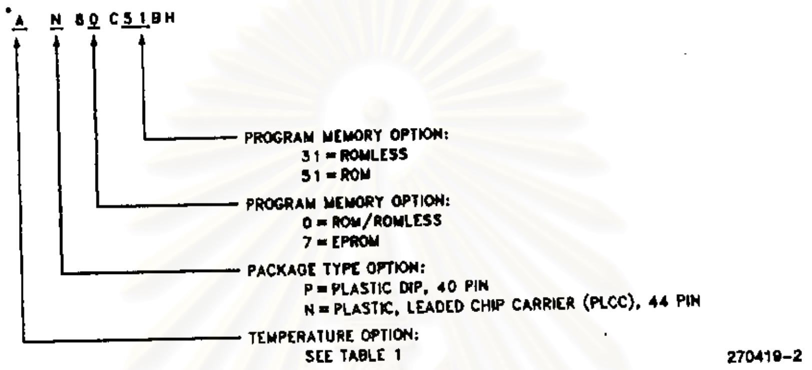
80C31BH/80C51BH/87C51 PRODUCT OPTIONS

Intel's extended and automotive temperature range products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

With the extended temperature range option, operational characteristics are guaranteed over the temperature range of −40°C to +85°C ambient. For the automotive temperature range option, operational characteristics are guaranteed over the temperature range of −40°C to +125°C ambient.

The automotive and extended temperature versions of the MCS 51 microcontroller product families are available with or without burn-in options as listed in Table 1.

As shown in Figure 2, temperature, burn-in, and package options are identified by a one- or two-letter prefix to the part number.



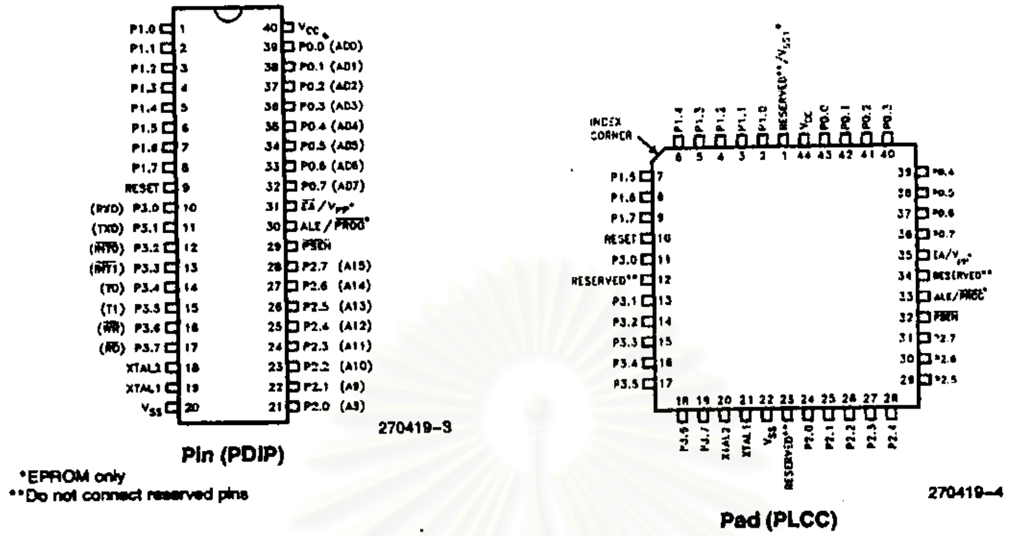
***Example:**
AN80C51 indicates an automotive temperature range version of the 80C51 in a PLCC package with 4 Kbyte ROM program memory.

Figure 2. MCS® 51 Microcontroller Product Family Nomenclature

Table 1. Temperature Options

Temperature Classification	Temperature Designation	Operating Temperature °C Ambient	Burn-In Options
Extended	T L	−40 to +85 −40 to +85	Standard Extended
Automotive	A B	−40 to +125 −40 to +125	Standard Extended

Diagrams are for pin reference only. Package sizes are not to scale.



*EPROM only
**Do not connect reserved pins

Figure 3. Pin Connections

PIN DESCRIPTION

V_{CC}: Supply voltage during normal, Idle, and Power Down operations.

