Learning

# 16-BIT MICROCOMPUTER TECHNOLOGY

## Part 4: The Programmable Interval Timer and Programmable Peripheral Interface

By George Meyerle

THE fourth part of this series on the construction of an 8088-based microcomputer compatible with the IBM-PC examines two circuit elements in detail—the 8253-5 Programmable Interval Timer and the 8255A-5 Programmable Peripheral Interface (PPI).

**8253-5 Programmable Interval Timer.** The 8253-5 consists of three independent 16-bit counters under software control (Fig. 21). The 1.19-MHz clock inputs (pins 9, 15, 18) are fed from *IC29* (Fig. 4) which divides PCLK (2.38 MHz) by two. (Note: Figs. 1 through 9 and Tables I through III appeared in Part 1 of this series.)

To cause a DMA memory refresh cycle every 15 μs, the monitor program initializes channel one as follows. During an initialization phase, it outputs a mode-control word to port 43 (see I/O map, Table II). This word selects the channel and type of counting operation desired. In this case, we have selected the "rate generator" mode of channel one. This simply means that the input clock will be divided by the constant entered into counter register one. There are six different modes of operation possible as listed in Table IV.

The program then loads a hexadecimal 12 (decimal 18) into count register one. This results in a pulse train output at pin 13 with a period of 15 μs (1/1.19/18). The RAM requires refreshing of the 128 row addresses (RAS) every 2 ms. This is done by the DMA controller, which has been initialized to perform dummy reads at one of the required 128 addresses every time the timer issues a pulse. The product of 15 μs and 128 is 1.92 ms, which is within the time required by the memory.

Channel two is used by both the speaker and cassette outputs. When used by the speaker, channel two is initialized in mode 3 (square-wave output). To get a 1-kHz tone, a 1-ms period is needed. We obtain this count by loading hexadecimal 0533 into count register two. This gives us a 1-kHz square wave at pin 17. To get the tone to the speaker, we have to output a logic 1 to the speaker data line (I/O port 8255 port B, bit 1, pin 19). While this line is high, the output from the timer is gated to the speaker through the 75477 buffer. The frequency and duration of the tone generated are programmable by both the timer and the 8255.

The cassette write routine also uses the channel two counter set for mode 3 (square-wave output). The count register is modified by the software program as follows. If a zero bit is to be outputted, a 500-μs square wave is generated. If a one bit is to be outputted, a 1000-μs square wave is generated. When the program is writing to cassette, it first turns the motor on by setting the MOTOR OFF line low (8255 port

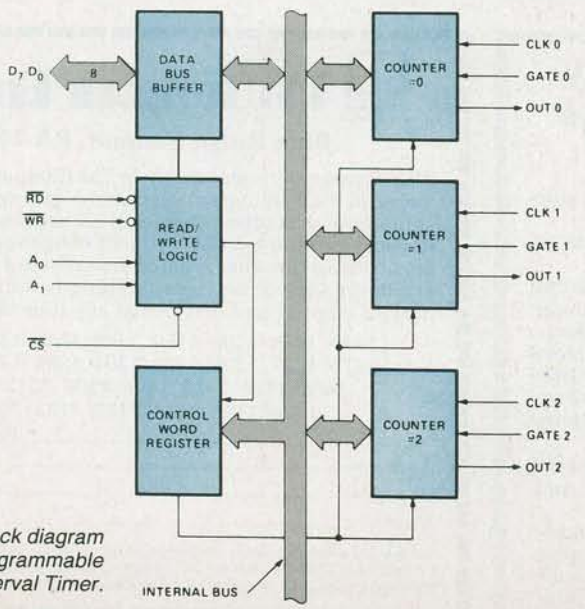


*Fig. 21. Block diagram of the 8253 Programmable Interval Timer.*

B, bit 3). This sets pin 3 of the 75477 low, energizing the relay. Then, using the timer, it generates a leader of 256 bytes of ones. This is followed by two sync bytes, 256 bytes of data, two bytes of error-checking code, 256 bytes of data, etc. until all the data specified is saved. After the last data block, a trailer string of four bytes of ones is written. The cassette motor is then turned off by setting the MOTOR OFF bit high. Although the cassette write program sounds busy, it is simply modifying the counter register to provide the tones that represent the data stored on tape. The cassette read is handled by the I/O port directly and will be discussed later.

Channel zero is not used in the basic Explorer 88. However, when the optional IBM keyboard is added, the program is changed to program the channel zero timer to interrupt the CPU at regular intervals. This allows keeping track of the "time of day" or other time-dependent programs. Channel zero is disabled during cassette operations.

