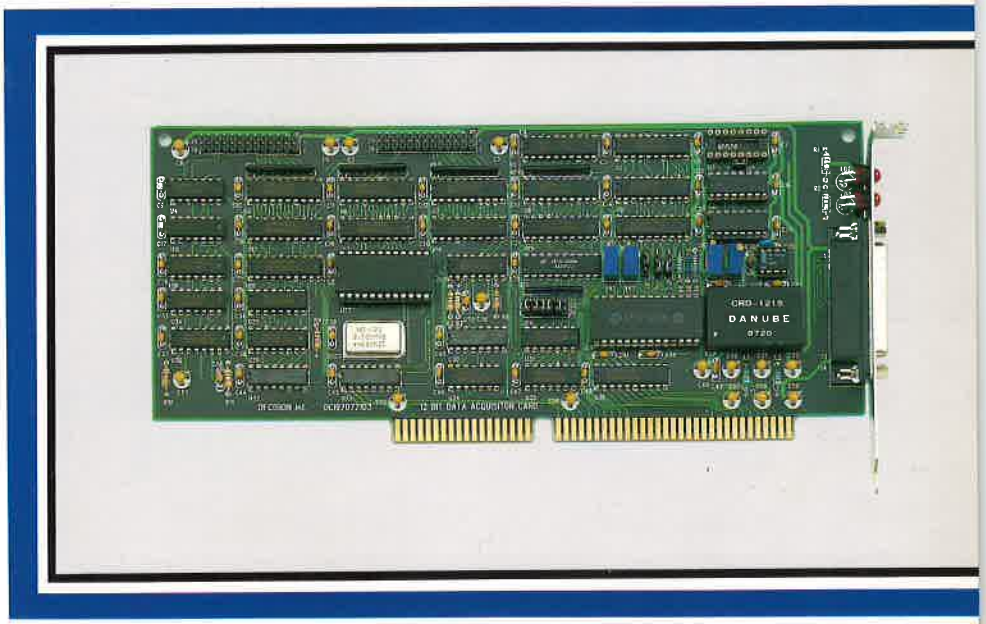
12 BIT DATA ACQUISITION CARD USER MANUAL



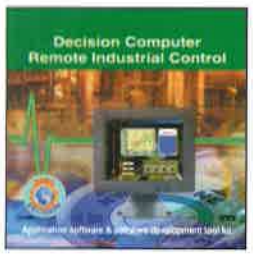
We provide software development tool kit package for free every shipment

Features:

- Send remote message to "PAGER" (DB CALL) with the use of Client/Server function.
- Transmit remote "VOICE MAIL" to telephone or portable phone with use of Client/Server function.
- Receive remote incoming phone call and play message with the use of Client/Server function.
- Detect remote DTMF (DUAL TONE MULTI-FREQUENCY) with the use of Client/Server function.
- Transmit and receive remote message to FAX with the use of Client/Server function.

Features:

- Send message to "PAGER"(DB Call)
- Send and receive "VOICE MAIL" from computer to telephone or portable phone.
- Send and receive E-MAIL.
- Send and receive FAX.
- Receive DTMF.
- OCX software accessory for Windows 95/98/NT.
- Standard COM2 and Decision Computer PC COM TeleCom 2/4/8 port RS-232 multi serial card.



Features:

- DII-Decision Industrial Interface: Device Driver for Decision Industrial Card for Windows 95, 98 , NT environment.
- DIC-Decision Industrial Controller: Software development tool kit which provides remote industrial control function via internet and World Wide Web (WWW).
- Turn-Point: Application software for remote temperature and humidity and digital input output control.

Features:

PC COM RemoteCom is an OCX programming development tool for application software. That will use to enhance RS-232/422 serial communication through Internet by stand-alone program or by browser through World Wide Web (WWW) under Windows 95/98/NT.

PC COM RemoteCom provides 1 to 16 RS-232 serial communication port, which link the computer and RS-232/422 serial peripheral device such as terminals, modems, RS-232 serial printer, card reader, plotters, ... etc.

User may call OCX functions to communicate with RemoteCOM both Server and Client using Internet communication; or encapsulate OCX function and remote serial ActiveX Control, then run development application program under Internet browser (IE and Netscape).

Features:

- All UART parameters are fully configurable.
- Review received data.
- Save received data.
- Monitor data of one or both devices at same time.
- Monitor all signals of one or both devices at same time.
- Monitor the interaction of both devices (Monitor mode).
- Monitor data and signals transparently (Transparent mode).

Software Development Tool Kits

(Using E-Mail Address to Find IP Address For Windows 95/98/NT)

12 BIT DATA ACQUISITION CARD

OPERATION MANUAL

DECISION
Computer International Co., Ltd.

98/03/22

Table of Contents

CHAPTERS

APPENDICES

CHAPTER 1
INTRODUCTION

The 12 bit data acquisition card is a high precision data conversion/acquisition system for PC/XT, PC/AT, PC/386, Pentium or compatibles. It contains 16 analog to digital channels with unipolar or bipolar input and 4 digital I/O channels with 8 bit line for each channel. The on board 8254 chip provides programmable interval timer/counter functions to trig A/D conversion. The 12 bit data acquisition card also provides interrupt driven for convention A/D input.

The features of the 12 bit data acquisition adapter are:

- Provides 16 A/D channels and the resolution is 12 bits.
- Unipolar or bipolar selectable.
- Input voltage range from 0 to 10V or 0 to 20V for unipolar, and from –5V to 5V or –10V to 10V for bipolar.
- Provides 4 digital input/digital output channels and the resolution is 8 bits.
- Provides three independent 16 bits counter to trig A/D conversion.
- I/O port address selectable.
- ISA bus interrupt programmable (IRQ3 to IRQ15).
- Gain control factor selectable from 1 to 8.

CHAPTER 2

UNPACKING INFORMATION

Check that your 12 bit data acquisition package includes the following items:

- 12 bit data acquisition adapter.
- User manual.
- Software utilities.
- Warranty form.

CHAPTER 3

SYSTEM REQUIREMENTS

Before installing your 12 bit data acquisition adapter, make sure that:

- The host computer is an IBM PC/AT, PC/386, PC/486, and Pentium compatibles.
- The five jumpers' blocks are correctly configured to coincide with the operating system you are using.

CHAPTER 4
HARDWARE INSTALLATION

Your 12 bit data acquisition adapter is designed to be inserted in any available slot in your PC/AT, PC/386, PC/486, Pentium or compatibles. In order to gain access to the expansion slots, follow the steps listed below:

1. Turn off all power to your computer and all peripheral devices before installing your 12 bit data acquisition adapter.
2. Remove the cover of the computer.
3. Insert the pre-configured 12 bit data acquisition adapter into any available slot. Make sure the adapter is firmly seated in the chosen slot.
4. Replace the cover of the computer.

Note:

1. You must adjust the A/D full scalar reference voltage by screwing the VR resistor. (see VR Full Scalar Adjustment).
2. You must setup everything including the connection of the analog input into the DB25 channel input before turning on the PC power, otherwise it may damage the card.

CHAPTER 5
HARDWARE CONFIGURATION

5.1 Introduction

The five jumper blocks on the 12 bit data acquisition adapter must be configured correctly in accordance with the operating system you are using.

JP1 (Jumper 1)
Determines A/D input voltage range.

JP2 (Jumper 2)
Determines unipolar or bipolar.

JP3 (Jumper 3)
Determines input mode.

JP4 (Jumper 4)
Determines I/O port address.

JP5 (Jumper 5)
Determines AD526 (gain control factor) is used.

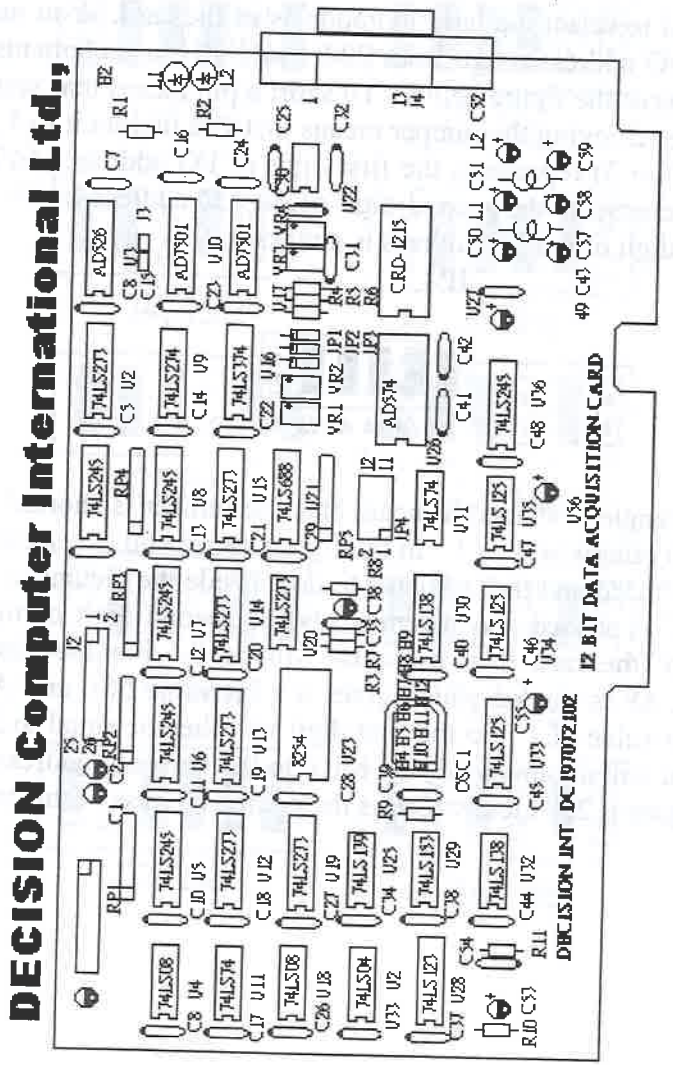
5.2 Configuration for Jumper

It is important to refer to the user manual supplied with your operating system to determine the correct configuration. Please contact your supplier if you have any difficulties with configuration.