V_{SS}: Circuit ground.

V_{SS1}: V_{SS1}—(EPROM PLCC only) secondary ground. Provided to reduce ground bounce and improve power supply bypassing.

NOTE:
This pin is not a substitute for the V_{SS} pin (pin 22). For ROM and ROMless, pin 1 is reserved—do not connect.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external memory. In this application it uses strong internal pullups when emitting 1s.

Port 0 also receives the code bytes during EPROM 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. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the datasheet) because of the internal pullups.

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

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) 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 address (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 some control signals and the high-order address bits during EPROM programming and program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the datasheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS 51 microcontroller family, as listed below:

Pin	Name	Alternate Function
P3.0	RXD	Serial Input Line
P3.1	TXD	Serial Output Line
P3.2	$\overline{\text{INT0}}$	External Interrupt 0
P3.3	$\overline{\text{INT1}}$	External Interrupt 1
P3.4	T0	Timer 0 External Input
P3.5	T1	Timer 1 External Input
P3.6	$\overline{\text{WR}}$	External Data Memory Write Strobe
P3.7	$\overline{\text{RD}}$	External Data Memory Read Strobe

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

RESET: Reset input. A logic high on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits a power-on reset to be generated using only an external capacitor to V_{CC}.

ALE/$\overline{\text{PROG}}$ (EPROM Only): Address Latch Enable output signal for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) during EPROM 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.

$\overline{\text{PSEN}}$: Program Store Enable is the Read strobe to External Program Memory. When the 87C51/80C51BH is executing from Internal Program Memory, $\overline{\text{PSEN}}$ is inactive (high). When the device is executing code from External Program Memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to External Data Memory.

$\overline{\text{EA}}$/V_{PP}: External Access enable. $\overline{\text{EA}}$ must be strapped to V_{SS} in order to enable the 87C51/80C51BH to fetch code from External Program Memory locations starting at 0000H up to 0FFFFH. [Note, however, that if either of the Lock Bits is programmed, the logic level at $\overline{\text{EA}}$ is internally latched during reset.] (EPROM only.)

$\overline{\text{EA}}$ must be strapped to V_{CC} for internal program execution.

V_{PP} (EPROM Only): This pin also receives the 12.75V programming supply voltage (V_{PP}) during EPROM programming.

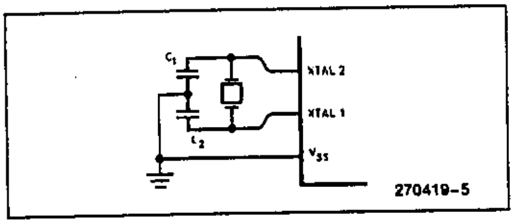


Figure 4. Using the On-Chip Oscillator

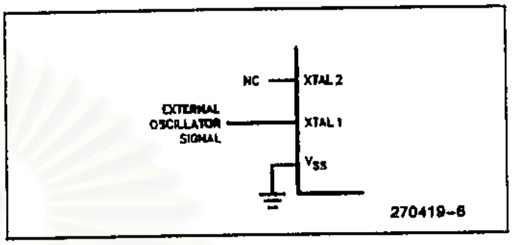


Figure 5. External Clock Drive

XTAL1: Input to the inverting oscillator amplifier and input to the internal clock generating circuits.

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 4.

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected, as shown in Figure 5. 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 high and low times specified on the Datasheet 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 hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to

Table 2. Status of the External Pins During Idle and Power Down

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	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

NOTE:
For more detailed information on the reduced power modes refer to current Embedded Applications Handbook, and Application Note AP-252, "Designing with the 80C51BH."

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.

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.

DESIGN CONSIDERATIONS

- At power on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up.
- Before entering the Power Down mode the contents of the Carry Bit and B.7 must be equal.
- When the Idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins in not inhibited. To eliminate the possibility of an unexpected write 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.
- An external oscillator may encounter as much as a 100 pF load at XTAL1 when it starts up. This is due to interaction between the amplifier and its feedback capacitance. Once the external signal meets the V_{IL} and V_{IH} specifications the capacitance will not exceed 20 pF.
- For EPROM versions exposure to light when the device is in operation may cause logic errors. For this reason, it is suggested that an opaque label be placed over the window when the die is exposed to ambient light.