**TABLE IV—8235-5 OPERATION MODES**

| MODE | OPERATION | MODE | OPERATION |
|---|---|---|---|
| 1 | **Interrupt on Terminal Count** After count bytes are loaded into selected count register, output goes low and remains low during the countdown. When terminal count is reached, output goes (and stays) high until selected count register is reloaded. | | put with a period equal to count loaded into count register. |
| 2 | **Programmable One-Shot** Output goes low on count following a gate input. Output will go high again on the terminal count. Since one-shot is triggerable, output is low for full count after gate input. | 4 | **Software-Triggered Strobe** Output remains high until terminal count has been reached. Then output goes low for one period of input clock. Count is inhibited while gate input is low and begins only after gate is brought high. |
| 3 | **Rate Generator** Generates a square-wave out- | 5 | **Hardware-Triggered Strobe** Output remains high until gate input is brought high. Then output goes low when count register terminal count is reached. Output remains low for one period of clock cycle. |

**8255A-5 Programmable Peripheral Interface.** The 8255, although used in a relatively straightforward manner in this design, is a very versatile CPU interface. It consists of 24 I/O lines that can be individually programmed in two groups of 12 and used in three different modes of operation (Fig. 22). In mode 0 (Fig. 23A), each group of 12 I/O pins can be programmed in sets of four to be either inputs or outputs. In mode 1 (Fig. 23B), each group can be programmed to have eight lines of input or output, with the remaining lines used for handshaking and interrupt control signals. In mode 2 (Fig. 23C), eight