IMPORTANT: CARE MUST BE TAKEN IN SELECTING THE CONFIGURATION OF JUMPERS TO ENSURE YOU

DO NOT DUPLICATE SETTINGS OF OTHER EQUIPMENT ALREADY INSTALLED IN YOUR COMPUTER. DUPLICATION OF SETTINGS WILL RESULT IN A MALFUNCTION OF ONE OR BOTH DEVICES.

☞ Please refer to the following settings for each switch and jumper block. If you are installing more than one board, do not duplicate jumper settings for any parameter.



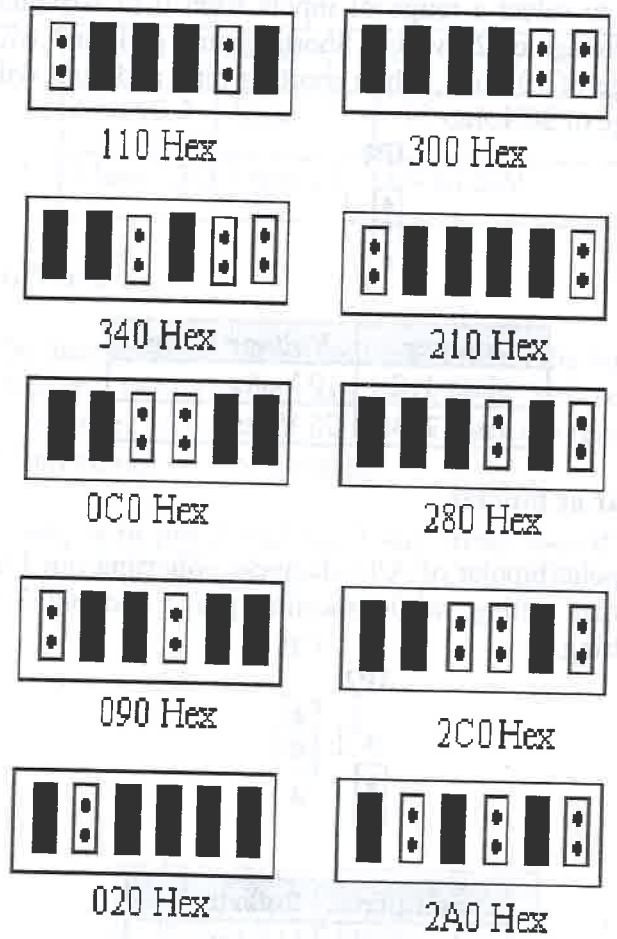
1. I/O Port Address

JP4 is used to select the base I/O address of the card. You may set the base I/O address range from 000H to 3F0H in increments of F Hex. Observe the figure below. To short a pin means that you set it as 0, while removing the jumper means that you just set it to 1. (A9, A8) (*i.e.* 0 to 3) represents the first digit of I/O address. (A7, A6, A5, A4) represents the second digit of the I/O address (*i.e.* 0 to F). The third digit of the I/O address is always 0.

JP4

If the rectangle is shaded it means that the jumper is shorted with a value of 0 otherwise it is 1. In the figure above you can see that A4 to A8 is shaded and that A9 have to dots inside the rectangle. Since A4 to A7 is shaded and it represents the second digit of the base address of the card then its value will be 0. For the first digit however, A8 is shaded which gives it a bit value of 0 and A9 will have a bit value of 1. So the first digit will then be equal to 2. The third digit will automatically be equal to 0. The base address of the above figure is 200 Hex, which is the default address of the card.

Addressing Example:



2. Voltage Range

JP1 is used to select a range of inputs from 0 to 10 volts or to a maximum voltage of 20 volts. Shorting pin1 and pin2 will have a voltage range of 10 volts, while shorting pin2 and pin3 will have a voltage range of 20 volts.

JP1

Jumper	Voltage Range
short 1, 2	10 Volts
short 2, 3	20 Volts

3. Unipolar or bipolar

Selects unipolar/bipolar of A/D channels. Shorting pin 1 and pin 2 selects bipolar setting, while shorting pin 2 and pin 3 selects a unipolar setting.

JP2

Jumper	Polarity
short 1, 2	bipolar
short 2, 3	unipolar

The combination of voltage range and polarity are shown in the following:

JP1	JP2	Voltage Range
Short 1,2	Short 1,2	-5V to 5V
Short 1,2	Short 2,3	0V to 10V
Short 2,3	Short 1,2	-10V to 10V
Short 2,3	Short 2,3	0V to 20V

4. Input Mode

Setting the jumper to pin 1 and pin 2 will ground the input value, so the data that will be converted should also be 0. This setting is used whenever we need to calibrate the VR or variable resistors so that the minimum voltage will be attained.

Shorting jumper to pin 2 and pin 3 will make use of the channel inputs to be read and converted.

JP3

Jumper	Input Mode
short 1, 2	Ground
short 2, 3	Channel Input

5. Programmable Gain Control Factor (AD526)

The JP5 is used to select programmable gain control factor, when not short the jumper, it enable AD526, otherwise short the pin means no AD526 work.

JP5

Jumper	AD526
short	no AD526
not short	AD526

5.3 I/O Address Specification

The I/O address specification are shown in the following:

For READ input

port* + *0: input A/D low byte data.
port* + *1: input A/D high byte data.
port* + *2: read back control signal for data conversion.
port* + *4: digital I/O channel 1.
port* + *5: digital I/O channel 2.
port* + *6: digital I/O channel 3.
port* + *7: digital I/O channel 4.
port* + *8: counter 0 I/O buffer (8254 IC).
port* + *9: counter 1 I/O buffer (8254 IC).
port* + *A: counter 2 I/O buffer (8254 IC).
port* + *B: counter control register (8254 IC).

For WRITE output

port* + *0: select A/D channel number and enable/disable the selected channel.
port* + *1: select IRQ and select control method.
port* + *2: start data conversion.
port* + *3: output signal to reset/retrigger IRQ.
port* + *4: digital I/O channel 1.
port* + *5: digital I/O channel 2.
port* + *6: digital I/O channel 3.
port* + *7: digital I/O channel 4.
port* + *8: counter 0 I/O buffer (8254 IC).
port* + *9: counter 1 I/O buffer (8254 IC).
port* + *A: counter 2 I/O buffer (8254 IC).
port* + *B: counter control register (8254 IC).

5.4 VR Full Scalar Adjustment

VR Number	Function
VR1	A/D bipolar offset voltage
VR2	AD574 reference voltage
VR3	A/D unipolar offset voltage
VR4	Offset voltage of LF398

VR1 is used to adjust the A/D bipolar offset voltage. JP2 should be set to pin 1 and 2 short before measuring the voltage input. After doing so, you can check if the values you get is half the input voltage by placing the multimeter probe to ground and pin 12 of AD574 (U26).

VR2 is used to adjust the reference voltage of AD574. The probe of the multimeter should be at the AD574 pin 10 (U26) and ground.

The value that you will read from the multimeter is 10 volts, otherwise you should adjust the variable resistor (VR2).

VR3 is used to adjust the A/D unipolar offset voltage. JP2 should be set to pin 2 and 3 short before measuring the voltage input. After doing so, you can check if the values you get is the same as the input voltage. You can test it by placing the multimeter probe to ground and pin 12 of AD574 (U26).

VR4 is used to adjust the offset voltage of LF398, to do so, one must short JP3 to pin 1 and 2, and make sure that JP2 is shorted at pin 2 and 3 (*i.e.* in the unipolar mode). After doing so, you should place the multimeter probe or any digital voltage reading device to ground and to LF398 pin 5 (U22) or JP1 pin 2 position. The value that you will get should be zero voltage, if not, you can adjust the VR4 to get the desired value.

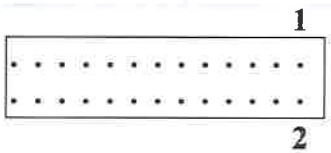
5.5 Pin Assignments

1. DB25 Pin Assignments for A/D (J3)

Pin	Function	Pin	Function
1	+ 12 V	14	-12 V
2	/EXTRG	15	No Connection
3	GND	16	CH15
4	CH14	17	CH13
5	CH12	18	CH11
6	CH10	19	CH9
7	CH8	20	CH7
8	CH6	21	CH5
9	CH4	22	CH3
10	CH2	23	CH1
11	CH0	24	GND
12	GND	25	-5 V
13	+5 V		

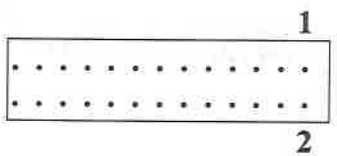
2. Digital I/O Pin Assignment for J1

Pin	Function	Pin	Function
1	+12 V	14	2D1
2	GND	15	2D2
3	+12 V	16	2D3
4	GND	17	2D4
5	1D0	18	2D5
6	1D1	19	2D6
7	1D2	20	2D7
8	1D3	21	J2-21
9	1D4	22	J2-22
10	1D5	23	+5 V
11	1D6	24	GND
12	1D7	25	-12 V
13	2D0	26	GND



3. Digital I/O Pin Assignment for J2

Pin	Function	Pin	Function
1	+12 V	14	4D1
2	GND	15	4D2
3	+12 V	16	4D3
4	GND	17	4D4
5	3D0	18	4D5
6	3D1	19	4D6
7	3D2	20	4D7
8	3D3	21	J3-21
9	3D4	22	J3-22
10	3D5	23	+5 V
11	3D6	24	GND
12	3D7	25	-12 V
13	4D0	26	GND



CHAPTER 6

SOFTWARE CONFIGURATION

6.1 Input Programming

1. port + 0

MSB							LSB
7	6	5	4	3	2	1	0

Input A/D low byte data.

2. port + 1

MSB							LSB
0	0	0	0	11	10	9	8

Input A/D high byte data.

