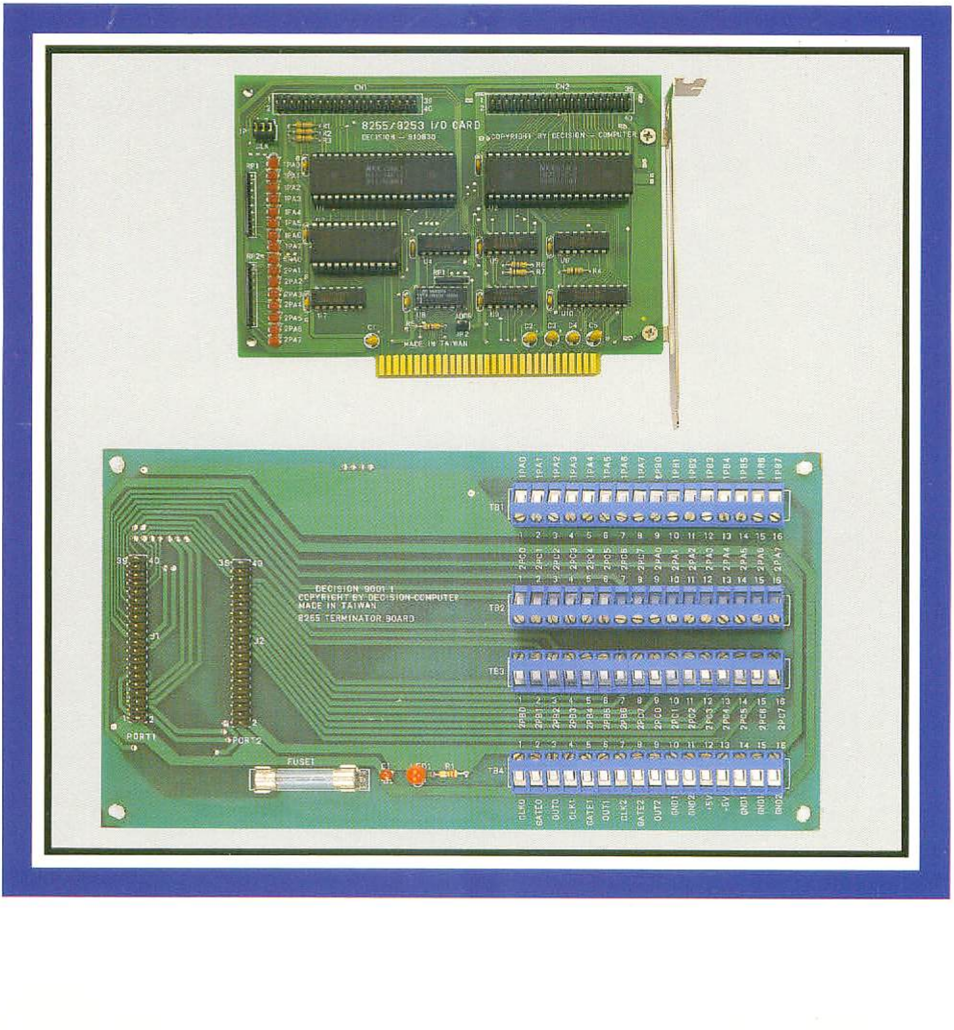
DCI SmartLab
8255/8253 I/O CARD
AND TERMINATOR BOARD
OPERATION MANUAL



DCI SmartLab.
8255/8253 I/O CARD
AND TERMINATOR BOARD
OPERATION MANUAL

CONTENTS:

Rev. Date: July, 01, 1990

CHAPTER 1 INTRODUCTION

The 8255 I/O card is a programmable peripheral interface for PC/XT. PC/AT, 80386, 80486 or compatible. The interface card contains 8253 chip and 8255 chips, the 8253 chip provides programmable interval timer/counter functions and the 8255 chips provide programmable input/output functions.

The features of the 8255 I/O Card are:

* Programmable I/O control functions.
* Up to 48 I/O lines.
* Maximum of 2MHZ count rate.
* Three independent 16 bits counter.
* Support several operating modes which are programmable.
* Sixteen LED indicate when I/O is operating.
* Port address selectable.

Package contents:

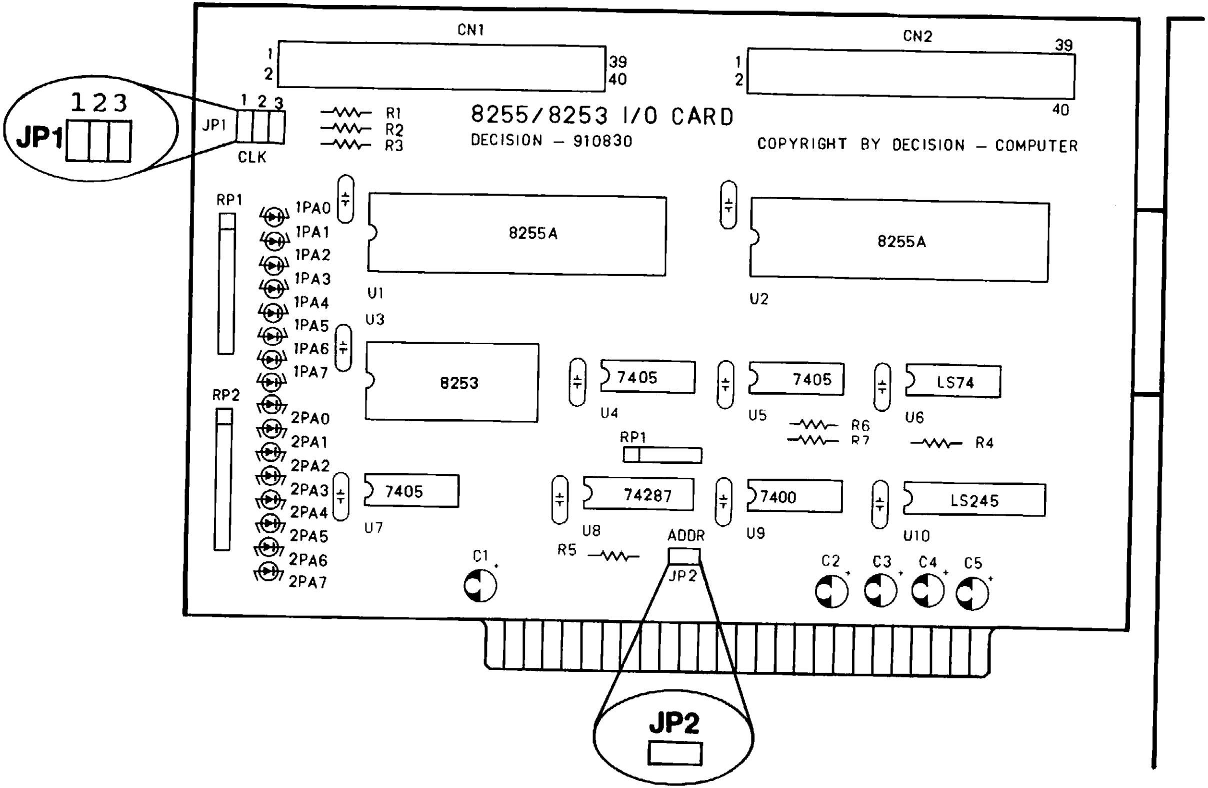
* 8255/8253 I/O card.
* 8255/8253 user's manual.
* 8255/8253 terminator board (option).
* Two expansion flat cables (option).

