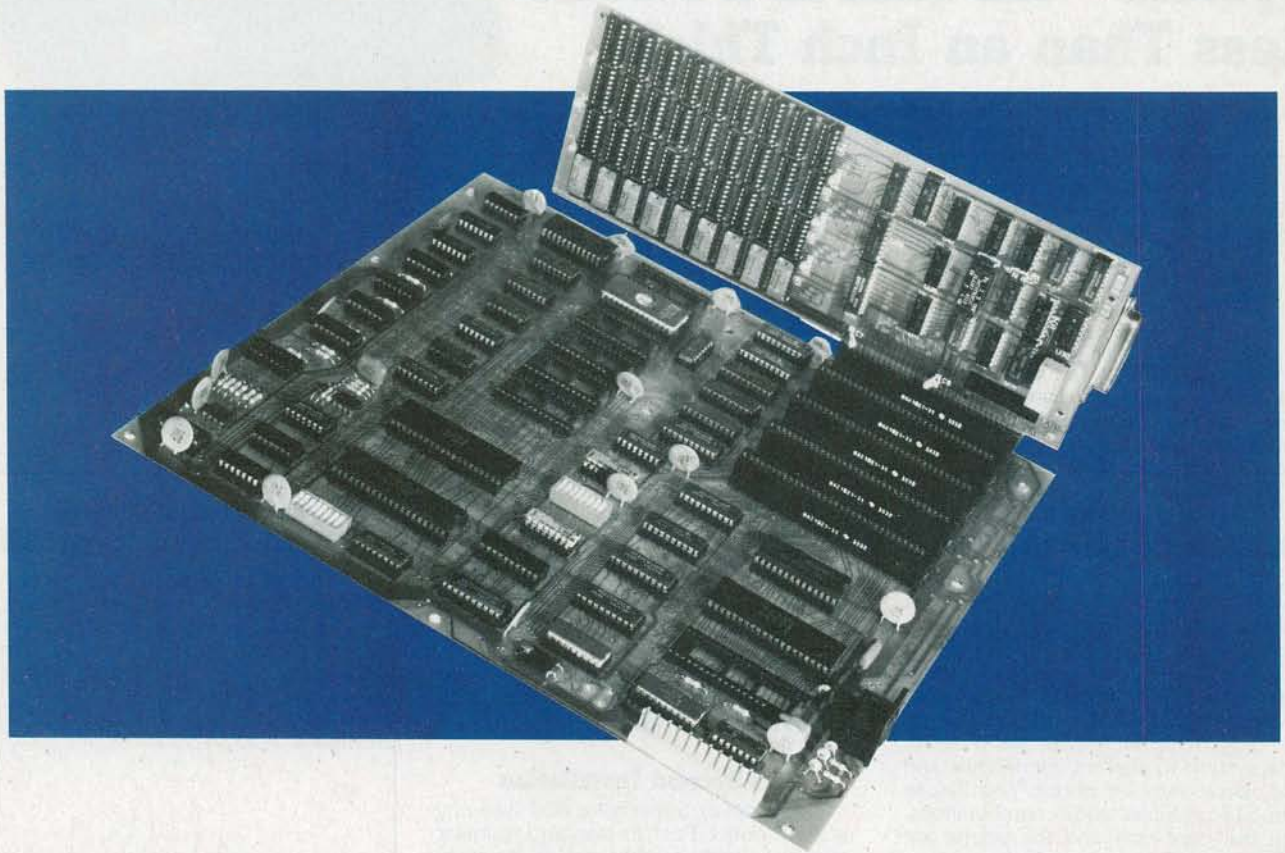
Learning

16-BIT MICROCOMPUTER TECHNOLOGY

Part 5: The Dynamic RAM and Asynchronous Communications Port

By George Meyerle



THE fifth part of our series on the construction of an 8088-based micro-computer compatible with the IBM-PC wraps up our discussion of the major circuit elements of the system.

64/256K Dynamic RAM Section. The use of dynamic RAM in open bus systems has caused many problems in the past. The IBM approach, however, uses the system board to control refresh via the DMA controller plus the addition of a 9th bit for parity-error testing. Hopefully, this puts to rest the objections of even the most critical skeptics. The only negative point is that, due to refresh requirements, the system is slowed down by about 7%.

