Memory Interface

In this chapter, we explain how to interface both ROM and RAM to the Intel family of microprocessors.

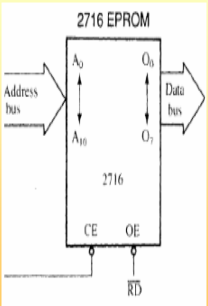
Objectives

- Upon completion of this chapter, you will be able to
  - Decode the memory address and use the outputs of the decoder to select various memory components.
  - Use programmable logic devices to decode memory addresses.
  - Explain how to interface both ROM and RAM to a microprocessor.
  - Interface memory to an 8-, 16-, 32- and 64-bit data bus.
  - Explain the operation of a dynamic RAM controller.
  - Interface dynamic RAM to the microprocessor.

Memory Devices

Memory Pin Connections:

- Address connections:
  - Address inputs that select a memory location within the memory device.
- Data connections:
  - Data outputs or input/outputs at which data are entered for storage or extracted for reading.
- Selection connections:
  - Signal inputs that selects or enables the memory device. (e.g. chip enable $\overline{CE}$, chip select $\overline{CS}$, select S etc.)
- Control connections:
  - Signal inputs that controls the operation of the memory device (e.g. write enable $\overline{WE}$, read enable $\overline{RE}$, output enable $\overline{OE}$, read/write R/$\overline{W}$ etc.)



Memory Devices

ROM Memory:

- The read-only memory (ROM) is permanently programmed so data are always present, even when power is disconnected.
- It is used to store programs and data that are resident to the system and must not change when power is disconnected.
- It is also called nonvolatile memory.
- The EPROM (erasable programmable ROM) is erasable if exposed to high-intensity ultraviolet light for about minutes.
- The PROM (programmable ROM) is programmed in the field by burning open tiny fuses, but once programmed it can't be erased.

Memory Devices

ROM Memory:

- The EEPROM (electrically erasable programmable ROM) is electrically erasable in the system, but require more time to erase than a normal RAM.
- EEPROM is also called flash memory.
- This type of memory component requires wait states to operate properly with the 8088/8086 processors because of its rather long access time.

Memory Devices

Static RAM:

- It retains data for as long as DC power is applied.
- No special action (except power) is required to retain stored data.
- The access time of a SRAM is much shorter than that of a ROM, and no wait state is required in general.
- Examples include 4016(2Kx8) and 62256(32Kx8).

Memory Devices

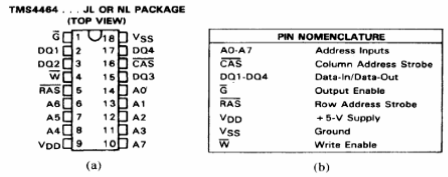
Dynamic RAM:

- It is essentially the same as SRAM, except that it retains data for only 2 to 4 ms on an integrated capacitor.
- The contents of the DRAM must be refreshed by being completely written.
- The address inputs are usually multiplexed to reduce the number of address bits
- DRAM is larger size than SRAM

Memory Devices

Dynamic RAM Example:

TMS4464, 64K×4 bits of data, 8 address inputs, use RAS and CAS to tell the memory if the address providing is the upper half or the lower half of the 16-bit address.

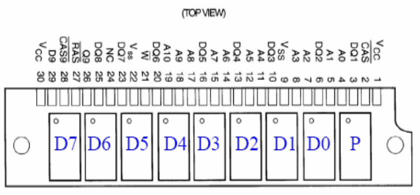


The pin-out of the TMS4464, 64K × 4 dynamic RAM (DRAM).

Memory Devices

Dynamic RAM Example:

- DRAM is usually places on small circuit boards called SIMMs (Single In-line Memory Modules).

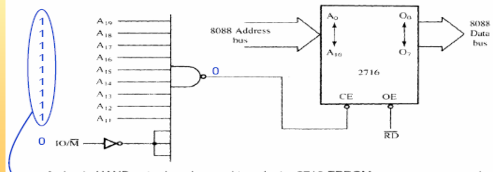


The pin-outs of a 30-pin SIMM organized as 4M × 9

Address Decoding

The previously explained address decoding techniques for I/O devices are equally applicable to memory devices.