CHAPTER 2 HARDWARE CONFIGURATION

2.1 Configuration for DIP Switch

Before you use the 8255 I/O card, you must ensure that the I/O address and the clock are set correctly. Observe the figure belows the proper settings for the 8255 I/O card are described in the following:

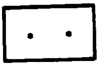
DECISION-COMPUTER International Co., Ltd.



1. I/O address

JP2: Short (default)

JP2



Select &H300 - &H30F as I/O port address. The I/O address correspond to three 8 bits ports and three counters are:

&H300: Port 1A input/output buffer.

&H301: Port 1B input/output buffer.

&H302: Port 1C input/output buffer.

&H303: Port 1 control register.

&H304: Port 2A input/output buffer.

&H305: Port 2B input/output buffer.

&H306: Port 2C input/output buffer.

&H307: Port 2 control register.

&H308: Counter 0 input/output buffer.

&H309: Counter 1 input/output buffer.

&H30A: Counter 2 input/output buffer.

&H30B: Counter control register.

I/O Address

JP2: Open

JP2 • •

Select &H360 - &H36F as I/O port address. The I/O address correspond to three 8 bits ports and three counters are:

&H360: Port 1A input/output buffer.

&H361: Port 1B input/output buffer.

&H362: Port 1C input/output buffer.

&H363: Port 1 control register.

&H364: Port 2A input/output buffer.

&H365: Port 2B input/output buffer.

&H366: Port 2C input/output buffer.

&H367: Port 2 control register.

&H368: Counter 0 input/output buffer.

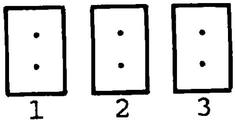
&H369: Counter 1 input/output buffer.

&H36A: Counter 2 input/output buffer.

&H36B: Counter control register.

2. Clock Selection

JP1



JP1-1 = Short (default) :

Select internal clock to counter 0.

JP1-2 = Short (default) :

Select internal clock to counter 1.

JP1-3 = Short (default) :

Select internal clock to counter 2.

JP1-1 = Open :

Select external clock to counter 0.

JP1-2 = Open :

Select external clock to counter 1.

JP1-3 = Open :

Select external clock to counter 2.

2.2 Hardware Installation

Your 8255 I/O card is designed to be inserted in any available slot in your PC/XT or compatibles. In order to gain access to the expansion slots and the program switches on the mainboard, follow the steps listed below:

1. Set the 8255 I/O card switch.
2. Turn off all power of your computer and all peripheral devices before installing your 8255 I/O card.
3. Remove the cover of the computer.
4. Insert your preconfigured card into any available slot. Make sure your I/O card is firmly seated in the chosen slot.
5. Replace the cover of the computer.
6. You are now ready to use your 8255 I/O card for several applications.

2.3 Pin Assignment

Port 1

PIN	1	GND	PIN	2	GND
	3	GND		4	PA3
	5	PA1		6	PA2
	7	CLK0		8	PA0
	9	GATE0		10	OUT0
	11	OUT2		12	CLK2
	13	CLK1		14	GATE2
	15	OUT1		16	GATE1
	17	PA5		18	PA4
	19	PA7		20	PA6
	21	PC6		22	PC7
	23	PC4		24	PC5
	25	PC1		26	PC0
	27	PB7		28	PC2
	29	PB6		30	PC3
	31	PB5		32	PB0
	33	PB4		34	PB1
	35	PB3		36	PB2
	37	+5V		38	-5V
	39	+12V		40	-12V

Pin Assignment

Port 2

PIN	1	GND	PIN	2	GND
	3	GND		4	GND
	5	GND		6	GND
	7	GND		8	GND
	9	GND		10	GND
	11	GND		12	GND
	13	PA0		14	PA1
	15	PA2		16	PA3
	17	PA4		18	PA5
	19	PA6		20	PA7
	21	PC7		22	PC6
	23	PC5		24	PC4
	25	PC0		26	PC1
	27	PC2		28	PB7
	29	PC3		30	PB6
	31	PB0		32	PB5
	33	PB1		34	PB4
	35	PB2		36	PB3
	37	+5V		38	-5V
	39	+12V		40	-12V

CHAPTER 3 DIAGNOSTIC

1. BASIC Version

```
100 REM 8255 I/O CARD TESTING PROGRAM
110 SCREEN 0,0,0: WIDTH 80,25: KEY OFF: CLS
120 LOCATE 10,10: PRINT "8255 I/O CARD TESTING"
130 LOCATE 12,10: PRINT "TWO 8255 PORT A,B,C OUTPUT SQUARE WAVE"
140 LOCATE 14,10: PRINT "8253 COUNTER 0 DIVIDE BY 2"
150 LOCATE 16,10: PRINT "COUNTER 1 DIVIDE BY 50"
160 LOCATE 18,10: PRINT "COUNTER 2 DIVIDE BY 100"
250 REM 8253 TESTING
255 PORT = &H300
260 OUT PORT+11,&H36
270 OUT PORT+11,&H76
280 OUT PORT+11,&HB6
290 OUT PORT+8,&H2: OUT PORT+8,&H0
300 OUT PORT+9,&H32: OUT PORT+9,&H0
310 OUT PORT+10,&H64: OUT PORT+10,&H0
320 PORT=&H300
330 OUT PORT+3,&H80
340 A=0: GOSUB 500
345 FOR K = 0 TO 1000: NEXT K
350 A=&HFF: GOSUB 500
360 PORT=PORT+4
370 OUT PORT+3,&H80
380 A=0: GOSUB 500
385 FOR K=0 TO 1000: NEXT K
390 A=&HFF: GOSUB 500
400 GOTO 320
500 FOR I=0 TO 2
510 OUT PORT+I,A
520 NEXT I
530 RETURN
```