PROGRAM MEMORY LOCK (EPROM Only)

The 87C51 contains two program memory lock schemes: Encrypted Verify and Lock Bits.

Encrypted Verify: The 87C51 implements a 32-byte EPROM array that can be programmed by the customer, and which can then be used to encrypt the program code bytes during EPROM verification. The EPROM verification procedure is performed as usual, except that each code byte comes out logically X-NORed with one of the 32 key bytes. The key bytes are gone through in sequence. Therefore, to read the ROM code, one has to know the 32 key bytes in their proper sequence.

Lock Bits: Also on the chip are two Lock Bits which can be left unprogrammed (U) or can be programmed (P) to obtain the following additional features:

Bit 1	Bit 2	Additional Features
U	U	none
P	U	• Externally fetched code can not access Internal Program Memory. • Further programming disabled.
U	P	(Reserved for Future definition.)
P	P	• Externally fetched code can not access internal Program Memory. • Further programming disabled. • Program verification is disabled.

When Lock Bit 1 is programmed, the logic level at the $\overline{\text{EA}}$ 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 $\overline{\text{EA}}$ be in agreement with the current logic level at that pin in order for the device to function properly.

ONCE MODE

The ONCE ("on-circuit emulation") mode facilitates testing and debugging of systems using the 87C51 without the 87C51 having to be removed from the circuit. The ONCE mode is invoked by:

1. Pull ALE low while the device is in reset and $\overline{\text{PSEN}}$ is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{\text{PSEN}}$ are weakly pulled high. The oscillator circuit remains active. While the 87C51 is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias −40°C to +125°C

Storage Temperature .... −65°C to +150°C

Voltage on $\overline{EA}/V_{PP}$ Pin to V_{SS} .... 0V to +13.0V

Voltage on Any Other Pin to V_{SS} .. −0.5V to +6.5V

I_{OL} per I/O pin .... 15 mA

Power Dissipation......................... 1.5W
(Based on package heat transfer limitations, not device power consumption).

Typical Junction Temperature (T_J) .... +135°C
(Based upon ambient temperature at +125°C)

Typical Thermal Resistance Junction-to-Ambient (θ_{JA}):

PDIP .... 75°C/W

PLCC.............................. 46°C/W

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.*

DC CHARACTERISTICS:

(T_A = −40°C to +125°C; V_{CC} = 5V ±10% (5V ±20% EPROM Only); V_{SS} = 0V)

Symbol	Parameter	Min	Typ(1)	Max (87C51/80C51BH)	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$)	−0.5		0.2 V_{CC}−0.25	V	
V_{IL1}	Input Low Voltage to $\overline{EA}$	0		0.2 V_{CC}−0.45	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	0.2V_{CC}+1.0		V_{CC}+0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}+0.1		V_{CC}+0.5	V	
V_{OL}	Output Low Voltage (Ports 1, 2, 3)			0.45(7)	V	I_{OL} = 1.6 mA(2)
V_{OL1}	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$)			0.45(7)	V	I_{OL} = 3.2 mA(2)
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$)	2.4			V	I_{OH} = −60 μA
		0.9 V_{CC}			V	I_{OH} = −10 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4			V	I_{OH} = −800 μA
		0.9 V_{CC}			V	I_{OH} = −80 μA(3)
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)			−75	μA	V_{IN} = 0.45 V
I_{TL}	Logical 1-to-0 transition current (Ports 1, 2, 3)			−750	μA	(4)
I_{LI}	Input Leakage Current (Port 0)			±10	μA	V_{IN} = V_{IL} or V_{IH}
I_{CC}	Power Supply Current: Active Mode @ 12 MHz (5) Idle Mode @ 12 MHz (5) Power Down Mode		11.5 1.3 3	25/20 6/5 100/75	mA mA μA	(6) V_{CC} = 2.2V to 5.5V
RRST	Internal Reset Pulldown Resistor	50		300	KΩ	
CIO	Pin Capacitance			10	pF	

NOTES:

1. "Typicals" are based on a limited *number* of samples taken from early manufacturing lots and are not guaranteed. The values listed are at room temp, 5V.
2. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
3. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.