- Use simple combinational logic devices:

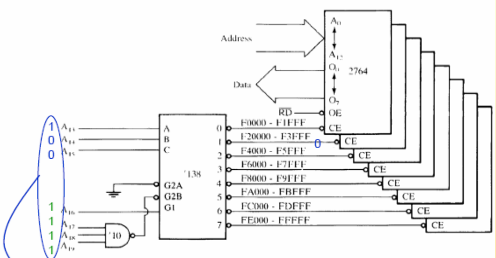


A simple NAND gate decoder used to select a 2716 EPROM memory component for memory locations FF800H–FFFFFH

1111 1111 1XXX XXXX XXXXb

Address Decoding

- Use line decoder such as 74138 and 74139:

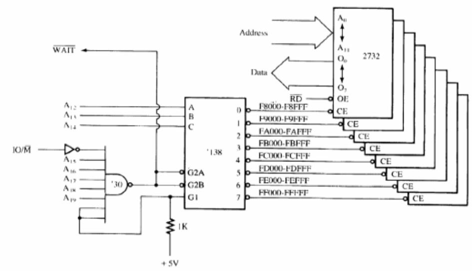


A circuit that uses eight 2764 EPROMs for a 64K × 8 section of memory in an 8088 microprocessor-based system. The addresses selected in this circuit are F0000H–FFFFFH.

1111 001X XXXX XXXX XXXXb

Example

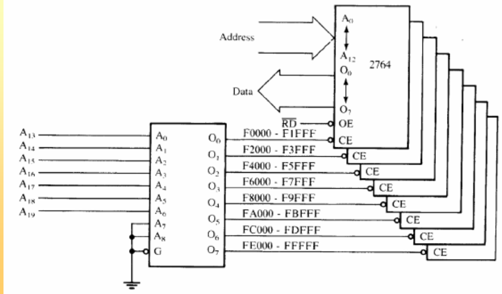
- Interfacing EPROM to the CPU:



Eight 2732 EPROMs interfaced to the 8088 microprocessor. Note that the output of the NAND gate is used to cause a wait state whenever this section of the memory is selected.

Address Decoding

- Use PROM decoder:



A memory system using the TPB28L42, 512 × 8 PROM as an address

Address Decoding

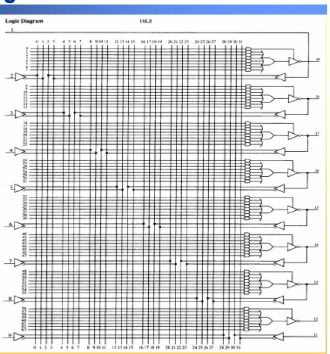
The 82S147 PROM programming pattern for the illustrated circuit.

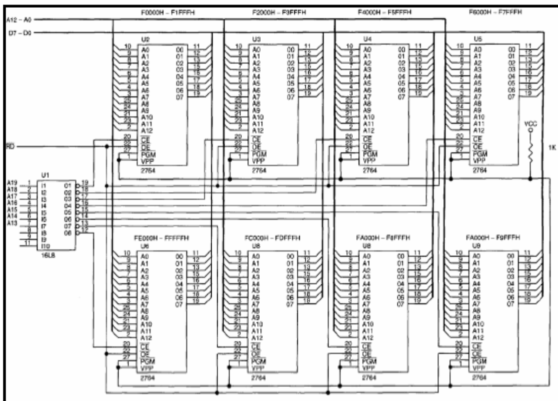
Inputs										Outputs							
OE	A8	A7	A6	A5	A4	A3	A2	A1	A0	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	1	1	0	0	1	1	0	1	1	1	1	1	1
0	0	0	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1
0	0	0	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1
0	0	0	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1
0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1
0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
all	other	combinations								1	1	1	1	1	1	1	1

Address Decoding

- Use PLD programmable decoders:

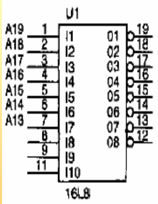
The PAL 16L8 (Advanced Micro Devices, Inc., 1988.)