2. PASCAL Version

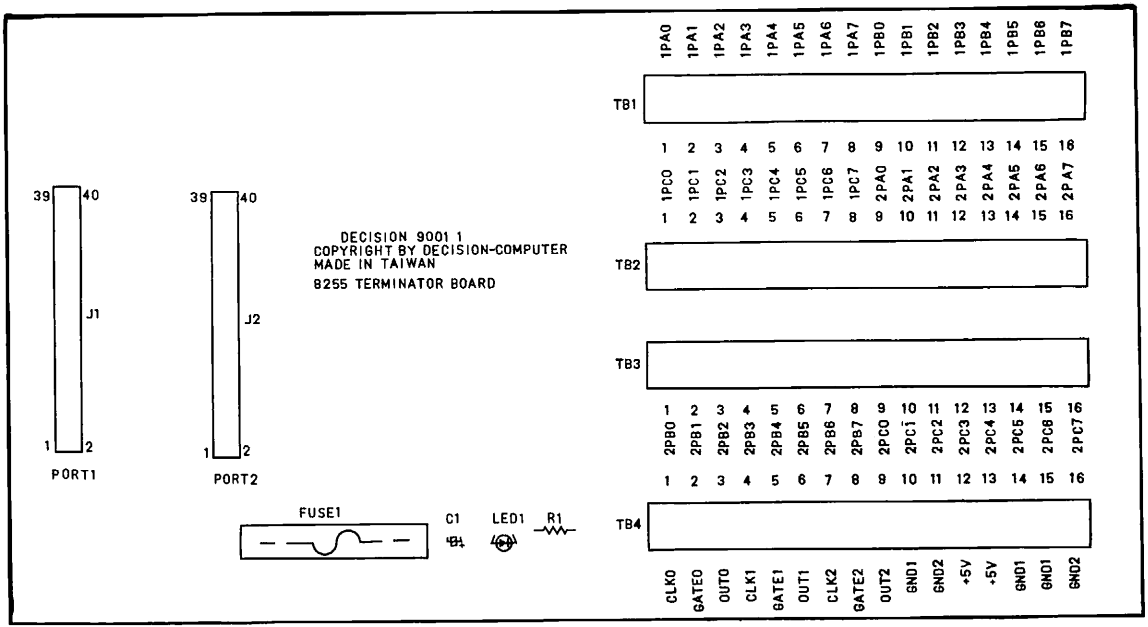
```
program diagnostic( input, output );
uses
          Crt;

var
          a, i, test : integer;

procedure subtest;
begin
          for i := 0 to 2 do
                    port[test+i] := a;
end;

begin
          { 8255 I/O Card Testing Program )
          clrscr;
          gotoxy(10,10);
          writeln('8255 I/O CARD TESTING');
          gotoxy(10,12);
          writeln('TWO 8255 PORT A, B, C OUTPUT SQUARE WAVE');
          gotoxy(10,14);
          writeln('8253 COUNTER 0 DIVIDE BY 2');
          gotoxy(10,16);
          writeln('8253 COUNTER 1 DIVIDE BY 50');
          gotoxy(10,18);
          writeln('8253 COUNTER 2 DIVIDE BY 100');

          { 8253 Testing }

          test := $300;
          port[test+11] := $36;
          port[test+11] := $76;
          port[test+11] := $B6;
          port[test+8] := $02;     port[test+8] := $0;
          port[test+9] := $32;   port[test+9] := $0;
          port[test+10] := $64;  port[test+10] := $0;

          repeat
                    test := $300;
                    port[test+3] := $80;
                    a := 0;                   subtest;
                    for  i := 0  to 1000 do;
                    a := $ff;     subtest;
                    test := test + 4;
                    port[test+3] := $80;
                    a := 0;                   subtest;
                    for i := 0 to 1000 do;
                    a := $ff;     subtest;
          until keypressed;
end.
```

3. C version

```
#include <stdio.h>
#include <conio.h>

int test,i,a;

void subtest()
{
              for(i=0;i<=2;i++) outportb(test+i,a);
}
main(){
              clrscr();
              gotoxy(10,10);
              puts(“8255 I/O CARD TESTING”);
              gotoxy(10,12);
              puts(“TWO 8255 PORT A,B,C OUTPUT SQUARE WAVE”);
              gotoxy(10,14);
              puts(“8253 COUNTER  0 DIVIDE BY 2”);
              gotoxy(10,16);
              puts(“8253 COUNTER  1 DIVIDE BY 50”);
              gotoxy(10,18);
              puts(“8253 COUNTER  222 DIVIDE BY 100”);

              test = 0x300;
              outportb(test+11,0x36);
              outportb(test+11,0x76);
              outportb(test+11,0xb6);
              outportb(testt+8,0x02);
              outportb(test+8,0x00);
              outportb(test+9,0x32);
              outportb(test+9,0x00);
              outportb(test+10,0x64);
              outportb(test+10,0x00);

              do{
                            test = 0x300;
                            outportb(test+3,0x80);
                            a = 0;
                            subtest();
                            for(i=0;i<=1000;i++);
                            a = 0xff;
                            subtest();
              }while(!kbhit());
 }
```

APPENDIX A TERMINATOR BOARD

The 8255 terminator board provides expansion signal connection for convenience purpose. When power control on the normal condition, the LED will light. The layout of 8255 terminator board is shown in the follows.


1PA0
1PA1
1PA2
1PA3
1PA4
1PA5
1PA6
1PA7
1PB0
1PB1
1PB2
1PB3
1PB4
1PB5
1PB6
1PB7
TB1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
1PC0
1PC1
1PC2
1PC3
1PC4
1PC5
1PC6
1PC7
2PA0
2PA1
2PA2
2PA3
2PA4
2PA5
2PA6
2PA7
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
TB2
TB3
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
2PB0
2PB1
2PB2
2PB3
2PB4
2PB5
2PB6
2PB7
2PC0
2PC1
2PC2
2PC3
2PC4
2PC5
2PC6
2PC7
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
TB4
CLK0
GATE0
OUT0
CLK1
GATE1
OUT1
CLK2
GATE2
OUT2
GND1
GND2
+5V
+5V
GND1
GND1
GND2
39
40
J1
1
2
PORT1
39
40
J2
1
2
PORT2
DECISION 9001 1
COPYRIGHT BY DECISION-COMPUTER
MADE IN TAIWAN
8255 TERMINATOR BOARD
FUSE1
C1
LED1
R1

The signal assignment is shown in the follows.

