LAMPIRAN

Lampiran 1

```
Program Pengendali_Kecepatan_Motor_AC_Melalui_IBM_PC;

uses crt,Graph,dos;

const
     Porta      = $300;
     Portb      = $301;
     Portc      = $302;
     Portcw     = $303;
     Cw         = $99;
     jumlahdata = 1000;
     sampling   = 14;

var lag,rpm,t                      : Real;
    data_time,data_rpm             : array[1..jumlahdata] of real;
    data_adc                       : array[1..jumlahdata] of integer;
    data_temp                      : array[0..sampling] of integer;
    set_up,N,h_grid,v_grid,i       : Integer;
    tol,data,xp,yp,xp1,yp1,j       : integer;
    grDriver,grMode,max_speed      : Integer;
    temp,k,l                       : integer;
    o                              : char;
    rev                            : pointer;
    tulisan_sementara              : string;
    tulisan_sementara2             : string;
    tulisan_sementara3             : string;
    er1,er2,de         : integer;
    kurangi,tambahi,tetap:boolean;
    error,arah:                    string;

Procedure inisial_PPI;
begin
  port[Portcw] := CW;
End;

procedure internal_timer; interrupt;
begin
   data_temp[j]:=port[porta];
   inc(j);
   port[$20]:= $20; {end of interrupt}
end;

procedure kotak(x1,y1,x2,y2,w:integer);  {Buat Tabel}
begin
   Setcolor(w);
   rectangle(x1,y1,x2,y2);
end;


PROCEDURE judul;
```

```
BEGIN
setbkcolor(7);
kotak(0,0,639,479,1);
kotak(10,10,630,470,1);
kotak(12,12,60,468,1);
settextstyle(triplexfont,1,2);
setcolor(1);
outtextxy(15,50,'AC MOTOR SPEED CONTROLLER ');
settextstyle(defaultfont,1,1);
outtextxy(50,110,'COMPUTER APPLICATION WITH PPI');
kotak(62,12,628,60,1);
kotak(67,30,623,55,1);
kotak(62,62,290,468,1);
kotak(67,92,285,463,1);
kotak(292,62,628,468,1);
kotak(297,92,623,463,1);
settextstyle(defaultfont,0,1);
setcolor(1);
outtextxy(240,18,'P R O G R A M   M E N U');
outtextxy(120,75,'ADC DATA TABLE');
outtextxy(400,75,'ADC DATA GRAPH' );
settextstyle(triplexfont,0,1);
setcolor(15);
outtextxy(120,30,'RUN');
outtextxy(270,30,'SPEED SETUP');
outtextxy(480,30,'EXIT');
setcolor(4);
outtextxy(120,30,'R');
outtextxy(270,30,'S');
outtextxy(480,30,'E');
END;

function realToStr(I: longint;koma:integer): String;
{ Convert any integer type to a string }
var
 S: string[11];
 a,b:integer;
begin
 Str(I, S);
 a:=length(s);
 if a<(koma+1) then for b:=1 to (koma-a+1) do  s:='0'+s;
 insert('.',s,(length(s)-koma+1));
 realToStr := S;
end;

function IntToStr(I: Longint): String;
{ Convert any integer type to a string }
var
 S: string[11];
begin
```

```
 Str(I, S);
 IntToStr := S;
end;

procedure dasar(x1,y1,x2,y2,hg,vg,w,bawah,atas:integer; awal,akhi
r:real);
var a,b,c,d,f:integer;
    e,g:real;
    S:String;
begin
rectangle(x1,y1,x2,y2);
c:=abs(x2-x1) div vg;
d:=abs(y2-y1) div hg;
e:=abs(akhir-awal)/vg;
f:=abs(atas-bawah) div hg;
for a:=1 to hg do
begin
s:=inttostr(bawah+f*a);
line(x1,y1+(a*d),x2,y1+(a*d));
settextstyle(smallfont,1,3);
outtextxy(x1-15,y2-((a)*d)-10,s);
end;
outtextxy(x1-15,y2-5,'0');
for a:=1 to vg do
begin
s:=realtostr(round((awal+e*a)*100),2);
line(x1+(a*c),y2,x1+(a*c),y1);
settextstyle(smallfont,0,3);
outtextxy(x1+((a)*c)-10,y2+10,s);
end;
s:=realtostr(round((awal)*100),2);
outtextxy(x1-10,y2+10,s);
end;

PROCEDURE awal_grafik(awal:boolean);    {Buat Grafik}
var       s:string;
BEGIN
cleardevice;
judul;
kotak(67,92,285,463,1);
kotak(297,92,623,463,1);
if not(awal) then
 begin
 setcolor(7);
 dasar(340,120,600,360,h_grid,v_grid,7,0,3000,t-lag,t);
 end;
setcolor(1);
dasar(340,120,600,360,h_grid,v_grid,7,0,3000,t,t+lag);
settextstyle(defaultfont,1,1);
outtextxy(315,140,'  Kecepatan Motor (RPM) ');
```

```
settextstyle(defaultfont,0,1);
outtextxy(440,395,'Waktu (detik)');

setcolor(4);
s:=inttostr(set_up);
settextstyle(smallfont,0,5);
outtextxy(80,125,'Speed Setup = '+s+' RPM');
i:=0;
END;


PROCEDURE proses;
VAR awal:boolean;
    s:string;
    tunda,range:integer;
BEGIN
awal:=true;
kotak(67,92,285,463,1);
kotak(297,92,623,463,1);
i:=26;t:=0;N:=1;j:=0;
o:='*';
er1:=round(rpm-(set_up/10));
er2:=0;
de:=er1-er2;
tunda:=0;
tambahi:=false;
kurangi:=false;
tetap:=true;
error:='0';
Getintvec($1c,rev);
Setintvec($1c,@internal_timer);
REPEAT
    IF keypressed THEN o:=readkey;
    IF i>25 THEN
    BEGIN
       awal_grafik(awal);
       awal:=false;
       t:=t+lag;
    END;

       setcolor(0);
       settextstyle(smallfont,0,5);
        temp:=0;
        for k:=1 to sampling do
        begin
        temp:=temp+data_temp[k];
        end;

       rpm:=(temp/sampling)*1.5;                        {kalibrasi rpm}
       if rpm>300 then rpm:=300;
```

```
  er1:=round(rpm-(set_up/10));
  de:=er1-er2;
  er2:=er1;
  range:=5;
  if (er1>70) then error:='2';
  if (er1<=70) and (er1>range) then error:='1';
  if (er1<=range) and (er1>=-range) then error:='0';
  if (er1<-range) and (er1>=-70) then error:='3';
  if (er1<-170) then error:='4';

  setcolor(4);
  settextstyle(smallfont,0,5);

  if (de>70) then arah:='naik2';
  if (de<=70) and (de>=0) then arah :='naik1';
  if (de<0) and (de>=-70) then arah :='turun1';
  if (de<-70) then arah :='turun2';

if error='2' then
  begin
  if arah='naik2' then begin tunda:=90;kurangi:=true; end;
  if arah='naik1' then begin tunda:=70;kurangi:=true; end;
  if arah='turun1' then begin tunda:=30;kurangi:=true;end;
  if arah='turun2' then begin tunda:=5;kurangi:=true;end;
  end;
if error='1' then
  begin
  if arah='naik2' then begin tunda:=50;kurangi:=true; end;
  if arah='naik1' then begin tunda:=40;kurangi:=true; end;
  if arah='turun1' then begin tunda:=5;kurangi:=true;end;
  if arah='turun2' then begin tunda:=15;tetap:=true; end;
  end;
 if error='0' then
  begin
  if arah='naik2' then begin tunda:=1;kurangi:=true; end;
  if arah='naik1' then begin tunda:=15;tetap:=true; end;
  if arah='turun1' then begin tunda:=15;tetap:=true;end;
  if arah='turun2' then begin tunda:=1;tambahi:=true; end;
  end;

  if error='3' then
  begin
  if arah='naik2' then begin tunda:=15;tetap:=true; end;
  if arah='naik1' then begin tunda:=5;tambahi:=true; end;
  if arah='turun1' then begin tunda:=30;tambahi:=true;end;
  if arah='turun2' then begin tunda:=50;tambahi:=true; end;
  end;
  if error='4' then
  begin
```

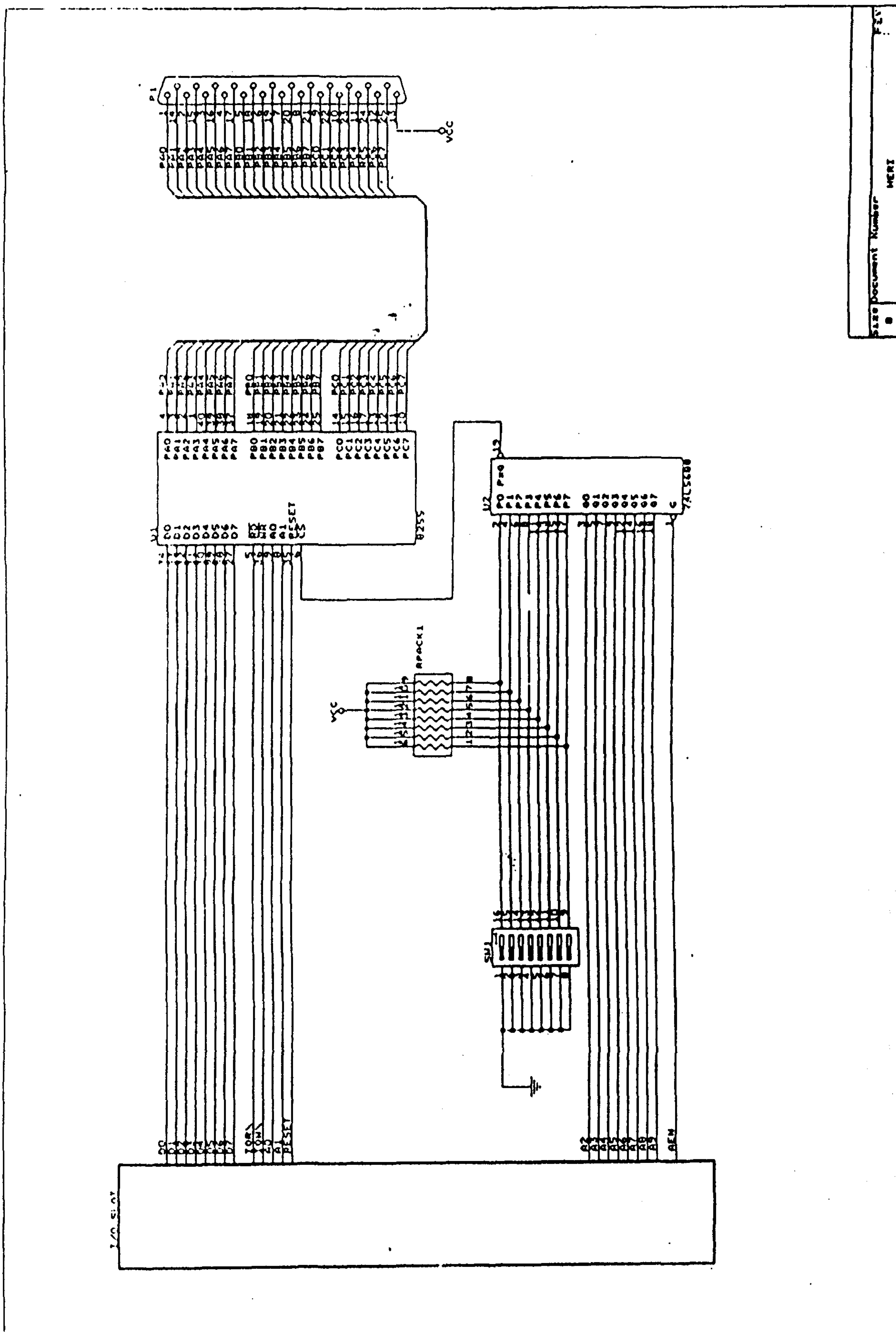
```
if arah='naik2' then begin tunda:=5;tambahi:=true; end;
if arah='naik1' then begin tunda:=20;tambahi:=true; end;
if arah='turun1' then begin tunda:=70;tambahi:=true;end;
if arah='turun2' then begin tunda:=90;tambahi:=true; end;
end;

if kurangi then
begin Port[PortB]:=1;delay(tunda);Port[PortB]:=3;
setcolor(0);
settextstyle(smallfont,0,5);
outtextxy(80,155,'kecepatan   = '+tulisan_sementara3);
setcolor(4);
tulisan_sementara3:='lebih';
settextstyle(smallfont,0,5);
outtextxy(80,155,'kecepatan   = '+tulisan_sementara3);

end;     {CW}

if tetap then
begin Port[PortB]:=3;delay(10);
setcolor(0);
settextstyle(smallfont,0,5);
outtextxy(80,155,'kecepatan   = '+tulisan_sementara3);
setcolor(4);
tulisan_sementara3:='tetap';
settextstyle(smallfont,0,5);
outtextxy(80,155,'kecepatan   = '+tulisan_sementara3);

end;     {stop}

if tambahi then
begin Port[PortB]:=0;delay(tunda);Port[PortB]:=3;
setcolor(0);
settextstyle(smallfont,0,5);
outtextxy(80,155,'kecepatan   = '+tulisan_sementara3);
setcolor(4);
tulisan_sementara3:='kurang';
settextstyle(smallfont,0,5);
outtextxy(80,155,'kecepatan   = '+tulisan_sementara3);

end;     {CCW}

tambahi:=false;
kurangi:=false;
tetap  :=false;

xp:=340+round(260/25*i);
yp:=360-trunc(rpm*0.8);
if i>0 then
begin
```

```
        setcolor(14);
        line(xp1,yp1,xp,yp);
        end;
        setcolor(0);
        settextstyle(smallfont,0,5);
        outtextxy(80,140,'Speed        = '+tulisan_sementara+' RPM'
);
        setcolor(4);
        tulisan_sementara:=inttostr(round(rpm*10));
        settextstyle(smallfont,0,5);
        outtextxy(80,140,'Speed        = '+tulisan_sementara+' RPM'
);

        setcolor(0);
        settextstyle(smallfont,0,5);
        outtextxy(80,110,'Data_ADC     = '+tulisan_sementara2);
        setcolor(4);
        tulisan_sementara2:=inttostr(round(temp/sampling));
        settextstyle(smallfont,0,5);
        outtextxy(80,110,'Data_ADC     = '+tulisan_sementara2);

        xp1:=xp; yp1:=yp;
        i:=i+1;
        port[$21] := port[$21] and $DF;  {enable Int1ch}
        while j<sampling do begin end;
        port[$21]  := port[$21] or $20; {disable Int_1ch}
        j:=0;
        UNTIL o=#27;
        port[$21]  := port[$21] or $20; {disable Int_1ch}
        setintvec($1c,rev);
        Port[PortB]:=3;       {STOP}
        cleardevice;
        judul;
        END;

function tanya_integer(x,y:integer;z:string):integer;
var s:string;
    o:char;
    I, Code: Integer;
begin
outtextxy(x,y,z);
s:='';
repeat
if keypressed then
begin
o:=readkey;
setcolor(7);
outtextxy(x+8*length(z),y,s);
if o =#8 then Delete(s,length(s),1);
if o in ['0','1','2','3','4','5','6','7','8','9'] then s:=s+o;
```

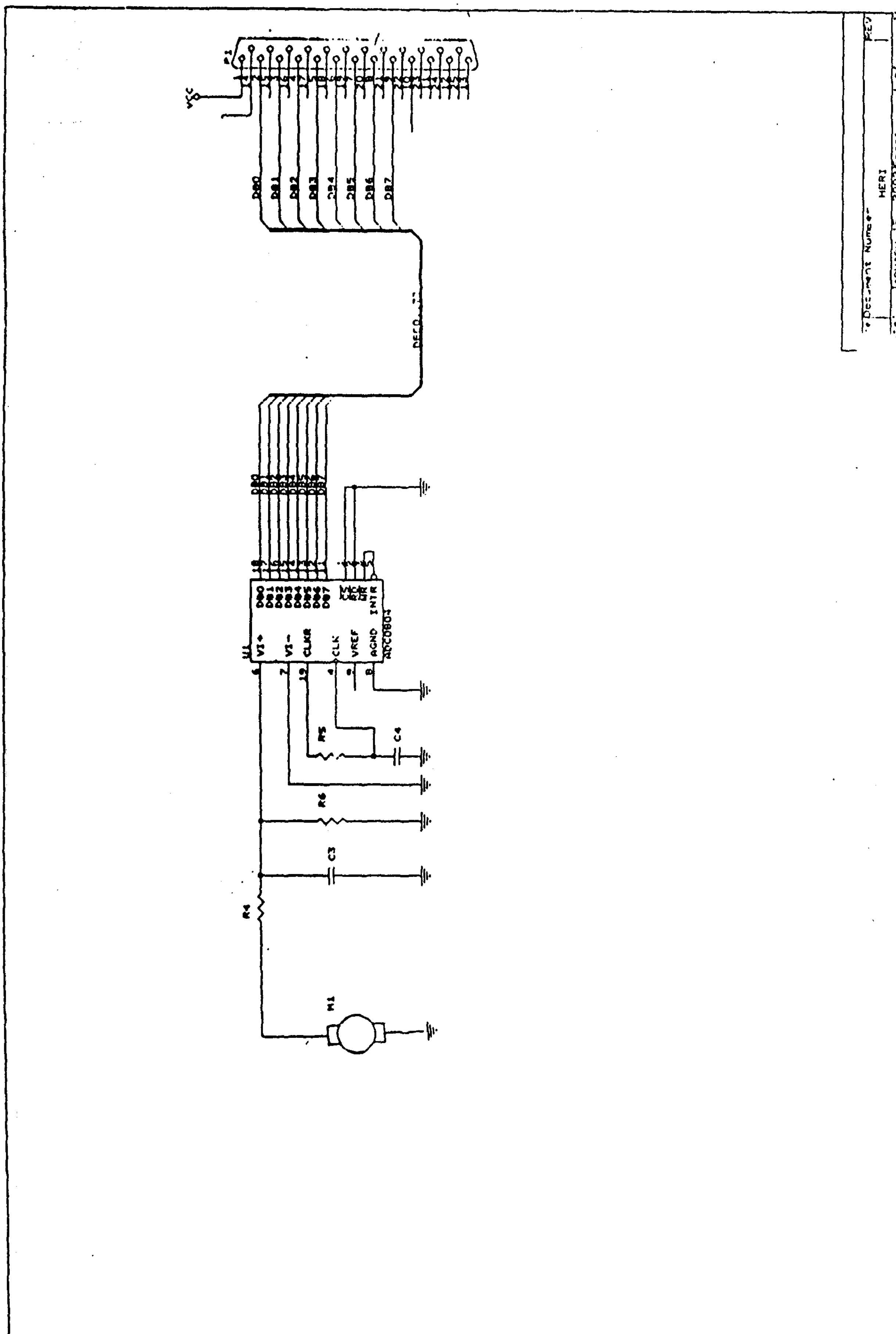
```
setcolor(1);
outtextxy(x+8*length(z),y,s);
end;
until (o=#13) or (length(s)>3);
setcolor(7);
outtextxy(x,y,z);
outtextxy(x+8*length(z),y,s);
outtextxy(310,140,'Speed Set Up ');
setcolor(1);
Val(s, I, Code);
if i>max_speed then i:=0;
tanya_integer:=i;
end;

PROCEDURE set_grid;
BEGIN
cleardevice;
judul;
kotak(297,92,623,463,1);
settextstyle(defaultfont,0,1);
outtextxy(310,140,'Speed Set Up ');
set_up:=tanya_integer(310,160,'Speed = ');
kotak(297,92,623,463,1);
END;

{---------- PROGRAM UTAMA ----------------------------------------
-----------}
BEGIN
clrscr;
  v_grid:=10;h_grid:=8;
  max_speed:=3600;
  tol:=10;
  lag:=18;
  Inisial_PPI;
  Port[PortB]:=0;
  delay(1250);
  Port[PortB]:=3;
{  delay(10000);
  delay(10000);
  Port[PortB]:=1;
  delay(200);}
  Port[PortB]:=3;
  grDriver := Detect;
  InitGraph(grDriver, grMode,'d:\tp\bgi');
  judul;
  o:='*';
  REPEAT
   IF keypressed THEN
    BEGIN
    o:=readkey;
```

```
    IF upcase(o)='R' THEN proses;;
    IF upcase(o)='S' THEN set_grid;
    END;
  UNTIL (upcase(o)='E') {or (o=#27)};
  Port[PortB]:=1;
  delay(3000);
  Port[PortB]:=3;
  closegraph;
 END.
```

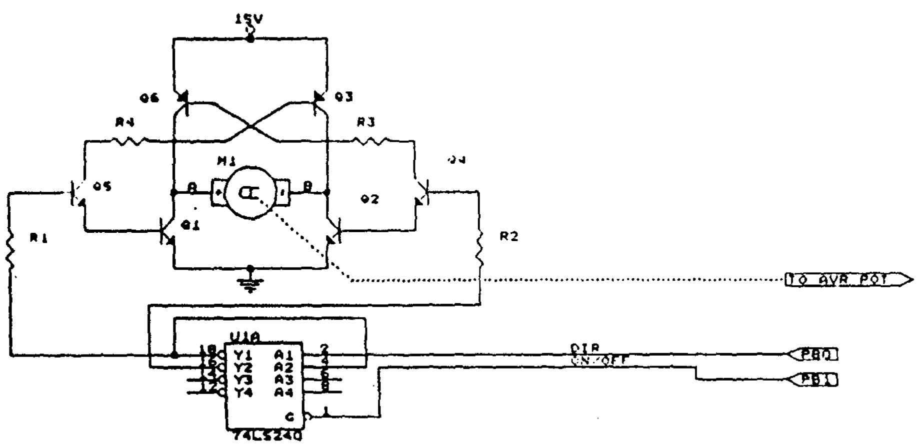
Lampiran 2
Rangkaian PPI



Lampiran 3
Rangkaian ADC



Lampiran 3
Rangkaian Penggerak Motor



MOTOR DRIVER		
Size	Document Number	REV
A	HERI HARSONO	
Date: January 21, 2002	Sheet 3 of 3	

Lampiran 4
Data Sheet Komponen yang digunakan

National Semiconductor Corporation

DM54ALS689/DM74ALS689 8-Bit Comparator

General Description

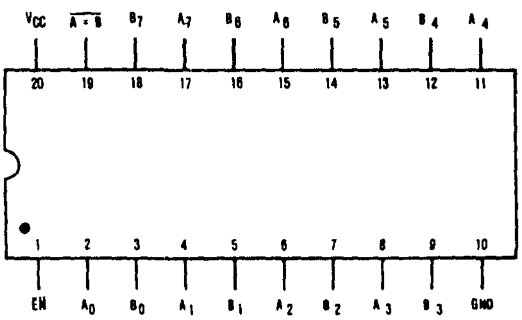
This comparator performs an "equal to" comparison of two eight-bit words with provision for expansion or external enabling. The matching of the two 8-bit inputs plus a logic LOW on the $\overline{EN}$ input produces the output $\overline{A = B}$. The ALS689 has an open collector output for wire AND cascading.