3. port + 2

MSB							LSB
/hold	R2	R1	R0	C3	C2	C1	C0

Read back control signal for data conversion. Please refer next section for more details.

4. port + 4

MSB							LSB
1D7	1D6	1D5	1D4	1D3	1D2	1D1	1D0

Digital input channel 1.

5. port + 5

MSB							LSB
2D7	2D6	2D5	2D4	2D3	2D2	2D1	2D0

Digital input channel 2.

6. port + 6

MSB							LSB
3D7	3D6	3D5	3D4	3D3	3D2	3D1	3D0

Digital input channel 3.

7. port + 7

MSB							LSB
4D7	4D6	4D5	4D4	4D3	4D2	4D1	4D0

Digital input channel 4.

8. port + 8

Counter 0 I/O buffer (8254 IC).

9. port + 9

Counter 1 I/O buffer (8254 IC).

10. port + A

Counter 2 I/O buffer (8254 IC).

11. port + B

Counter control register (8254 IC).

6.2 Output Programming

1. port + 0

MSB							LSB
x	R2	R1	R0	C3	C2	C1	C0

Select A/D channel number and enable/disable the selected channel. The R0 to R2 are used to select gain control factor, and the C0 to C4 are used to select A/D input channel. If JP5 is short, the Ro to R2 are disable.

R2	R1	R0	Gain Control Factor
0	0	0	*1
0	0	1	*2
0	1	0	*3
0	1	1	*4
1	0	0	*5
1	0	1	*6
1	1	0	*7
1	1	1	*8

The gain control factor is used to scale your input voltage. For example, if you select unipolar and its voltage range from 0 to 20V, and the gain control factor is *8, then your input voltage range is from 0 to 2.5V, because whole the input voltage was scale 8 times.

C3	C2	C1	C0	Input Channel
0	0	0	0	CH0
0	0	0	1	CH1
0	0	1	0	CH2
0	0	1	1	CH3
0	1	0	0	CH4
0	1	0	1	CH5
0	1	1	0	CH6
0	1	1	1	CH7
1	0	0	0	CH8
1	0	0	1	CH9
1	0	1	0	CH10
1	0	1	1	CH11
1	1	0	0	CH12
1	1	0	1	CH13
1	1	1	0	CH14
1	1	1	1	CH15

2. port + 1

MSB							LSB
ENX	X	T1	T0	SL3	SL2	SL1	SL0

Select IRQ and select control method. The T1 and T2 are used to select trig method for A/D converter, and the SL0 to SL3 are used to select interrupt line. When user select external trig(T1=1 and T0=0), he must connect J3 pin 2.

ENX	Enable/Disable
0	Gate Disable
1	Gate Enable

T1	T0	Selection
0	0	Software trigger by port address + 2
0	1	Trig by 8254
1	0	External trig from J3-2
1	1	No trigger

When user select trig by 8254 (T1=0 and T0 =1), he must enable the gate by set ENX bit. The clock rate of 8254 is 2M, it is connected to counter 0, then the output of counter 0 is connected to counter 1, so that user need divide the clock by counter 0 then divide it by counter 1. The divided clock rate of counter 1 is used to trig A/D conversion.

SL3	SL2	SL1	SL0	Interrupt Selection
0	x	x	x	No selection
1	0	0	0	IRQ3
1	0	0	1	IRQ4
1	0	1	0	IRQ5
1	0	1	1	IRQ7
1	1	0	0	IRQ10
1	1	0	1	IRQ11
1	1	1	0	IRQ12
1	1	1	1	IRQ15

3. port + 2

Start data conversion. When you select the software trigger (T1=0, T0=0) to start the A/D converter process, you can write any value to this address and start A/D converter process. It needs about 200ns to convert the signal.

4. port + 3

Output signal to reset/retrigger IRQ. When the A/D chip finish the converter process, they generate interrupt request selected by your choice channel. After your program finish the job, you must write any value to this address to clear the interrupt request signal.

5. port + 4

MSB							LSB
1D7	1D6	1D5	1D4	1D3	1D2	1D1	1D0

Digital output channel 1.

6. port + 5

MSB							LSB
2D7	2D6	2D5	2D4	2D3	2D2	2D1	2D0

Digital output channel 2.

7. port + 6

MSB							LSB
3D7	3D6	3D5	3D4	3D3	3D2	3D1	3D0

Digital output channel 3.

8. port + 7

MSB							LSB
4D7	4D6	4D5	4D4	4D3	4D2	4D1	4D0

Digital output channel 4.

9. port + 8

counter 0 I/O buffer (8254 IC).

10. port + 9

counter 1 I/O buffer (8254 IC).

11. port + A

counter 2 I/O buffer (8254 IC).

12. port + B

counter control register (8254 IC).

6.3 Programming Examples

If you want port+4, port+5, port+6, and port+7 as input channel, you must write FFH value to this port. In the following, we show a C language example.

This program uses the minimum resources of the 12 bit data acquisition. It only uses IRQ5 and an analog channel and checks digital channels in the card, but before you can use the option for digital I/O you must have a 2 x 26 pin cable connector placed in J1 and J2.

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <ctype.h>
#include <process.h>
#include <stdlib.h>

#ifdef __cplusplus
    #define __CPPARGS ...
#else
    #define __CPPARGS
#endif

int channel = 0, address = 0x200;
unsigned int hibyte, lobyte;
float ad;
long int countint;

void main();

void interrupt (*old_interrupt)(__CPPARGS);    //interrupt
handler

void interrupt dataacq_interrupt(__CPPARGS)  //interrupt
service routine
{
countint++;  //interrupt counter
outportb(address, channel);  //selects the channel that will
send a signal
outportb(address + 3, 0x00);  //resets the IRQ
hibyte = inportb(address + 1); //captures hibyte data
lobyte = inportb(address);  //captures lobyte data
outportb(0x20, 0x20); //end of interrupt
}


void one_channel()
```



```
{
int looping, OldMask1, NewMask1;
char select_irq, get_ch;

countint = 0;
clrscr();
old_interrupt = getvect(0x0d);    //save old interrupt service
routine
disable();                          //disable interrupts
setvect(0x0d, dataacq_interrupt);  //use new ISR
enable();                        //enable interrupts
OldMask1 = inportb(0x21);          //save old interrupt mask
NewMask1 = OldMask1 & 0xDF;       //0xDF masks the interrupt
to be used
outportb(0x21, NewMask1);         //output new interrupt mask

//controls output of counter to 100hz
outportb(address + 11, 0x36);
outportb(address + 8, 0xff);
outportb(address + 8, 0x03);
outportb(address + 11, 0x76);
outportb(address + 9, 0x14);
outportb(address + 9, 0x00);

clrscr();
gotoxy(21,3);printf("Channel              Received Value");
gotoxy(20,22);printf("Press 'Space' To Pause and Any Key To
Quit");
outportb(address + 1, 0x9A); // output to selected irq
outportb(address + 3, 0x00);  //reset IRQ

do
    {
    gotoxy(1,1);printf("Interrupt Count: %ld", countint);
//loop counter

    while(kbhit())//if key is pressed
        {
        get_ch = toupper(getch());
        if(get_ch == 0x20)  //if space bar
            {
            getch();
            }
        else
            {
```

```
                disable(); //disable interrupt
                setvect(0x0d, old_interrupt);  //restore old
ISR
                enable(); //enable interrupt
                outportb(0x21, OldMask1);   //restore old mask
                main();
                }
            }
          ad = ((hibyte & 0x0f) << 8) + (lobyte & 0xff);
               //convert  hex   value to decimal
       ad = (ad * 10) / 0xfff;

       for(int z=0;z<=50;z++)      //loop counter
       for(int y=0;y<=100;y++)

       gotoxy(24,5);printf("%d", channel);  //display channel
       gotoxy(45, 5);printf("%.3f ", ad);  //display data

          }
while(!kbhit());
disable();
setvect(0x0d, old_interrupt);
enable();
outportb(0x21, OldMask1);
main();
}


void digital_io()
{
int four_digital_io, all_hi, digital_addr, count_by_5,
digital_send;
int data_read;
long int loop_count = 0;
char esc_key;

do
     {
     clrscr();
     gotoxy(28,20);printf("Press Any Key To Quit");
     gotoxy(1,1);printf("Loops: %ld    ", loop_count);

     for(count_by_5 = 0; count_by_5 <= 255; count_by_5 +=5)
          {
          delay(5);
```



```
          while(kbhit())
               {
               main();
               }
          gotoxy(1,4);printf("Transfer Data J3-1D to J2-3D");
          outportb(address + 4, count_by_5);
          outportb(address + 6, 0xff);
          data_read = inportb(address + 6);
          gotoxy(1,5);printf("Digital Input = %d
",data_read);
          if(count_by_5 == data_read)
               {
               gotoxy(25,5);printf("...Pass  ");
               }
          else
               {
               gotoxy(25,5);printf("...Error");
               gotoxy(25,20);printf("Press Any Key To
Continue");
               esc_key = getch();
               if(esc_key == 0x1b)
                    {
                    main();
                    }
               }
          }
     loop_count++;
     }
while(!kbhit());
}

void main()
{
char main_select;

clrscr();