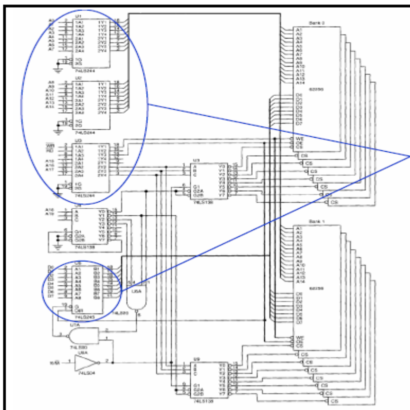


A PAL16L8 that decodes 8 2764 (8K × 8) memory devices

Address Decoding Example




```
TITLE    Address Decoder
CHIP     DECODER1 PAL16L8

;pins 1   2   3   4   5   6   7   8  9  10
    A19 A18 A17 A16 A15 A14 A13 NC NC GND

;pins 11 12 13 14 15 16 17 18 19  20
      NC O8 O7 O6 O5 O4 O3 O2 O1 VCC

EQUATIONS

/O1 = A19 * A18 * A17 * A16 * /A15 * /A14 * /A13
/O2 = A19 * A18 * A17 * A16 * /A15 * /A14 * A13
/O3 = A19 * A18 * A17 * A16 * /A15 * A14 * /A13
/O4 = A19 * A18 * A17 * A16 * /A15 * /A14 * A13
/O5 = A19 * A18 * A17 * A16 * A15 * /A14 * /A13
/O6 = A19 * A18 * A17 * A16 * A15 * A14 * A13
/O7 = A19 * A18 * A17 * A16 * A15 * /A14 * /A13
/O8 = A19 * A18 * A17 * A16 * A15 * /A14 * A13
```

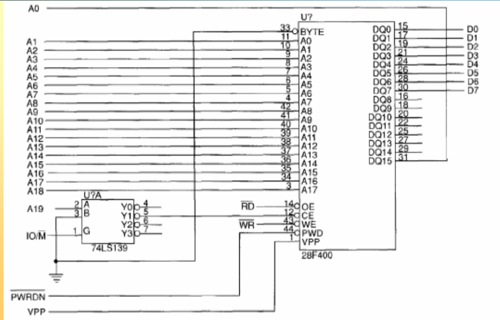
A 512K byte static memory system using 16 62255 SRAMS

Buffering is necessary to the memory when many devices are connected to the same address and data buses in a single system.

Interfacing Flash Memory to CPU

- Differences between a flash memory device and SRAM
  - the flash memory device requires a 12 V programming voltage (V_{pp}) to erase and write new data
  - the amount of time required to accomplish a write operation (~0.4s in flash memory c.w. ~10ns in SRAM)

Interfacing Flash Memory to CPU



The 28F400 flash memory device interfaced to the 8088 microprocessor

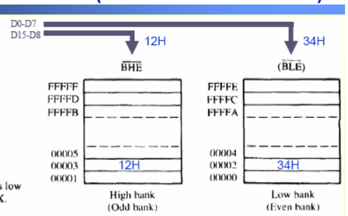
Interfacing Flash Memory to CPU

- The Intel 28F400 flash memory device can be used as either a 512Kx8 (byte mode) memory device or as a 256Kx16 (word mode) memory device, which is selected by the control pin BYTE.
- The pin DQ15 of 28F400 functions as the least significant address input when the device operates in the byte mode.

16-bit Memory Interface (8086/80286/80386SX)

Example:
Address =
0000 0000 0000 0000 001-

Case 1: $\overline{BHE}$=0, $\overline{BLE}$=1
Case 2: $\overline{BHE}$=1, $\overline{BLE}$=0
Case 3: $\overline{BHE}$=0, $\overline{BLE}$=0



Note: A_0 is labeled $\overline{BLE}$ (Bus low enable) on the 80386SX.