Features

- Switching specifications at 50 pF
- Switching specifications guaranteed over full temperature and V_{CC} range
- Advanced oxide-isolated, ion-implanted Schottky TTL process
- Functionally and pin for pin compatible with LS family TTL counterpart
- Improved output transient handling capability

Connection Diagram

Dual-In-Line Package



TL/F/6238-1

Order Number DM54ALS689J, DM74ALS689WM or DM74ALS689N
See NS Package Number J20A, M20B or N20A

Function Table

Inputs		Output
$\overline{EN}$	Data	$\overline{A = B}$
L	A = B	L
L	A ≠ B	H
H	X	H

H = High Level, L = Low Level, X = Don't Care

Absolute Maximum Ratings

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

Supply Voltage	7V
Input Voltage	7V
Off State Output Voltage	7V
Operating Free Air Temperature Range	
DM54ALS	−55°C to +125°C
DM74ALS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: *The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.*

Recommended Operating Conditions

Symbol	Parameter	DM54ALS689			DM74ALS689			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.5	5	5.5	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
V_{OH}	High Level Output Voltage			5.5			5.5	V
I_{OL}	Low Level Output Current			12			24	mA
T_A	Free Air Operating Temperature	−55		125	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range. All typical values are measured at V_{CC} = 5V, T_A = 25°C.

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{IK}	Input Clamp Voltage	V_{CC} = 4.5V, I_I = −18 mA				−1.5	V
I_{OH}	High Level Output Current	V_{CC} = 5.5V, V_{OH} = 5.5V				0.1	mA
V_{OL}	Low Level Output Voltage	V_{CC} = 4.5V	54/74ALS I_{OL} = 12 mA		0.25	0.4	V
			74ALS I_{OL} = 24 mA		0.35	0.5	V
I_I	Max High Input Current	V_{CC} = 5.5V, V_{IH} = 7V				0.1	mA
I_{IH}	High Level Input Current	V_{CC} = 5.5V, V_{IH} = 2.7V				20	μA
I_{IL}	Low Level Input Current	V_{CC} = 5.5V, V_{IL} = 0.4V				−0.1	mA
I_{CC}	Supply Current	V_{CC} = 5.5V (Note 1)			12	19	mA

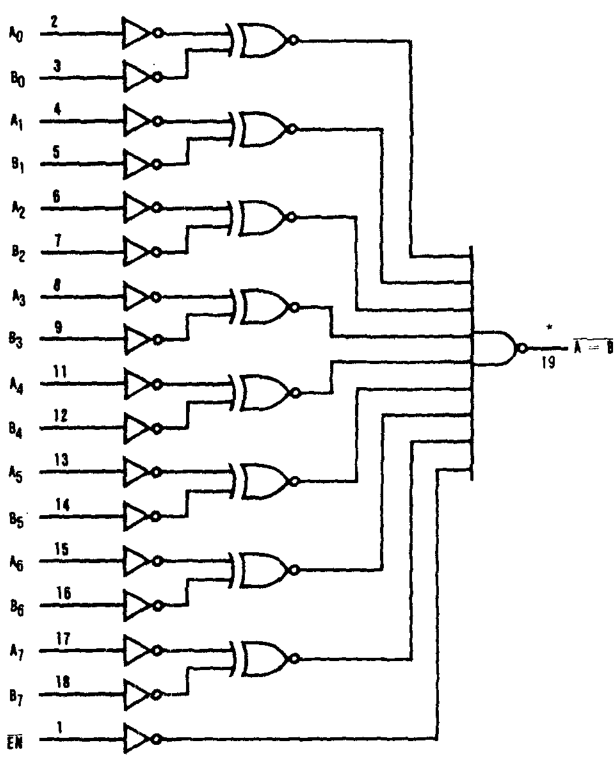
Note 1: I_{CC} is measured with $\overline{EN}$ grounded, A and B inputs at 4.5V.

Switching Characteristics *over recommended operating free air temperature range (Note 1)*

ymbol	Parameter	Conditions	From (Input)	To (Output)	DM54ALS689		DM74ALS689		Units
					Min	Max	Min	Max	
'LH	Propagation Delay Time Low to High Level Output	V_{CC} = 4.5V to 5.5V C_L = 50 pF R_L = 667Ω	A or B Data	$\overline{A = B}$	10	30	10	25	ns
'HL	Propagation Delay Time High to Low Level Output		A or B Data	$\overline{A = B}$	5	25	5	23	ns
'LH	Propagation Delay Time Low to High Level Output		$\overline{EN}$	$\overline{A = B}$	8	30	8	25	ns
PHL	Propagation Delay Time High to Low Level Output		$\overline{EN}$	$\overline{A = B}$	8	30	8	25	ns

Note 1: See Section 1 for test waveforms and output load.

Logic Diagram



TL/F/6238-2

*Output is open collector

- 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.

82C55A FUNCTIONAL DESCRIPTION

Group A and Group B Controls

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

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.

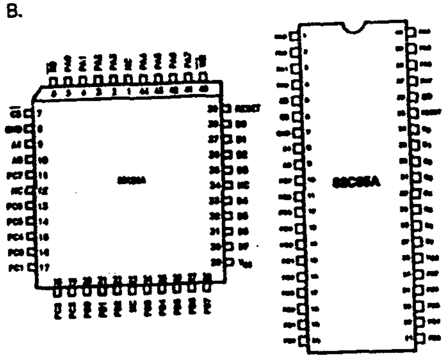
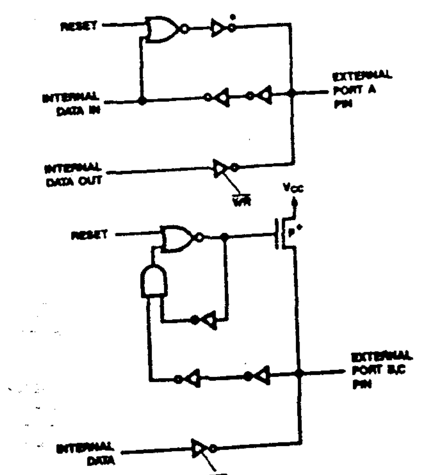


Figure 2. 82C55A Pinout

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 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.



*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

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

Symbol	Pin Number Dip	Pin Number PLCC	Type	Name and Function
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

PC2, PC4, and PC6 bit positions as illustrated by Figure 18.

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.

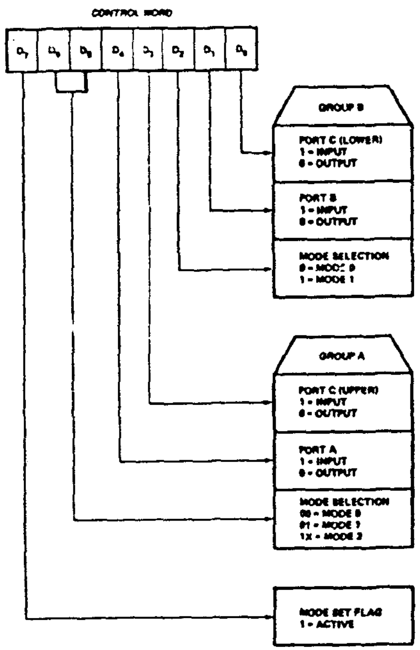


Figure 6. Mode Definition Format

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

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.

INPUT CONFIGURATION

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
GROUP A					GROUP B		

OUTPUT CONFIGURATIONS

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
GROUP A					GROUP B		

Figure 17a. MODE 1 Status Word Format

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$			
GROUP A					GROUP B		

(Defined By Mode 0 or Mode 1 Selection)

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

lute 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 above 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.

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.

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

FUNCTIONAL DESCRIPTION

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provides this function for 8-bit microprocessors.

The 8279 has two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.

The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279. These modes include:

Input Modes

- Scanned Keyboard—with encoded (8 x 8 key keyboard) or decoded (4 x 8 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key rollover.
- Scanned Sensor Matrix—with encoded (8 x 8 matrix switches) or decoded (4 x 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.
- Strobed Input—Data on return lines during control line strobe is transferred to FIFO.

Output Modes

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit (B_0 = D_0, A_3 = D_7).
- Right entry or left entry display formats.

Other features of the 8279 include:

- Mode programming from the CPU.
- Clock Prescaler
- Interrupt output to signal CPU when there is keyboard or sensor data available.
- An 8 byte FIFO to store keyboard information.
- 16 byte internal Display RAM for display refresh. This RAM can also be read by the CPU.

PRINCIPLES OF OPERATION

The following is a description of the major elements of the 8279 Programmable Keyboard/Display Interface device. Refer to the block diagram in Figure 3.

I/O Control and Data Buffers

The I/O control section uses the $\overline{CS}$, A_0, $\overline{RD}$ and $\overline{WR}$ lines to control data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by $\overline{CS}$. The character of the information, given or desired by the CPU, is identified by A_0. A logic one means the information is a command or status. A logic zero means the information is data. $\overline{RD}$ and $\overline{WR}$ determine the direction of data flow through the Data Buffers. The Data Buffers are bi-directional buffers that connect the internal bus to the external bus. When the chip is not selected ($\overline{CS}$ = 1), the devices are in a high impedance state. The drivers input during $\overline{WR}$ • $\overline{CS}$ and output during $\overline{RD}$ • $\overline{CS}$.

Control and Timing Registers and Timing Control

These registers store the keyboard and display modes and other operating conditions programmed by the CPU. The modes are programmed by presenting the proper command on the data lines with A_0 = 1 and then sending a $\overline{WR}$. The command is latched on the rising edge of $\overline{WR}$. The command is then decoded and the appropriate function is set. The timing control contains the basic timing counter chain. The first counter is a ÷ N prescaler that can be programmed to yield an internal frequency of 100 kHz which gives a 5.1 ms keyboard scan time and a 10.3 ms debounce time. The other counters divide down the basic internal frequency to provide the proper key scan, row scan, keyboard matrix scan, and display scan times.

Scan Counter

The scan counter has two modes. In the encoded mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the keyboard and display. In the decoded mode, the scan counter decodes the least significant 2 bits and provides a decoded 1 of 4 scan. Note that when the keyboard is in decoded scan, so is the display. This means that only the first 4 characters in the Display RAM are displayed.

In the encoded mode, the scan lines are active high outputs. In the decoded mode, the scan lines are active low outputs.

BUS

REA

WI

The 8279 is packaged in a 40 pin DIP. The following is a functional description of each pin.

Table 1. Pin Description

Symbol	Pin No.	Name and Function
DB_0–DB_7	19–12	**BI-DIRECTIONAL DATA BUS:** All data and commands between the CPU and the 8279 are transmitted on these lines.
CLK	3	**CLOCK:** Clock from system used to generate internal timing.
RESET	9	**RESET:** A high signal on this pin resets the 8279. After being reset the 8279 is placed in the following mode: 1) 16 8-bit character display—left entry. 2) Encoded scan keyboard—2 key lockout. Along with this the program clock prescaler is set to 31.
CS	22	**CHIP SELECT:** A low on this pin enables the interface functions to receive or transmit.
A_0	21	**BUFFER ADDRESS:** A high on this line indicates the signals in or out are interpreted as a command or status. A low indicates that they are data.
$\overline{RD}$, $\overline{WR}$	10–11	**INPUT/OUTPUT READ AND WRITE:** These signals enable the data buffers to either send data to the external bus or receive it from the external bus.
IRQ	4	**INTERRUPT REQUEST:** In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM. The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.
V_{SS}, V_{CC}	20, 40	**GROUND AND POWER SUPPLY PINS.**
SL_0–SL_3	32–35	**SCAN LINES:** Scan lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4).
RL_0–RL_7	38, 39, 1, 2, 5–8	**RETURN LINE:** Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.
SHIFT	36	**SHIFT:** The shift input status is stored along with the key position on key closure in the Scanned Keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.
CNTL/STB	37	**CONTROL/STROBED INPUT MODE:** For keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in the Strobed Input mode. (Rising Edge). It has an active internal pullup to keep it high until a switch closure pulls it low.
OUT A_0–OUT A_3 OUT B_0–OUT B_3	27–24 31–28	**OUTPUTS:** These two ports are the outputs for the 16 x 4 display refresh registers. The data from these outputs is synchronized to the scan lines (SL_0–SL_3) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8-bit port.
$\overline{BD}$	23	**BLANK DISPLAY:** This output is used to blank the display during digit switching or by a display blanking command.

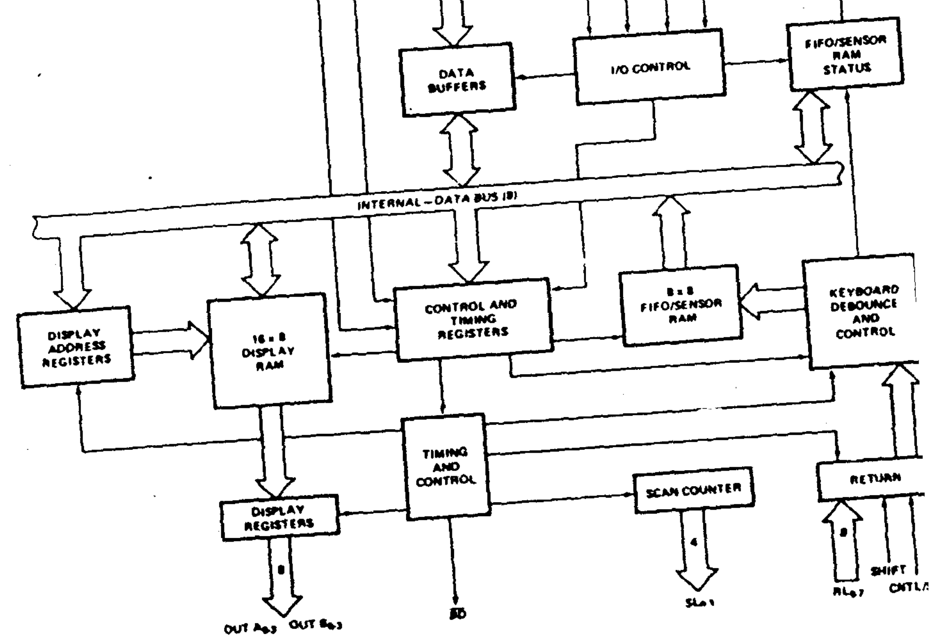


Figure 3. Internal Block Diagram

Return Buffers and Keyboard Debounce and Control

The 8 return lines are buffered and latched by the Return Buffers. In the keyboard mode, these lines are scanned, looking for key closures in that row. If the debounce circuit detects a closed switch, it waits about 10 ms to check if the switch remains closed. If it does, the address of the switch in the matrix plus the status of SHIFT and CONTROL are transferred to the FIFO. In the scanned Sensor Matrix modes, the contents of the return lines is directly transferred to the corresponding row of the Sensor RAM (FIFO) each key scan time. In Strobed Input mode, the contents of the return lines are transferred to the FIFO on the rising edge of the CNTL/STB line pulse.

FIFO/Sensor RAM and Status

This block is a dual function 8 x 8 RAM. In Keyboard or Strobed Input modes, it is a FIFO. Each new entry is written into successive RAM positions and each is then read in order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error. The status can be read by an $\overline{RD}$ with $\overline{CS}$ low and A_0 high. The status logic al[...] provides an IRQ signal when the FIFO is not emp[...] In Scanned Sensor Matrix mode, the memory i[...] Sensor RAM. Each row of the Sensor RAM is loa[...] with the status of the corresponding row of senso[...] the sensor matrix. In this mode, IRQ is high [...] change in a sensor is detected.

Display Address Registers and Displ[...] RAM

The Display Address Registers hold the addre[...] the word currently being written or read by the [...] and the two 4-bit nibbles being displayed. The r[...] write addresses are programmed by CPU comm[...] They also can be set to auto increment after [...] read or write. The Display RAM can be directly [...] by the CPU after the correct mode and addr[...] set. The addresses for the A and B nibbles are [...] matically updated by the 8279 to match data [...] by the CPU. The A and B nibbles can be e[...] independently or as one word, according [...] mode that is set by the CPU. Data entry to th[...] play can be set to either left or right entry. Se[...] face Considerations for details.

8279 Commands

The following commands program the 8279 operating modes. The commands are sent on the Data Bus with $\overline{CS}$ low and A_0 high and are loaded to the 8279 on the rising edge of $\overline{WR}$.

Keyboard/Display Mode Set

	MSB							LSB
Code:	0	0	0	D	D	K	K	K

Where DD is the Display Mode and KKK is the Keyboard Mode.

DD

0	0	8 8-bit character display—Left entry
0	1	16 8-bit character display—Left entry*
1	0	8 8-bit character display—Right entry
1	1	16 8-bit character display—Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

0	0	0	Encoded Scan Keyboard—2 Key Lockout*
0	0	1	Decoded Scan Keyboard—2-Key Lockout
0	1	0	Encoded Scan Keyboard—N-Key Rollover
0	1	1	Decoded Scan Keyboard—N-Key Rollover
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed Input, Encoded Display Scan
1	1	1	Strobed Input, Decoded Display Scan

*Default after reset.

Program Clock

Code:	0	0	1	P	P	P	P	P

All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (pin 3) by a programmable integer. Bits PPPPP determine the value of this integer which ranges from 2 to 31. Choosing a divisor that yields 100 kHz will give the specified scan and debounce times. For instance, if Pin 3 of the 8279 is being clocked by a 2 MHz signal, PPPPP should be set to 10100 to divide the clock by 20 to yield the proper 100 kHz operating frequency.

Code:	0	1	0	AI	X	A	A	A	X = Don't Care

The CPU sets the 8279 for a read of the FIFO/Sensor RAM by first writing this command. In the Scan Keyboard Mode, the Auto-Increment flag (AI) and the RAM address bits (AAA) are irrelevant. The 8279 will automatically drive the data bus for each subsequent read ($A_0 = 0$) in the same sequence in which the data first entered the FIFO. All subsequent reads will be from the FIFO until another command is issued.

In the Sensor Matrix Mode, the RAM address bits AAA select one of the 8 rows of the Sensor RAM. If the AI flag is set (AI = 1), each successive read will be from the subsequent row of the sensor RAM.

Read Display RAM

Code:	0	1	1	AI	A	A	A	A

The CPU sets up the 8279 for a read of the Display RAM by first writing this command. The address bits AAAA select one of the 16 rows of the Display RAM. If the AI flag is set (AI = 1), this row address will be incremented after each following read *or write* to the Display RAM. Since the same counter is used for both reading and writing, this command sets the next read or *write* address and the sense of the Auto-Increment mode for both operations.

Write Display RAM

Code:	1	0	0	AI	A	A	A	A

The CPU sets up the 8279 for a write to the Display RAM by first writing this command. After writing the command with $A_0 = 1$, all subsequent writes with $A_0 = 0$ will be to the Display RAM. The addressing and Auto-Increment functions are identical to those for the Read Display RAM. However, this command does not affect the source of subsequent Data Reads; the CPU will read from whichever RAM (Display of FIFO/Sensor) which was last specified. If, indeed, the Display RAM was last specified, the Write Display RAM will, nevertheless, change the next Read location.

Display Write Inhibit/Blanking

					A	B	A	B
Code:	1	0	1	X	IW	IW	BL	BL

The IW Bits can be used to mask nibble A and nibble B in applications requiring separate 4-bit display ports. By setting the IW flag (IW = 1) for one of the ports, the port becomes marked so that entries to the Display RAM from the CPU do not affect that port. Thus, if each nibble is input to a BCD decoder,

out affecting the other digit being ... portant to note that bit B_0 corresponds to bit D_0 on the CPU bus, and that bit A_3 corresponds to bit D_7.

If the user wishes to blank the display, the BL flags are available for each nibble. The last Clear command issued determines the code to be used as a "blank." This code defaults to all zeros after a reset. Note that both BL flags must be set to blank a display formatted with a single 8-bit port.

Clear

Code:	1	1	0	C_D	C_D	C_D	C_F	C_A

The C_D bits are available in this command to clear all rows of the Display RAM to a selectable blanking code as follows:

C_D C_D C_D

0	X	All Zeros (X = Don't Care)
1	0	AB = Hex 20 (0010 0000)
1	1	All Ones

Enable clear display when = 1 (or by $C_A = 1$)

During the time the Display RAM is being cleared (~160 μs), it may not be written to. The most significant bit of the FIFO status word is set during this time. When the Display RAM becomes available again, it automatically resets.

If the C_F bit is asserted ($C_F = 1$), the FIFO status is cleared and the interrupt output line is reset. Also, the Sensor RAM pointer is set to row 0.

C_A, the Clear All bit, has the combined effect of C_D and C_F; it uses the C_D clearing code on the Display RAM and also clears FIFO status. Furthermore, it resynchronizes the internal timing chain.

End Interrupt/Error Mode Set

Code:	1	1	1	E	X	X	X	X	X = Don't care

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM. (The IRQ line would have been raised upon the detection of a change in a sensor value. This would have also inhibited further writing into the RAM until reset).

For the N-key rollover mode—if the E bit is programmed to "1" the chip will operate in the special Error mode. (For further details, see Interface Considerations Section.)

Status Word

The status word contains the FIFO status, error, and

Interface Considerations ... word.

Data Read

Data is read when A_0, $\overline{CS}$ and $\overline{RD}$ are all low. ... source of the data is specified by the Read FIF... Read Display commands. The trailing edge o... will cause the address of the RAM being read ... incremented if the Auto-Increment flag is set. ... reads always increment (if no error occurs) inde... dent of AI.

Data Write

Data that is written with A_0, $\overline{CS}$ and $\overline{WR}$ low ... ways written to the Display RAM. The addre... specified by the latest Read Display or Write Di... command. Auto-Incrementing on the rising ... of $\overline{WR}$ occurs if AI is set by the latest d... command.

INTERFACE CONSIDERATIONS

Scanned Keyboard Mode, 2-Key Lockout

There are three possible combinations of con... that can occur during debounce scanning. W... key is depressed, the debounce logic is set. ... depressed keys are looked for during the ne... scans. If none are encountered, it is a sing... depression and the key position is entered i... FIFO along with the status of CNTL and SHIF... If the FIFO was empty, IRQ will be set to sig... CPU that there is an entry in the FIFO. If th... was full, the key will not be entered and th... flag will be set. If another closed switch is e... tered, no entry to the FIFO can occur. If a... keys are released before ths one, then it wil... tered to the FIFO. If this key is released bef... other, it will be entirely ignored. A key is en... the FIFO only once per depression, no mat... many keys were pressed along with it or ... order they were released. If two keys are de... within the debounce cycle, it is a simultane... pression. Neither key will be recognized ... key remains depressed alone. The last ke... treated as a single key depression.