1. TB1

PORT 1	PA0
PORT 1	PA1
PORT 1	PA2
PORT 1	PA3
PORT 1	PA4
PORT 1	PA5
PORT 1	PA6
PORT 1	PA7
PORT 1	PB0
PORT 1	PB1
PORT 1	PB2
PORT 1	PB3
PORT 1	PB4
PORT 1	PB5
PORT 1	PB6
PORT 1	PB7

2. TB2

PORT 1	PC0
PORT 1	PC1
PORT 1	PC2
PORT 1	PC3
PORT 1	PC4
PORT 1	PC5
PORT 1	PC6
PORT 1	PC7
PORT 2	PA0
PORT 2	PA1
PORT 2	PA2
PORT 2	PA3
PORT 2	PA4
PORT 2	PA5
PORT 2	PA6
PORT 2	PA7

3. TB3

PORT 2	PB0
PORT 2	PB1
PORT 2	PB2
PORT 2	PB3
PORT 2	PB4
PORT 2	PB5
PORT 2	PB6
PORT 2	PB7
PORT 2	PC0
PORT 2	PC1
PORT 2	PC2
PORT 2	PC3
PORT 2	PC4
PORT 2	PC5
PORT 2	PC6
PORT 2	PC7

4. TB4

CLK0
GATE0
OUT0
CLK1
GATE1
OUT1
CLK2
GATE2
OUT2
GND1
GND2
+5V
-5V
GND1
GND1
GND2

APPENDIX B DATA SHEET

The 8253 and 8255 data sheet are list in the following:

FUNCTIONAL DESCRIPTION

General

The 8253 is a programmable interval timér/counter specifically designed for use with the Intel™ Microcomputer systems. Its function is that of a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8253 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 systems software, the programmer configures the 8253 to match his requirements, initializes one of the counters of the 8253 with the desired quantity, then upon command the 8253 will count-out the delay and interrupt the CPU when it has completed its tasks. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- Programmable Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock
- Digital One-Shot
- Complex Motor Controller

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions.

1. Programming the MODES of the 8253.
2. Loading the count registers,
3. Reading the count values

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation. It is enabled or disabled by CS so that no operation can occur to change the function unless the device has been selected by the system logic.

$\overline{RD}$ (Read)

A "low" on this input informs the 8253 that the CPU is inputting data in the form of a counters value

$\overline{WR}$ (Write)

A "low" on this input informs the 8253 that the CPU is outputting data in the form of mode information or loading counters.

A0, A1

These inputs are normally connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for mode selection.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8253. No reading or writing will occur unless the device is selected. The $\overline{CS}$ input has no effect upon the actual operation of the counters

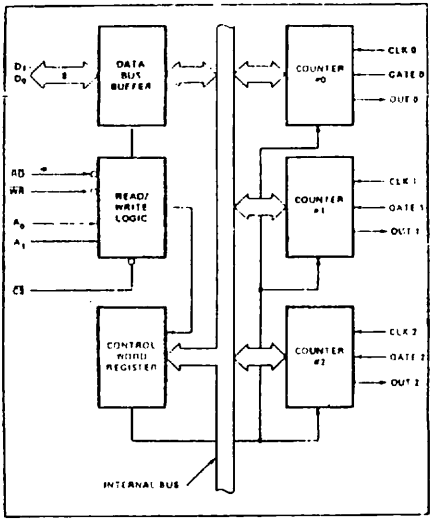


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Funejions

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State

Control Word Register

The Control Word Register is selected when A0. A1 are 11 It then accepts information from the data bus buffer and stores it in a register The information stored in this register controls the operational MODE of each counter, selection of binary or BCD counting and the loading of each count register

The Control Word Register can only be written into, no read operation of its contents is available

Counter #0, Counter #1, Counter #2

These three functional blocks are identical in operation so only a single Counter will be described Each Counter consists of a single 16-bit, pre-settable DOWN counter The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES stored in the Control Word Register.

The counters are fully independent and each can have separate Mode configuration and counting operation, binary or BCD Also, there are special features in the control word that handle the loading of the count value so that software overhead can be minimized for these functions

The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

8253 SYSTEM INTERFACE

The 8253 is a component of the Intel™ Microcomputer Systems and Interfaces in the same manner as all other peripherals of the family. It 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

Basically, the select inputs A0, A1 connect to the A0. A1 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 Intel® 8205 for larger systems.

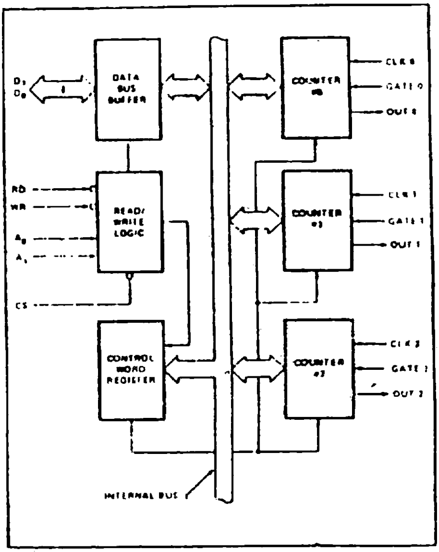


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

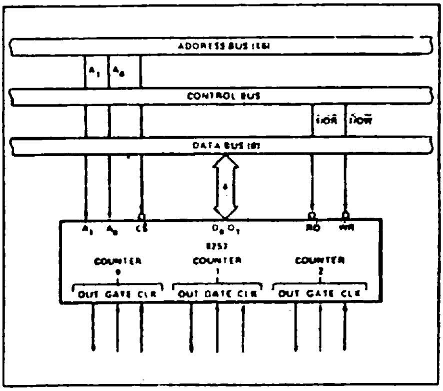


Figure 5. 8253 System Interface

OPERATIONAL DESCRIPTION

General

The complete functional definition of the 8253 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize each counter of the 8253 with the desired MODE and quantity information. Prior to initialization, the MODE, count, and output of all counters is undefined. These control words program the MODE, Loading sequence and selection of binary or BCD counting

Once programmed, the 8253 is ready to perform whatever timing tasks it is assigned to accomplish.

The actual counting operation of each counter is completely independent and additional logic is provided on-chip so that the usual problems associated with efficient monitoring and management of external, asynchronous events or rates to the microcomputer system have been eliminated.

Programming the 8253

All of the MODES for each counter are programmed by the systems software by simple I/O operations

Each counter of the 8253 is individually programmed by writing a control word into the Control Word Register (A0, A1 = 11)

Control Word Format

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Definition of Control

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

RL — Read/Load:

RL1	RL0	
0	0	Counter Latching operation (see READ/WRITE Procedure Section)
1	0	Read/Load most significant byte only
0	1	Read/Load least significant byte only.
1	1	Read/Load 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 6

BCD:

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

Counter Loading

The count register is not loaded until the count value is written (one or two bytes, depending on the mode selected by the RL bits), followed by a rising edge and a falling edge of the clock. Any read of the counter prior to that falling clock edge may yield invalid data.