//main menu
gotoxy(25,7);printf("12 BIT DATA ACQUISITION CARD");
gotoxy(28,9);printf("[1] Test Analog Input");
gotoxy(28,10);printf("[2] Digital I/O");
gotoxy(28,11);printf("[3] Exit");
gotoxy(30,13);printf("Choose Option: ");

do
```

```
    {
    gotoxy(45,13);main_select = toupper(getch());
    switch(main_select)
        {
        case '1':
            //menu select for analog input function
            one_channel();
            break;
        case '2':
            digital_io();
            break;
        case '3':
                    //Terminates the program whenever the
                    user chooses 5 from the main menu
            exit(0);
            break;
        }
    }
while(1);
}
```


APPENDIX A

WARRANTY INFORMATION

A.1 Copyright

Each piece of 12 BIT DATA ACQUISITION CARD package permits user to use 12 BIT DATA ACQUISITION CARD only on a single computer, a registered user may use the program on a different computer, but may not use the program on more than one computer at the same time.

Corporate licensing agreements allow duplication and distribution of specific number of copies within the licensed institution. Duplication of multiple copies is not allowed except through execution of a licensing agreement. Welcome call for details.

A.2 Warranty Information

DECISION warrants that for a period of one year from the date of purchase (unless otherwise specified in the warranty card) that the goods supplied will perform according to the specifications defined in the user manual. Furthermore that the 12 BIT DATA ACQUISITION CARD product will be supplied free from defects in

materials and workmanship and be fully functional under normal usage.

In the event of the failure of a 12 BIT DATA ACQUISITION CARD product within the specified warranty period, DECISION will, at its option, replace or repair the item at no additional charge. This limited warranty does not cover damage resulting from incorrect use, electrical interference, accident, or modification of the product.

All goods returned for warranty repair must have the serial number intact. Goods without serial numbers attached will not be covered by the warranty.

Transportation costs for goods returned must be paid by the purchaser. Repaired goods will be dispatched at the expense of 12 BIT DATA ACQUISITION CARD.

To ensure that your 12 BIT DATA ACQUISITION CARD product is covered by the warranty provisions, it is necessary that you return the Warranty card.

Under this Limited Warranty, DECISION's obligations will be limited to repair or replacement only, of goods found to be defective as specified above during the warranty period. DECISION is not liable to the purchaser for any damages or losses of any kind, through the use of, or inability to use, the 12 BIT DATA ACQUISITION CARD product.

DECISION reserves the right to determine what constitutes warranty repair or replacement.

Return Authorization: It is necessary that any returned goods are clearly marked with an RA number that has been issued by DECISION. Goods returned without this authorization will not be attended to.

APPENDIX B
DATA SHEET

82C54

CMOS Programmable Interval Timer

DISTINCTIVE CHARACTERISTICS

- Compatible with all Intel and most other microprocessors
- High-speed, zero-wait-state operation with 10-MHz 8086/88 and 80186/188
- Three independent 16-bit counters
- Handles inputs from DC to 8 MHz
  - 10 MHz for 82C54-2
  - 12.5 MHz for 82C54-12
- Low-power CMOS
  - I_{CC} = 10 μA commercial standby current I_{CC}
- Completely TTL compatible
- Six programmable counter modes
- Binary or BCD counting
- Status read-back command
- Available in 24-pin DIP and 28-lead PLCC

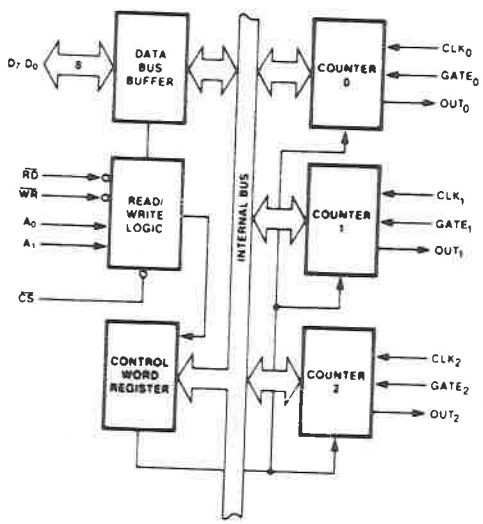
GENERAL DESCRIPTION

The AMD 82C54 is a high-performance, CMOS version of the industry-standard 8254 counter/timer which is designed to solve the timing-control problems common in microcomputer system design. It provides three independent 16-bit Counters — each capable of handling clock inputs up to 12.5 MHz. All modes are software-programmable. The 82C54 is pin-compatible with the NMOS 8254 and is a superset of the 8253.

Six programmable-timer modes allow the 82C54 to be used as an event counter, elapsed time indicator, programmable one-shot, and in many other applications as well.

The 82C54 is fabricated with AMD's CMOS technology providing low-power consumption with performance equal to or greater than the equivalent NMOS product. The 82C54 is available in 24-pin DIPs (plastic and ceramic) and 28-pin plastic leaded chip carrier (PLCC) packages.

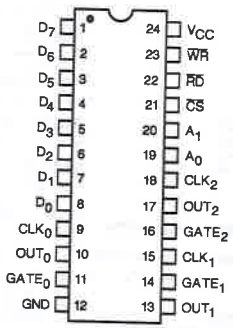
BLOCK DIAGRAM



BD006111

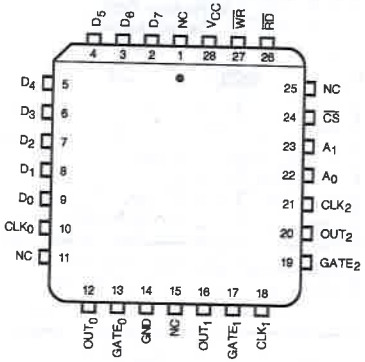
CONNECTION DIAGRAMS
Top View

DIPs



CD009390

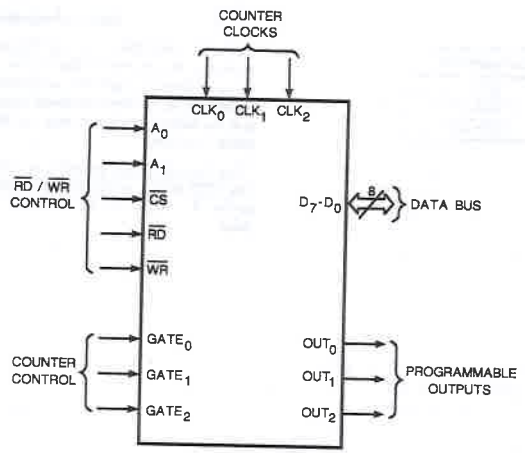
PLCC



CD009401

Note: Pin 1 is marked for orientation.

LOGIC SYMBOL

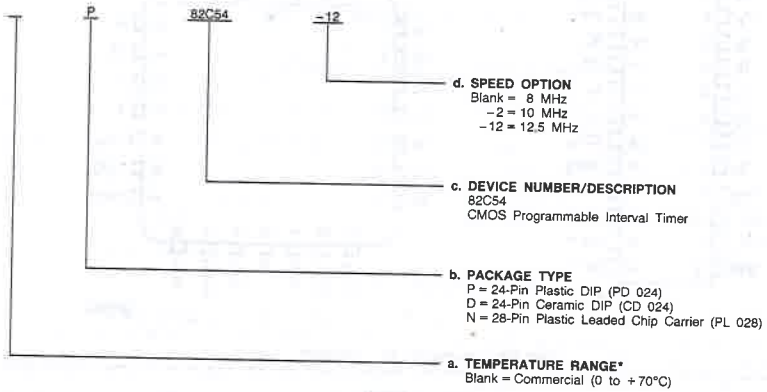


LS002340

V_{CC} = Power Supply
GND = Ground

ORDERING INFORMATION

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Temperature Range**
b. Package Type
c. Device Number
d. Speed Option



Valid Combinations	
P, D, N	82C54
	82C54-2
	82C54-12

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (order # 09275A/0) for electrical performance characteristics.

PIN DESCRIPTION

$D_7 - D_0$ Data Bus Lines (Bidirectional, Three-state)
Connected to system data bus.

CLK_0 Clock 0 (Input)
Clock input of Counter 0.

OUT_0 Out 0 (Output)
Output of Counter 0.

$GATE_0$ Gate 0 (Input)
Gate input of Counter 0.

CLK_1 Clock 1 (Input)
Clock input of Counter 1.

OUT_1 Out 1 (Output)
Output of Counter 1.

$GATE_1$ Gate 1 (Input)
Gate input of Counter 1.

CLK_2 Clock 2 (Input)
Clock input of Counter 2.

OUT_2 Out 2 (Output)
Output of Counter 2.

$GATE_2$ Gate 2 (Input)
Gate input of Counter 2.

A_1, A_0 Addresses (Input)
Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus.

A_1	A_0	Selects
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

$\overline{CS}$ Chip Select (Input, Active LOW)
A LOW on this input enables the 82C54 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.

$\overline{RD}$ Read Control (Input, Active LOW)
This input is LOW during CPU read operations.

$\overline{WR}$ Write Control (Input, Active LOW)
This input is LOW during CPU write operations.

V_{CC} +5-Volt Power Supply

GND Ground

NC No Connect

FUNCTIONAL DESCRIPTION

General

The 82C54 is a programmable interval counter/timer designed for use with AMD microcomputer systems. It is a general-purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 82C54 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in software, the programmer configures the 82C54 to match his/her requirements and programs one of the counters for the desired delay; after which the 82C54 will interrupt the CPU. Software overhead is minimal and variable length delays can easily be accommodated.

Some of the other counter/timer functions — common to microcomputers — which can be implemented with the 82C54 are:

- Real-time clock
- Event counter
- Digital one-shot
- Programmable-rate generator
- Square-wave generator
- Binary-rate multiplier
- Complex-waveform generator
- Complex-motor controller

Data Bus Buffer

This three-state, bidirectional 8-bit buffer is used to interface the 82C54 to the system bus (see Figure 1).

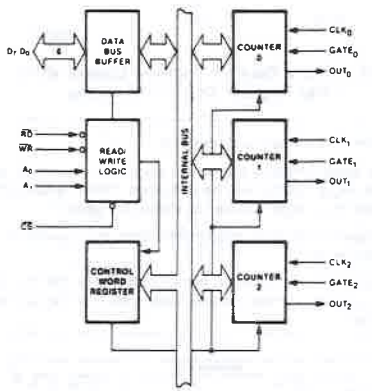


Figure 1. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 82C54. A_1 and A_0 select one of the three Counters or the Control Word Register to be read from/written into. A LOW on the $\overline{RD}$ input tells the 82C54 that the CPU is reading one of the Counters. A LOW on the $\overline{WR}$ input tells the 82C54 that the CPU is writing either a Control Word or an initial count. Both $\overline{RD}$ and $\overline{WR}$ are qualified by $\overline{CS}$; $\overline{RD}$ and $\overline{WR}$ are ignored unless the 82C54 has been selected by holding $\overline{CS}$ LOW.