The high (odd) and low (even) 8-bit memory banks of the 8086/8028J6/83865X microprocessors

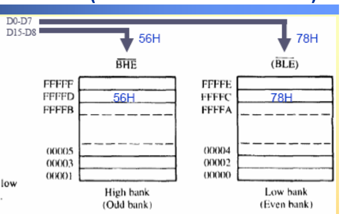
Memory bank selection using BHE and BLE (A_0)

$\overline{BHE}$	$\overline{BLE}$ (A_0)	Function
0	0	Both banks enabled for a 16-bit transfer
0	1	High bank enabled for an 8-bit transfer
1	0	Low bank enabled for an 8-bit transfer
1	1	No banks enabled

16-bit Memory Interface (8086/80286/80386SX)

Example:
Address =
1111 1111 1111 1111 110-

Case 1: $\overline{BHE}$=0, $\overline{BLE}$=1
Case 2: $\overline{BHE}$=1, $\overline{BLE}$=0
Case 3: $\overline{BHE}$=0, $\overline{BLE}$=0



Note: A_0 is labeled $\overline{BLE}$ (Bus low enable) on the 80386SX.

The high (odd) and low (even) 8-bit memory banks of the 8086/8028J6/83865X microprocessors

Memory bank selection using BHE and BLE (A_0)

$\overline{BHE}$	$\overline{BLE}$ (A_0)	Function
0	0	Both banks enabled for a 16-bit transfer
0	1	High bank enabled for an 8-bit transfer
1	0	Low bank enabled for an 8-bit transfer
1	1	No banks enabled

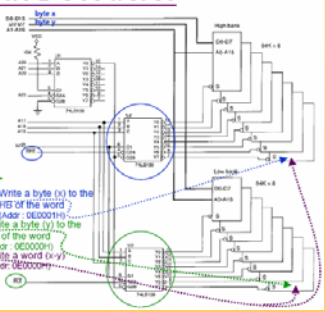
16-bit Memory Interface (8086/80286/80386SX)

- Bank selection can be accomplished in two ways:
  - Issue a separate write signal to select a write to each bank of memory.
  - Use separate decoders for each bank.
- No separate read signal is required for each memory bank as processors can select the data they need at any time from half of the data bus.

16-bit Memory Interface (8086/80286/80386SX)

Separate Bank Decoders:

- It's more expensive and less effective.



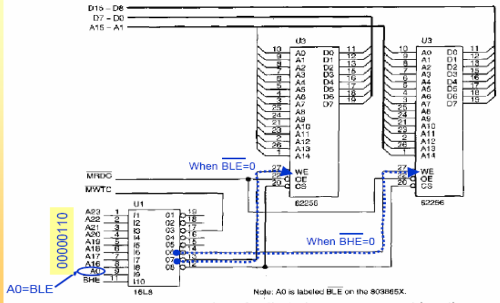
Example:
Address =
0000 1110 0000 0000 0000 000-

Case 1: $\overline{BHE}$=0, $\overline{BLE}$=1

Case 2: $\overline{BHE}$=1, $\overline{BLE}$=0

Case 3: $\overline{BHE}$=0, $\overline{BLE}$=0

Example of a 16-bit Memory Interface

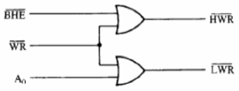


A 16-bit memory decoder that places memory at locations 060000H–06FFFFH

16-bit Memory Interface (8086/80286/80386SX)

Separate Bank Write Strobes:

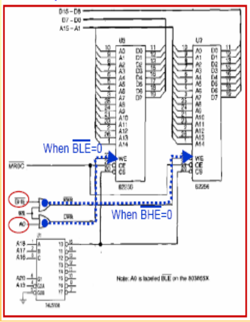
- It's more effective and less expensive.



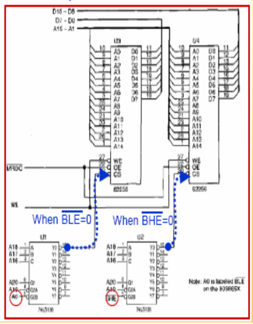

```
TITLE       Address Decoder
CHIP        DECODER2 PAL16L8

;pins 1   2   3   4   5   6   7   8   9  10
     A23 A22 A21 A20 A19 A18 A17 A16 A0 GND

;pins 11  12  13  14  15 16 17   18 19 20
      BHE SEL LWR HWR NC NC MWTC NC NC VCC

EQUATIONS
/SEL = /A23 * /A22 * /A21 * /A20 * /A19 * A18 * A17 * /A16
/LWR = /MWTC * /AO
/HWR = /MWTC * /BHE
```