NOTES: (Continued)

4. Pins of Ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V_{IN} is approximately 2V.

5. ICCMAX at other frequencies is given by:

Active Mode: 87C51 ICCMAX = 0.94 × FREQ + 13.71
80Cx1BH ICCMAX = 1.47 × FREQ + 2.36
Idle Mode: ICCMAX = 0.14 × FREQ + 3.81

where FREQ is the external oscillator frequency in MHz. ICCMAX is given in mA. See Figure 6.

6. See Figures 7 through 10 for I_{CC} test conditions. Minimum V_{CC} for Power Down is 2.0V.

7. 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, and 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.

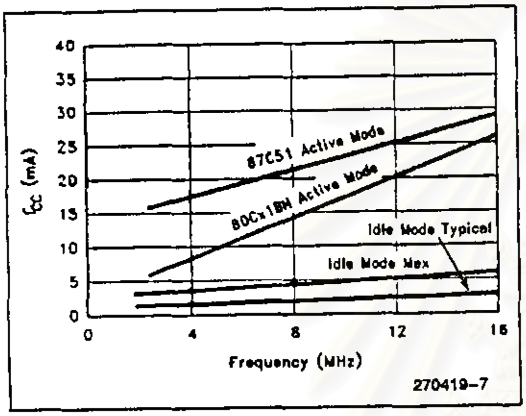


Figure 6. I_{CC} vs. FREQ. Valid only within frequency specifications of the device under test.

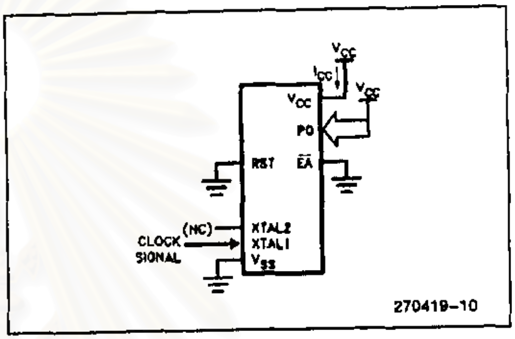


Figure 8. I_{CC} Test Condition, Idle Mode. All other pins are disconnected.

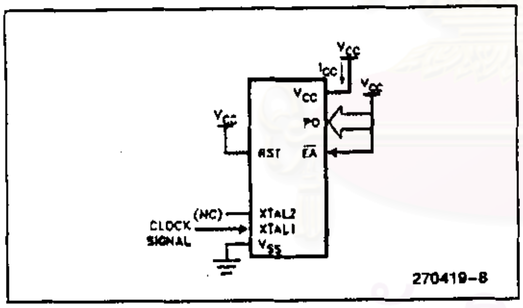


Figure 7. I_{CC} Test Condition, Active Mode. All other pins are disconnected.

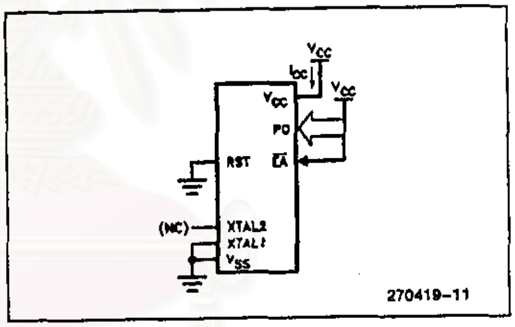


Figure 9. I_{CC} Test Condition, Power Down Mode. All other pins are disconnected.