MODE Definition

MODE 0: Interrupt on Terminal Count. The output will be initially low after the mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached the output will go high and remain high until the selected count register is reloaded with the mode or a new count is loaded. The counter continues to decrement after terminal count has been reached.

Rewriting a counter register during counting results in the following:

(1) Write 1st byte stops the current counting.
(2) Write 2nd byte starts the new count.

MODE 1: Programmable One-Shot. The output will go low on the count following the rising edge of the gate input..

The output will go high on the terminal count. If a new count value is loaded while the output is low it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output [illegible] remain low for the full count after any rising ed[illegible] of the gate input.

MODE 2: Rate Generator. Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator. Similar to MODE 2 except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by 2 until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for (N + 1)/2 counts and low for (N – 1)/2 counts.

MODE 4: Software Triggered Strobe. After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again..

If the count register is reloaded during counting, the new count will be loaded on the next CLK pulse. The count will be inhibited while the GATE input is low.

MODE 5: Hardware Triggered Strobe. The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of any trigger.

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	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	— —	Enables counting
5		Initiates counting	— —

Figure 5. Gate Pin Operations Summary

8253 READ/WRITE PROCEDURE

Write Operations

The systems software must program each counter of the 8253 with the mode and quantity desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection, e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent. (SC0, SC1)

The loading of the Count Register with the actual count value, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded it must be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will actually be decremented. Loading all zeroes into a count register will result in the maximum count (2¹⁶ for Binary or 10⁴ for BCD). In MODE 0 the new count will not restart until the load has been completed. It will accept one of two bytes depending on how the MODE control words (RL0, RL1) are programmed. Then proceed with the restart operation.

	MODE Control Word Counter n
LSB	Count Register byte Counter n
MSB	Count Register byte Counter n

Note: Format shown is a simple example of loading the 8253 and does not imply that it is the only format that can be used.

Figure 8. Programming Format

			A1	A0
No. 1		MODE Control Word Counter 0	1	1
No. 2		MODE Control Word Counter 1	1	1
No. 3		MODE Control Word Counter 2	1	1
No. 4	LSB	Count Register Byte Counter 1	0	1
No. 5	MSB	Count Register Byte Counter 1	0	1
No. 6	LSB	Count Register Byte Counter 2	1	0
No. 7	MSB	Count Register Byte Counter 2	1	0
No. 8	LSB	Count Register Byte Counter 0	0	0
No. 9	MSB	Count Register Byte Counter 0	0	0

Note: The exclusive addresses of each counter's count register make the task of programming the 8253 a very simple matter, and maximum effective use of the device will result if this feature is fully utilized.

Figure 9. Alternate Programming Formats

Read Operations

In most counter applications it becomes necessary to read the value of the count in progress and make a computational decision based on this quantity. Event counters are probably the most common application that uses this function. The 8253 contains logic that will allow the programmer to easily read the contents of any of the three counters without disturbing the actual count in progress.

There are two methods that the programmer can use to read the value of the counters. The first method involves the use of simple I/O read operations of the selected counter. By controlling the A0, A1 inputs to the 8253 the programmer can select the counter to be read (remember that no read operation of the mode register is allowed A0, A1-11). The only requirement with this method is that in order to assure a stable count reading the actual operation of the selected counter must be inhibited either by controlling the Gate input or by external logic that inhibits the clock input. The contents of the counter selected will be available as follows:

first I/O Read contains the least significant byte (LSB;

second I/O Read contains the most significant byte (MSB)

Due to the Internal logic of the 8253 it is absolutely necessary to complete the entire reading procedure. If two bytes are programmed to be read then two bytes must be read before any loading WR command can be sent to the same counter.

Read Operation Chart

A1	A0	RD	
0	0	0	Read Counter No. 0
0	1	0	Read Counter No. 1
1	0	0	Read Counter No. 2
1	1	0	Illegal

Reading While Counting

In order for the programmer to read the contents of any counter without effecting or disturbing the counting operation the 8253 has special internal logic that can be accessed using simple WR commands to the MODE register. Basically, when the programmer wishes to read the contents of a selected counter "on the fly" he loads the MODE register with a special code which latches the present count value into a storage register so that its contents contain an accurate, stable quantity. The programmer then issues a normal read command to the selected counter and the contents of the latched register is available.

MODE Register for Latching Count

A0, A1 = 11

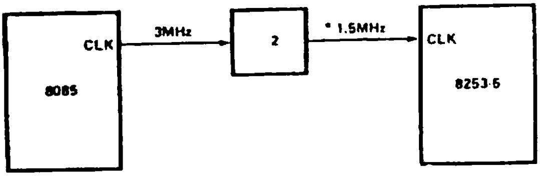
D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1,SC0 — specify counter to be latched.

D5,D4 — 00 designates counter latching operation

X — don't care.

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed. This command has no effect on the counter's mode.



*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2 MHz or less

Figure 10. MCS-85™ Clock Interface*

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel® microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

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

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communiction between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

8255A BASIC OPERATION

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
					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
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS → 3-STATE

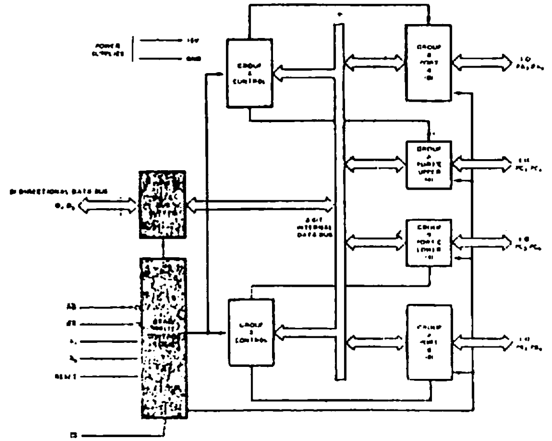


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode"; "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

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

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

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

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.