Example of a 16-bit Memory Interface

Separate bank write strobes

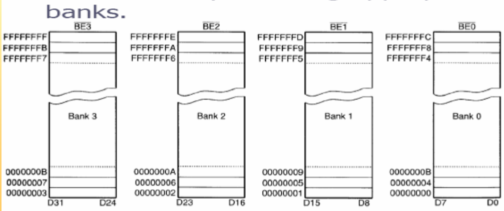


Separate bank decoders



32-bit Memory Interface (80386DX and 80486)

- Bank selection can be accomplished by the bank selection signals $\overline{BE3}$, $\overline{BE2}$, $\overline{BE1}$ and $\overline{BE0}$.
- A 8-, 16- or 32-bit number can be transferred by selecting appropriate banks.

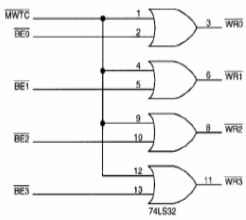


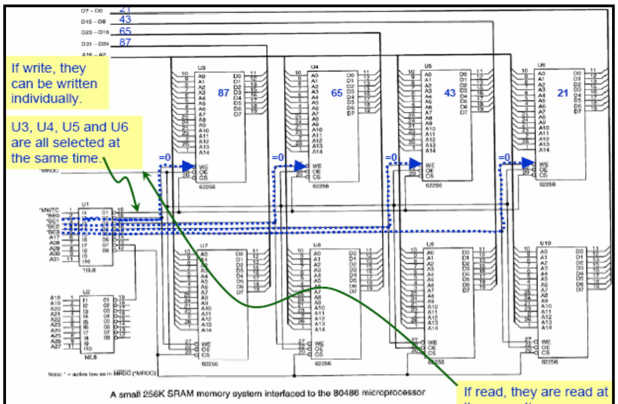
The memory organization for the 80386DX and 80486 microprocessors

32-bit Memory Interface (80386DX and 80486)

- Separate write strobe signals can be developed to select a write to each memory bank.

Bank write signals for the 80386DX and 80486 microprocessors



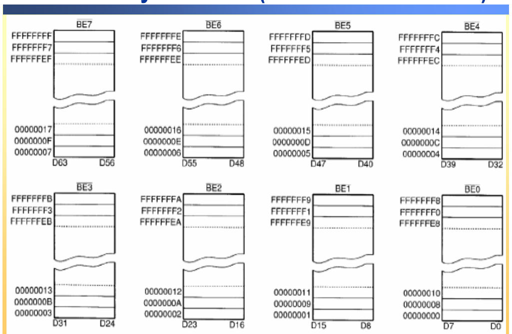


A small 256K SRAM memory system interfaced to the 80486 microprocessor

64-bit Memory Interface (Pentium Processors)

- Bank selection is accomplished by the bank selection signals $\overline{BE7}$, $\overline{BE6}$, $\overline{BE5}$, $\overline{BE4}$, $\overline{BE3}$, $\overline{BE2}$, $\overline{BE1}$ and $\overline{BE0}$.
- A 8-, 16-, 32- or 64-bit number can be transferred by selecting appropriate banks.

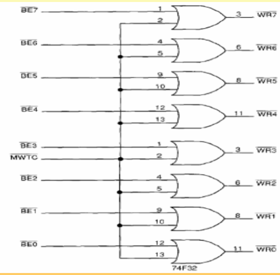
64-bit Memory Interface (Pentium Processors)



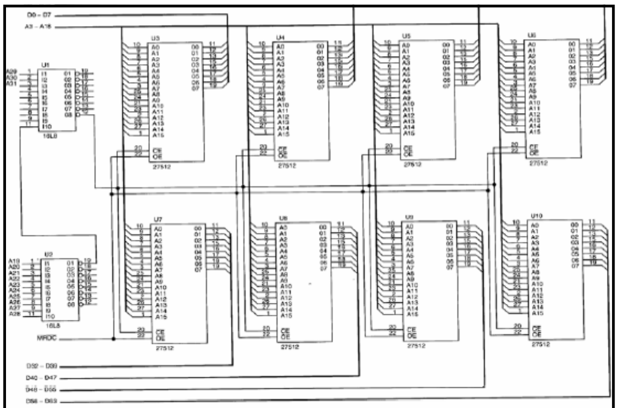
The memory organization of the Pentium microprocessor

64-bit Memory Interface (Pentium Processors)

The separate write strobe signals can be obtained by combining the bank selection signals with the /MWTC signal, which is generated by combining the M/IO and W/R.