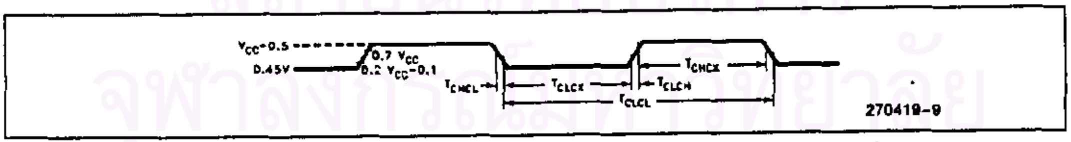


Figure 10. Clock Signal Waveform for I_{CC} tests in Active and Idle Modes. $T_{CLCH} = T_{CHCL} = 5$ ns.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A:Address.
C:Clock.
D:Input data.
H:Logic level HIGH.
I:Instruction (program memory contents).
L:Logic level LOW, or ALE.
P:$\overline{\text{PSEN}}$.
Q:Output data.
R:$\overline{\text{RD}}$ signal.
T:Time.
V:Valid.
W:$\overline{\text{WR}}$ signal.
X:No longer a valid logic level.
Z:Float.

For example,

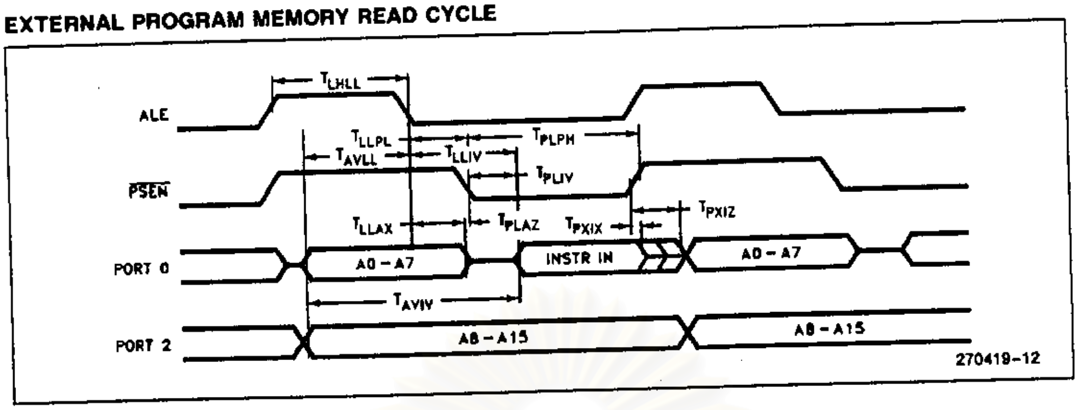
T_{AVLL} = Time from Address Valid to ALE Low.
T_{LLPL} = Time from ALE Low to PSEN Low.

AC CHARACTERISTICS: (T_A = −40°C to +125°C; V_{CC} = 5V ±10% (5V ±20% EPROM Only); V_{SS} = 0V; Load Capacitance for Port 0, ALE, and $\overline{\text{PSEN}}$ = 100 pF; Load Capacitance for All Other Outputs = 80 pF)