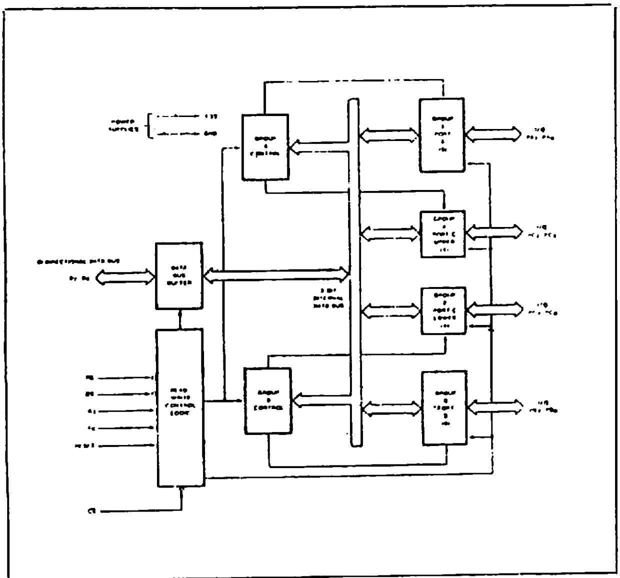
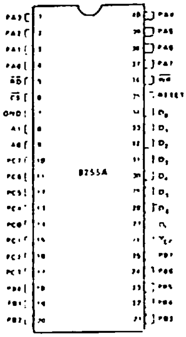


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions

PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (BI DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
Vcc	+5 VOLTS
GND	0 VOLTS

8255A OPERATIONAL DESCRIPTION

Mode Selection

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

Mode 0 – Basic Input/Output
Mode 1 – Strobed Input/Output
Mode 2 – Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

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

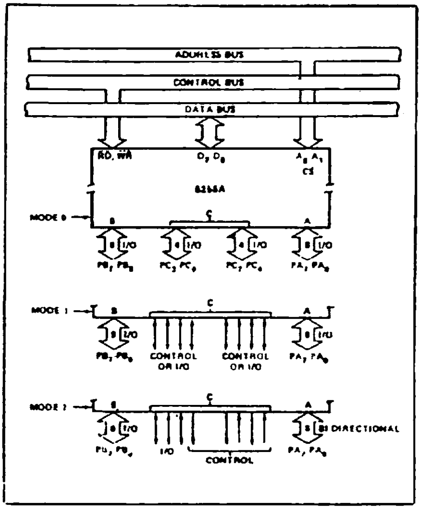


Figure 5. Basic Mode Definitions and Bus Interface

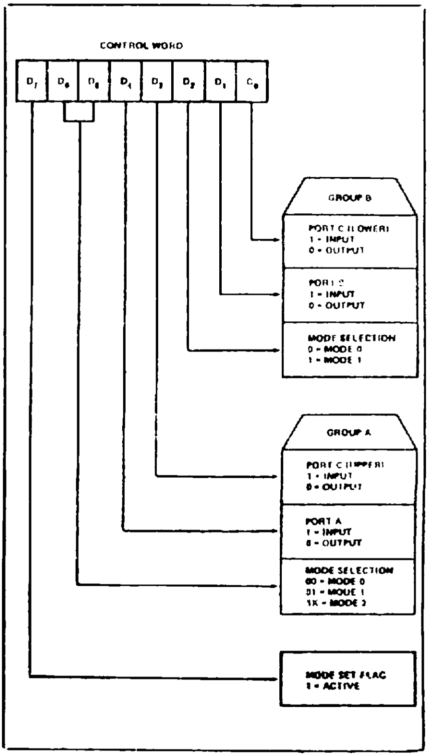


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

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

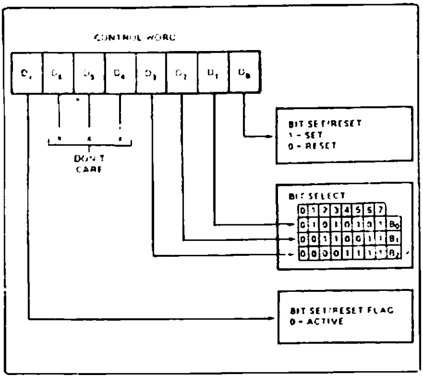


Figure 7. Bit Set/Reset Format

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

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

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

INTE flip-flop definition:

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

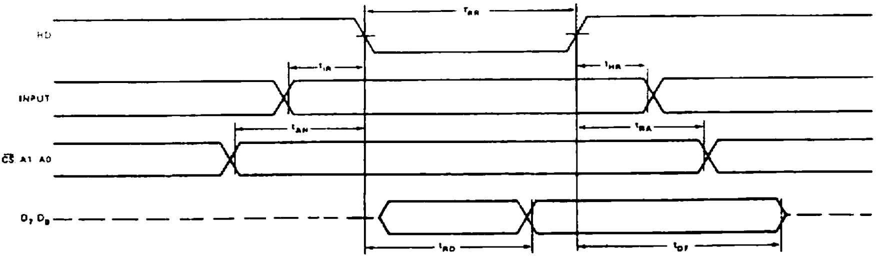
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

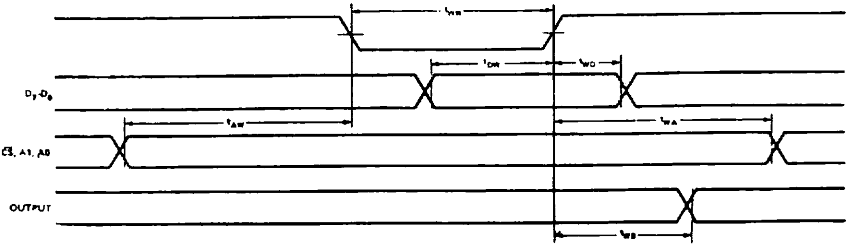
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

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



MODE 0 (Basic Input)

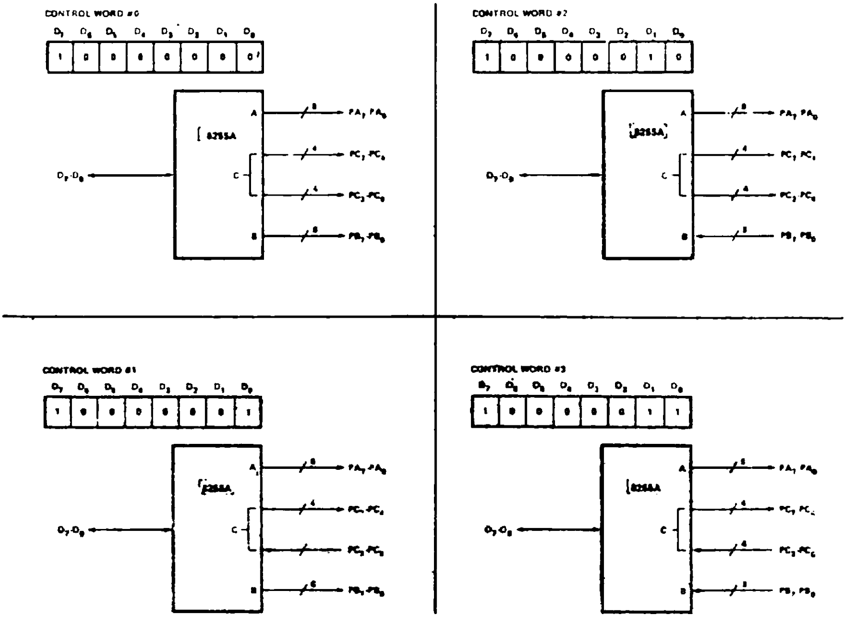


MODE 0 (Basic Output)