Scanned Keyboard Mode, N-Key Rollover

With N-key Rollover each key depression ... independently from all others. When a k... pressed, the debounce circuit waits 2 scans and then checks to see if the key is still down. If it is, the key is entered into the FIFO. Any number of keys can be depressed and another can be recognized and entered into the FIFO. If a simultaneous depression occurs, the keys are recognized and entered according to the order the keyboard scan found them.

Scanned Keyboard—Special Error Modes

For N-key rollover mode the user can program a special error mode. This is done by the "End Interrupt/Error Mode Set" command. The debounce cycle and key-validity check are as in normal N-key mode. If during a *single debounce cycle*, two keys are found depressed, this is considered a simultaneous multiple depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. (See "FIFO STATUS" for further details.) The error flag is reset by sending the normal CLEAR command with CF = 1.

Sensor Matrix Mode

In Sensor Matrix mode, the debounce logic is inhibited. The status of the sensor switch is inputted directly to the Sensor RAM. In this way the Sensor RAM keeps an image of the state of the switches in the sensor matrix. Although debouncing is not provided, this mode has the advantage that the CPU knows how long the sensor was closed and when it was released. A keyboard mode can only indicate a validated closure. To make the software easier, the designer should functionally group the sensors by row since this is the format in which the CPU will read them.

The IRQ line goes high if any sensor value change is detected at the end of a sensor matrix scan. The IRQ line is cleared by the first data read operation if the Auto-Increment flag is set to zero, or by the End Interrupt command if the Auto-Increment flag is set to one.

NOTE:
Multiple changes in the matrix Addressed by (SL_{0-3} = 0) may cause *multiple interrupts*. (SL_0 = 0 in the Decoded Mode.) Reset may cause the 8279 to see multiple changes.

Data Format

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted). CNTL is the MSB of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.

MSB			LSB
CNTL	SHIFT	SCAN	RETURN

SCANNED KEYBOARD DATA FORMAT

In Sensor Matrix mode, the data on the return lines is entered directly in the row of the Sensor RAM that corresponds to the row in the matrix being scanned. Therefore, each switch position maps directly to a Sensor RAM position. The SHIFT and CNTL inputs are ignored in this mode. Note that switches are not necessarily the only thing that can be connected to the return lines in this mode. Any logic that can be triggered by the scan lines can enter data to the return line inputs. Eight multiplexed input ports could be tied to the return lines and scanned by the 8279.

MSB							LSB
RL_7	RL_6	RL_5	RL_4	RL_3	RL_2	RL_1	RL_0

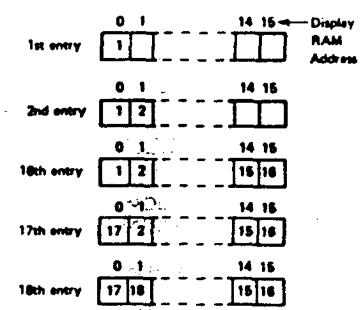
In Strobed Input mode, the data is also entered to the FIFO from the return lines. The data is entered by the rising edge of a CNTL/STB line pulse. Data can come from another encoded keyboard or simple switch matrix. The return lines can also be used as a general purpose strobed input.

MSB							LSB
RL_7	RL_6	RL_5	RL_4	RL_3	RL_2	RL_1	RL_0

Display

Left Entry

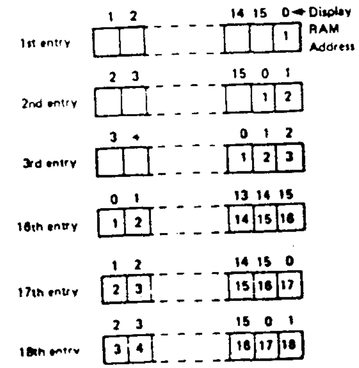
Left Entry mode is the simplest display format in that each display position directly corresponds to a byte (or nibble) in the Display RAM. Address 0 in the RAM is the left-most display character and address 15 (or address 7 in 8 character display) is the right most display character. Entering characters from position zero causes the display to fill from the left. The 17th (9th) character is entered back in the left most position and filling again proceeds from there.



Left Entry Mode (Auto Increment)

Right Entry

Right entry is the method used by most electronic calculators. The first entry is placed in the right most display character. The next entry is also placed in the right most character after the display is shifted left one character. The left most character is shifted off the end and is lost.

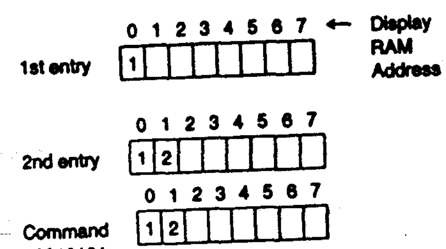


Right Entry Mode (Auto Increment)

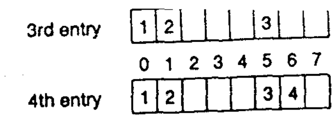
Note that now the display position and register address do not correspond. Consequently, entering a character to an arbitrary position in the Auto Increment mode may have unexpected results. Entry starting at Display RAM address 0 with sequential entry is recommended.

Auto Increment

In the Left Entry mode, Auto Incrementing causes the address where the CPU will next write to be incremented by one and the character appears in the next location. With non-Auto Incrementing the entry is both to the same RAM address and display position. Entry to an arbitrary address in the Auto Increment mode has no undesirable side effects and the result is predictable:

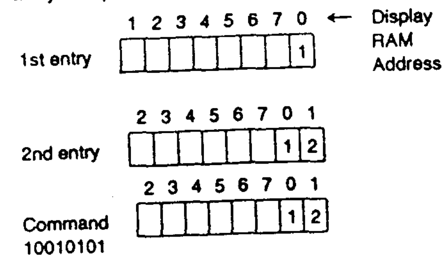


Enter next at Location 5 Auto Increment

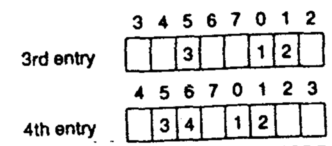


LEFT ENTRY MODE (AUTO INCREMENT)

In the Right Entry mode, Auto Incrementing and non Incrementing have the same effect as in the Left Entry except if the address sequence is interrupted.

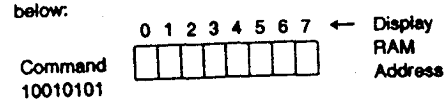


Enter next at Location 5 Auto Increment

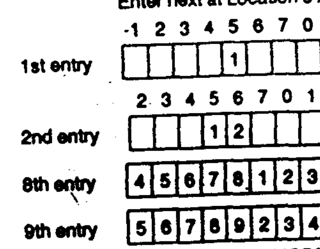


RIGHT ENTRY MODE (AUTO INCREMENT)

Starting at an arbitrary location operates as shown below:



Enter next at Location 5 Auto Increment



RIGHT ENTRY MODE (AUTO INCREMENT)

Entry appears to be from the initial entry point.

8/16 Character Display Formats

If the display mode is set to an 8 character display acters vs. 10.3 ms for 16 characters with 100 kHz internal frequency).

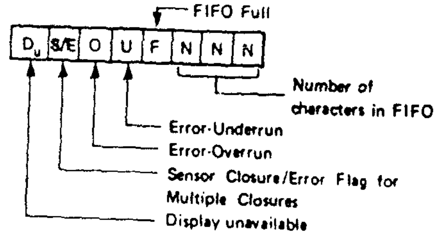
G. FIFO Status

FIFO status is used in the Keyboard and Strobed Input modes to indicate the number of characters in the FIFO and to indicate whether an error has occurred. There are two types of errors possible: overrun and underrun. Overrun occurs when the entry of another character into a full FIFO is attempted. Underrun occurs when the CPU tries to read an empty FIFO.

The FIFO status word also has a bit to indicate that the Display RAM was unavailable because a Clear Display or Clear All command had not completed its clearing operation.

In a Sensor Matrix mode, a bit is set in the FIFO status word to indicate that at least one sensor closure indication is contained in the Sensor RAM.

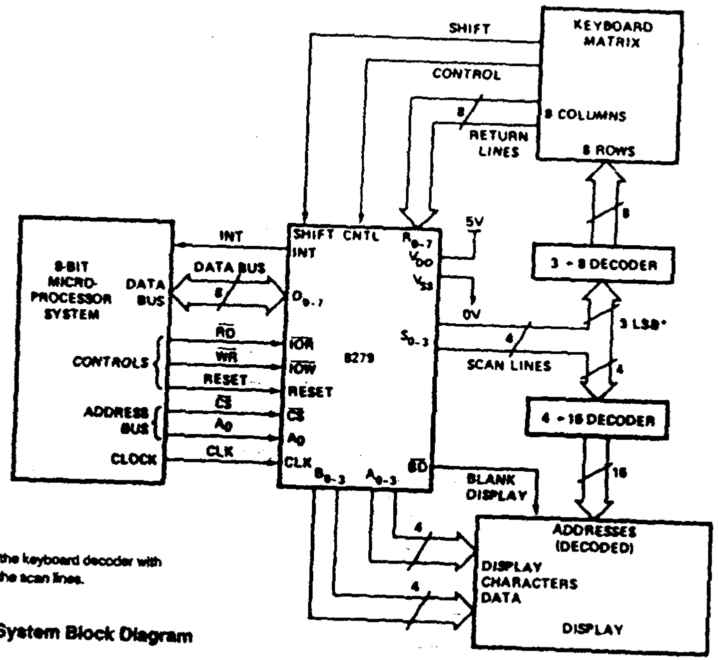
In Special Error Mode the S/E bit is showing the error flag and serves as an indication to whether a simultaneous multiple closure error has occurred.



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature .... 0°C to 70°C
Storage Temperature .... −65°C to 125°C
Voltage on any Pin with Respect to Ground .... −0.5V to +7V
Power Dissipation .... 1 Watt

Notice: Stresses above 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 above 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.



*Do not drive the keyboard decoder with the MSB of the scan lines.

Figure 4. System Block Diagram

Symbol	Parameter	Min	Max	Unit	
V_{IL1}	Input Low Voltage for Return Lines	−0.5	1.4	V	
V_{IL2}	Input Low Voltage for All Others	−0.5	0.8	V	
V_{IH1}	Input High Voltage for Return Lines	2.2		V	
V_{IH2}	Input High Voltage for All Others	2.0		V	
V_{OL}	Output Low Voltage		0.45	V	(Note 1)
V_{OH1}	Output High Voltage on Interrupt Line	3.5		V	(Note 2)
V_{OH2}	Other Outputs	2.4			I_{OH} = −400 µA 82 −100 µA 82
I_{IL1}	Input Current on Shift, Control and Return Lines		+10 −100	µA µA	$V_{IN} = V_{CC}$ V_{IN} = 0V
I_{IL2}	Input Leakage Current on All Others		±10	µA	$V_{IN} = V_{CC}$ to 0V
I_{OFL}	Output Float Leakage		±10	µA	$V_{OUT} = V_{CC}$ to 0.45
I_{CC}	Power Supply Current		120	mA	
C_{IN}	Input Capacitance		10	pF	f_C = 1 MHz Unmeas Pins Returned to V_{SS}
C_{OUT}	Output Capacitance		20	pF	

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{SS} = 0V (Note 3)*

Bus Parameters

READ CYCLE

Symbol	Parameter	8279		8279-5		U
		Min	Max	Min	Max	
t_{AR}	Address Stable Before $\overline{READ}$	50		0		
t_{RA}	Address Hold Time for $\overline{READ}$	5		0		
t_{RR}	$\overline{READ}$ Pulse Width	420		250		
t_{RD}(4)	Data Delay from $\overline{READ}$		300		150	
t_{AD}(4)	Address to Data Valid		450		250	
t_{DF}	$\overline{READ}$ to Data Floating	10	100	10	100	
t_{RCY}	Read Cycle Time	1		1		
t_{AW}	Address Stable Before $\overline{WRITE}$	50		0		
t_{WA}	Address Hold Time for $\overline{WRITE}$	20		0		

WRITE CYCLE

Symbol	Parameter	8279		8279-5	
		Min	Max	Min	Max
t_{WW}	$\overline{WRITE}$ Pulse Width	400		250	
t_{DW}	Data Set Up Time for $\overline{WRITE}$	300		150	
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		0	
t_{WCY}	Write Cycle Time	1		1	

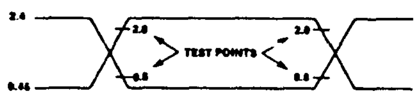
OTHER TIMINGS

Symbol	Parameter	8279		8279-5		Unit
		Min	Max	Min	Max	
$t_{\phi W}$	Clock Pulse Width	230		120		ns
t_{CY}	Clock Period	500		320		ns

Keyboard Scan Time .... 5.1 ms
Keyboard Debounce Time .... 10.3 ms
Key Scan Time .... 80 μs
Display Scan Time .... 10.3 ms
Digit-on Time .... 480 μs
Blanking Time .... 160 μs
Internal Clock Cycle(5) .... 10 μs

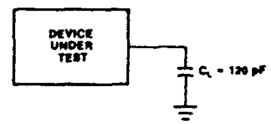
NOTES:
1. 8279, I_{OL} = 1.6 mA; 8279-5, I_{OL} = 2.2 mA.
2. I_{OH} = −100 μA
3. 8279, V_{CC} = +5V ±5%; 8279-5, V_{CC} = +5V ±10%
4. 8279, C_L = 100 pF; 8279-5, C_L = 150 pF.
5. The Prescaler should be programmed to provide a 10 μs internal clock cycle.
6. Sampled not 100% tested. T_A = 25°C.
* For Extended Temperature EXPRESS, use M8279A electrical parameters.

A.C. TESTING INPUT, OUTPUT WAVEFORM



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 0.8V for a Logic "0".

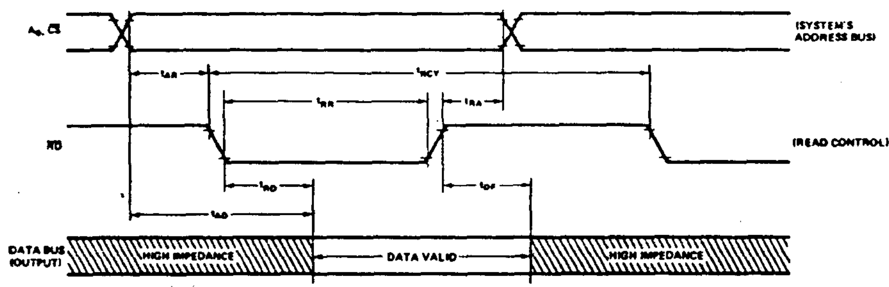
A.C. TESTING LOAD CIRCUIT



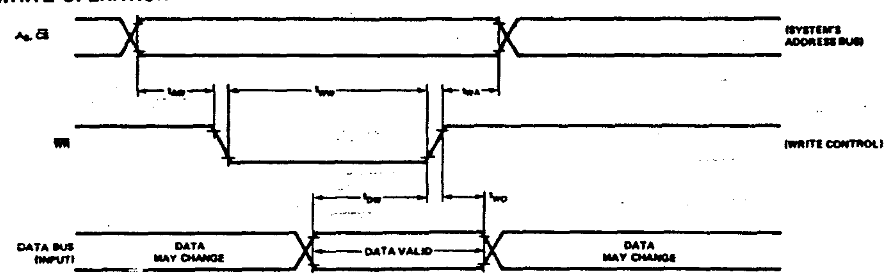
C_L = 120 pF
C_L Includes Jig Capacitance

WAVEFORMS

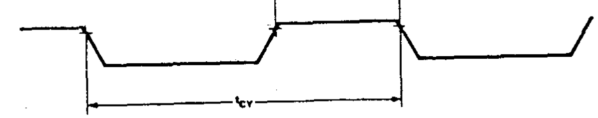
READ OPERATION



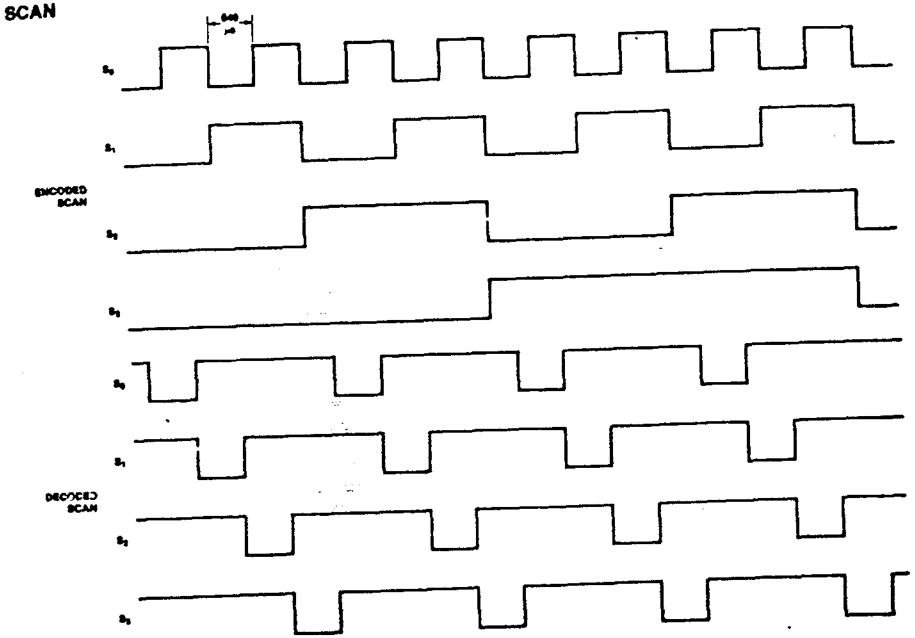
WRITE OPERATION



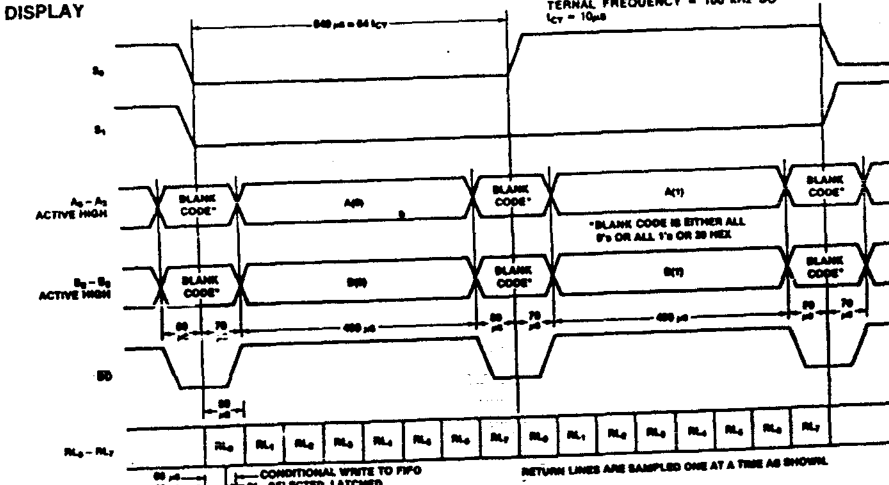
CLOCK INPUT



SCAN



DISPLAY



NOTE:
Shown is encoded scan left entry
S_2–S_3 are not shown but they are simply S_1 divided by 2 and 4.

FEATURES

- Provides a direct interface between Z80 microcomputer systems and peripheral devices.
- Two ports with interrupt-driven handshake for fast response.
- Four programmable operating modes: Output, Input, Bidirectional (Port A only), and Bit Control
- Programmable interrupts on peripheral status conditions.
- NMOS version for high cost performance solutions.
- CMOS version for the designs requiring low power consumption.
- NMOS Z0842004 - 4 MHz, Z0842006 - 6.17 MHz.
- CMOS Z84C2004 - DC to 4 MHz, Z84C2006 - DC to 6.17 MHz, Z84C2008 - DC to 8 MHz.
- Standard Z80 Family bus-request and prioritized interrupt-request daisy chains implemented without external logic.
- The eight Port B outputs can drive Darlington transistors (1.5 mA at 1.5V).
- 6 MHz version supports 6.144 MHz CPU clock operation.

GENERAL DESCRIPTION

The Z80 PIO Parallel I/O Circuit (hereinafter referred to as the Z80 PIO or PIO) is a programmable, dual-port device that provides a TTL-compatible interface between peripheral devices and the Z80 CPU (Figures 1 and 2). Note the QFP package is only available in CMOS version. The CPU configures the Z80 PIO to interface with a wide range of

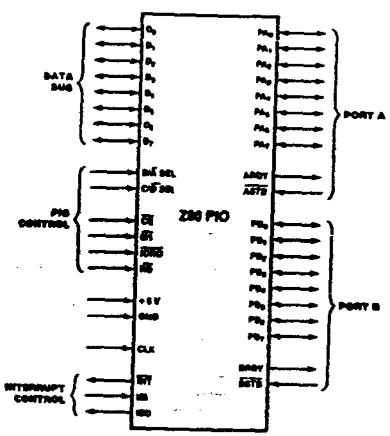


Figure 1. Pin Functions

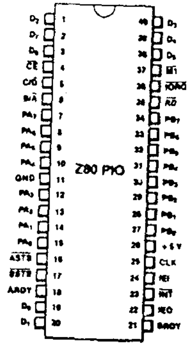


Figure 2a. 40-pin Dual-In-Line Package (DIP), Pin Assignments

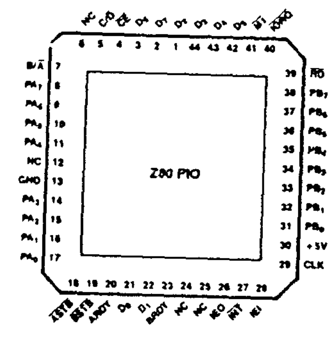


Figure 2b. 44-pin Chip Carrier, Pin Assignments

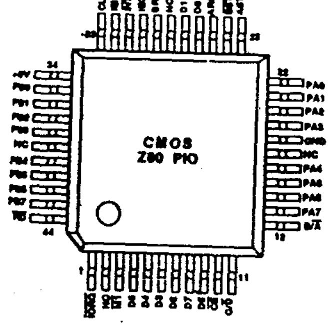


Figure 2c. 44-pin Quad Flat Pack Pin Assignments.

peripheral devices that are compatible with the Z80 PIO include most keyboards, paper tape readers and punches, printers, and PROM programmers.

One characteristic of the Z80 peripheral controllers that separates them from other interface controllers is that all data transfer between the peripheral device and the CPU is accomplished under interrupt control. Thus, the interrupt logic of the PIO permits full use of the efficient interrupt capabilities of the Z80 CPU during I/O transfers. All logic necessary to implement a fully nested interrupt structure is included in the PIO (Figure 3).