Control Word Register

The Control Word Register (see Figure 2) is selected by the Read/Write Logic when A_1, $A_0 = 11$. If the CPU then does a write operation to the 82C54, the data is stored in the Control Word Register and is interpreted as a control word used to define the operation of the Counters.

The Control Word Register can only be written to; status information is available with the Read-Back Command.

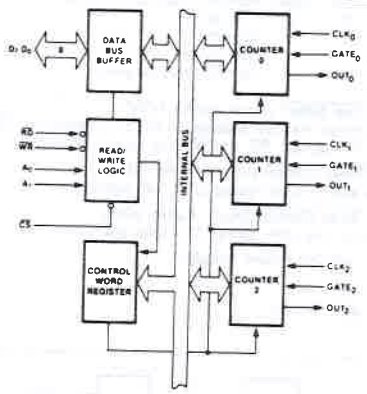


Figure 2. Block Diagram Showing Control Word and Register Counter Functions

Counter 0, Counter 1, Counter 2

These three functional blocks are identical in operation, so only a single Counter will be described. The internal block diagram of a single Counter is shown in Figure 3.

The Counters are fully independent; each may operate in a different mode.

The Control Word Register is shown in the figure; it is not part of the Counter, but its contents determine how the Counter operates.

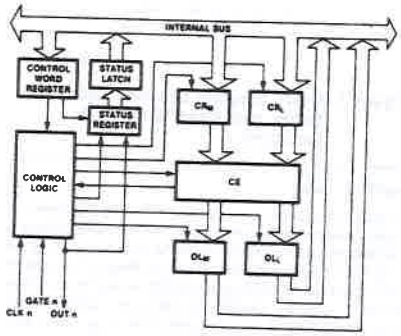


Figure 3. Internal Block Diagram of a Counter

The Status Register — shown in Figure 3 — when latched, contains the current contents of the Control Word Register and status of the output and null-count flag (see detailed explanation of the Read-Back Command).

The actual Counter is labeled CE (for "Counting Element"). It is a 16-bit, presettable synchronous-down Counter.

OL_M and OL_L are two 8-bit latches. OL stands for "Output Latch;" the subscripts M and L stand for "Most significant byte" and "Least significant byte," respectively. Both are normally referred to as one unit and called just OL. These latches normally "follow" the CE, but if a suitable Counter-Latch Command is sent to the 82C54, the latches "latch" the present count until read by the CPU and then return to "following" the CE. One latch at a time is enabled by the Counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read.

Similarly, there are two 8-bit registers called CR_M and CR_L (for "Count Register"). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CR_M and CR_L are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one-byte counts either most signficant byte only), the other byte will be zero. Note that the CE cannot be written into — whenever a count is written, it is written into the CR.

The Control Logic is also shown in the diagram. CLK_n, $GATE_n$, and OUT_n are all connected to the outside world through the Control Logic.

82C54 System Interface

The 82C54 is treated by the systems software as an array of peripheral I/O ports; three are Counters and the fourth is a Control Register for Mode programming (see Figure 4).

Basically, the select inputs, A_0 and A_1, connect to the A_0, A_1 address-bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an AMD 8205, for larger systems.

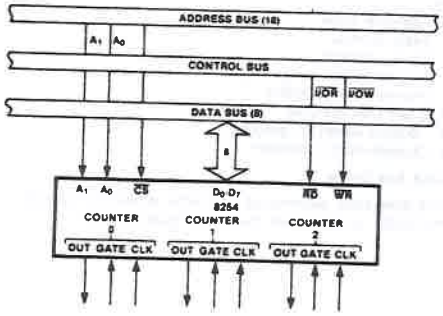


Figure 4. 82C54 System Interface

Operational Description

General

After power-up, the state of the 82C54 is undefined. The mode, count values, and output of all Counters are undefined.

How each Counter operates is determined when it is programmed. Each Counter must be programmed before it can be used. Unused Counters need not be programmed.

Programming the 82C54

Counters are programmed by writing a Control Word and then an initial count. The control-word format is shown in Figure 5.

All Control Words are written into the Control Word Register, which is selected when A_1, $A_0 = 11$. The Control Word itself specifies which Counter is being programmed.

By contrast, initial counts are written into the Counters, not the Control Word Register. The A_1 and A_0 inputs are used to select the Counter to be written into. The format of the initial count is determined by the Control Word used.

Control Word Format

A_1, $A_0 = 11$ $\overline{CS} = 0$ $\overline{RD} = 1$ $\overline{WR} = 0$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (See Read Operations)

RW — Read/Write:

RW1	RW0	
0	0	Counter-Latch Command (see Read Operations).
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16 bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE: Don't care bits (X) should be 0 to insure compatiblity with future AMD products.

Figure 5. Control-Word Format

Write Operations

The programming procedure for the 82C54 is very flexible. Only two conventions need to be remembered:

1) For each Counter, the Control Word must be written before the initial count is written.

2) The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

Since the Control Word Register and the three Counters have separate addresses (selected by the A_1, A_0 inputs), and each Control Word specifies the Counter it applies to (SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the conventions above is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed mode in any way. Counting will be affected as described in the Mode Definitions

section. The new count must follow the programmed count format.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count.

		A_1	A_0
Control Word	— Counter 0	1	1
LSB of count	— Counter 0	0	0
MSB of count	— Counter 0	0	0
Control Word	— Counter 1	1	1
LSB of count	— Counter 1	0	1
MSB of count	— Counter 1	0	1
Control Word	— Counter 2	1	1
LSB of count	— Counter 2	1	0
MSB of count	— Counter 2	1	0

		A_1	A_0
Control Word	— Counter 2	1	1
Control Word	— Counter 1	1	1
Control Word	— Counter 0	1	1
LSB of count	— Counter 2	1	0
MSB of count	— Counter 2	1	0
LSB of count	— Counter 1	0	1
MSB of count	— Counter 1	0	1
LSB of count	— Counter 0	0	0
MSB of count	— Counter 0	0	0

		A_1	A_0
Control Word	— Counter 0	1	1
Counter Word	— Counter 1	1	1
Control Word	— Counter 2	1	1
LSB of count	— Counter 2	1	0
LSB of count	— Counter 1	0	1
LSB of count	— Counter 0	0	0
MSB of count	— Counter 0	0	0
MSB of count	— Counter 1	0	1
MSB of count	— Counter 2	1	0

		A_1	A_0
Control Word	— Counter 1	1	1
Control Word	— Counter 0	1	1
LSB of count	— Counter 1	0	1
Control Word	— Counter 2	1	1
LSB of count	— Counter 0	0	0
MSB of count	— Counter 1	0	1
LSB of count	— Counter 2	1	0
MSB of count	— Counter 0	0	0
MSB of count	— Counter 2	1	0

NOTE: In all four examples, all Counters are programmed to read/write two-byte counts. These are only four of many possible programming sequences.

Figure 6. A Few Possible Programming Sequences

Read Operations

It is often desirable to read the value of a Counter without disturbing the count in progress — this is easily done in the 82C54.

There are three possible methods for reading the Counters: 1) a simple read operation, 2) the Counter-Latch Command, or 3) the Read-Back Command.

The first method is to perform a simple read operation. To read the Counter, which is selected with the A_1, A_0 inputs, the CLK input of the selected Counter must be inhibited by using either the Gate input or external logic. Otherwise the count may be in the process of changing when it is read, giving an undefined result.

Counter-Latch Command

The second method uses the "Counter-Latch Command." Like a Control Word, this command is written to the Control Word Register which is selected when A_1, A_0 = 11. Also like a Control Word, the SC0, SC1 bits select one of the three Counters, but two other bits, D_5 and D_4, distinguish this command from a Control Word.

A_1, A_0 = 11; $\overline{CS}$ = 0; $\overline{RD}$ = 1; $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1, SC0 – specify counter to be latched