MODE 0 Port Definition

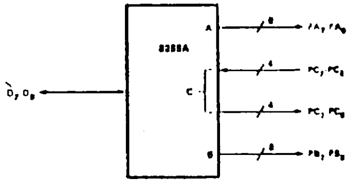
A		B		GROUP A			GROUP B	
D4	D3	D1	D0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations



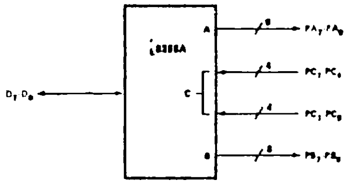
CONTROL WORD #4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



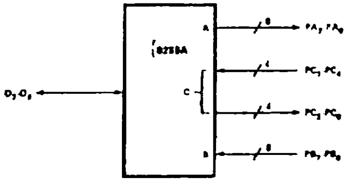
CONTROL WORD #5

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



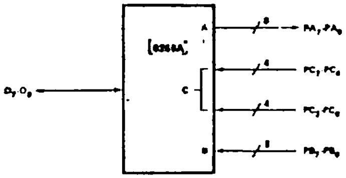
CONTROL WORD #6

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



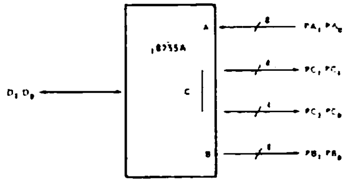
CONTROL WORD #7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



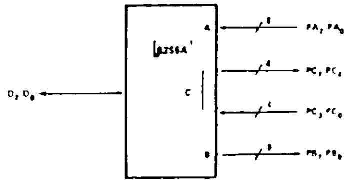
CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



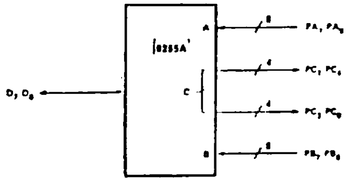
CONTROL WORD #9

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



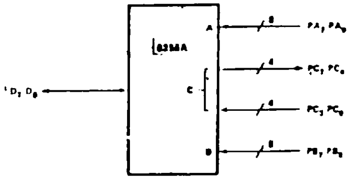
CONTROL WORD #10

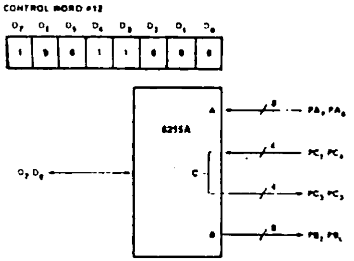
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0

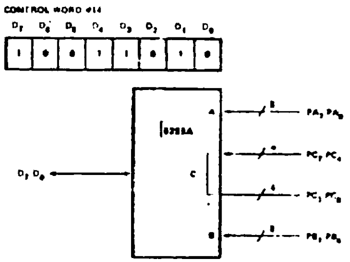


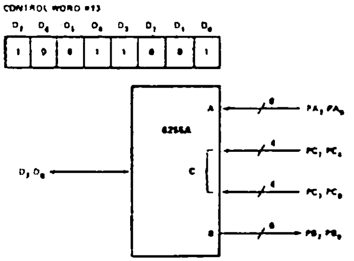
CONTROL WORD #11

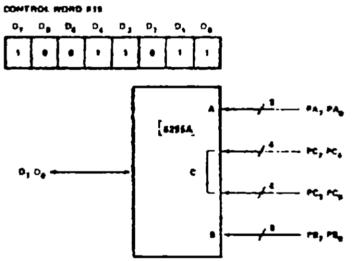
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1











Operating Modes

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

Mode 1 Basic Functional Definitions:

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

Input Control Signal Definition

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

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

INTR (Interrupt Request)