Another feature of the PIO is the ability to interrupt the CPU upon occurrence of specified status conditions in the peripheral device. For example, the PIO can be programmed to interrupt if any specified peripheral alarm conditions should occur. This interrupt capability reduces the time the processor must spend in polling peripheral status.

The Z80 PIO interfaces to peripherals via two independent general-purpose I/O ports, designated Port A and Port B. Each port has eight data bits and two handshake signals, Ready and Strobe, which control data transfer. The Ready output indicates to the peripheral that the port is ready for a data transfer. Strobe is an input from the peripheral that indicates when a data transfer has occurred.

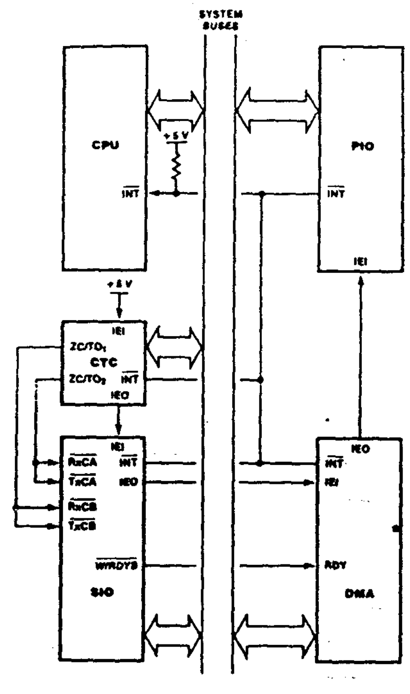


Figure 3. PIO in a Typical Z80 Family Environment

Operating Modes. The Z80 PIO ports can be programme to operate in four modes: Output (Mode 0), Input (Mode 1 Bidirectional (Mode 2) and Bit Control (Mode 3).

Either Port A or Port B can be programmed to output data i Mode 0. Both ports have output registers that ar individually addressed by the CPU; data can be written t either port at any time. When data is written to a port, a active Ready output indicates to the external device tha data is available at the associated port and is ready fo transfer to the external device. After the data transfer, th external device responds with an active Strobe input, whic generates an interrupt, if enabled.

Either Port A or Port B can be programmed to input data i Mode 1. Each port has an input register addressed by th CPU. When the CPU reads data from a port, the PIO sets th Ready signal, which is detected by the external device. Th external device then places data on the I/O lines and strobe the I/O port, which latches the data into the Port Inpu Register, resets Ready, and triggers the Interrupt Request, i enabled. The CPU can read the input data at any time which again sets Ready.

Mode 2 is bidirectional and uses only Port A, plus th interrupts and handshake signals from both ports. Port E must be set to Mode 3 and masked off from generating interrupts. In operation, Port A is used for both data inpu and output. Output operation is similar to Mode 0 excep that data is allowed out onto the Port A bus only when ASTE is Low. For input, operation is similar to Mode 1, except tha the data input uses the Port B handshake signals and the Port B interrupt, if enabled.

Both ports can be used in Mode 3. In this mode, the individual bits are defined as either input or output bits. This provides up to eight separate, individually defined bits fo each port. During operation, Ready and Strobe are no used. Instead, an interrupt is generated if the condition o one input changes, or if all inputs change. The requirements for generating an interrupt are defined during the programming operation; the active level is specified as either High or Low, and the logic condition is specified as either one input active (OR) or all inputs active (AND). Fo example, if the port is programmed for active Low inputs and the logic function is AND, then all inputs at the specified por must go Low to generate an interrupt.

Data outputs are controlled by the CPU and can be writter or changed at any time.

- Individual bits can be masked off.
- The handshake signals are not used in Mode 3; Ready is held Low, and Strobe is disabled.
- When using the Z80 PIO interrupts, the Z80 CPU interrupt mode must be set to Mode 2.

INTERNAL STRUCTURE

The internal structure of the Z80 PIO consists of a Z80 CPU bus interface, internal control logic, Port A I/O logic, Port B I/O logic, and interrupt control logic (Figure 4). The CPU bus interface logic allows the Z80 PIO to interface directly to the Z80 CPU with no other external logic. The internal control

... obtain an output digital code change. The
...ation error have to be accounted for in
the test results. A plot of this natural
... in *Figure 5* where, for clarity, both the
...nd the error voltage are normalized to

...Ds

	INPUT VOLTAGE VALUE WITH 10.24 V_{REF}	
	MS GROUP	LS GROUP
5/256	9.600	0.600
	8.960	0.560
3/256	8.320	0.520
	7.680	0.480
1/256	7.040	0.440
	6.400	0.400
/256	5.760	0.360
	5.120	0.320
'256	4.480	0.280
	3.840	0.240
'256	3.200	0.200
	2.560	0.160
256	1.920	0.120
	1.280	0.080
256	0.640	0.040
	0	0

H/5670-8

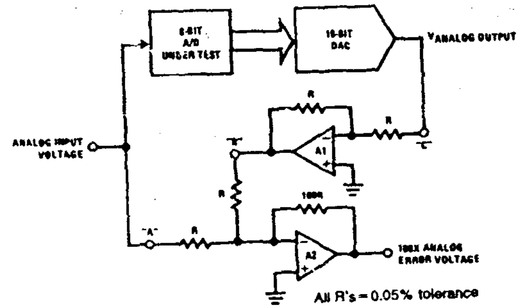
...ation Error

nary counting sequence from zero to full...

The techniques described so far are suitable for an engineering evaluation or a quick check on performance. For a higher speed test system, or to obtain plotted data, a digital-to-analog converter is needed for the test set-up. An accurate 10-bit DAC can serve as the precision voltage source for the A/D. Errors of the A/D under test can be provided as either analog voltages or differences in two digital words.

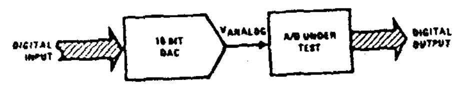
A basic A/D tester which uses a DAC and provides the error as an analog output voltage is shown in *Figure 6*. The 2 op amps can be eliminated if a lab DVM with a numerical subtraction feature is available to directly readout the difference voltage, "A−C".

output code transitions can be detected as the 10-bit DAC is incremented. This provides ¼ LSB steps for the 8-bit A/D under test. If the results of this test are automatically plotted with the analog input on the X axis and the error (in LSB's) as the Y axis, a useful transfer function of the A/D under test results. For acceptance testing, the plot is not necessary and the testing speed can be increased by establishing internal limits on the allowed error for each code.



TL/H/5670-16

FIGURE 6. A/D Tester with Analog Error Output

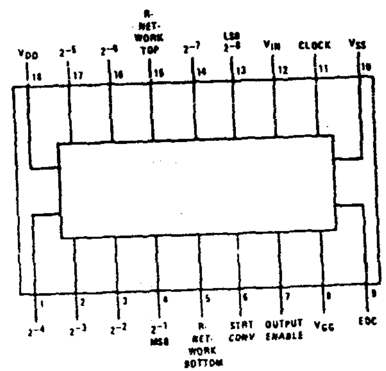


TL/H/5670-17

FIGURE 7. Basic "Digital" A/D Tester

Connection Diagram

Dual-In-Line Package

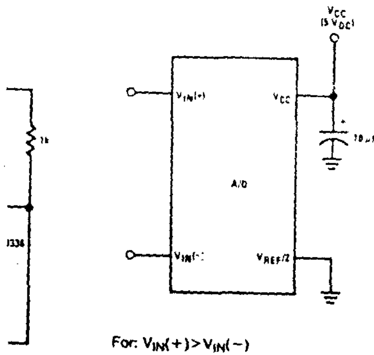


TL/H/5670-9

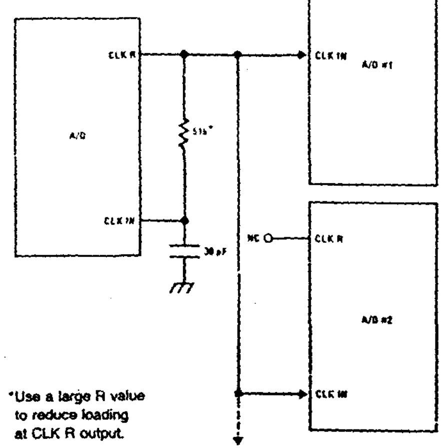
Top View

Order Number ADC0800PD or ADC0800PCD
See NS Package Number D18A

A µP Interfaced Comparator

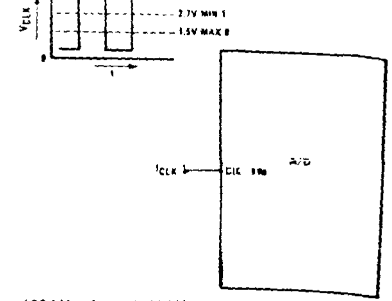


For: $V_{IN}(+) > V_{IN}(-)$
Output = FF_{HEX}
For: $V_{IN}(+) < V_{IN}(-)$
Output = 00_{HEX}



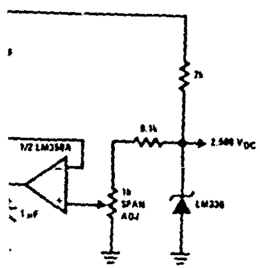
*Use a large R value to reduce loading at CLK R output.

IF MORE THAN 5 ADDITIONAL A/Ds, USE A CMOS BUFFER (NOT T^2L)

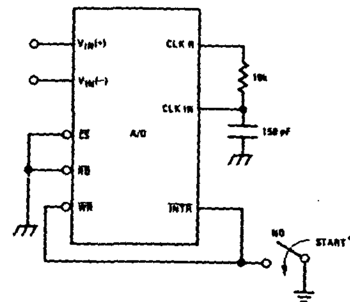


100 kHz ≤ f_{CLK} ≤ 1460 kHz

...ed Range

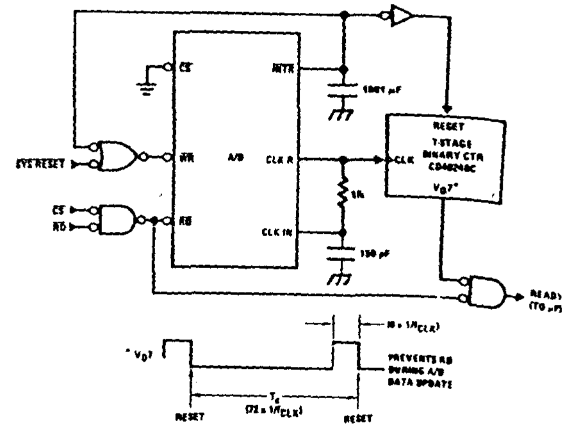


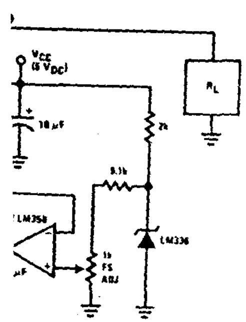
Self-Clocking in Free-Running Mode



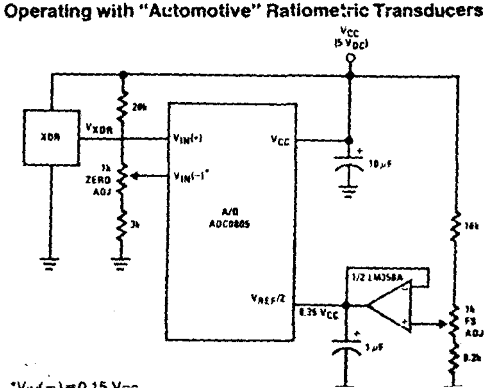
*After power up, a momentary grounding of the $\overline{WR}$ input is needed to guarantee operation.

µP Interface for Free-Running A/D

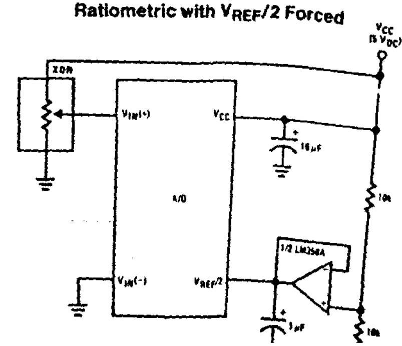


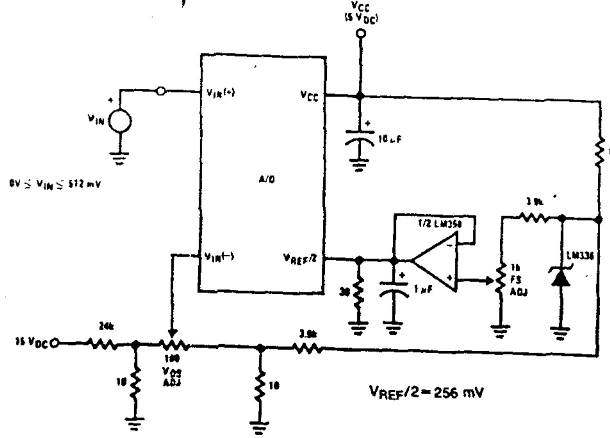


Operating with "Automotive" Ratiometric Transducers



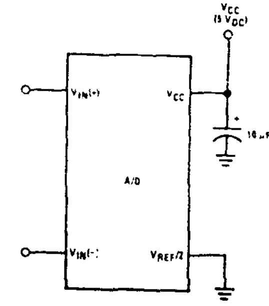
Ratiometric with $V_{REF}/2$ Forced





$V_{REF}/2 = 256$ mV

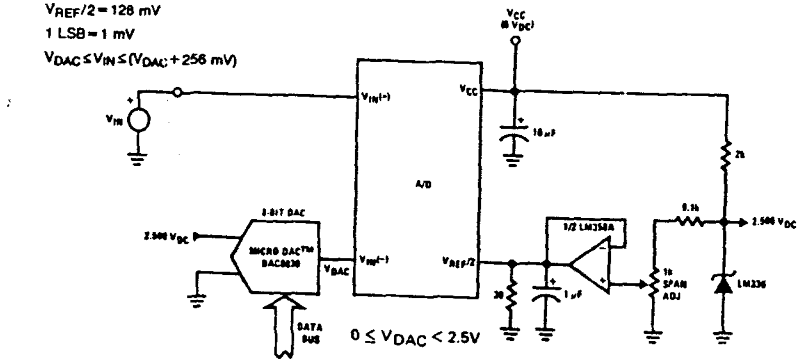
A μP Interfaced Comparator



For: $V_{IN}(+) > V_{IN}(-)$
Output = FF_{HEX}
For: $V_{IN}(+) < V_{IN}(-)$
Output = 00_{HEX}

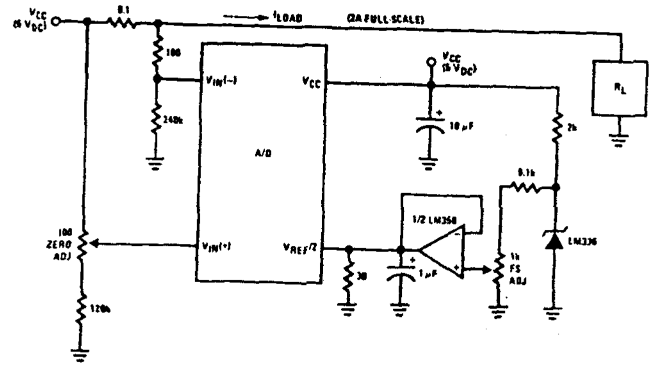
1 mV Resolution with μP Controlled Range

$V_{REF}/2 = 128$ mV
1 LSB = 1 mV
$V_{DAC} \le V_{IN} \le (V_{DAC} + 256$ mV)

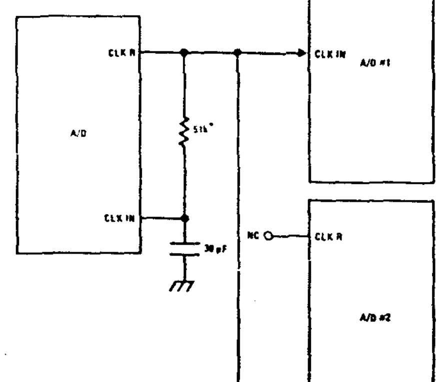


$0 \le V_{DAC} < 2.5V$

Digitizing a Current Flow

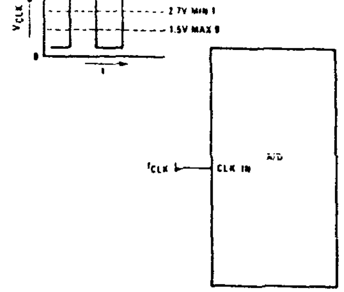


TL/H/5671-6



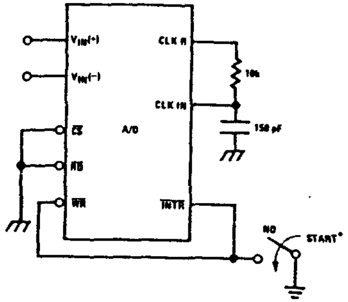
*Use a large R value to reduce loading at CLK R output.

IF MORE THAN 5 ADDITIONAL A/Ds, USE A CMOS BUFFER (NOT T^2L)



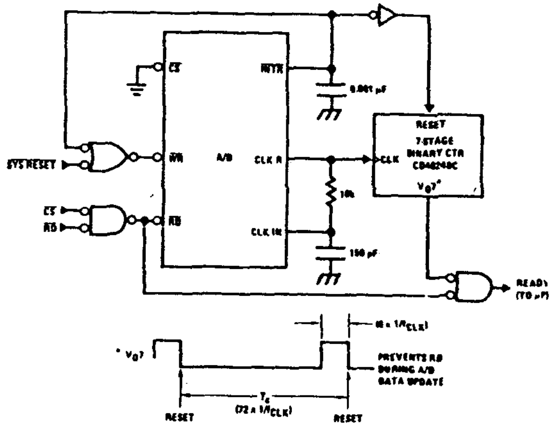
$100 \text{ kHz} \le f_{CLK} \le 1460 \text{ kHz}$

Self-Clocking in Free-Running Mode

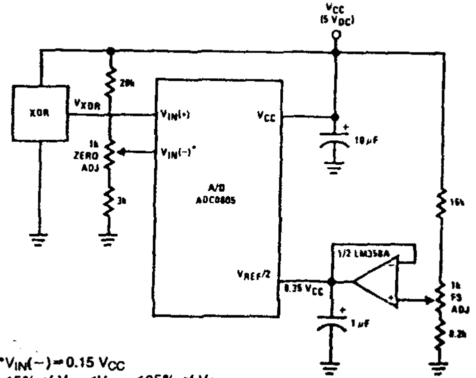


*After power up, a momentary grounding of the $\overline{WR}$ input is needed to guarantee operation.

μP Interface for Free-Running A/D

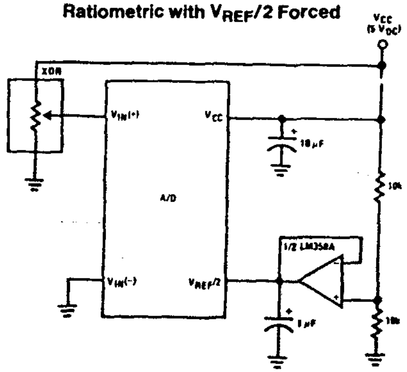


Operating with "Automotive" Ratiometric Transducers



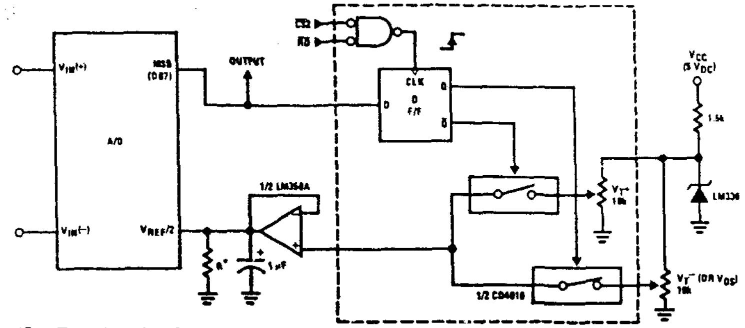
$^*V_{IN}(-) = 0.15 V_{CC}$
15% of $V_{CC} \le V_{XDR} \le 85\%$ of V_{CC}

Ratiometric with $V_{REF}/2$ Forced



TL/H/5671-7

µP Compatible Differential-Input Comparator with Pre-Set V_{OS} (with or without Hysteresis)

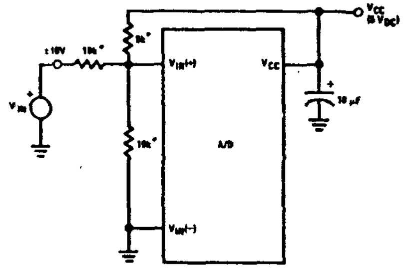


*See *Figure 5* to select R value

DB7 = "1" for $V_{IN}(+) > V_{IN}(-) + (V_{REF}/2)$

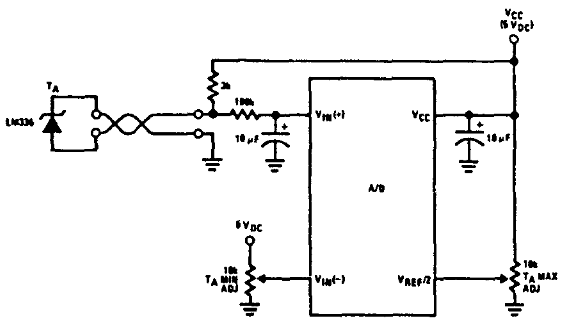
Omit circuitry within the dotted area if hysteresis is not needed

Handling ±10V Analog Inputs

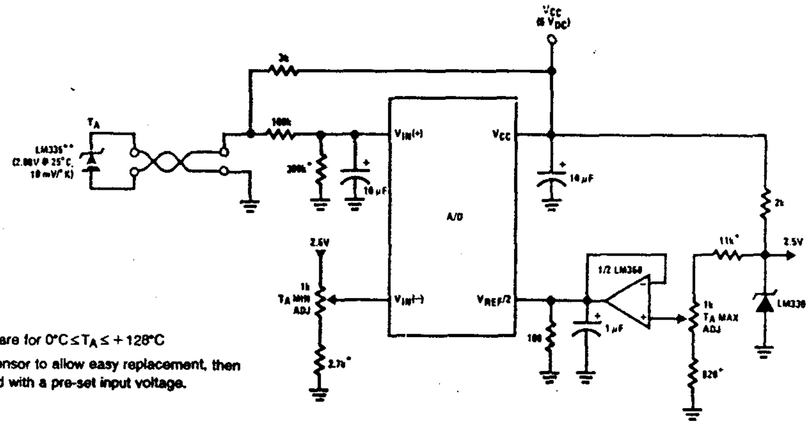


*Beckman Instruments #694-3-R10K resistor array

Low-Cost, µP Interfaced, Temperature-to-Digital Converter



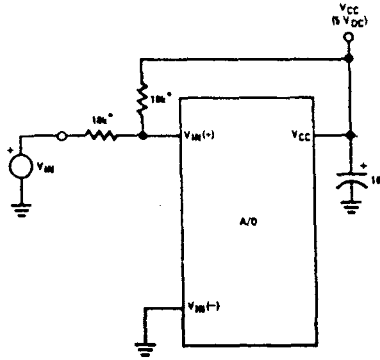
µP Interfaced Temperature-to-Digital Converter



*Circuit values shown are for 0°C ≤ T_A ≤ +128°C

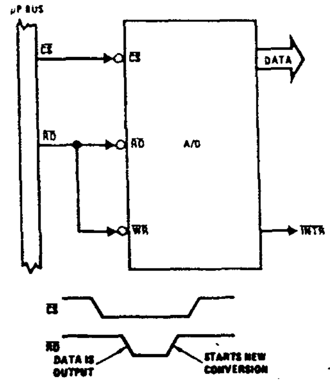
**Can calibrate each sensor to allow easy replacement, then A/D can be calibrated with a pre-set input voltage.