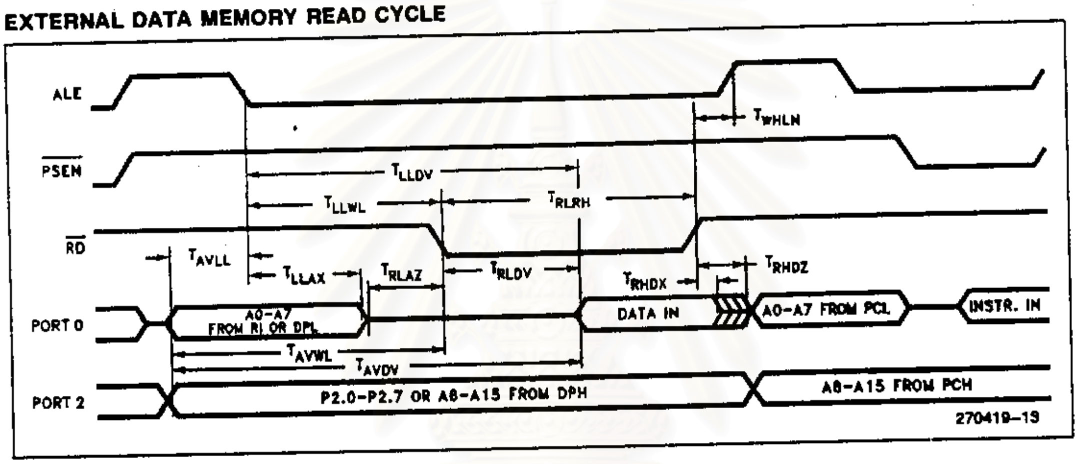
EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/T_{CLCL}$	Oscillator Frequency 87C51/80C51BH/80C31BH			3.5	12–16	MHz
T_{LHLL}	ALE Pulse Width	127		$2T_{CLCL}-40$		ns
T_{AVLL}	Address Valid to ALE Low	28		$T_{CLCL}-55$		ns
T_{LLAX}	Address Hold After ALE Low	48		$T_{CLCL}-35$		ns
T_{LLIV}	ALE Low to Valid Instr In		224		$4T_{CLCL}-110$	ns
T_{LLPL}	ALE Low to $\overline{\text{PSEN}}$ Low	43		$T_{CLCL}-40$		ns
T_{PLPH}	$\overline{\text{PSEN}}$ Pulse Width	205		$3T_{CLCL}-45$		ns
T_{PLIV}	$\overline{\text{PSEN}}$ Low to Valid Instr In		135		$3T_{CLCL}-115$	ns
T_{PXIX}	Input Instr Hold After $\overline{\text{PSEN}}$	0		0		ns
T_{PXIZ}	Input Instr Float After $\overline{\text{PSEN}}$		59		$T_{CLCL}-25$	ns
T_{AVIV}	Address Valid to Valid Instr In		312		$5T_{CLCL}-105$	ns
T_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
T_{RLRH}	$\overline{\text{RD}}$ Pulse Width	400		$6T_{CLCL}-100$		ns
T_{WLWH}	$\overline{\text{WR}}$ Pulse Width	400		$6T_{CLCL}-100$		ns
T_{RLDV}	$\overline{\text{RD}}$ Low to Valid Data In		252		$5T_{CLCL}-165$	ns
T_{RHDX}	Data Hold After $\overline{\text{RD}}$ High	0		0		ns
T_{RHDZ}	Data Float After $\overline{\text{RD}}$ High		97		$2T_{CLCL}-70$	ns
T_{LLDV}	ALE Low to Valid Data In		517		$8T_{CLCL}-150$	ns
T_{AVDV}	Address Valid to Valid Data In		585		$9T_{CLCL}-165$	ns
T_{LLWL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3T_{CLCL}-50$	$3T_{CLCL}+50$	ns
T_{AVWL}	Address Valid to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4T_{CLCL}-130$		ns
T_{QVWX}	Data Valid to $\overline{\text{WR}}$ Transition	23		$T_{CLCL}-60$		ns
T_{WHQX}	Data Hold After $\overline{\text{WR}}$ High	33		$T_{CLCL}-50$		ns
T_{RLAZ}	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
T_{WHLH}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$T_{CLCL}-40$	$T_{CLCL}+40$	ns

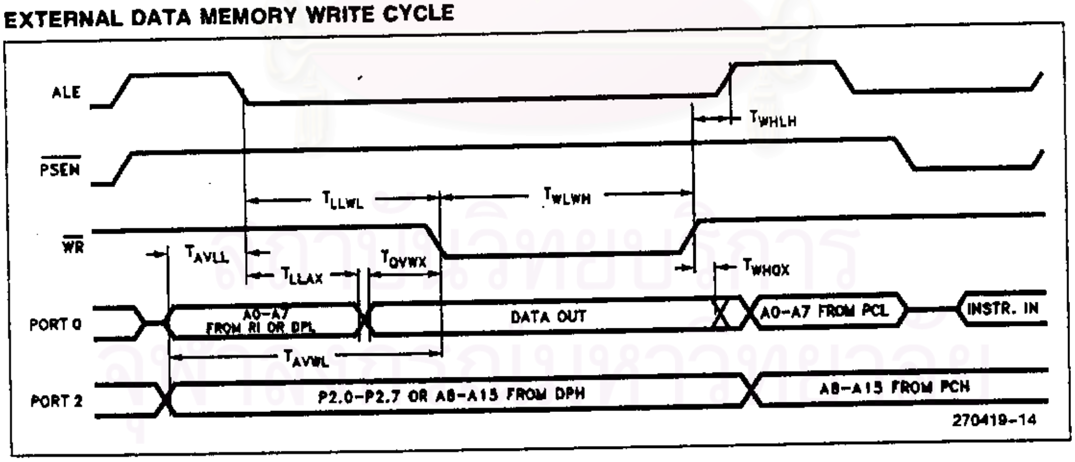
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY READ CYCLE