To better understand the concept of dynamic RAM, let's examine the principal elements of a RAM chip (Fig. 24). The 64 RAM chips used in this design are organized as 64K by 1 data bits. The memory cells actually consist of 65,436 tiny capacitors (arranged in a 256-row by 256-column matrix), with one capacitor for each memory location. When a capacitor is charged up, the cell is considered to be storing a "one", when a capacitor is discharged, the cell is storing a "zero." Since the capacitor is not perfect, it will eventually lose its charge, thus changing the value of the stored data. This is of course, unacceptable and is prevented by *refreshing*, which is simply a system of recharging or discharging the capacitors periodically before the data values change.

The RAM chips, using sense/refresh amps, automatically refresh an entire column of addresses during a row address read. Each row is read every 2 ms by the DMA controller, which performs a dummy read every 15 μs on a different (1 of 256) row address. Note that 16 address lines are necessary to address 64K bits and that the RAM chip has only eight address inputs, labled MA0 through MA7. This means that the memory board must multiplex the 16-bit address bus into two 8-bit slices and generate address strobe signals RAS and CAS (row and column address strobes), which are used to latch first the row address and then the column address into the RAM chips.

Referring to Figs. 25 and 26, you'll note that RASON and $\overline{\text{CASON}}$ are generated during all memory read or write cycles. $\overline{\text{CASON}}$ is delayed by 100 ns relative to RASON and continues 20 ns after RASON. The 60-ns $\overline{\text{ROW}}$/COLUMN delay output controls the address multiplexer logic, which presents the RAM chip with first the row addresses and then the column addresses. These addresses are then latched into the RAM chip logic by the $\overline{\text{RAS0}}$ through $\overline{\text{RAS3}}$ and $\overline{\text{CAS0}}$ through $\overline{\text{CAS3}}$. The CAS signals, which are not required during refresh (to save power), also act as a chip select, enabling the read and write functions.

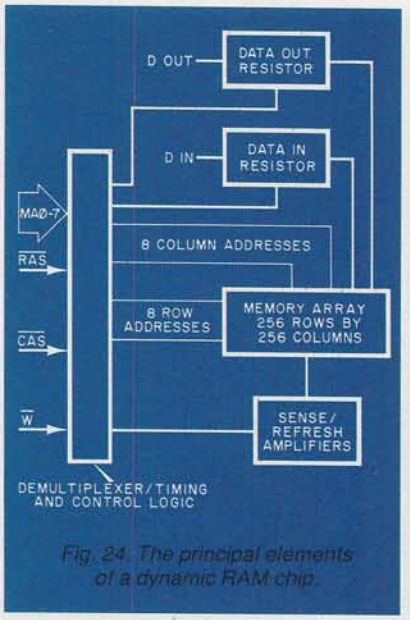


Fig. 24. The principal elements of a dynamic RAM chip.

Refreshing occurs even if the board is not addressed because $\overline{\text{DACK0}}$, which is generated by the DMA controller, generates the $\overline{\text{RFSH}}$ signal, which acts as the RAS signal during refresh. $\overline{\text{RAS0}}$ through $\overline{\text{RAS3}}$ and $\overline{\text{CAS0}}$ through $\overline{\text{CAS3}}$, are only generated when the board and corresponding memory bank (0 through 3) are selected. When writing to RAM a parity bit is generated by the 74LS280 (9-bit parity generator/checker), which is written to RAM as a ninth data bit. When the data is read, the parity bit is checked by the 74LS280, which forces the entire system to stop if a parity error has occurred. The RAM chips are read during the rising edge of $\overline{\text{RAS0}}$ through $\overline{\text{RAS3}}$ at which time the data must be valid. RAM chips that are slower than 250 ns cannot be used. The actual write timing occurs sometime after $\overline{\text{CAS}}$ and is determined by the RAM chips. All that is required is that the data bus be stable during the write cycle.

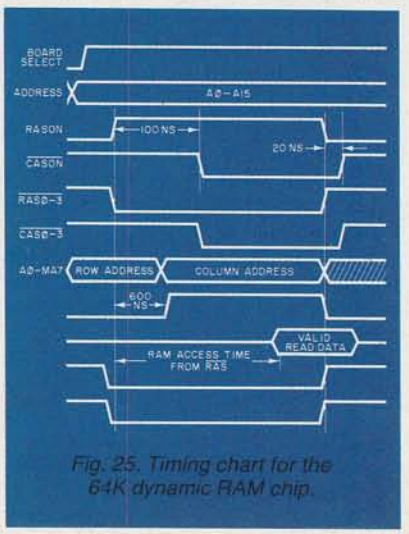


Fig. 25. Timing chart for the 64K dynamic RAM chip.