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Read-Back-Command

D_5, D_4 – 00 designates Counter-Latch Command
X – Don't Care

NOTE: Don't care bits (X) should be 0 to insure compatibility with future AMD products.

Figure 7. Counter-Latching Command Format

The selected Counter's "Output Latch" (OL) latches the count at the time the Counter-Latch Command is received. This count is held in the latch until it is read by the CPU (or until the Counter is re-programmed). The count is then unlatched automatically and the OL return to "following" the Counting Element (CE). This allows reading the contents of the Counters "on the fly" without affecting counting in progress. Multiple Counter-Latch Commands may be used to latch more than one Counter. Each latched Counter's OL holds its count until it is read. Counter-Latch Commands do not affect the programmed mode of the Counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter-Latch Command is ignored. The count read will be the count at the time the first Counter-Latch Command was issued.

With either method, the count must be read according to the programmed format; specifically, if the Counter is programmed for two-byte counts, two bytes must be read. The two bytes do not have to be read one right after the other; read, write, or programming operations of other Counters may be inserted between them.

Another feature of the 82C54 is that reads and writes of the same Counter may be interleaved; for example, if the Counter is programmed for two-byte counts, the following sequence is valid:

1) Read least significant byte,
2) Write new least significant byte,
3) Read most significant byte,
4) Write new most significant byte.

If a Counter is programmed to read/write two-byte counts, the following precaution applies: A program must not transfer control between reading the first and second byte to another routine which also reads from that same Counter. Otherwise, an incorrect count may be read.

Read-Back Command

The third method uses the Read-Back Command. This command allows the user to check the count value, programmed mode, and current state of the Out pin and Null-Count Flag of the selected Counter(s).

The command is written into the Control Word Register and has the format shown in Figure 8. The command applies to the Counters selected by setting their corresponding bits D_3, D_2, D_1 = 1.

A_0, A_1 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counter(s)
D_3: 1 = Select Counter 2
D_2: 1 = Select Counter 1
D_1: 1 = Select Counter 0
D_0: Reserved for future expansion; must be 0

Figure 8. Read-Back Command Format

The Read-Back Command may be used to latch multiple Counter Output Latches (OL) by setting the Count bit D_5 = 0 and selecting the desired Counter(s). This single command is functionally equivalent to several Counter-Latch Commands, one for each Counter latched. Each Counter's latched count is held until it is read (or the Counter is reprogrammed). That Counter is automatically unlatched when read, but other Counters remain latched until they are read. If multiple-count Read-Back Commands are issued to the same Counter without reading the count, all but the first are ignored. In other words, the count which will be read is the count at the time the first Read-Back Command was issued.

The Read-Back Command may also be used to latch status information of selected Counter(s) by setting Status bit D_4 = 0. Status must be latched to be read; status of a Counter is accessed by a read from that Counter.

The Counter status format is shown in Figure 9. Bits D_5 through D_0 contain the Counter's programmed mode exactly as written in the last mode Control Word. Output bit D_7 contains the current state of the Out pin. This allows the user to monitor the Counter's output via software, possibly eliminating some hardware from a system.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

D_7 1 = Out Pin is 1
0 = Out Pin is 0
D_6 1 = Null count
0 = Count available for reading
D_5 – D_0 = Counter Programmed Mode (See Figure 5)

Figure 9. Status Byte

Null-Count bit D_6 indicates when the last count written to the Counter Register (CR) has been loaded into the Counting Element (CE). The exact time this happens depends on the mode of the Counter and is described in the Mode Definitions section, but until the count is loaded into the CE, it cannot be read from the Counter. If the count is latched or read before this time, the count value will not reflect the new count just written. The operation of Null Count is shown in Figure 10.

THIS ACTION:	CAUSES:
A. Write to the Control Word Register:[1]	Null count = 1
B. Write to the Count Register (CR);[2]	Null count = 1
C. New count is loaded into CE (CR → CE);	Null count = 0

[1] Only the Counter specified by the Control Word will have its null count set to 1. Null count bits of other counters are unaffected.

[2] If the Counter is programmed for two-byte counts (least significant byte then most significant byte) null count goes to 1 when the second byte is written.

Figure 10. Null Count Operation

If multiple Status-Latch operations of the Counter(s) are performed without reading the status, all but the first are ignored. In other words, the status that will be read is the status of the Counter at the time the first Status Read-Back Command was issued.

Both count and status of the selected Counter(s) may be latched simultaneously by setting both Count and Status bits D_5, D_4 = 0. This is functionally the same as issuing two separate Read-Back Commands at once, and the above

discussions apply here also. Specifically, if multiple count and/or status Read-Back Commands are issued to the same Counters) without any intervening reads, all but the first are ignored. This is illustrated in Figure 11.

If both count and status of a Counter are latched, the first read operation of that Counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the Counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

Command D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Description	Results
1	1	0	0	0	0	1	0	Read-back count and status of Counter 0	Count and status latched for Counter 0
1	1	1	0	0	1	0	0	Read-back status of Counter 1	Status latched for Counter 1
1	1	1	0	1	1	0	0	Read-back status of Counters 2, 1	Status latched for Counter 2, but not Counter 1
1	1	0	1	1	0	0	0	Read-back count of Counter 2	Count latched for Counter 2
1	1	0	0	0	1	0	0	Read-back count and status of Counter 1	Count latched for Counter 1, but not status
1	1	1	0	0	0	1	0	Read-back status of Counter 1	Command ignored, status already latched for Counter 1

Figure 11. Read-Back Command Example

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Write into Counter 0
0	1	0	0	1	Write into Counter 1
0	1	0	1	0	Write into Counter 2
0	1	0	1	1	Write Control Word
0	0	1	0	0	Read from Counter 0
0	0	1	0	1	Read from Counter 1
0	0	1	1	0	Read from Counter 2
0	0	1	1	1	No-Operation (3-State)
1	X	X	X	X	No-Operation (3-State)
0	1	1	X	X	No-Operation (3-State)

Figure 12. Read/Write Operations Summary

Mode Definitions

The following are defined for use in describing the operation of the 82C54:

CLK Pulse = a rising edge, then a falling edge — in that order — of a Counter's CLK input.

Trigger = a rising edge of a Counter's GATE input.

Counter Loading = the transfer of a count from the CR to the CE (refer to the Functional Description section).

Mode 0: Interrupt on Terminal Count

Mode 0 is typically used for event counting. After the Control Word is written, Out is initially LOW and will remain LOW until the Counter reaches zero. Out then goes HIGH and remains HIGH until a new count or a new Mode 0 Control Word is written into the Counter.