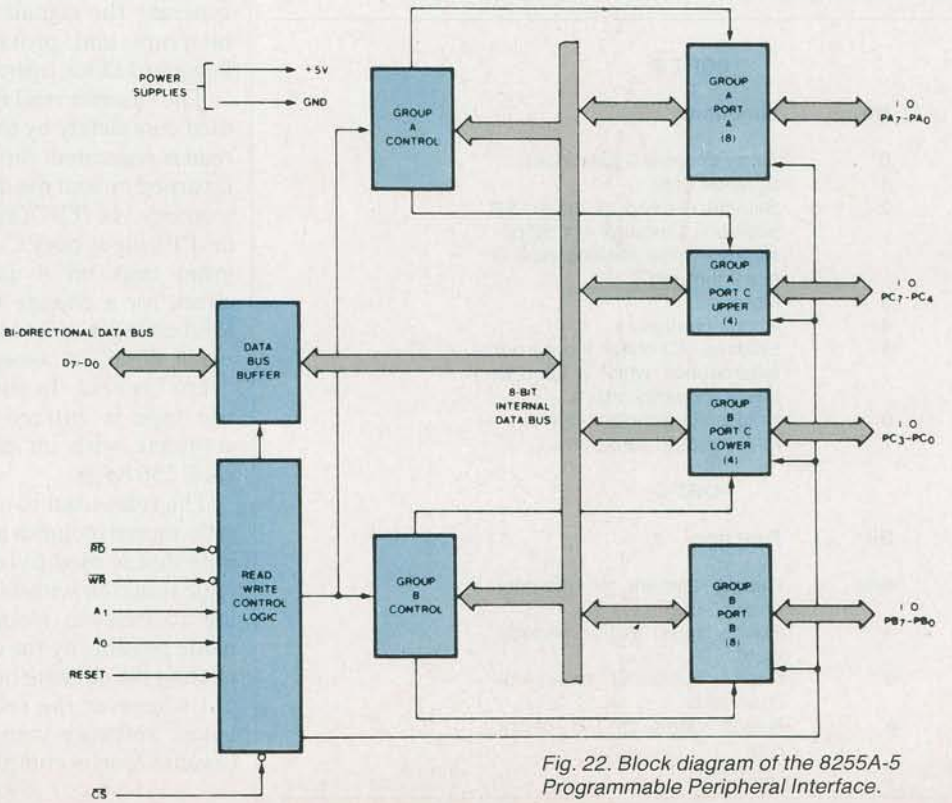


*Fig. 22. Block diagram of the 8255A-5 Programmable Peripheral Interface.*

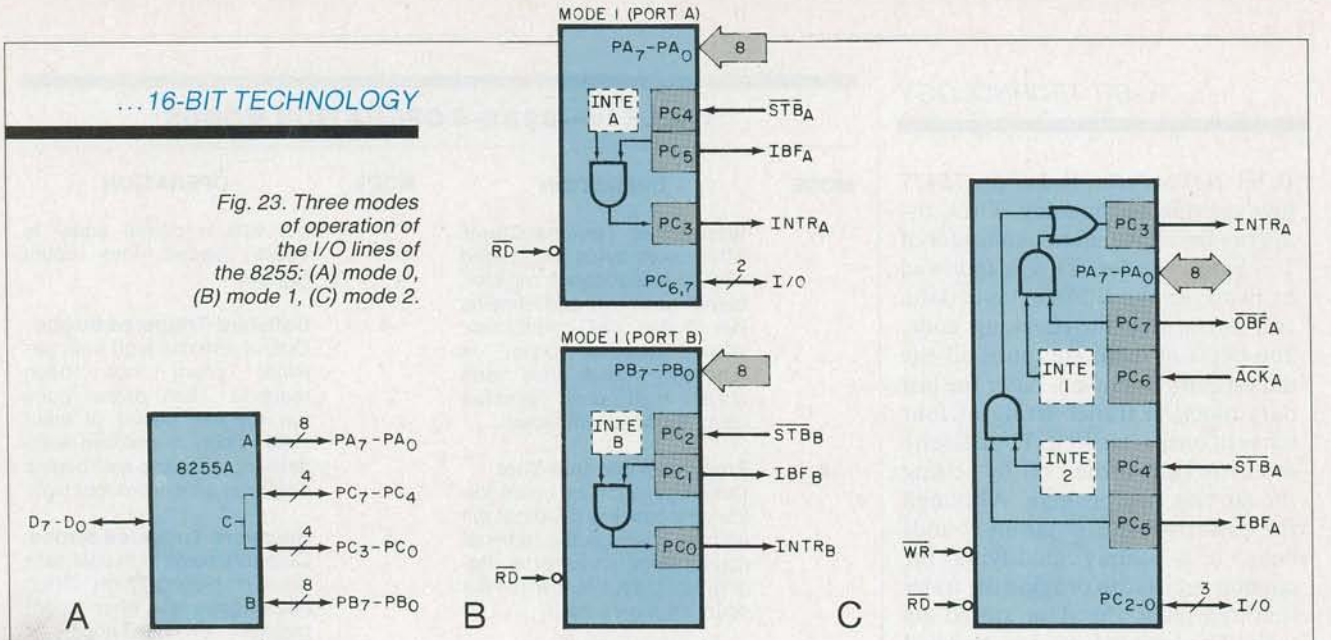


*Fig. 23. Three modes of operation of the I/O lines of the 8255: (A) mode 0, (B) mode 1, (C) mode 2.*

lines are used for a bidirectional bus and five lines for handshaking (borrowing one line from the other group).

The PPI is used here in mode 0 only. By writing a 99 hex to I/O port 63 hex, it is initialized so that ports A and C are inputs and port B is an output. (Complete programming details are available in the 8255 spec sheet.) Referring to Fig. 5, let's examine how the two 8-position mode switches are connected. The switches advise the system monitor what hardware or memory has been connected to the system. Mode switch *S1* is connected to port A, which is configured to be an input port, through buffer *IC19*. Port A is also connected to *IC20*, which is a shift register used by the optional IBM compatible keyboard to enter data. The status of port B bit 7 determines whether switch *S1* or the shift register is to be read. Port B (set up to be an output port) and port C (set up to be an input port) have bit assignments as shown in Table V.

The optional keyboard generates serial data that is loaded into *IC20*. When the data is ready to be read, an interrupt request is generated at *IC40*. The keyboard service routines then read the data via PPI port A and use PPI port B bits 6 and 7 to generate the signals that reset the interrupt and provide for proper keyboard clock operation.

The cassette read function is handled completely by the PPI. When a read is requested, the cassette motor is turned on and the data is read into memory via *IC37* (Fig. 4) and bit 4 on PPI input port C. The read program tests bit 4 continuously to check for a change in level. If the level changes in less than about 700 μs, a "one" is assumed; if not, a "zero" is read. In this way, the entire tape is entered into memory complete with an error test after each 256 bytes.

The relay used to control the cassette motor includes an extra switch pole that is used to test the cassette logic function without actually having to make a recording. This is made possible by the extra pole connecting the cassette output to the input whenever the relay is not activated. Software then tests that the cassette loop is complete.

*(To be continued.)*

## TABLE V—8255A-5 PORT ASSIGNMENTS

**PORT B**

| Bit | Function |
|---|---|
| 0 | Timer channel 2 gate input. |
| 1 | Speaker data. |
| 2 | Selects reading of either *S2* positions 1 through 4 or *S2* position 5 when reading port C bits 0 through 3. |
| 3 | Motor off. |
| 4 | No connection. |
| 5 | Enables I/O check input from external bus, which is used to check for parity errors. |
| 6 | Hold keyboard clock low. |
| 7 | *S1*/keyboard sense line. |

**PORT C**

| Bit | Function |
|---|---|
| 0-3 | Report amount of memory above 64K. |
| 4 | Reads data from cassette input. |
| 5 | Reads status of timer out channel 2. |
| 6 | Reads status of I/O check line. |
| 7 | Not used. |