TL/H/5671-8



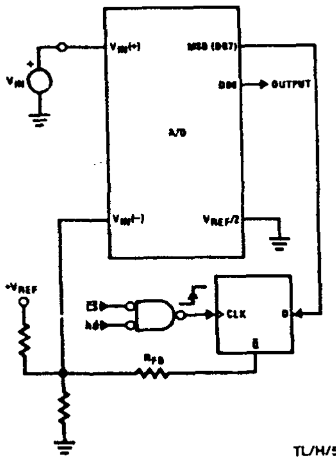
TL/H/5671-33

*Beckman Instruments #694-3-R10K resistor array



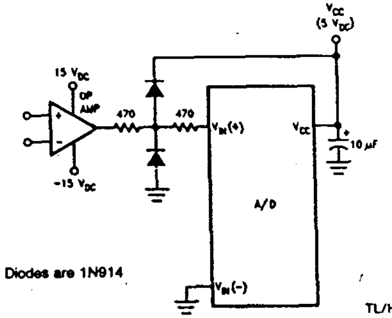
TL/H/5671-34

µP Interfaced Comparator with Hysteresis



TL/H/5671-35

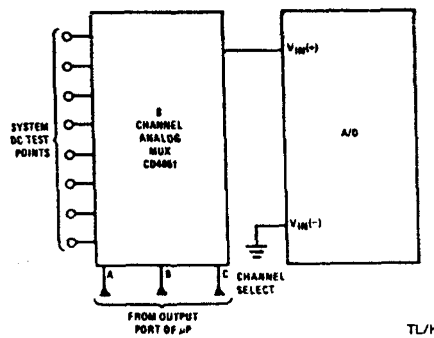
Protecting the Input



Diodes are 1N914

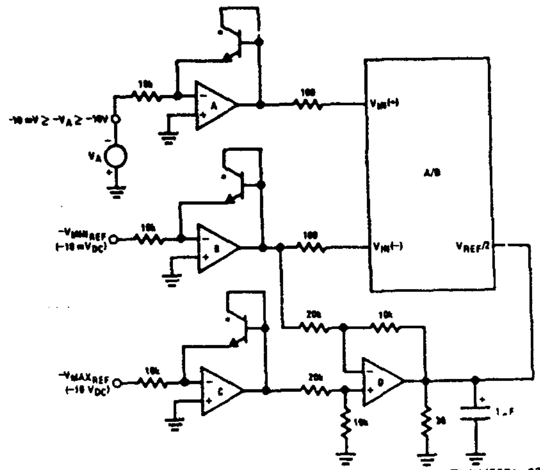
TL/H/5671-9

Analog Self-Test for a System



TL/H/5671-36

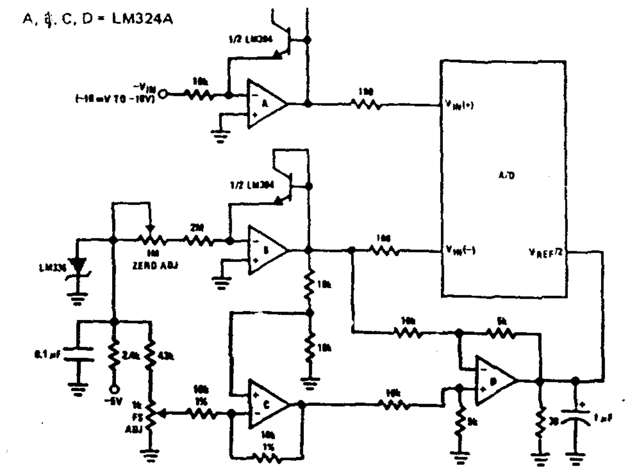
A Low-Cost, 3-Decade Logarithmic Converter



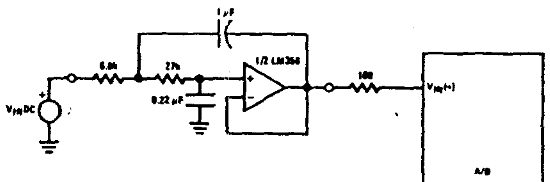
*LM389 transistors

A, B, C, D = LM324A quad op amp

TL/H/5671-37

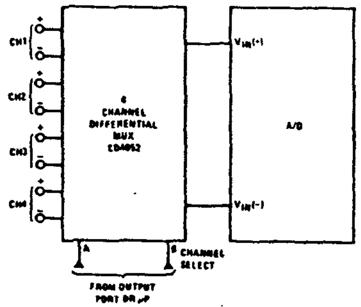


Noise Filtering the Analog Input

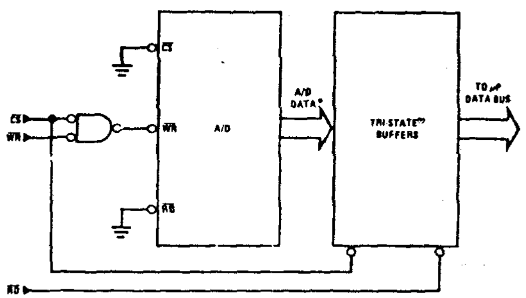


$f_C = 20$ Hz
Uses Chebyshev implementation for steeper roll-off unity-gain, 2nd order, low-pass filter
Adding a separate filter for each channel increases system response time if an analog multiplexer is used

Multiplexing Differential Inputs

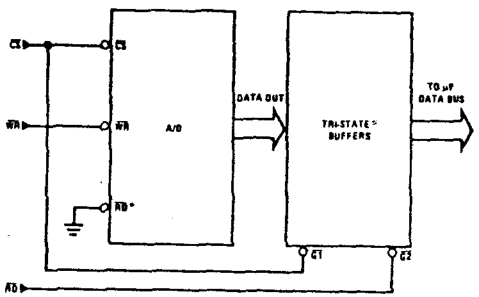


Output Buffers with A/D Data Enabled



*A/D output data is updated 1 CLK period prior to assertion of $\overline{INTR}$

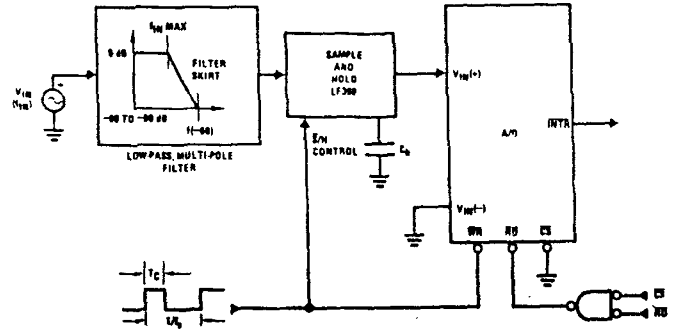
Increasing Bus Drive and/or Reducing Time on Bus



TL/H/5671-10

*Allows output data to set-up at falling edge of $\overline{CS}$

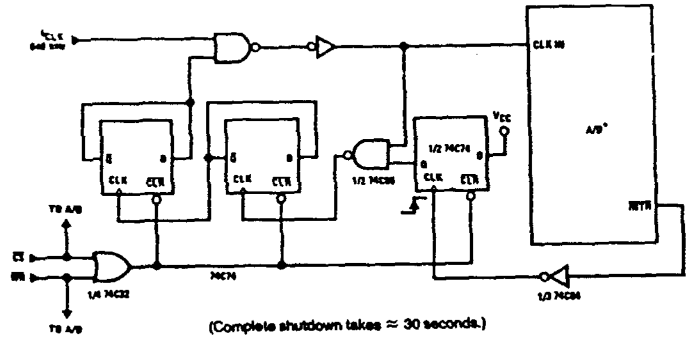
Sampling an AC Input Signal



Note 1: Oversample whenever possible [keep fs > 2f(−60)] to eliminate input frequency folding (aliasing) and to allow for the skirt response of the filter.

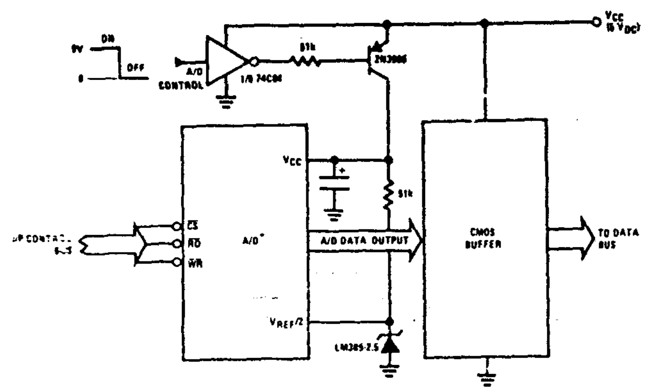
Note 2: Consider the amplitude errors which are introduced within the passband of the filter.

70% Power Savings by Clock Gating



(Complete shutdown takes ≈ 30 seconds.)

Power Savings by A/D and V_{REF} Shutdown



TL/H/5671-11

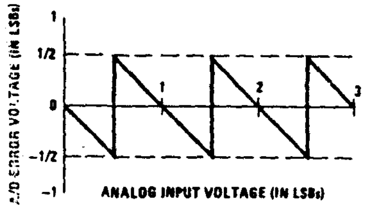
*Use ADC0801, 02, 03 or 05 for lowest power consumption.
Note: Logic inputs can be driven to V_{CC} with A/D supply at zero volts.
Buffer prevents data bus from overdriving output of A/D when in shutdown mode.

... which are obtained from the column: "Input Voltage Value with a 10.240 V_{REF}" of both the MS and LS groups, the value of the digital display can be determined. For example, for an output LED display of "1011 0110" or "B6" (in hex) the voltage values from the table are 7.04 + 0.24 or

... an A/D, to obtain an output digital code change. The effects of this quantization error have to be accounted for in the interpretation of the test results. A plot of this natural error source is shown in *Figure 5* where, for clarity, both the analog input voltage and the error voltage are normalized to LSBs.

TABLE I. DECODING THE DIGITAL OUTPUT LEDs

HEX	BINARY				FRACTIONAL BINARY VALUE FOR		INPUT VOLTAGE VALUE WITH 10.24 V_{REF}	
					MS GROUP	LS GROUP	MS GROUP	LS GROUP
F	1	1	1	1	15/16	15/256	9.600	0.600
E	1	1	1	0	7/8	7/128	8.960	0.560
D	1	1	0	1	13/16	13/256	8.320	0.520
C	1	1	0	0	3/4	3/64	7.680	0.480
B	1	0	1	1	11/16	11/256	7.040	0.440
A	1	0	1	0	5/8	5/128	6.400	0.400
9	1	0	0	1	9/16	9/256	5.760	0.360
8	1	0	0	0	1/2	1/32	5.120	0.320
7	0	1	1	1	7/16	7/256	4.480	0.280
6	0	1	1	0	3/8	3/128	3.840	0.240
5	0	1	0	1	5/16	5/256	3.200	0.200
4	0	1	0	0	1/4	1/64	2.560	0.160
3	0	0	1	1	3/16	3/256	1.920	0.120
2	0	0	1	0	1/8	1/128	1.280	0.080
1	0	0	0	1	1/16	1/256	0.640	0.040
0	0	0	0	0			0	0



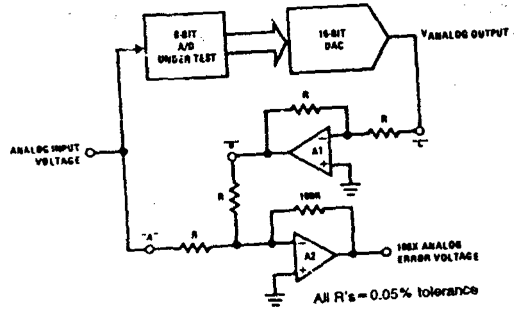
TL/H/5670-8

FIGURE 5. Error Plot of a Perfect A/D Showing Effects of Quantization Error

... nary counting sequence ...

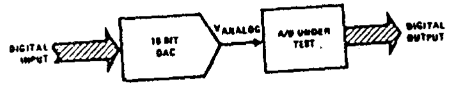
The techniques described so far are suitable for an engineering evaluation or a quick check on performance. For a higher speed test system, or to obtain plotted data, a digital-to-analog converter is needed for the test set-up. An accurate 10-bit DAC can serve as the precision voltage source for the A/D. Errors of the A/D under test can be provided as either analog voltages or differences in two digital words.

A basic A/D tester which uses a DAC and provides the error as an analog output voltage is shown in *Figure 6*. The 2 op amps can be eliminated if a lab DVM with a numerical subtraction feature is available to directly readout the difference voltage, "A–C".



TL/H/5670-16

FIGURE 6. A/D Tester with Analog Error Output



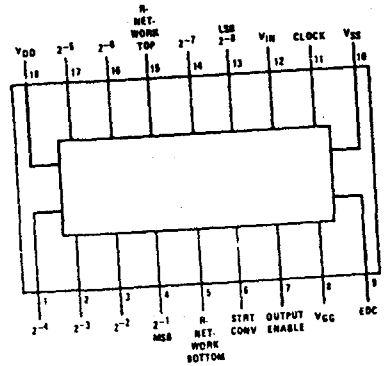
TL/H/5670-17

FIGURE 7. Basic "Digital" A/D Tester

... output ... incremented. This provides ¼ LSB steps for the 8-bit A/D under test. If the results of this test are automatically plotted with the analog input on the X axis and the error (in LSB's) as the Y axis, a useful transfer function of the A/D under test results. For acceptance testing, the plot is not necessary and the testing speed can be increased by establishing internal limits on the allowed error for each code.

Connection Diagram

Dual-In-Line Package



TL/H/5670-9

Top View

Order Number ADC0800PD
or ADC0800PCD
See NS Package Number D18A

3

ADC0801, ADC0802, ADC0803, ADC0804, ADC0805 8-Bit μP Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE® output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

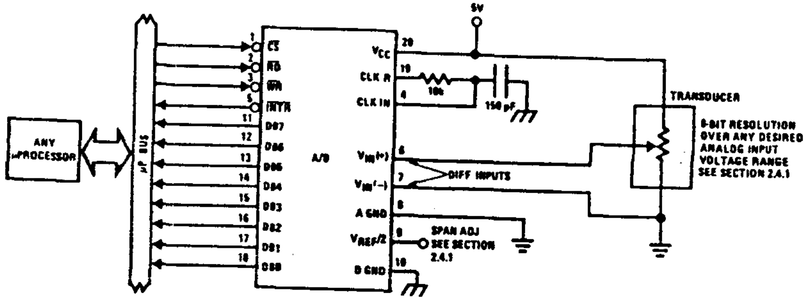
Features

- Compatible with 8080 μP derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5 V_{DC}, 2.5 V_{DC}, or analog span adjusted voltage reference

Key Specifications

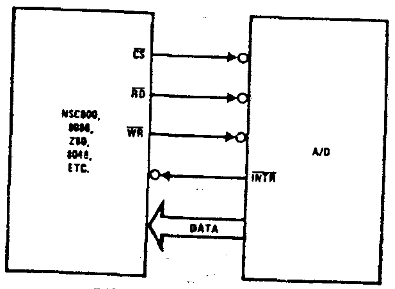
- Resolution: 8 bits
- Total error: ±¼ LSB, ±½ LSB and ±1 LSB
- Conversion time: 100 μs

Typical Applications



TL/H/5671–1

8080 Interface



TL/H/5671–31

Error Specification (Includes Full-Scale, Zero Error, and Non-Linearity)			
Part Number	Full-Scale Adjusted	$V_{REF}/2 = 2.500\ V_{DC}$ (No Adjustments)	$V_{REF}/2$ = No Connection (No Adjustments)
ADC0801	±¼ LSB		
ADC0802		±½ LSB	
ADC0803	±½ LSB		
ADC0804		±1 LSB	
ADC0805			±1 LSB

contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage	
Logic Control Inputs	−0.3V to +18V
At Other Input and Outputs	−0.3V to (V_{CC}+0.3V)
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
Package Dissipation at T_A=25°C	875 mW
ESD Susceptibility (Note 10)	800V

Operating Ratings (Notes 1 & 2)

Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
ADC0801/02LJ	$-55°C \le T_A \le +125°C$
ADC0801/02/03/04LCJ	$-40°C \le T_A \le +85°C$
ADC0801/02/03/05LCN	$-40°C \le T_A \le +85°C$
ADC0804LCN	$0°C \le T_A \le +70°C$
ADC0802/03/04LCV	$0°C \le T_A \le +70°C$
ADC0802/03/04LCWM	$0°C \le T_A \le +70°C$
Range of V_{CC}	4.5 V_{DC} to 6.3 V_{DC}

Electrical Characteristics

The following specifications apply for V_{CC}=5 V_{DC}, $T_{MIN} \le T_A \le T_{MAX}$ and f_{CLK}=640 kHz unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
ADC0801: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			±¼	LSB
ADC0802: Total Unadjusted Error (Note 8)	$V_{REF}/2$=2.500 V_{DC}			±½	LSB
ADC0803: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			±½	LSB
ADC0804: Total Unadjusted Error (Note 8)	$V_{REF}/2$=2.500 V_{DC}			±1	LSB
ADC0805: Total Unadjusted Error (Note 8)	$V_{REF}/2$-No Connection			±1	LSB
$V_{REF}/2$ Input Resistance (Pin 9)	ADC0801/02/03/05 ADC0804 (Note 9)	2.5 0.75	8.0 1.1		kΩ kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(−)	Gnd−0.05		V_{CC}+0.05	V_{DC}
DC Common-Mode Error	Over Analog Input Voltage Range		±1/16	±⅛	LSB
Power Supply Sensitivity	V_{CC}=5 V_{DC} ±10% Over Allowed $V_{IN}(+)$ and $V_{IN}(-)$ Voltage Range (Note 4)		±1/16	±⅛	LSB

AC Electrical Characteristics

The following specifications apply for V_{CC}=5 V_{DC} and T_A=25°C unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T_C	Conversion Time	f_{CLK}=640 kHz (Note 6)	103		114	μs
T_C	Conversion Time	(Note 5, 6)	66		73	$1/f_{CLK}$
f_{CLK}	Clock Frequency Clock Duty Cycle	V_{CC}=5V, (Note 5) (Note 5)	100 40	640	1460 60	kHz %
CR	Conversion Rate in Free-Running Mode	$\overline{INTR}$ tied to $\overline{WR}$ with $\overline{CS}$=0 V_{DC}, f_{CLK}=640 kHz	8770		9708	conv/s
$t_{W(\overline{WR})L}$	Width of $\overline{WR}$ Input (Start Pulse Width)	$\overline{CS}$=0 V_{DC} (Note 7)	100			ns
t_{ACC}	Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid)	C_L=100 pF		135	200	ns
t_{1H}, t_{0H}	TRI-STATE Control (Delay from Rising Edge of $\overline{RD}$ to Hi-Z State)	C_L=10 pF, R_L=10k (See TRI-STATE Test Circuits)		125	200	ns
t_{WI}, t_{RI}	Delay from Falling Edge of $\overline{WR}$ or $\overline{RD}$ to Reset of $\overline{INTR}$			300	450	ns
C_{IN}	Input Capacitance of Logic Control Inputs			5	7.5	pF
C_{OUT}	TRI-STATE Output Capacitance (Data Buffers)			5	7.5	pF
CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
$V_{IN}(1)$	Logical "1" Input Voltage (Except Pin 4 CLK IN)	V_{CC}=5.25 V_{DC}	2.0		15	V_{DC}

Symbol	Parameter	Conditions	Min	Typ	Max	Units
CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
V_{IN} (0)	Logical "0" Input Voltage (Except Pin 4 CLK IN)	V_{CC} = 4.75 V_{DC}			0.8	V_{DC}
I_{IN} (1)	Logical "1" Input Current (All Inputs)	V_{IN} = 5 V_{DC}		0.005	1	μA_{DC}
I_{IN} (0)	Logical "0" Input Current (All Inputs)	V_{IN} = 0 V_{DC}	−1	−0.005		μA_{DC}
CLOCK IN AND CLOCK R						
V_T+	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	V_{DC}
V_T-	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	V_{DC}
V_H	CLK IN (Pin 4) Hysteresis $(V_T+)-(V_T-)$		0.6	1.3	2.0	V_{DC}
V_{OUT} (0)	Logical "0" CLK R Output Voltage	I_O = 360 μA V_{CC} = 4.75 V_{DC}			0.4	V_{DC}
V_{OUT} (1)	Logical "1" CLK R Output Voltage	I_O = −360 μA V_{CC} = 4.75 V_{DC}	2.4			V_{DC}
DATA OUTPUTS AND $\overline{INTR}$						
V_{OUT} (0)	Logical "0" Output Voltage Data Outputs $\overline{INTR}$ Output	 I_{OUT} = 1.6 mA, V_{CC} = 4.75 V_{DC} I_{OUT} = 1.0 mA, V_{CC} = 4.75 V_{DC}			 0.4 0.4	 V_{DC} V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	I_O = −360 μA, V_{CC} = 4.75 V_{DC}	2.4			V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	I_O = −10 μA, V_{CC} = 4.75 V_{DC}	4.5			V_{DC}
I_{OUT}	TRI-STATE Disabled Output Leakage (All Data Buffers)	V_{OUT} = 0 V_{DC} V_{OUT} = 5 V_{DC}	−3		 3	μA_{DC} μA_{DC}
I_{SOURCE}		V_{OUT} Short to Gnd, T_A = 25°C	4.5	6		mA_{DC}
I_{SINK}		V_{OUT} Short to V_{CC}, T_A = 25°C	9.0	16		mA_{DC}
POWER SUPPLY						
I_{CC}	Supply Current (Includes Ladder Current) ADC0801/02/03/04LCJ/05 ADC0804LCN/LCV/LCWM	f_{CLK} = 640 kHz, $V_{REF}/2$ = NC, T_A = 25°C and $\overline{CS}$ = 5V		 1.1 1.9	 1.8 2.5	 mA mA

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. The separate A Gnd point should always be wired to the D Gnd.