8250-B Asynchronous Communications Port. The final major circuit block that we'll discuss is the 8250-B asynchronous communications port. This serial I/O port, which is the major link between the computer and the outside world, is part of this kit project, though an option with an IBM-PC.(It's essential if a modem, serial printer, or terminal emulation software is to be used.) The Explorer 88 also has a BIOS program that can be used with either a standard video terminal or an IBM-compatible keyboard and either a color or monochrome video board.

The 8250-B performs serial-to-parallel conversions on data received from peripheral devices and parallel-to-serial conversions on data received from the CPU. The functional characteristics such as the band rate selection, parity generation, error detection, are programmed into the port by the initialization program executed during power-up or system reset.

The monitor program used in the Explorer 88 programs the 8250 to be a standard terminal interface with the baud rate determined by the switch settings on *S1* (see Part 1; Fig. 5). If an IBM compatible keyboard is used, the 8250-B is not initialized until activated by a user program. It is important to realize that the functional characteristics of the 8250-B, being under software control, can be changed at any time by the user. The 8250-B is connected to the system as an I/O port at addresses 3F8-3FF. These are the addresses reserved by IBM for the first of two serial communicastions ports. If you wish to change the port address, it is suggested that you use 1F8-1FF. (Simply invert address line A9 going to the 74LS30 port decoder shown in Part 3, Fig. 16, and add the necessary initializing software.)

To get an idea of just how powerful this chip is, here is a look at some of its features:

- Adds or deletes start, stop, and parity bits to or from a serial data stream
- Independently controlled transmit, receive, line status, and data set interrupts
- Programmable baud-rate generator with 18 steps from 50 to 56000 baud (only supported to 9600 baud by IBM)
- Includes all modem control functions
- Fully programmable serial interface characteristics with 5-, 6-, 7-, or 8-bit characters; even-, odd-, or no-parity bit