Gate = 1 enables counting; Gate = 0 disables counting. Gate has no effect on Out.

After the Control Word and initial count are written to a Counter, the initial count will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, Out does not go HIGH until N + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte disables counting. Out is set LOW immediately (no CLK pulse required).

2) Writing the second byte allows the new count to be loaded on the next CLK pulse. This allows the counting sequence to be synchronized by software. Again, Out does not go HIGH until N + 1 CLK pulses after the new count of N is written.

If an initial count is written while Gate = 0, it will still be loaded on the next CLK pulse. When Gate goes HIGH, Out will go HIGH N CLK pulses later; no CLK pulse is needed to load the Counter, as this has already been done.

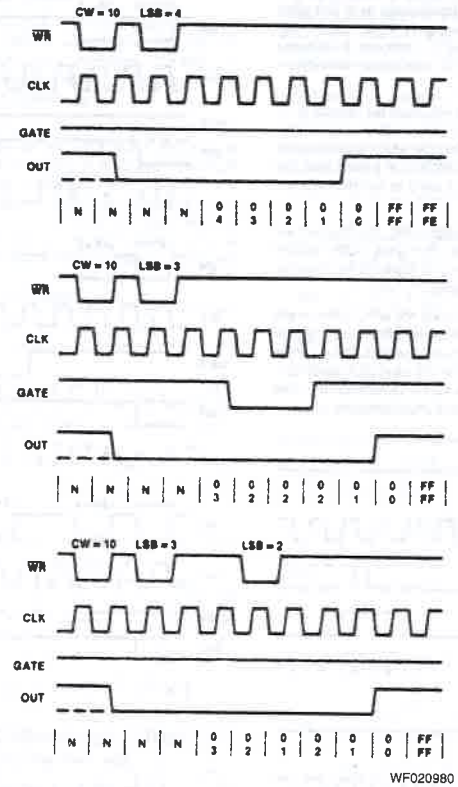


Figure 13. Mode 0

Notes: The following conventions apply to all Mode-Timing Diagrams:

1. Counters are being programmed for binary (not BCD) counting and for reading/writing least significant byte (LSB) only.
2. The Counter is always selected ($\overline{CS}$ always LOW).
3. CW stands for "Control Word;" CW = 10 means a control word of 10, hex, is written to the Counter.
4. LSB stands for "Least Significant Byte" of count.
5. Numbers below diagrams are count values. The lower number is the LSB; the upper number is the MSB. Since the Counter is programmed to read/write LSB only, the MSB cannot be read. "N" stands for an undefined count; vertical lines show transitions between count values.

Mode 1: Hardware Retriggerable One-Shot

Out will be initially HIGH. Out will go LOW on the CLK pulse following a trigger to begin the one-shot pulse, and will remain LOW until the Counter reaches zero. Out will then go HIGH and remain HIGH until the CLK pulse after the next trigger.

After writing the Control Word and initial count, the Counter is armed. A trigger results in loading the Counter and setting Out LOW on the next CLK pulse, thus starting the one-shot pulse. An initial count of N will result in a one-shot pulse N CLK cycles in duration. The one-shot is retriggerable, hence, Out will remain LOW for N CLK pulses after any trigger. The one-shot pulse can be repeated without rewriting the same count into the Counter. Gate has no effect on Out.

If a new count is written to the Counter during a one-shot pulse, the current one-shot is not affected unless the Counter is retriggered. In that case, the Counter is loaded with the new count and the one-shot pulse continues until the new count expires.

Mode 2: Rate Generator

This mode functions like a divide-by-N Counter. It is typically used to generate a real-time clock interrupt. Out will initially be

HIGH. When the initial count has decremented to 1, Out goes LOW for one CLK pulse. Out then goes HIGH again, the Counter reloads the initial count and the process is repeated indefinitely. For an initial count of N, the sequence repeats every N CLK cycles.

Gate = 1 enables counting; Gate = 0 disables counting. If Gate goes LOW during an output pulse, Out is set HIGH immediately. A trigger reloads the Counter with the initial count on the next CLK pulse; Out goes LOW N CLK pulses after the trigger. Thus, the Gate input can be used to synchronize the Counter.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. Out goes LOW N CLK pulses after the initial count is written. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count, but before the end of the current period, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current counting cycle. In Mode 2, a count of 1 is illegal.

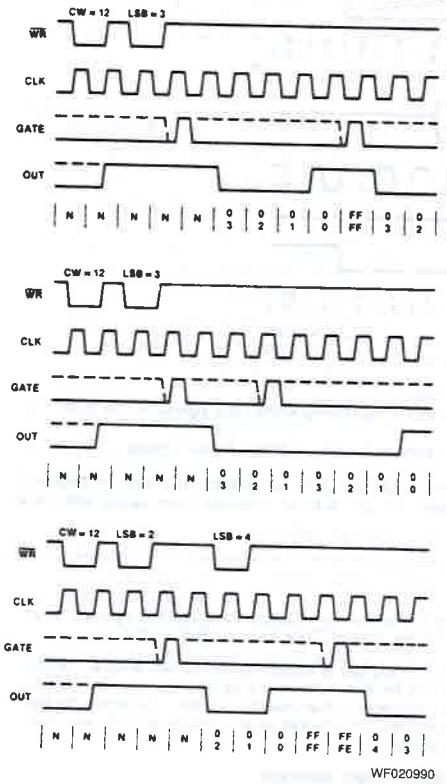
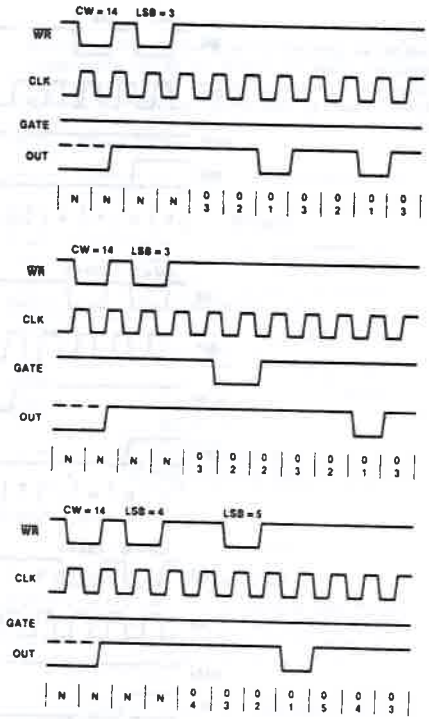


Figure 14. Mode 1



NOTE: A Gate transition should not occur one clock prior to terminal count.

Figure 15. Mode 2

Mode 3: Square-Wave Mode

Mode 3 is typically used for baud-rate generation. Mode 3 is similar to Mode 2 except for the duty cycle of Out. Out will initially be HIGH. When half the initial count has expired, Out goes LOW for the remainder of the count. Mode 3 is periodic; the sequence above is repeated indefinitely. An initial count of N results in a square wave with a period of N CLK cycles.

Gate = 1 enables counting; Gate = 0 disables counting. If Gate goes LOW while Out is LOW, Out is set HIGH immediately; no CLK pulse is required. A trigger reloads the Counter with the initial count on the next CLK pulse. Thus, the Gate input can be used to synchronize the Counter.

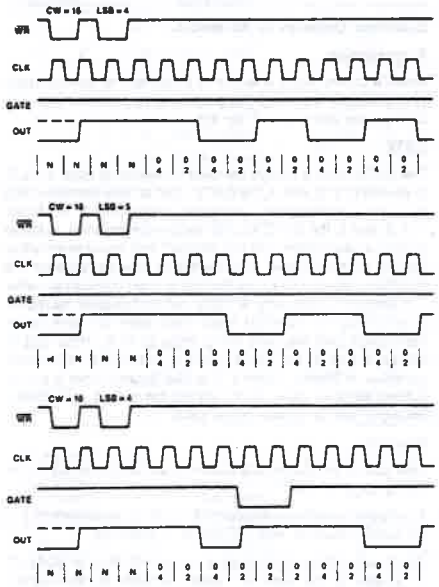
After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This allows the Counter to be synchronized by software also.

Writing a new count while counting does not affect the current counting sequence. If a trigger is received after writing a new count, but before the end of the current half-cycle of the square wave, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from the new count. Otherwise, the new count will be loaded at the end of the current half-cycle.

Mode 3 is implemented as follows:

Even counts: Out is initially HIGH. The initial count is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. When the count expires, Out changes value and the Counter is reloaded with the initial count. The above process is repeated indefinitely.

Odd counts: Out is initially HIGH. The initial count minus one (an even number) is loaded on one CLK pulse and then is decremented by two on succeeding CLK pulses. One CLK pulse after the count expires, Out goes LOW and the Counter is reloaded with the initial count minus one. Succeeding CLK pulses decrement the count by two. When the count expires, Out goes HIGH again and the Counter is reloaded with the initial count minus one. The above process is repeated indefinitely. For odd counts, Out will be HIGH for (N + 1)/2 counts and LOW for (N – 1)/2 counts.



NOTE: A Gate transition should not occur one clock prior to terminal count.

Figure 16. Mode 3

Mode 4: Software-Triggered Strobe

Out will be initially HIGH. When the initial count expires, Out will go LOW for one CLK pulse and then go HIGH again. The counting sequence is 'triggered' by writing the initial count.

Gate = 1 enables counting; Gate = 0 disables counting. Gate has no effect on Out.

After writing a Control Word and initial count, the Counter will be loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of N, Out does not strobe LOW until N + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it will be loaded on the next CLK pulse and counting will continue from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting,
2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be "retriggered" by software. Out strobes LOW N + 1 CLK pulses after the new count of N is written.

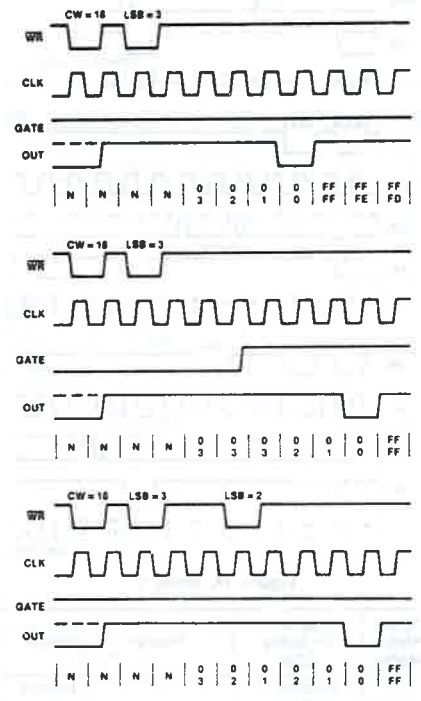


Figure 17. Mode 4

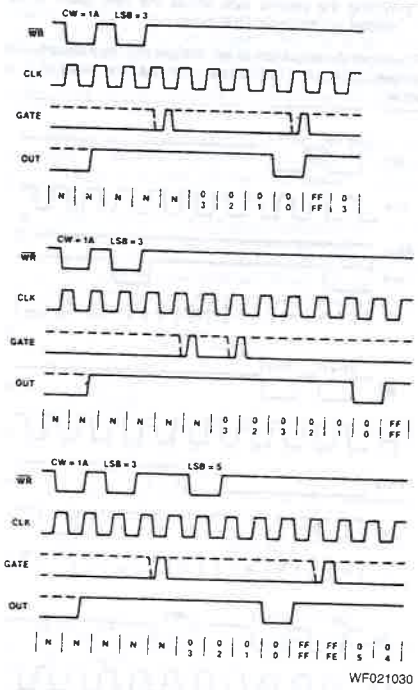
Mode 5: Hardware-Triggered Strobe (Retriggerable)

Out will initially be HIGH. Counting is triggered by a rising edge of Gate. When the initial count has expired, Out will go LOW for one CLK pulse and then go HIGH again.

After writing the Control Word and initial count, the Counter will not be loaded until the CLK pulse after a trigger. This CLK pulse does not decrement the count, so for an initial count of N, Out does not strobe LOW until N + 1 CLK pulses after a trigger.

A trigger results in the Counter being loaded with the initial count on the next CLK pulse. The counting sequence is retriggerable. Out will not strobe LOW for N + 1 CLK pulses after any trigger. Gate has no effect on Out.