Note 3: A zener diode exists, internally, from V_{CC} to Gnd and has a typical breakdown voltage of 7 V_{DC}.

Note 4: For $V_{IN}(-) \geq V_{IN}(+)$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct–especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Accuracy is guaranteed at f_{CLK} = 640 kHz. At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be extended so long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

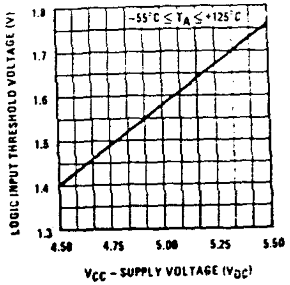
Note 6: With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process. The start request is internally latched, see *Figure 2* and section 2.0.

Note 7: The $\overline{CS}$ input is assumed to bracket the $\overline{WR}$ strobe input and therefore timing is dependent on the $\overline{WR}$ pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the $\overline{WR}$ pulse (see timing diagrams).

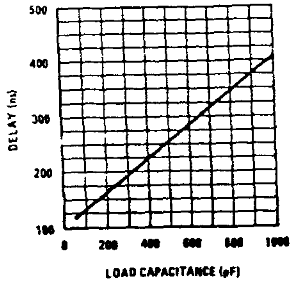
Note 8: None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and *Figure 5*.

Note 9: The $V_{REF}/2$ pin is the center point of a two resistor divider connected from V_{CC} to ground. Each resistor is 2.2k, except for the ADC0804LCJ where each resistor is 16k. Total ladder input resistance is the sum of the two equal resistors.

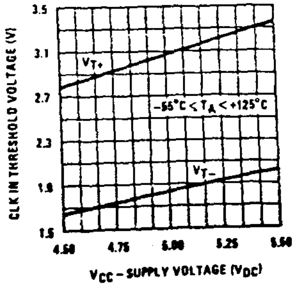
Logic Input Threshold Voltage vs. Supply Voltage



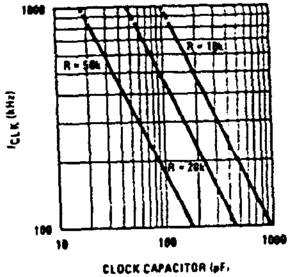
Delay From Falling Edge of $\overline{RD}$ to Output Data Valid vs. Load Capacitance



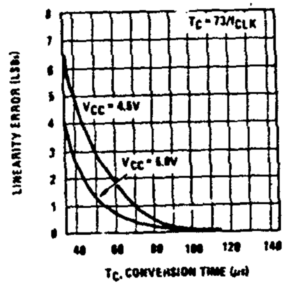
CLK IN Schmitt Trip Levels vs. Supply Voltage



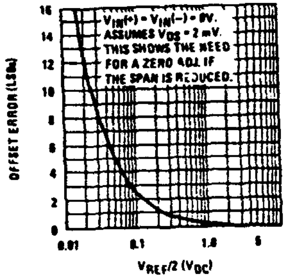
f_{CLK} vs. Clock Capacitor



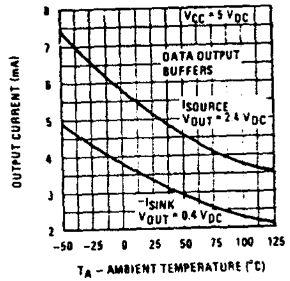
Full-Scale Error vs Conversion Time



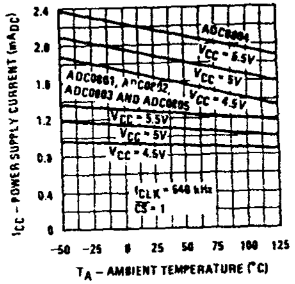
Effect of Unadjusted Offset Error vs. $V_{REF}/2$ Voltage



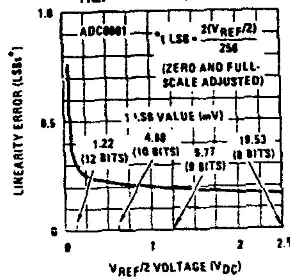
Output Current vs Temperature



Power Supply Current vs Temperature (Note 9)

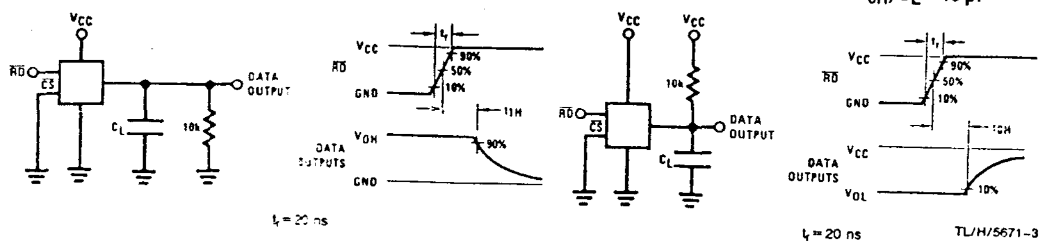


Linearity Error at Low $V_{REF}/2$ Voltages

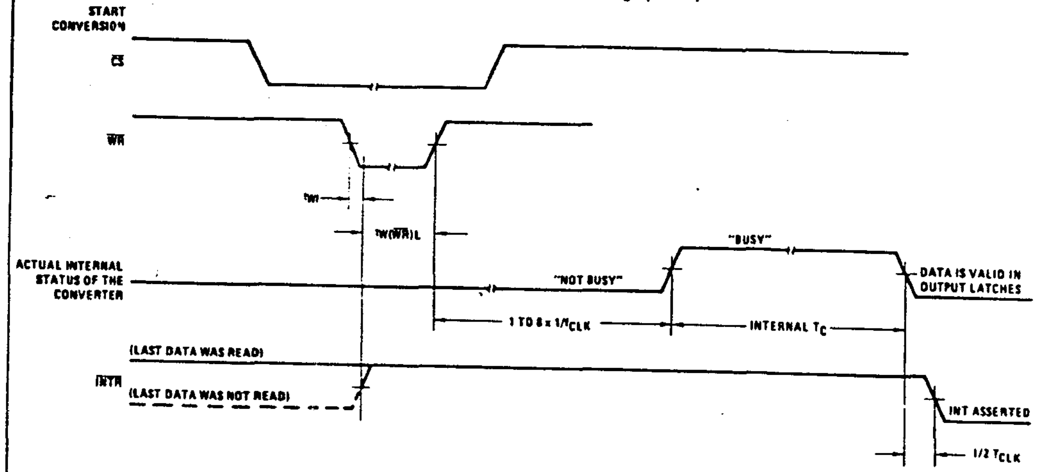


TL/H/5671-2

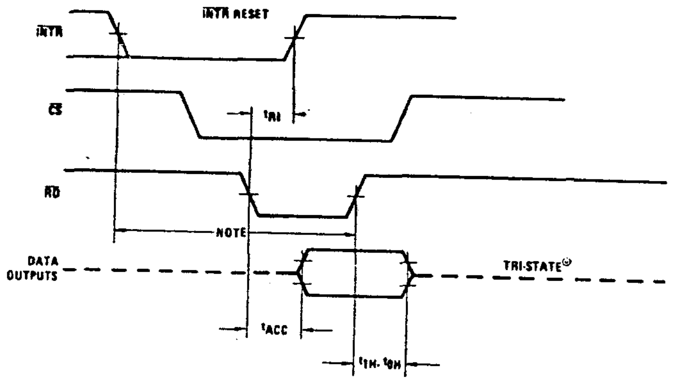
t_{1H} | t_{1H}, C_L = 10 pF | t_{0H} | t_{0H}, C_L = 10 pF



Timing Diagrams (All timing is measured from the 50% voltage points)

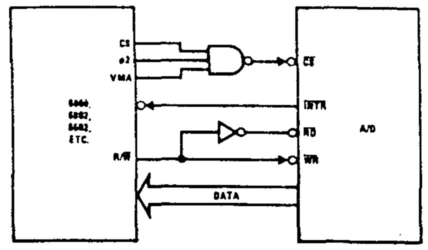


Output Enable and Reset $\overline{INTR}$

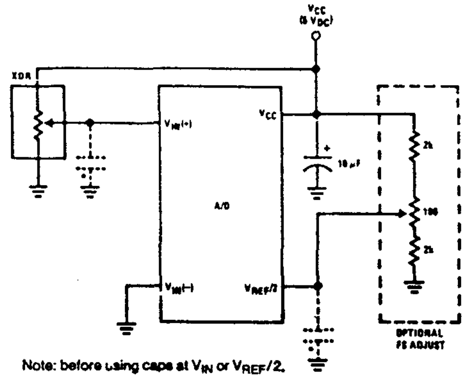


Note: Read strobe must occur 8 clock periods ($8/f_{CLK}$) after assertion of interrupt to guarantee reset of $\overline{INTR}$. TL/H/5671–4

6800 Interface

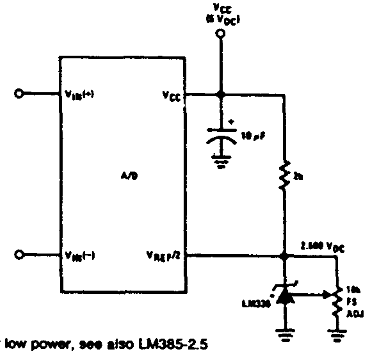


Ratiometric with Full-Scale Adjust



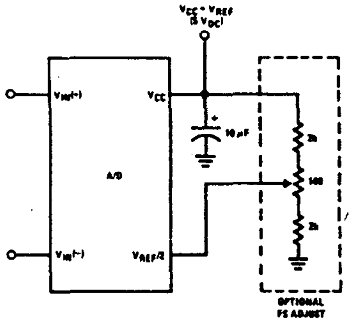
Note: before using caps at V_{IN} or $V_{REF}/2$, see section 2.3.2 Input Bypass Capacitors.

Absolute with a 2.500V Reference

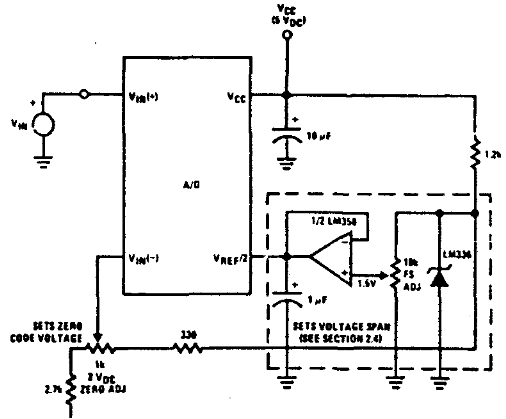


*For low power, see also LM385-2.5

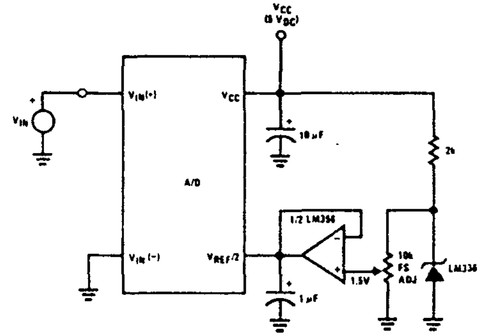
Absolute with a 5V Reference



Zero-Shift and Span Adjust: 2V ≤ V_{IN} ≤ 5V



Span Adjust: 0V ≤ V_{IN} ≤ 3V

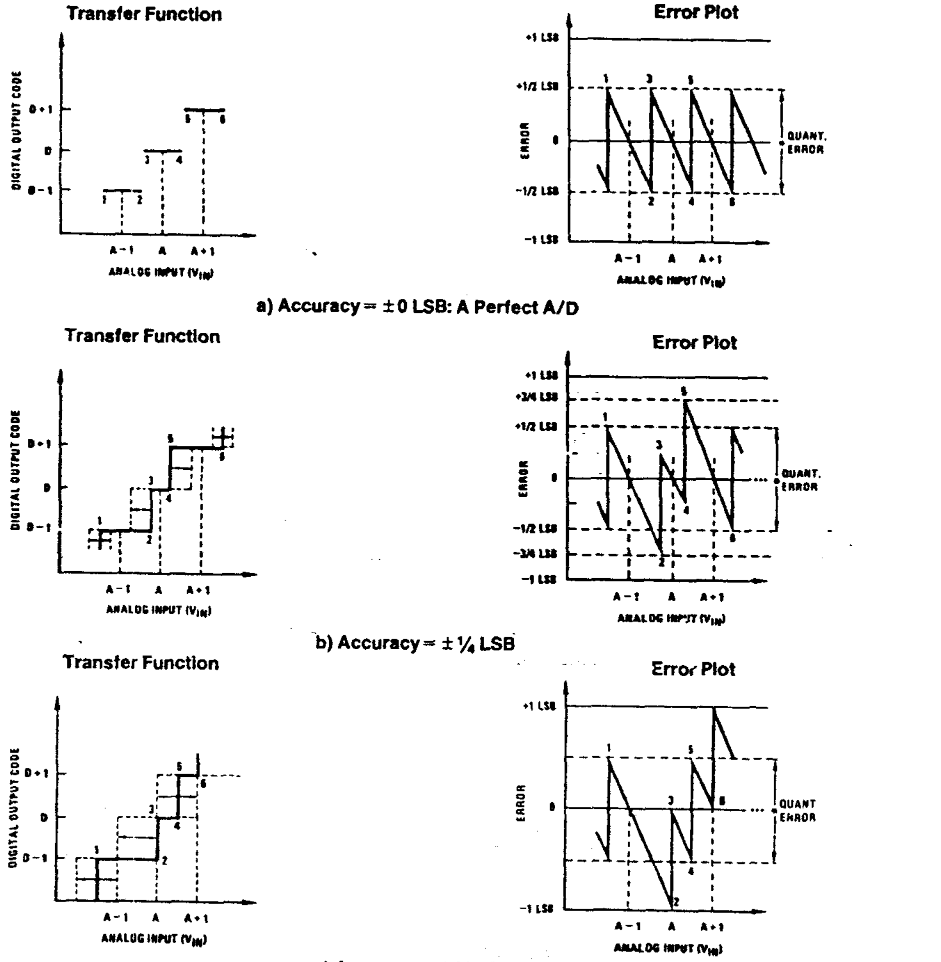


shown in *Figure 1a*. The horizontal scale is analog input voltage and the particular points labeled are in steps of 1 LSB (19.53 mV with 2.5V tied to the $V_{REF}/2$ pin). The digital output codes that correspond to these inputs are shown as D−1, D, and D+1. For the perfect A/D, not only will center-value (A−1, A, A+1, ...) analog inputs produce the correct output ditigal codes, but also each riser (the transitions between adjacent output codes) will be located ±½ LSB away from each center-value. As shown, the risers are ideal and have no width. Correct digital output codes will be provided for a range of analog input voltages that extend ±½ LSB from the ideal center-values. Each tread (the range of analog input voltage that provides the same digital output code) is therefore 1 LSB wide.

Figure 1b shows a worst case error plot for the ADC0801. All center-valued inputs are guaranteed to produce the correct output codes and the adjacent risers are guaranteed to be no closer to the center-value points than ±¼ LSB. In the code transition is indicated by the horizontal arrow and it is guaranteed to be no more than ½ LSB.

The error curve of *Figure 1c* shows a worst case error plot for the ADC0802. Here we guarantee that if we apply an analog input equal to the LSB analog voltage center-value the A/D will produce the correct digital code.

Next to each transfer function is shown the corresponding error plot. Many people may be more familiar with error plots than transfer functions. The analog input voltage to the A/D is provided by either a linear ramp or by the discrete output steps of a high resolution DAC. Notice that the error is continuously displayed and includes the quantization uncertainty of the A/D. For example the error at point 1 of *Figure 1a* is +½ LSB because the digital code appeared ½ LSB in advance of the center-value of the tread. The error plots always have a constant negative slope and the abrupt upside steps are always 1 LSB in magnitude.



a) Accuracy = ±0 LSB: A Perfect A/D

b) Accuracy = ±¼ LSB

c) Accuracy = ±½ LSB

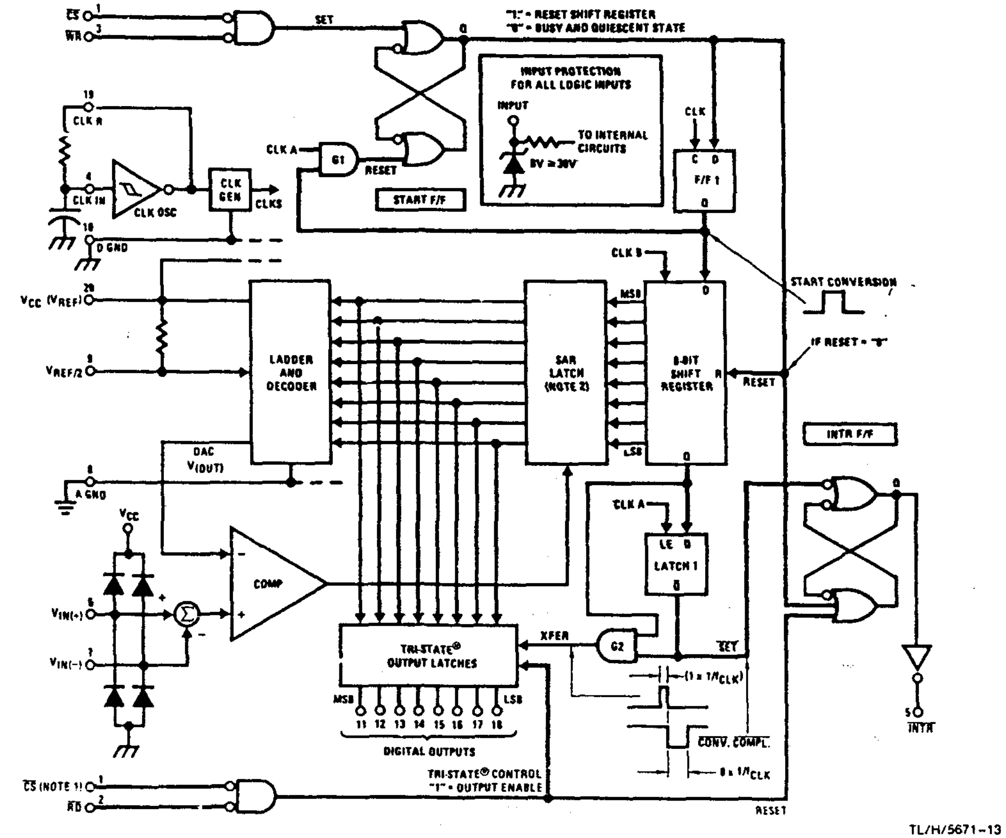
TL/H/5671–12

FIGURE 1. Clarifying the Error Specs of an A/D Converter

256R network. Analog switches are sequenced by successive approximation logic to match the analog difference input voltage [$V_{IN}(+) - V_{IN}(-)$] to a corresponding tap on the R network. The most significant bit is tested first and after 8 comparisons (64 clock cycles) a digital 8-bit binary code (1111 1111 = full-scale) is transferred to an output latch and then an interrupt is asserted ($\overline{INTR}$ makes a high-to-low transition). A conversion in process can be interrupted by issuing a second start command. The device may be operated in the free-running mode by connecting $\overline{INTR}$ to the $\overline{WR}$ input with $\overline{CS}=0$. To ensure start-up under all possible conditions, an external $\overline{WR}$ pulse is required during the first power-up cycle.

On the high-to-low transition of the $\overline{WR}$ input the internal SAR latches and the shift register stages are reset. As long as the $\overline{CS}$ input and $\overline{WR}$ input remain low, the A/D will remain in a reset state. *Conversion will start from 1 to 8 clock periods after at least one of these inputs makes a low-to-high transition.*

The converter is started by having $\overline{CS}$ and $\overline{WR}$ simultaneously low. This sets the start flip-flop (F/F) and the resulting "1" level resets the 8-bit shift register, resets the Interrupt (INTR) F/F and inputs a "1" to the D flop, F/F1, which is at the input end of the 8-bit shift register. Internal clock signals then transfer this "1" to the Q output of F/F1. The AND gate, G1, combines this "1" output with a clock signal to provide a reset signal to the start F/F. If the set signal is no longer present (either $\overline{WR}$ or $\overline{CS}$ is a "1") the start F/F is reset and the 8-bit shift register then can have the "1" clocked in, which starts the conversion process. If the set signal were to still be present, this reset pulse would have no effect (both outputs of the start F/F would momentarily be at a "1" level) and the 8-bit shift register would continue to be held in the reset mode. This logic therefore allows for wide $\overline{CS}$ and $\overline{WR}$ signals and the converter will start after at least one of these signals returns high and the internal clocks again provide a reset signal for the start F/F.

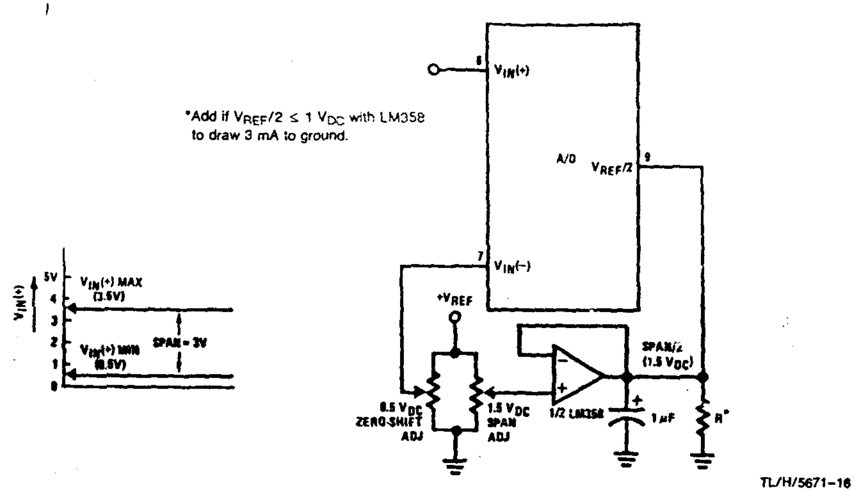


TL/H/5671–13

Note 1: $\overline{CS}$ shown twice for clarity.

Note 2: SAR = Successive Approximation Register.