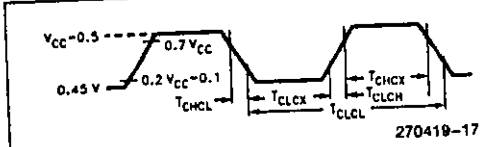
EXTERNAL DATA MEMORY WRITE CYCLE



EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
$1/T_{CLCL}$	Oscillator Frequency	3.5 3.5	12 16	MHz
T_{CHCX}	High Time	20		ns
T_{CLCX}	Low Time	20		ns
T_{CLCH}	Rise Time		20	ns
T_{CHCL}	Fall Time		20	ns

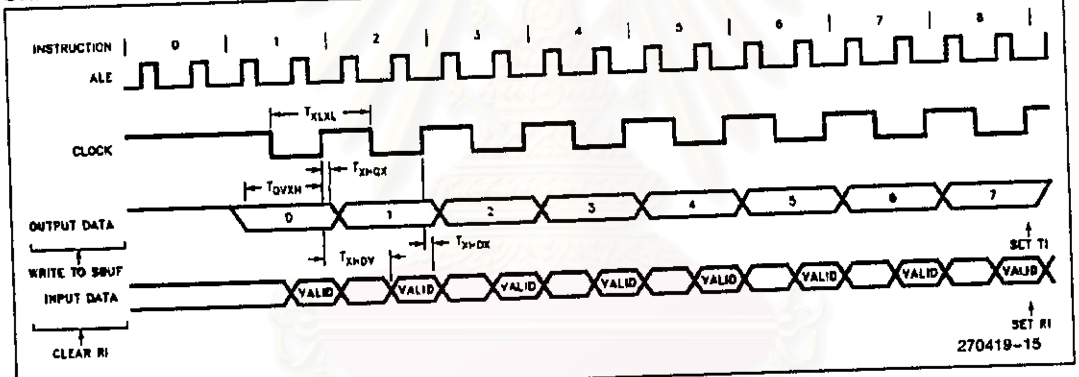
EXTERNAL CLOCK DRIVE WAVEFORM



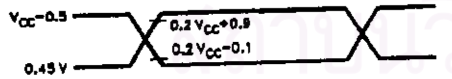
SERIAL PORT TIMING—SHIFT REGISTER MODE

Symbol	Parameter	12 MHz Oscillator		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

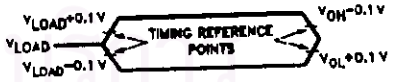


AC TESTING INPUT, OUTPUT WAVEFORMS



AC Inputs during testing are driven at $V_{CC}-0.5$ 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



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

ประวัติผู้เขียน

นาย ธเนส ศิริไตรวัฒนาพร เกิดวันที่ 23 ธันวาคม พ.ศ. 2515 ที่อำเภอเมือง จังหวัดบุรีรัมย์ สำเร็จการศึกษาปริญญาตรี อุตสาหกรรมศาสตรบัณฑิต สาขาเทคโนโลยีไฟฟ้าอุตสาหกรรม ภาควิชาเทคโนโลยีไฟฟ้าอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2537 และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2538

ประวัติผู้เขียน

นาย ธเนส ศิริไตรวัฒนาพร เกิดวันที่ 23 ธันวาคม พ.ศ. 2515 ที่อำเภอเมือง จังหวัดบุรีรัมย์ สำเร็จการศึกษาปริญญาตรี อุตสาหกรรมศาสตรบัณฑิต สาขาเทคโนโลยีไฟฟ้าอุตสาหกรรม ภาควิชาเทคโนโลยีไฟฟ้าอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2537 และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2538