If a new count is written during counting, the current counting sequence will not be affected. If a trigger occurs after the new count is written, but before the current count expires, the Counter will be loaded with the new count on the next CLK pulse and counting will continue from there.



WF021030

Figure 18. Mode 5

Signal Status Modes	LOW Or Going LOW	Rising	HIGH
0	Disables counting	—	Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2, 3	1) Disables counting 2) Sets output immediately HIGH	Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 19. Gate-Pin Operations Summary

Mode	Min. Count	Max. Count
0	1	0
1	1	0
2	2	0
3	2	0
4	1	0
5	1	0

NOTE: 0 is equivalent to 2^{16} for binary counting and 10^4 for BCD counting

Figure 20. Minimum and Maximum Initial Counts

Operation Common to All Modes

Programming

When a Control Word is written to a Counter, all Control Logic is immediately reset and Out goes to a known initial state; no CLK pulses are required for this.

GATE

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3, and 4, the GATE input is level-sensitive, and the logic level is sampled on the rising edge of CLK. In Modes 1, 2, 3, and 5, the GATE input is rising-edge sensitive. In these modes, a rising edge of GATE (trigger) sets an edge-sensitive flip-flop to the Counter. This flip-flop is then sampled on the next rising edge of CLK; the flip-flop is reset immediately after it is sampled. In this way, a trigger will be detected no matter when it occurs — a HIGH logic level does not have to be maintained until the next rising edge of CLK. Note that in Modes 2 and 3, the GATE input is both edge- and level-sensitive. In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

Counter

New counts are loaded and Counters are decremented on the falling edge of CLK.

The largest possible initial count is 0; this is equivalent to 2^{16} for binary counting and 10^4 for BCD counting.

The Counter does not stop when it reaches zero. In Modes 0, 1, 4, and 5, the Counter "wraps around" to the highest count — either FFFF hex for binary counting or 9999 for BCD counting — and continues counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to GND ..... −0.5 to +7.0 V
Power Dissipation ..... 1 W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... +4.5 to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

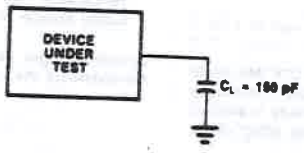
Parameter Symbol	Parameter Description	Test Conditions			Min.	Max.	Units
V_{IL}	Input LOW Voltage			C Devices	−0.5	0.8	V
V_{IH}	Input HIGH Voltage			C Devices	2.0	V_{CC}+0.5 V	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0 mA				.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = −400 µA			2.4		V
I_{IL}	Input Load Current	V_{IN} = V_{CC} to 0 V				±10	µA
I_{OFL}	Output Float Leakage Current	V_{OUT} = V_{CC} to 0.45 V				±10	µA
I_{CC}	Operating Power-Supply Current	CLK Freq =	8 MHz	C Devices		20	mA
			10 MHz	C Devices		20	
			12.5 MHz	C Devices		20	
I_{CCSB}	Standby Power-Supply Current	CLK Freq = DC, $\overline{CS}$ = HIGH, All Inputs/Data Bus HIGH, All Outputs Floating		C Devices		10	µA
I_{CCSB1}	V_{CC} Standby Power-Supply Current	CLK Freq = DC, $\overline{CS}$ = HIGH, All Other Inputs, Outputs, I/O Plus Floating				150	µA

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0 V)

Parameter Symbol	Parameter Description	Test Conditions		Min.	Max.	Units
C_{IN}	Input Capacitance	fc = 1 MHZ Unmeasured pins returned to GND	C Devices		10	pF
$C_{I/O}$	I/O Capacitance		C Devices		20	pF
C_{OUT}	Output Capacitance		C Devices		20	pF

See Section 6 for Thermal Characteristics Information.

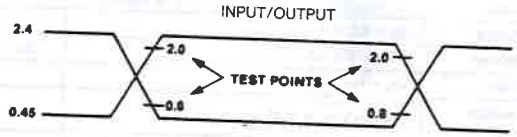
SWITCHING TEST CIRCUIT



TC003430

C_L = 150 pF
C_L includes jig capacitance

SWITCHING TEST WAVEFORM



WF021040

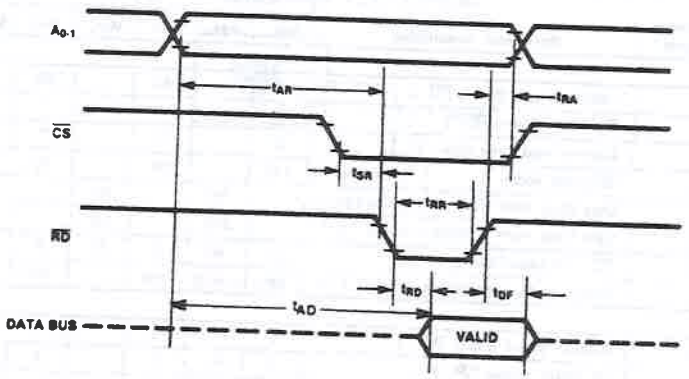
A.C. Testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0." Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for a logic "0."

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

No.	Parameter Symbol	Parameter Description		8 MHz Min.	8 MHz Max.	10 MHz Min.	10 MHz Max.	12.5 MHz Min.	12.5 MHz Max.	Units
Read Cycle										
1	t_{AR}	Address Stable Before $\overline{RD}$ ↓		45		30		25		ns
2	t_{SR}	$\overline{CS}$ Stable Before $\overline{RD}$ ↓		0		0		0		ns
3	t_{RA}	Address Hold Time After $\overline{RD}$ ↑		0		0		0		ns
4	t_{RR}	$\overline{RD}$ Pulse Width	C Devices	150		95		80		ns
5	t_{RD}	Data Delay from $\overline{RD}$ ↓	C Devices		120		85		70	ns
6	t_{AD}	Data Delay from Address			220		185		150	ns
7	t_{DF}	$\overline{RD}$ ↑ to Data Floating		5	90	5	65	5	55	ns
8	t_{RV}	Command Recovery Time		200		165		135		ns
Write Cycle										
9	t_{AW}	Address Stable Before $\overline{WR}$ ↓		0		0		0		ns
10	t_{SW}	$\overline{CS}$ Stable Before $\overline{WR}$ ↓		0		0		0		ns
11	t_{WA}	Address Hold Time After $\overline{WR}$ ↑		0		0		0		ns
12	t_{WW}	$\overline{WR}$ Pulse Width		150		95		80		ns
13	t_{DW}	Data Setup Time Before $\overline{WR}$ ↑		120		95		80		ns
14	t_{WD}	Data Hold Time After $\overline{WR}$ ↑		0		0		0		ns
15	t_{RV}	Command Recovery Time		200		165		135		ns
Clock and Gate Cycle										
16	t_{CLK}	Clock Period		125	DC	100	DC	80	DC	ns
17	t_{PWH}	HIGH Pulse Width (Note 3)	C Devices	60		30		25		ns
18	t_{PWL}	LOW Pulse Width (Note 3)		60		50		40		ns
19	t_R	Clock Rise Time			25		25		25	ns
20	t_F	Clock Fall Time			25		25		25	ns
21	t_{GW}	Gate Width HIGH		50		50		40		ns
22	t_{GL}	Gate Width LOW		50		50		40		ns
23	t_{GS}	Gate Setup Time to CLK ↑		50		40		30		ns
24	t_{GH}	Gate Hold Time After CLK ↑ (Note 2)		50		50		40		ns
25	t_{OD}	Output Delay from CLK ↓			150		100		80	ns
26	t_{ODG}	Output Delay from Gate ↓			120		100		80	ns
27	t_{WC}	CLK Delay for Loading		0	55	0	55	0	45	ns
28	t_{WG}	Gate Delay for Sampling		−5	50	−5	40	−5	35	ns
29	t_{WO}	Out Delay from Mode Write			260		240		200	ns
30	t_{CL}	CLK Set Up for Count Latch		−4	45	−4	40	−4	35	ns

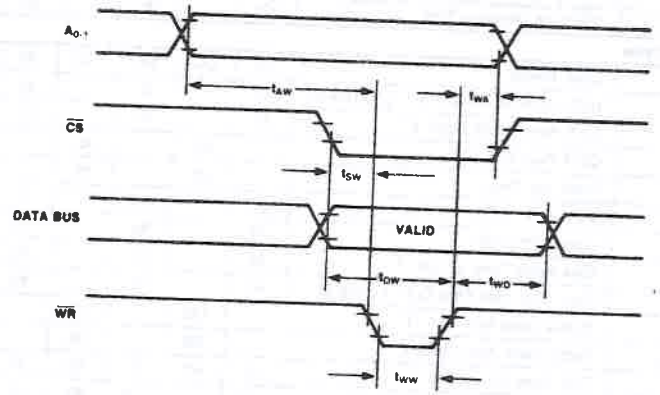
Notes: 1. Timings measured at V_{OH} = 2.0 V, V_{OL} = 0.8 V.
2. In Mode 1 and 5, triggers are sampled on each rising clock edge. A second trigger within 120 ns (70 ns for the 82C54-2) of the rising clock edge may not be detected.
3. LOW-going glitches that violate t_{PWH}, t_{PWL} may cause errors requiring Counter re-programming.

SWITCHING WAVEFORMS



WF021051

Read Cycle



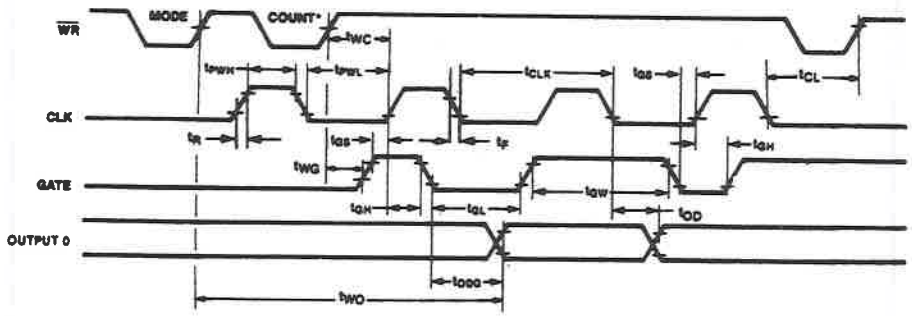
WF021061

Write Cycle

SWITCHING WAVEFORMS (Cont'd.)

WF021070

Recovery Cycle



WF021080

*Last byte of count being written

Clock and Gate Cycle

NOTE:

WE MANUFACTURE....

Data Acquisition and Measurement Products
Industrial Automation & Control Products
Multiport RS232/422/485 Serial Cards

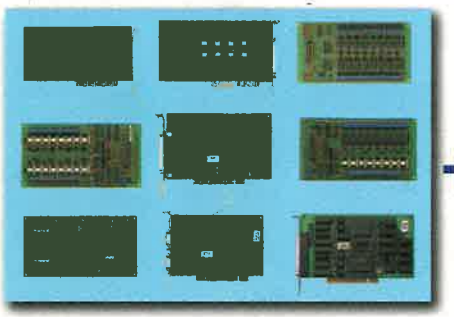
Via Internet & World Wide Web (WWW)
Remote Control & Communication Function

15 Years of Professional Experience for PC-Base Data Acquisition and Serial Communication Products.

Worldwide's
No.1 in Remote Control Technology!

WE MANUFACTURE....

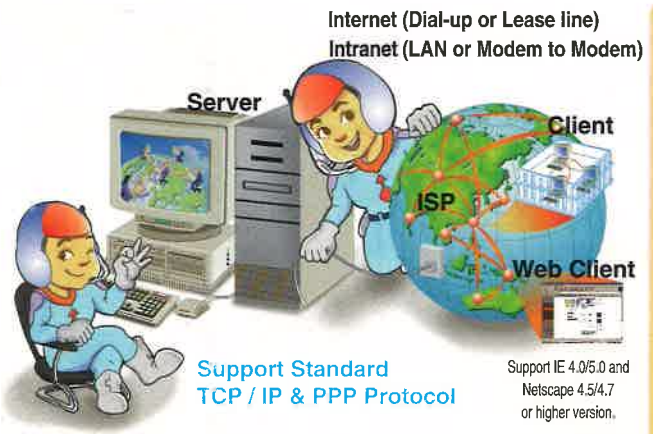
Data Acquisition and Measurement Products
Industrial Automation & Control Products
Multiport RS232/422/485 Serial Cards



We Provide Software Development Tool Kits

Under Window 95 / 98 / NT

For Internet, Intranet & World Wide Web (WWW) Remote Control & Communication Function



Support Standard TCP / IP & PPP Protocol

Support IE 4.0/5.0 and Netscape 4.5/4.7 or higher version.

8 Major Functions

1. Remote Sound
2. Chat
3. Remote CCD Camera
4. Internet Location Server
5. Remote Digital I/O
6. Remote AD/DA
7. Remote Serial Communication
8. Remote Telecommunication function