FIGURE 2. Block Diagram



a) Analog Input Signal Example

b) Accommodating an Analog Input from 0.5V (Digital Out = 00_{HEX}) to 3.5V (Digital Out = FF_{HEX})

FIGURE 5. Adapting the A/D Analog Input Voltages to Match an Arbitrary Input Signal Range

2.4.2 Reference Accuracy Requirements

The converter can be operated in a ratiometric mode or an absolute mode. In ratiometric converter applications, the magnitude of the reference voltage is a factor in both the output of the source transducer and the output of the A/D converter and therefore cancels out in the final digital output code. The ADC0805 is specified particularly for use in ratiometric applications with no adjustments required. In absolute conversion applications, both the initial value and the temperature stability of the reference voltage are important factors in the accuracy of the A/D converter. For $V_{REF}/2$ voltages of 2.4 V_{DC} nominal value, initial errors of ± 10 mV_{DC} will cause conversion errors of ± 1 LSB due to the gain of 2 of the $V_{REF}/2$ input. In reduced span applications, the initial value and the stability of the $V_{REF}/2$ input voltage become even more important. For example, if the span is reduced to 2.5V, the analog input LSB voltage value is correspondingly reduced from 20 mV (5V span) to 10 mV and 1 LSB at the $V_{REF}/2$ input becomes 5 mV. As can be seen, this reduces the allowed initial tolerance of the reference voltage and requires correspondingly less absolute change with temperature variations. Note that spans smaller than 2.5V place even tighter requirements on the initial accuracy and stability of the reference source.

In general, the magnitude of the reference voltage will require an initial adjustment. Errors due to an improper value of reference voltage appear as full-scale errors in the A/D transfer function. IC voltage regulators may be used for references if the ambient temperature changes are not excessive. The LM336B 2.5V IC reference diode (from National Semiconductor) has a temperature stability of 1.8 mV typ (6 mV max) over $0°C \leq T_A \leq +70°C$. Other temperature range parts are also available.

2.5 Errors and Reference Voltage Adjustments

2.5.1 Zero Error

The zero of the A/D does not require adjustment. If the minimum analog input voltage value, $V_{IN(MIN)}$, is not ground, a zero offset can be done. The converter can be made to output 0000 0000 digital code for this minimum input voltage by biasing the A/D $V_{IN}(-)$ input at this $V_{IN(MIN)}$ value (see Applications section). This utilizes the differential mode operation of the A/D.

The zero error of the A/D converter relates to the location of the first riser of the transfer function and can be measured by grounding the $V_{IN}(-)$ input and applying a small magnitude positive voltage to the $V_{IN}(+)$ input. Zero error is the difference between the actual DC input voltage that is necessary to just cause an output digital code transition from 0000 0000 to 0000 0001 and the ideal ½ LSB value (½ LSB = 9.8 mV for $V_{REF}/2 = 2.500\ V_{DC}$).

2.5.2 Full-Scale

The full-scale adjustment can be made by applying a differential input voltage that is 1½ LSB less than the desired analog full-scale voltage range and then adjusting the magnitude of the $V_{REF}/2$ input (pin 9 or the V_{CC} supply if pin 9 is not used) for a digital output code that is just changing from 1111 1110 to 1111 1111.

2.5.3 Adjusting for an Arbitrary Analog Input Voltage Range

If the analog zero voltage of the A/D is shifted away from ground (for example, to accommodate an analog input signal that does not go to ground) this new zero reference should be properly adjusted first. A $V_{IN}(+)$ voltage that equals this desired zero reference plus ½ LSB (where the LSB is calculated for the desired analog span, 1 LSB = analog span/256) is applied to pin 6 and the zero reference voltage at pin 7 should then be adjusted to just obtain the 00_{HEX} to 01_{HEX} code transition.

The full-scale adjustment should then be made (with the proper $V_{IN}(-)$ voltage applied) by forcing a voltage to the $V_{IN}(+)$ input which is given by:

$$V_{IN}(+)\ \text{fs adj} = V_{MAX} - 1.5\left[\frac{(V_{MAX} - V_{MIN})}{256}\right],$$

where:

V_{MAX} = The high end of the analog input range

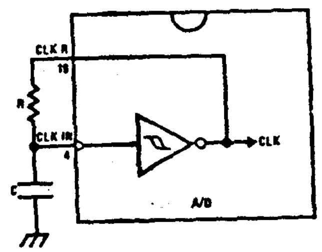
and

V_{MIN} = the low end (the offset zero) of the analog range. (Both are ground referenced.)

The $V_{REF}/2$ (or V_{CC}) voltage is then adjusted to provide a code change from FE_{HEX} to FF_{HEX}. This completes the adjustment procedure.

2.6 Clocking Option

The clock for the A/D can be derived from the CPU clock or an external RC can be added to provide self-clocking. The CLK IN (pin 4) makes use of a Schmitt trigger as shown in *Figure 6*.



$$f_{CLK} \approx \frac{1}{1.1\ RC}$$

$R \approx 10\ k\Omega$

TL/H/5671–17

FIGURE 6. Self-Clocking the A/D

Heavy capacitive or DC loading of the clock R pin should be avoided as this will disturb normal converter operation. Loads less than 50 pF, such as driving up to 7 A/D converter clock inputs from a single clock R pin of 1 converter, are allowed. For larger clock line loading, a CMOS or low power TTL buffer or PNP input logic should be used to minimize the loading on the clock R pin (do not use a standard TTL buffer).

2.7 Restart During a Conversion

If the A/D is restarted ($\overline{CS}$ and $\overline{WR}$ go low and return high) during a conversion, the converter is reset and a new conversion is started. The output data latch is not updated if the conversion in process is not allowed to finish, therefore the data of the previous conversion remains in this latch. The $\overline{INTR}$ output simply remains at the "1" level.

2.8 Continuous Conversions

For operation in the free-running mode an initializing pulse should be used, following power-up, to ensure circuit operation. In this application, the $\overline{CS}$ input is grounded and the $\overline{WR}$ input is tied to the $\overline{INTR}$ output. This $\overline{WR}$ and $\overline{INTR}$ node should be momentarily forced to logic low following a power-up cycle to guarantee operation.

2.9 Driving the Data Bus

This MOS A/D, like MOS microprocessors and memories, will require a bus driver when the total capacitance of the data bus gets large. Other circuitry, which is tied to the data bus, will add to the total capacitive loading, even in TRI-STATE (high impedance mode). Backplane bussing also greatly adds to the stray capacitance of the data bus.

There are some alternatives available to the designer to handle this problem. Basically, the capacitive loading of the data bus slows down the response time, even though DC specifications are still met. For systems operating with a relatively slow CPU clock frequency, more time is available in which to establish proper logic levels on the bus and therefore higher capacitive loads can be driven (see typical characteristics curves).

At higher CPU clock frequencies time can be extended for I/O reads (and/or writes) by inserting wait states (8080) or using clock extending circuits (6800).

Finally, if time is short and capacitive loading is high, external bus drivers must be used. These can be TRI-STATE buffers (low power Schottky such as the DM74LS240 series is recommended) or special higher drive current products which are designed as bus drivers. High current bipolar bus drivers with PNP inputs are recommended.

2.10 Power Supplies

Noise spikes on the V_{CC} supply line can cause conversion errors as the comparator will respond to this noise. A low inductance tantalum filter capacitor should be used close to the converter V_{CC} pin and values of 1 µF or greater are recommended. If an unregulated voltage is available in the system, a separate LM340LAZ-5.0, TO-92, 5V voltage regulator for the converter (and other analog circuitry) will greatly reduce digital noise on the V_{CC} supply.

2.11 Wiring and Hook-Up Precautions

Standard digital wire wrap sockets are not satisfactory for breadboarding this A/D converter. Sockets on PC boards can be used and all logic signal wires and leads should be grouped and kept as far away as possible from the analog signal leads. Exposed leads to the analog inputs can cause undesired digital noise and hum pickup, therefore shielded leads may be necessary in many applications.

ground points should be used. The power supply bypass capacitor and the self-clocking capacitor (if used) should both be returned to digital ground. Any $V_{REF}/2$ bypass capacitors, analog input filter capacitors, or input signal shielding should be returned to the analog ground point. A test for proper grounding is to measure the zero error of the A/D converter. Zero errors in excess of ¼ LSB can usually be traced to improper board layout and wiring (see section 2.5.1 for measuring the zero error).

3.0 TESTING THE A/D CONVERTER

There are many degrees of complexity associated with testing an A/D converter. One of the simplest tests is to apply a known analog input voltage to the converter and use LEDs to display the resulting digital output code as shown in *Figure 7*.

For ease of testing, the $V_{REF}/2$ (pin 9) should be supplied with 2.560 V_{DC} and a V_{CC} supply voltage of 5.12 V_{DC} should be used. This provides an LSB value of 20 mV.

If a full-scale adjustment is to be made, an analog input voltage of 5.090 V_{DC} (5.120–1½ LSB) should be applied to the $V_{IN}(+)$ pin with the $V_{IN}(-)$ pin grounded. The value of the $V_{REF}/2$ input voltage should then be adjusted until the digital output code is just changing from 1111 1110 to 1111 1111. This value of $V_{REF}/2$ should then be used for all the tests.

The digital output LED display can be decoded by dividing the 8 bits into 2 hex characters, the 4 most significant (MS) and the 4 least significant (LS). Table I shows the fractional binary equivalent of these two 4-bit groups. By adding the voltages obtained from the "VMS" and "VLS" columns in Table I, the nominal value of the digital display (when $V_{REF}/2$ = 2.560V) can be determined. For example, for an output LED display of 1011 0110 or B6 (in hex), the voltage values from the table are 3.520 + 0.120 or 3.640 V_{DC}. These voltage values represent the center-values of a perfect A/D converter. The effects of quantization error have to be accounted for in the interpretation of the test results.

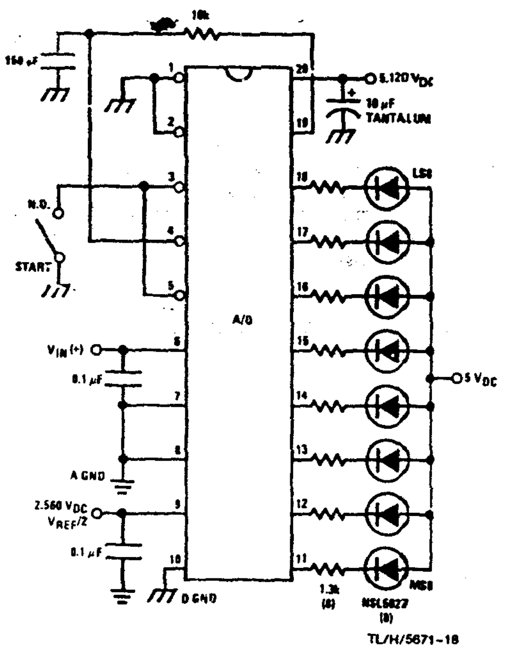


FIGURE 7. Basic A/D Tester

For a higher speed test system, or to obtain plotted data, a digital-to-analog converter is needed for the test set-up. An accurate 10-bit DAC can serve as the precision voltage source for the A/D. Errors of the A/D under test can be expressed as either analog voltages or differences in 2 digital words.

A basic A/D tester that uses a DAC and provides the error as an analog output voltage is shown in *Figure 8*. The 2 op amps can be eliminated if a lab DVM with a numerical subtraction feature is available to read the difference voltage, "A–C", directly. The analog input voltage can be supplied by a low frequency ramp generator and an X-Y plotter can be used to provide analog error (Y axis) versus analog input (X axis). The construction details of a tester of this type are provided in the NSC application note AN-179, "Analog-to-Digital Converter Testing".

For operation with a microprocessor or a computer-based test system, it is more convenient to present the errors digitally. This can be done with the circuit of *Figure 9*, where the output code transitions can be detected as the 10-bit DAC is incremented. This provides ¼ LSB steps for the 8-bit A/D under test. If the results of this test are automatically plotted with the analog input on the X axis and the error (in LSB's) as the Y axis, a useful transfer function of the A/D under test results. For acceptance testing, the plot is not necessary and the testing speed can be increased by establishing internal limits on the allowed error for each code.

4.0 MICROPROCESSOR INTERFACING

To dicuss the interface with 8080A and 6800 microprocessors, a common sample subroutine structure is used. The microprocessor starts the A/D, reads and stores the results of 16 successive conversions, then returns to the user's program. The 16 data bytes are stored in 16 successive memory locations. All Data and Addresses will be given in hexadecimal form. Software and hardware details are provided separately for each type of microprocessor.

4.1 Interfacing 8080 Microprocessor Derivatives (8048, 8085)

This converter has been designed to directly interface with derivatives of the 8080 microprocessor. The A/D can be mapped into memory space (using standard memory address decoding for $\overline{CS}$ and the $\overline{MEMR}$ and $\overline{MEMW}$ strobes) or it can be controlled as an I/O device by using the $\overline{I/O\ R}$ and $\overline{I/O\ W}$ strobes and decoding the address bits A0 → A7 (or address bits A8 → A15 as they will contain the same 8-bit address information) to obtain the $\overline{CS}$ input. Using the I/O space provides 256 additional addresses and may allow a simpler 8-bit address decoder but the data can only be input to the accumulator. To make use of the additional memory reference instructions, the A/D should be mapped into memory space. An example of an A/D in I/O space is shown in *Figure 10*.

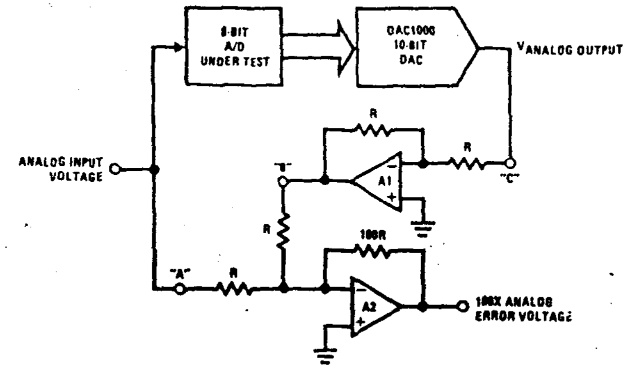
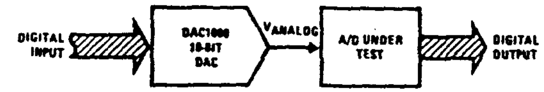


FIGURE 8. A/D Tester with Analog Error Output

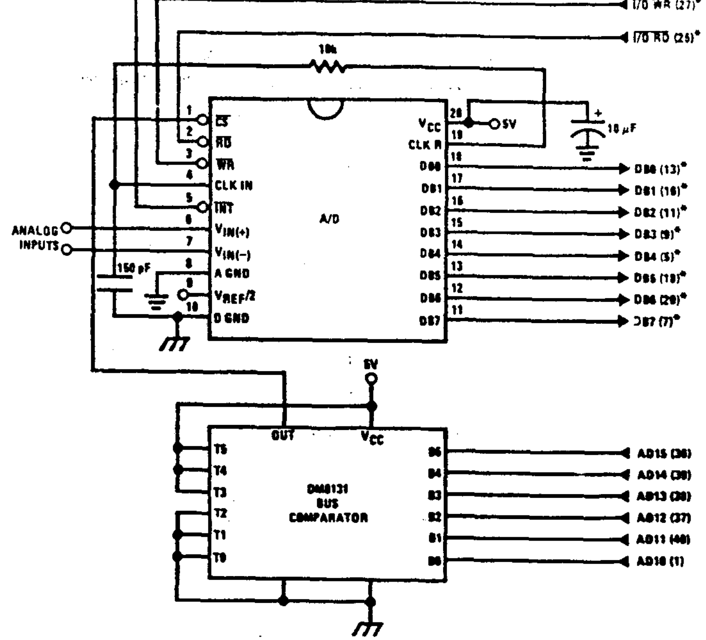


TL/H/5671-19

FIGURE 9. Basic "Digital" A/D Tester

TABLE I. DECODING THE DIGITAL OUTPUT LEDs

HEX	BINARY				FRACTIONAL BINARY VALUE FOR				OUTPUT VOLTAGE CENTER VALUES WITH $V_{REF}/2 = 2.560\ V_{DC}$	
					MS GROUP		LS GROUP		VMS GROUP*	VLS GROUP*
F	1	1	1	1		15/16		15/256	4.800	0.300
E	1	1	1	0	7/8		7/128		4.480	0.280
D	1	1	0	1		13/16		13/256	4.160	0.260
C	1	1	0	0	3/4		3/64		3.840	0.240
B	1	0	1	1		11/16		11/256	3.520	0.220
A	1	0	1	0	5/8		5/128		3.200	0.200
9	1	0	0	1		9/16		9/256	2/880	0.180
8	1	0	0	0	1/2		1/32		2/560	0.160
7	0	1	1	1		7/16		7/256	2.240	0.140
6	0	1	1	0	3/8		3/128		1.920	0.120
5	0	1	0	1		5/16		2/256	1.600	0.100
4	0	1	0	0	1/4		1/64		1/280	0.080
3	0	0	1	1		3/16		3/256	0.960	0.060
2	0	0	1	0	1/8		1/128		0.640	0.040
1	0	0	0	1		1/16		1/256	0.320	0.020
0	0	0	0	0					0	0



TL/H/5671-20

Note 1: *Pin numbers for the DP8228 system controller, others are INS8080A.

Note 2: Pin 23 of the INS8228 must be tied to +12V through a 1 kΩ resistor to generate the RST 7 instruction when an interrupt is acknowledged as required by the accompanying sample program.

FIGURE 10. ADC0801–INS8080A CPU Interface

SAMPLE PROGRAM FOR *FIGURE 10* ADC0801–INS8080A CPU INTERFACE

```
0038   C3 00 03   RST 7:       JMP  LD DATA
 •        •        •
 •        •        •
0100   21 00 02   START:       LXI H 0200H     ; HL pair will point to
                                               ; data storage locations
0103   31 00 04   RETURN:      LXI SP 0400H    ; Initialize stack pointer (Note 1)
0106   7D                      MOV A, L        ; Test # of bytes entered
0107   FE 0F                   CPI 0F H        ; If # = 16. JMP to
0109   CA 13 01                JZ CONT         ; user program
010C   D3 E0                   OUT E0 H        ; Start A/D
010E   FB                      EI              ; Enable interrupt
010F   00         LOOP:        NOP             ; Loop until end of
0110   C3 0F 01                JMP LOOP        ; conversion
0113      •       CONT:           •
 •        •        •              •
 •        •       (User program to •
 •        •       process data)   •
 •        •        •              •
 •        •        •              •
0300   DB E0      LD DATA:     IN E0 H         ; Load data into accumulator
0302   77                      MOV M, A        ; Store data
0303   23                      INX H           ; Increment storage pointer
0304   C3 03 01                JMP RETURN
```

Note 1: The stack pointer must be dimensioned because a RST 7 instruction pushes the PC onto the stack.

Note 2: All address used were arbitrarily chosen.

WR) can be directly wired to the digital control inputs of the A/D and the bus timing requirements are met to allow both starting the converter and outputting the data onto the data bus. A bus driver should be used for larger microprocessor systems where the data bus leaves the PC board and/or must drive capacitive loads larger than 100 pF.

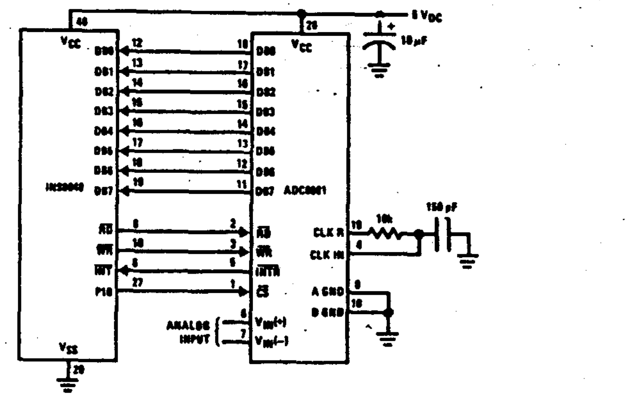
4.1.1 Sample 8080A CPU Interfacing Circuitry and Program

The following sample program and associated hardware shown in *Figure 10* may be used to input data from the converter to the INS8080A CPU chip set (comprised of the INS8080A microprocessor, the INS8228 system controller and the INS8224 clock generator). For simplicity, the A/D is controlled as an I/O device, specifically an 8-bit bi-directional port located at an arbitrarily chosen port address, E0. The TRI-STATE output capability of the A/D eliminates the need for a peripheral interface device, however address decoding is still required to generate the appropriate $\overline{CS}$ for the converter.

verter is 1 of 8 or less I/O mapped devices, no address decoding circuitry is necessary. Each of the 8 address bits (A0 to A7) can be directly used as $\overline{CS}$ inputs—one for each I/O device.

4.1.2 INS8048 Interface

The INS8048 interface technique with the ADC0801 series (see *Figure 11*) is simpler than the 8080A CPU interface. There are 24 I/O lines and three test input lines in the 8048. With these extra I/O lines available, one of the I/O lines (bit 0 of port 1) is used as the chip select signal to the A/D, thus eliminating the use of an external address decoder. Bus control signals $\overline{RD}$, $\overline{WR}$ and $\overline{INT}$ of the 8048 are tied directly to the A/D. The 16 converted data words are stored at on-chip RAM locations from 20 to 2F (Hex). The $\overline{RD}$ and $\overline{WR}$ signals are generated by reading from and writing into a dummy address, respectively. A sample interface program is shown below.



TL/H/5671-21

FIGURE 11. INS8048 Interface

SAMPLE PROGRAM FOR *FIGURE 11* INS8048 INTERFACE

```
04 10                JMP    10H        ; Program starts at addr 10
                     ORG    3H
04 50                JMP    50H        ; Interrupt jump vector
                     ORG    10H        ; Main program
99 FE                ANL    P1, #0FEH  ; Chip select
81                   MOVX   A, @R1     ; Read in the 1st data
                                       ; to reset the intr
89 01     START:     ORL    P1, #1     ; Set port pin high
B8 20                MOV    R0, #20H   ; Data address
B9 FF                MOV    R1, #0FFH  ; Dummy address
BA 10                MOV    R2, #10H   ; Counter for 16 bytes
23 FF     AGAIN:     MOV    A, #0FFH   ; Set ACC for intr loop
99 FE                ANL    P1, #0FEH  ; Send CS (bit 0 of P1)
91                   MOVX   @R1, A     ; Send WR out
05                   EN     I          ; Enable interrupt
96 21     LOOP:      JNZ    LOOP       ; Wait for interrupt
EA 1B                DJNZ   R2, AGAIN  ; If 16 bytes are read
00                   NOP               ; go to user's program
00                   NOP
                     ORG    50H
81        INDATA:    MOVX   A, @R1     ; Input data, CS still low
A0                   MOV    @R0, A     ; Store in memory
18                   INC    R0         ; Increment storage counter
89 01                ORL    P1, #1     ; Reset CS signal
27                   CLR    A          ; Clear ACC to get out of
```

4.2 Interfacing the Z-80

The Z-80 control bus is slightly different from that of the 8080. General $\overline{RD}$ and $\overline{WR}$ strobes are provided and separate memory request, $\overline{MREQ}$, and I/O request, $\overline{IORQ}$, signals are used which have to be combined with the generalized strobes to provide the equivalent 8080 signals. An advantage of operating the A/D in I/O space with the Z-80 is that the CPU will automatically insert one wait state (the $\overline{RD}$ and $\overline{WR}$ strobes are extended one clock period) to allow more time for the I/O devices to respond. Logic to map the A/D in I/O space is shown in *Figure 13*.