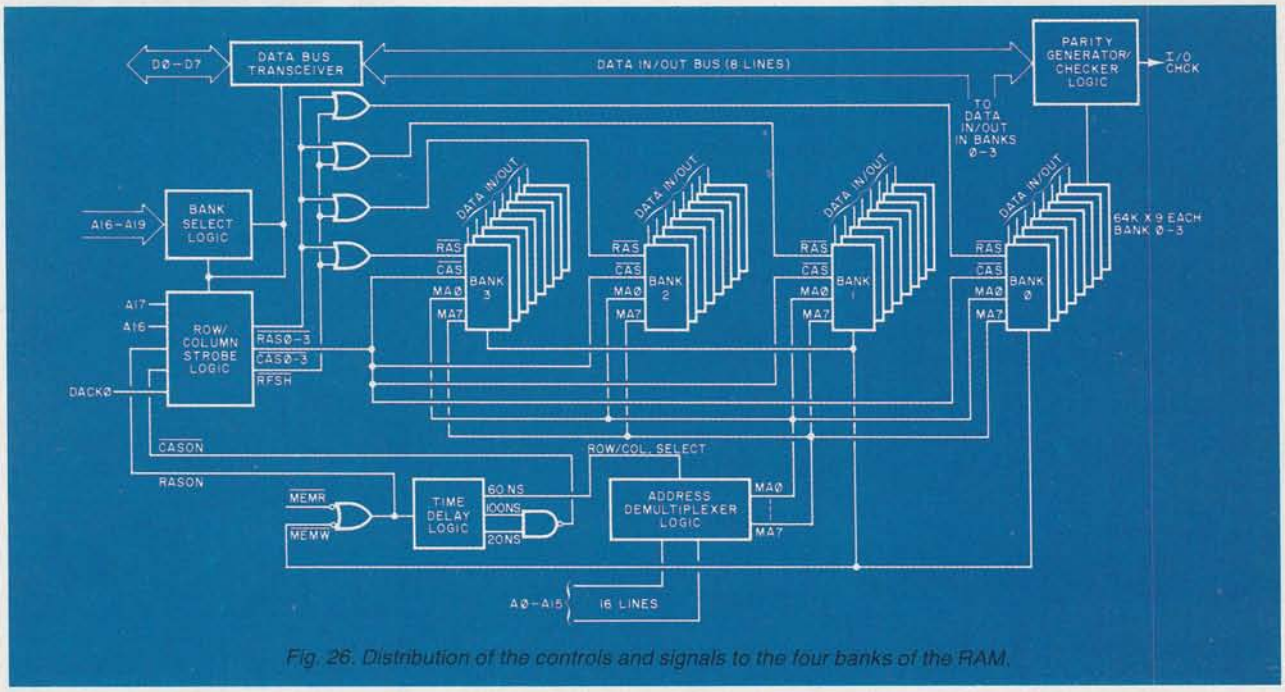


Fig. 26. Distribution of the controls and signals to the four banks of the RAM.

generation and detection; and 1-, 1.5-, or 2-stop bit generation

- Complete status reporting capabilities
- False-start bit detection
- Internal diagnostic capabilities with loopback for link fault detection
- Prioritized interrupt controls
- Requires only a crystal and line drivers to connect to any EIA RS232 peripheral.

Referring to the block diagram of the 8250-B (Fig. 27), the bi-directional data bus is used to load control words, load and receive data, and read status registers. The interrupt line is programmed to go high whenever a key is pressed. This signals the CPU that a character has been entered at the Keyboard. The interrupt line can also be programmed to be active when there is a receiver error, empty transmitter holding buffer, and modem status. The interrupt activity is controlled by the interrupt enable register, which is programmed during system start-up or reset. The $\overline{I/OR}$, $\overline{I/OW}$, and address lines A0 to A2 are used to read and write data, control, and status to and from the port. There are eight additional lines that are connected to the RS232 interface drivers. Their functions are as follows:

SOUT: Serial data is outputted to a peripheral via this line. This output, in keeping with the standard for ports that are considered an output type, is connected to *J2*, pin #2 (*J2* refers to the DB-25 connector on the 64K RAM/R5232C port board). Note, however, that a terminal, which is also considered an output type, expects to find data inputted at its pin#3. This means that leads #2 and #3, connecting the terminal to the port, must be flipped or reversed.

SIN: Serial data to the port is inputted via this line. It is connected to *J2*, pin #3.

The remaining six lines are considered control or handshaking signals.

RTS: (*J2*, pin #4, *Request to Send*) This output line, when high, informs the modem or data set that the port is ready to transmit data.

DTR: (*J2*, pin #20, *Data Terminal Ready)* When high, this line informs the modem or data set that the port is ready to communicate and receive data.

DSR: (*J2*, pin #8, *Data Set Ready)* This input signal, when high, informs the port that the modem or data set is ready to communicate. The program tests the contents of the modem-status

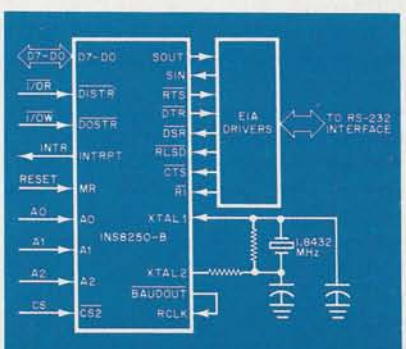


Fig. 27. Block diagram of the 8250-B.

register, which, when ready, reveals if the modem or data set is ready to communicate and if there has been any change in status since the last reading.

CTS: (*J2*, pin #5, *Clear to Send)* This active-high input signal is a modem-control signal that indicates if the modem will accept data from the port. The modem-control register also records if the input has changed state since the last reading of the register.

CD: *Carrier Detect* (also referred to as RLSD *Receiver Line Signal Detect)* This active-high signal indicates that the data carrier has been detected by the modem or data set. The status register, as above, reveals both the current status and if any carrier losses have occured since the last reading. The ability to detect carrier loss, is of course, vital when establishing reliable modem communications.

RI: *(Ring Indicator)* This active-high signal indicates that a telephone ringing signal has been received by the modem. The modem status register again reveals if a ring was detected since the last reading of the register.

It is important to remember that the pin numbers are different when connecting this port to devices that are not of an output type. Check the pin designations carefully before making any connections.

This completes our discussion of the major hardware elements of the Explorer 88. In future issues we will discuss other aspects of the system.

(To be continued)