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

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

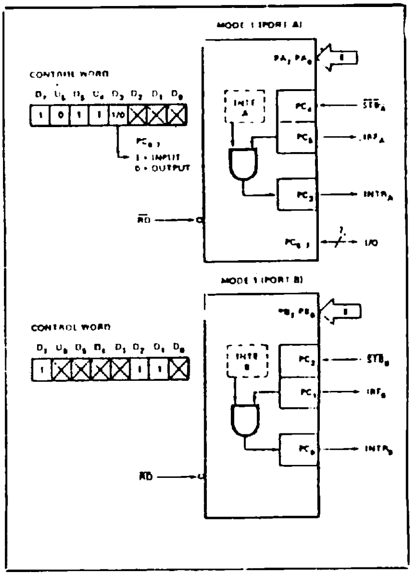


Figure 8. MODE 1 Input

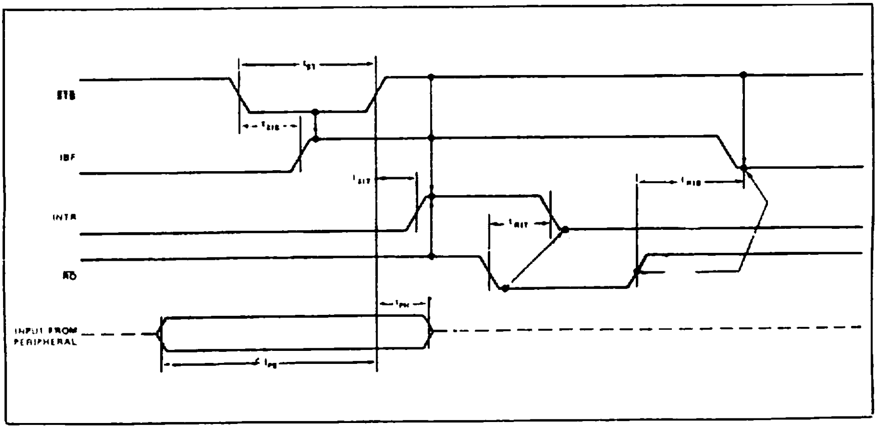


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

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

$\overline{\text{ACK}}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

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

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

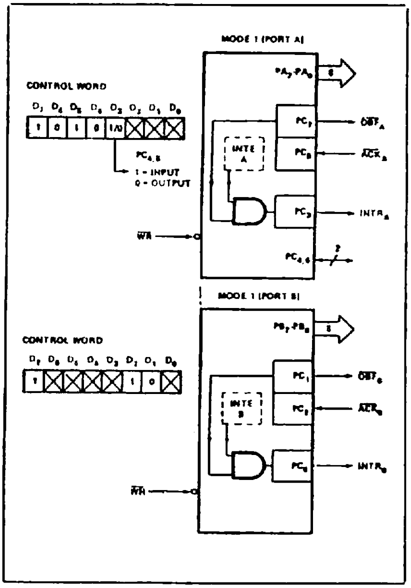


Figure 10. MODE 1 Output

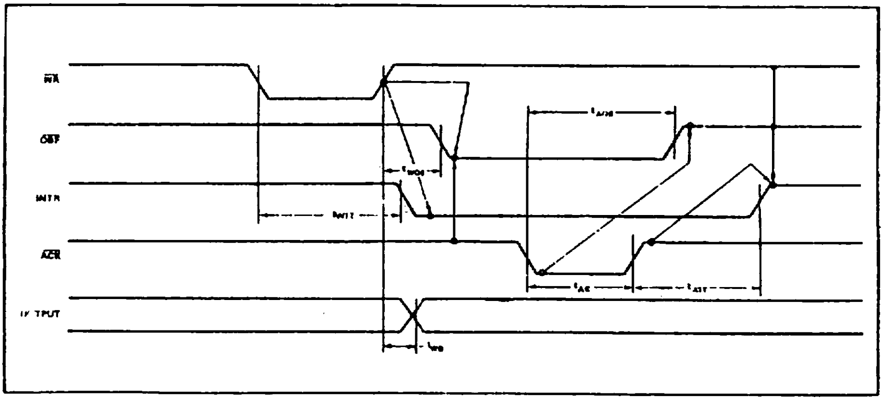


Figure 11. Mode 1 (Strobed Output)

Combinations of MODE 1

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

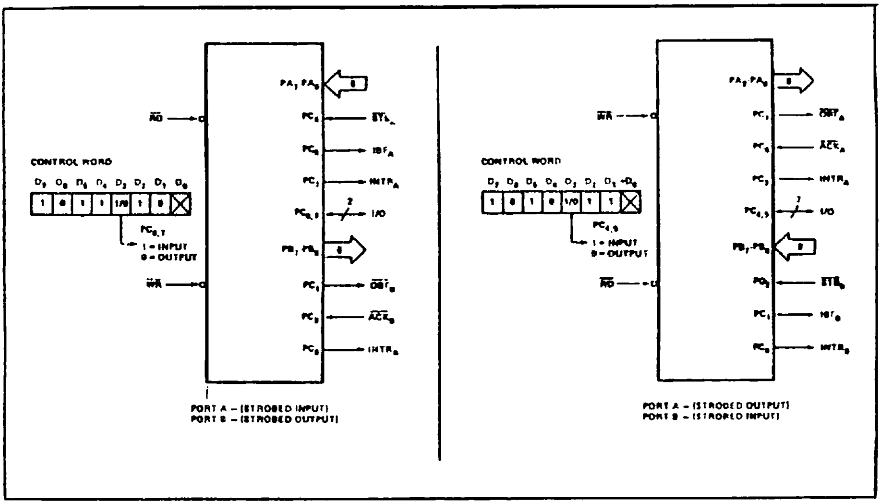


Figure 12. Combinations of MODE 1

Operating Modes

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

MODE 2 Basic Functional Definitions:

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

Bidirectional Bus I/O Control Signal Definition

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

Output Operations

OBF (Output Buffer Ful). The OBF output will go "low" to indicate that the CPU has written data out to port A.

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

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

Input Operations

$\overline{STB}$ (Strobe Input)

STB (Strobe Input). A "low" on this input loads data into the input latch.

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

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

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	←→	
PA_1	IN	OUT	IN	OUT	←→	
PA_2	IN	OUT	IN	OUT	←→	
PA_3	IN	OUT	IN	OUT	←→	
PA_4	IN	OUT	IN	OUT	←→	
PA_5	IN	OUT	IN	OUT	←→	
PA_6	IN	OUT	IN	OUT	←→	
PA_7	IN	OUT	IN	OUT	←→	
PB_0	IN	OUT	IN	OUT	——	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	——	
PB_2	IN	OUT	IN	OUT	——	
PB_3	IN	OUT	IN	OUT	——	
PB_4	IN	OUT	IN	OUT	——	
PB_5	IN	OUT	IN	OUT	——	
PB_6	IN	OUT	IN	OUT	——	
PB_7	IN	OUT	IN	OUT	——	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs —

All input lines can be accessed during a normal Port C read.

If Programmed as Outputs —

Bits in C upper (PC_7-PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3-PC_0) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

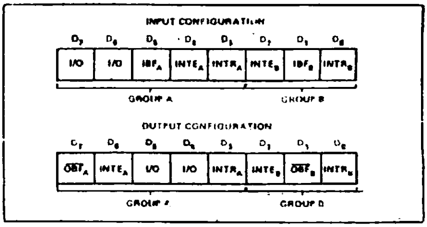
Source Current Capability on Port B and Port C

Any set of eight output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

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



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	$INTR_B$

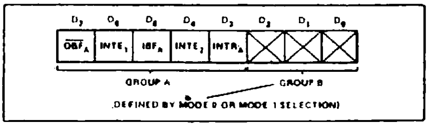
GROUP A — GROUP B

OUTPUT CONFIGURATION

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 17. 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 18. MODE 2 Status Word Format

APPENDIX C WARRANTY INFORMATION

SmartLab 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 DCI SmartLab 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 SmartLab porduct within the specified warranty period, SmartLab 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 SmartLab.

To ensure that your SmartLab product is covered by the warranty provisions, it is necessary that you return the Warranty card.

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

SmartLab 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 SmartLab. Goods returned without this authorization will not be attended to.

APPENDIX D WINDOW LIBRARY FOR 8255/8253 I/O CARD

The port.dll file provides I/O functions for 8255/8253 I/O cards, each function is specified as follows. Where port and data are defined as integer.

```
 1. int DCI_inb(port)
    Read one byte from port address.
 2. int DCI_inw(port)
    Read one word (16 bits) from port address.
 3. DCI_outb(port,data)
    Write data (byte) to port.
 4. DCI_outw(port,data)
    Write data (word) to port.
```

Under visual BASIC, to declare the function and call the function is specified as follows.

```
 1. General declaration
    Declare Sub DCI_outb Lib "a:\port.dll"
    (ByVal port As Integer, ByVal data As Integer)

    Declare Function DCI_inb Lib "a:\port.dll"
    (ByVal port As Integer) As Integer

 2. Library call
    DCI_outb port,data        /*for output*/
    res = DCI_inb(port+4)     /*for input*/
```

Under Borland C :

```
 1. General declaration
    hModule = LoadLibrary("a:\port.dll");
    myin = GetProcAddress(hModule,"DCI_inb");
    myout = GetProcAddress(hModule,"DCI_outb");

 2. Library call
    myout(port,odata);
    idata = myin(port+4);
```

We provide sample program in the distribution diskette. For Visual BASIC, please refer "8255_2.mak" and "8255_2.frm" files. For Borland C, please refer "8255_2.c" and "8255_2.def" files.