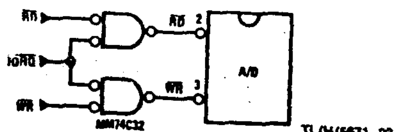


FIGURE 13. Mapping the A/D as an I/O Device for Use with the Z-80 CPU

Additional I/O advantages exist as software DMA routines are available and use can be made of the output data transfer which exists on the upper 8 address lines (A8 to A15) during I/O input instructions. For example, MUX channel selection for the A/D can be accomplished with this operating mode.

4.3 Interfacing 6800 Microprocessor Derivatives (6502, etc.)

The control bus for the 6800 microprocessor derivatives does not use the $\overline{RD}$ and $\overline{WR}$ strobe signals. Instead it employs a single R/$\overline{W}$ line and additional timing, if needed, can be derived fom the $\phi 2$ clock. All I/O devices are memory mapped in the 6800 system, and a special signal, VMA, indicates that the current address is valid. *Figure 14* shows an interface schematic where the A/D is memory mapped in the 6800 system. For simplicity, the $\overline{CS}$ decoding is shown using ½ DM8092. Note that in many 6800 systems, an al... brought out to the common bus at pin 21. This can be tied directly to the $\overline{CS}$ pin of the A/D, provided that no other devices are addressed at HX ADDR: 4XXX or 5XXX.

The following subroutine performs essentially the same function as in the case of the 8080A interface and it can be called from anywhere in the user's program.

In *Figure 15* the ADC0801 series is interfaced to the M6800 microprocessor through (the arbitrarily chosen) Port B of the MC6820 or MC6821 Peripheral Interface Adapter, (PIA). Here the $\overline{CS}$ pin of the A/D is grounded since the PIA is already memory mapped in the M6800 system and no $\overline{CS}$ decoding is necessary. Also notice that the A/D output data lines are connected to the microprocessor bus under program control through the PIA and therefore the A/D $\overline{RD}$ pin can be grounded.

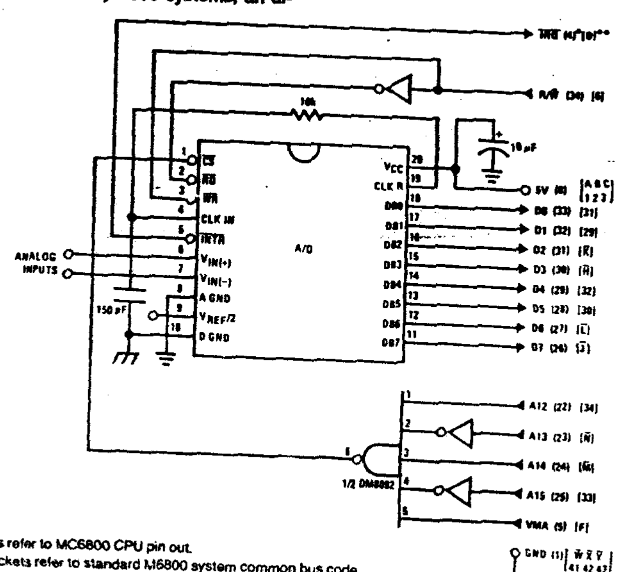
A sample interface program equivalent to the previous one is shown below *Figure 15*. The PIA Data and Control Registers of Port B are located at HEX addresses 8006 and 8007, respectively.

5.0 GENERAL APPLICATIONS

The following applications show some interesting uses for the A/D. The fact that one particular microprocessor is used is not meant to be restrictive. Each of these application circuits would have its counterpart using any microprocessor that is desired.

5.1 Multiple ADC0801 Series to MC6800 CPU Interface

To transfer analog data from several channels to a single microprocessor system, a multiple converter scheme presents several advantages over the conventional multiplexer single-converter approach. With the ADC0801 series, the differential inputs allow individual span adjustment for each channel. Furthermore, all analog input channels are sensed simultaneously, which essentially divides the microprocessor's total system servicing time by the number of channels, since all conversions occur simultaneously. This scheme is shown in *Figure 16*.



Note 1: Numbers in parentheses refer to MC6800 CPU pin out.

Note 2: Number or letters in brackets refer to standard M6800 system common bus code.

Functional Description (Continued)

SAMPLE PROGRAM FOR *FIGURE 14* ADC0801-MC6800 CPU INTERFACE

0010	DF 36	DATAIN	STX	TEMP2	; Save contents of X
0012	CE 00 2C		LDX	#$002C	; Upon $\overline{IRQ}$ low CPU
0015	FF FF F8		STX	$FFF8	; jumps to 002C
0018	B7 50 00		STAA	$5000	; Start ADC0801
001B	0E		CLI		
001C	3E	CONVRT	WAI		; Wait for interrupt
001D	DE 34		LDX	TEMP1	
001F	8C 02 0F		CPX	#$020F	; Is final data stored?
0022	27 14		BEQ	ENDP	
0024	B7 50 00		STAA	$5000	; Restarts ADC0801
0027	08		INX		
0028	DF 34		STX	TEMP1	
002A	20 F0		BRA	CONVRT	
002C	DE 34	INTRPT	LDX	TEMP1	
002E	B6 50 00		LDAA	$5000	; Read data
0031	A7 00		STAA	X	; Store it at X
0033	3B		RTI		
0034	02 00	TEMP1	FDB	$0200	; Starting address for ; data storage
0036	00 00	TEMP2	FDB	$0000	
0038	CE 02 00	ENDP	LDX	#$0200	; Reinitialize TEMP1
003B	DF 34		STX	TEMP1	
003D	DE 36		LDX	TEMP2	
003F	39		RTS		; Return from subroutine ; To user's program

Note 1: In order for the microprocessor to service subroutines and interrupts, the stack pointer must be dimensioned in the user's program.

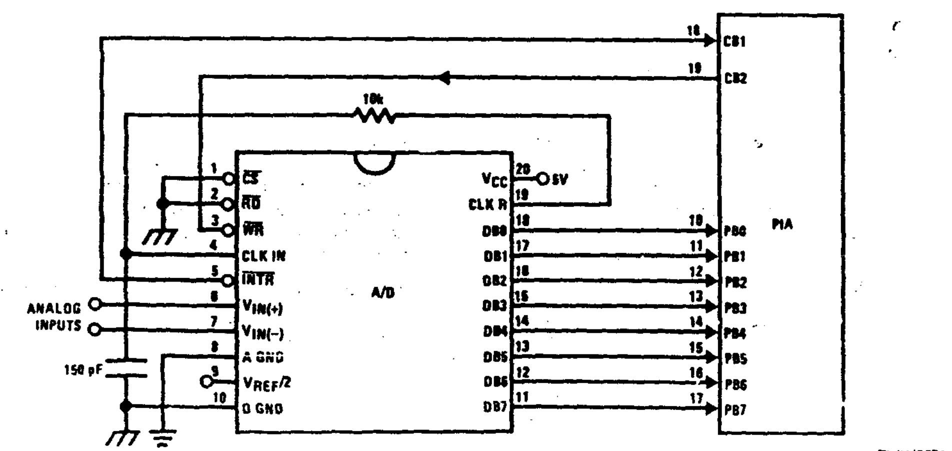


FIGURE 15. ADC0801–MC6820 PIA Interface

Functional Description (Continued)

SAMPLE PROGRAM FOR *FIGURE 15* ADC0801–MC6820 PIA INTERFACE

```
0010  CE 00 38   DATAIN  LDX   #$0038   ; Upon IRQ low CPU
0013  FF FF F8           STX   $FFF8    ; jumps to 0038
0016  B6 80 06           LDAA  PIAORB   ; Clear possible IRQ flags
0019  4F                 CLRA
001A  B7 80 07           STAA  PIACRB
001D  B7 80 06           STAA  PIAORB   ; Set Port B as input
0020  0E                 CLI
0021  C6 34              LDAB  #$34
0023  86 3D              LDAA  #$3D
0025  F7 80 07   CONVRT  STAB  PIACRB   ; Starts ADC0801
0028  B7 80 07           STAA  PIACRB
002B  3E                 WAI            ; Wait for interrupt
002C  DE 40              LDX   TEMP1
002E  8C 02 0F           CPX   #$020F   ; Is final data stored?
0031  27 0F              BEQ   ENDP
0033  08                 INX
0034  DF 40              STX   TEMP1
0036  20 ED              BRA   CONVRT
0038  DE 40      INTRPT  LDX   TEMP1
003A  B6 80 06           LDAA  PIAORB   ; Read data in
003D  A7 00              STAA  X        ; Store it at X
003F  3B                 RTI
0040  02 00      TEMP1   FDB   $0200    ; Starting address for
                                        ; data storage
0042  CE 02 00   ENDP    LDX   #$0200   ; Reinitialize TEMP1
0045  DF 40              STX   TEMP1
0047  39                 RTS            ; Return from subroutine
                 PIAORB  EQU   $8006    ; To user's program
                 PIACRB  EQU   $8007
```

The following schematic and sample subroutine (DATA IN) may be used to interface (up to) 8 ADC0801's directly to the MC6800 CPU. This scheme can easily be extended to allow the interface of more converters. In this configuration the converters are (arbitrarily) located at HEX address 5000 in the MC6800 memory space. To save components, the clock signal is derived from just one RC pair on the first converter. This output drives the other A/Ds.

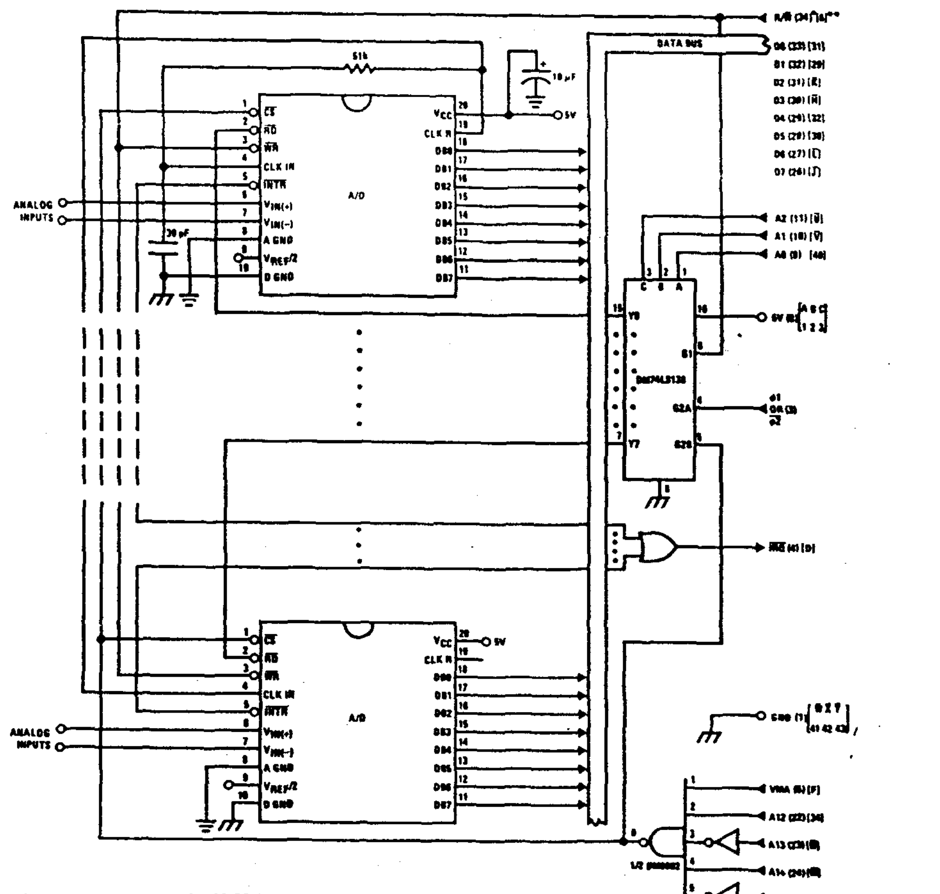
All the converters are started simultaneously with a STORE instruction at HEX address 5000. Note that any other HEX address of the form 5XXX will be decoded by the circuit, pulling all the $\overline{CS}$ inputs low. This can easily be avoided by using a more definitive address decoding scheme. All the interrupts are ORed together to insure that all A/Ds have completed their conversion before the microprocessor is interrupted.

The subroutine, DATA IN, may be called from anywhere in the user's program. Once called, this routine initializes the CPU, starts all the converters simultaneously and waits for the interrupt signal. Upon receiving the interrupt, it reads the converters (from HEX addresses 5000 through 5007) and stores the data successively at (arbitrarily chosen) HEX addresses 0200 to 0207, before returning to the user's program. All CPU registers then recover the original data they had before servicing DATA IN.

5.2 Auto-Zeroed Differential Transducer Amplifier and A/D Converter

The differential inputs of the ADC0801 series eliminate the need to perform a differential to single ended conversion for a differential transducer. Thus, one op amp can be eliminated since the differential to single ended conversion is provided by the differential input of the ADC0801 series. In general, a transducer preamp is required to take advantage of the full A/D converter input dynamic range.

Functional Description (Continued)



Note 1: Numbers in parentheses refer to MC6800 CPU pin out.

Note 2: Numbers of letters in brackets refer to standard M6800 system common bus code.

TL/H/5671–26

FIGURE 16. Interfacing Multiple A/Ds in an MC6800 System

SAMPLE PROGRAM FOR *FIGURE 16* INTERFACING MULTIPLE A/D's IN AN MC6800 SYSTEM

```
ADDRESS  HEX CODE           MNEMONICS               COMMENTS
0010     DF 44      DATAIN  STX   TEMP      ; Save Contents of X
0012     CE 00 2A           LDX   #$002A    ; Upon IRQ LOW CPU
0015     FF FF F8           STX   $FFF8     ; Jumps to 002A
0018     B7 50 00           STAA  $5000     ; Starts all A/D's
001B     0E                 CLI
001C     3E                 WAI             ; Wait for interrupt
001D     CE 50 00           LDX   #$5000
0020     DF 40              STX   INDEX1    ; Reset both INDEX
0022     CE 02 00           LDX   #$0200    ; 1 and 2 to starting
0025     DF 42              STX   INDEX2    ; addresses
0027     DE 44              LDX   TEMP
0029     39                 RTS             ; Return from subroutine
002A     DE 40      INTRPT  LDX   INDEX1    ; INDEX1 → X
```

19. It must be noted that the ADC0801 series will output an all zero code when it converts a negative input [$V_{IN}(-) \geq V_{IN}(+)$]. Also, a logic inversion exists as all of the I/O ports are buffered with inverting gates.

Basically, if the data read is zero, the differential output voltage is negative, so a bit in Port B is cleared to pull V_X more negative which will make the output more positive for the next conversion. If the data read is not zero, the output voltage is positive so a bit in Port B is set to make V_X more positive and the output more negative. This continues for 8 approximations and the differential output eventually converges to within 5 mV of zero.

The actual program is given in *Figure 20*. All addresses used are compatible with the BLC 80/10 microcomputer system. In particular:

- Port A and the ADC0801 are at port address E4
- Port B is at port address E5
- Port C is at port address E6
- PPI control word port is at port address E7
- Program Counter automatically goes to ADDR:3C3D upon acknowledgement of an interrupt from the ADC0801

5.3 Multiple A/D Converters in a Z-80 Interrupt Driven Mode

In data acquisition systems where more than one A/D converter (or other peripheral device) will be interrupting program execution of a microprocessor, there is obviously a need for the CPU to determine which device requires servicing. *Figure 21* and the accompanying software is a method of determining which of 7 ADC0801 converters has completed a conversion ($\overline{INTR}$ asserted) and is requesting an interrupt. This circuit allows starting the A/D converters in any sequence, but will input and store valid data from the converters with a priority sequence of A/D 1 being read first, A/D 2 second, etc., through A/D 7 which would have the lowest priority for data being read. Only the converters whose INT is asserted will be read.

The key to decoding circuitry is the DM74LS373, 8-bit D type flip-flop. When the Z-80 acknowledges the interrupt, the program is vectored to a data input Z-80 subroutine. This subroutine will read a peripheral status word from the DM74LS373 which contains the logic state of the $\overline{INTR}$ outputs of all the converters. Each converter which initiates an interrupt will place a logic "0" in a unique bit position in the status word and the subroutine will determine the identity of the converter and execute a data read. An identifier word (which indicates which A/D the data came from) is stored in the next sequential memory location above the location of the data so the program can keep track of the identity of the data entered.

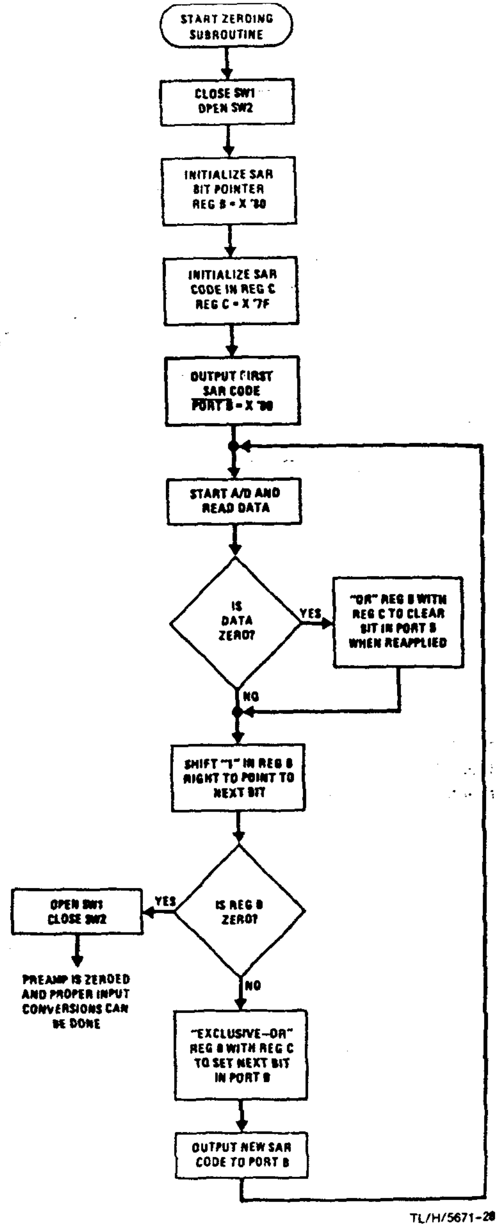


FIGURE 19. Flow Chart for Auto-Zero Routine

```
3D02  D3E7    Out Control Port                            ; Program PPI
3D04  2601    MVI H 01         Auto-Zero Subroutine
3D06  7C      MOV A,H
3D07  D3E6    OUT C                                       ; Close SW1 open SW2
3D09  0680    MVI B 80                                    ; Initialize SAR bit pointer
3D0B  3E7F    MVI A 7F                                    ; Initialize SAR code
3D0D  4F      MOV C,A          Return
3D0E  D3E5    OUT B                                       ; Port B = SAR code
3D10  31AA3D  LXI SP 3DAA      Start                      ; Dimension stack pointer
3D13  D3E4    OUT A                                       ; Start A/D
3D15  FB      IE
3D16  00      NOP              Loop                       ; Loop until INT asserted
3D17  C3163D  JMP Loop
3D1A  7A      MOV A,D          Auto-Zero
3D1B  C600    ADI 00
3D1D  CA2D3D  JZ Set C                                    ; Test A/D output data for zero
3D20  78      MOV A,B          Shift B
3D21  F600    ORI 00                                      ; Clear carry
3D23  1F      RAR                                         ; Shift "1" in B right one place
3D24  FE00    CPI 00                                      ; Is B zero? If yes last
3D26  CA373D  JZ Done                                     ; approximation has been made
3D29  47      MOV B,A
3D2A  C3333D  JMP New C
3D2D  79      MOV A,C          Set C
3D2E  B0      ORA B                                       ; Set bit in C that is in same
3D2F  4F      MOV C,A                                     ; position as "1" in B
3D30  C3203D  JMP Shift B
3D33  A9      XRA C            New C                      ; Clear bit in C that is in
3D34  C30D3D  JMP Return                                  ; same position as "1" in B
3D37  47      MOV B,A          Done                       ; then output new SAR code.
3D38  7C      MOV A,H                                     ; Open SW1, close SW2 then
3D39  EE03    XRI 03                                      ; proceed with program. Preamp
3D3B  D3E6    OUT C                                       ; is now zeroed.
3D3D          •                Normal
              •
              •
              Program for processing
              proper data values
3C3D  DBE4    IN A             Read A/D Subroutine        ; Read A/D data
3C3F  EEFF    XRI FF                                      ; Invert data
3C41  57      MOV D,A
3C42  78      MOV A,B                                     ; Is B Reg = 0? If not stay
3C43  E6FF    ANI FF                                      ; in auto zero subroutine
3C45  C21A3D  JNZ Auto-Zero
3C48  C33D3D  JMP Normal
```

Note: All numerical values are hexadecimal representations.

FIGURE 20. Software for Auto-Zeroed Differential A/D

5.3 Multiple A/D Converters in a Z-80® Interrupt Driven Mode (Continued)

The following notes apply:

1) It is assumed that the CPU automatically performs a RST 7 instruction when a valid interrupt is acknowledged (CPU is in interrupt mode 1). Hence, the subroutine starting address of X0038.

2) The address bus from the Z-80 and the data bus to the Z-80 are assumed to be inverted by bus drivers.

3) A/D data and identifying words will be stored in sequential memory locations starting at the arbitrarily chosen address X 3E00.

4) The stack pointer must be dimensioned in the main program as the RST 7 instruction automatically pushes the PC onto the stack and the subroutine uses an additional

5) The peripherals of concern are mapped into I/O space with the following port assignments:

HEX PORT ADDRESS	PERIPHERAL
00	MM74C374 8-bit flip-flop
01	A/D 1
02	A/D 2
03	A/D 3
04	A/D 4
05	A/D 5
06	A/D 6
07	A/D 7

BIODATA

Nama : **HERI HARSONO**

Nrp : 5103097036

TTL : Surabaya, 11 Pebruari 1979

Agama : Katolik.

Alamat : Sidoyoso II Blok II No 37

SURABAYA

Riwayat Pendidikan :

- SDK 'Materdai II' Lumajang 1985-1991
- SMPK 'Soegijopranata' Lumajang 1991-1994
- SMUK 'Soegijopranata' Lumajang 1994-1997
- Universitas Katolik Widya Mandala Surabaya Fakultas Teknik Jurusan Teknik Elektro tahun 1997. pada bulan April 2002 mengikuti ujian skripsi.