The generation of the write strobes for the Pentium microprocessor


A small 512K-byte EPROM memory interfaced to the Pentium microprocessor

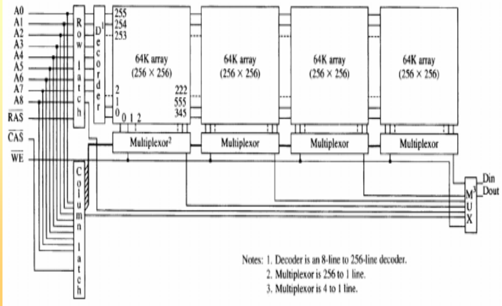
Dynamic RAM

- As RAM memory is often very large, sometimes it is desirable to use DRAMs instead of SRAMs to reduce the system cost.
- A DRAM retains data for only 2-4 ms and requires the multiplexing of address inputs.
- DRAM must be refreshed periodically.

Dynamic RAM

- A refresh can be accomplished by doing a read, a write, or a special refresh cycle that doesn't read or write data.
- A refresh cycle which is totally internal to the DRAM is called either hidden refresh, transparent refresh, or sometimes cycle stealing.
- In a hidden refresh, an $\overline{RAS}$-only cycle strobes a row address into the DRAM to select a row of bits to be refreshed, and, at the same time, the $\overline{RAS}$ input also causes the selected row to be read out internally and rewritten into the selected bits.

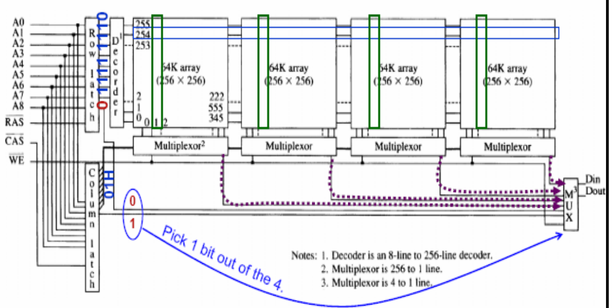
Internal Structure of a Dynamic RAM



The internal structure of a 256K × 1 DRAM. Note that each of the internal 256 words are 1,025-bits wide.

Step 1. Give a 9-bit row addr (e.g.0 1111 1110B)
Step 2: Give a 9-bit col addr (e.g.1 0000 0001B)
Step 3: Get a bit

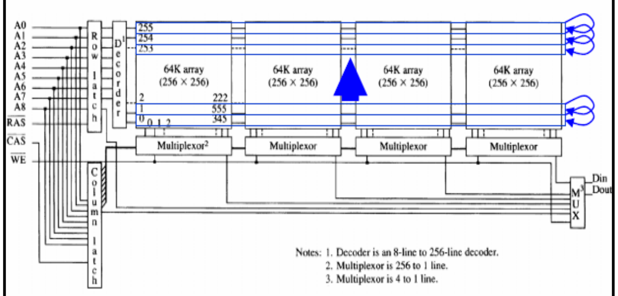
Read Example



The internal structure of a 256K × 1 DRAM. Note that each of the internal 256 words are 1,025-bits wide.

Refresh Example

Step 0: DRAM controller initializes ROW addr counter = 0
Step 1. Give a 9-bit row addr by a DRAM controller
Step 2: read & write the row internally
Step 3: the DRAM controller increases the ROW addr by 1
Step 4: IF ROW addr<256 GOTO step 1, else GOTO step 0



The internal structure of a 256K × 1 DRAM. Note that each of the internal 256 words are 1,025-bits wide.

Refresh Overhead

- The row address is usually obtained from a binary counter.
- An overhead is required to perform the refresh when exploiting the DRAM.
- In a 8088/8086 system running at a 5 MHz clock, it takes 800 ns to do a refresh.
- If there are 256 rows to be refreshed within 4 ms, then the refresh cycle must be activated at least once every 4ms/256 = 15.6μs to meet the specification.
- In other words, it takes 800ns to refresh every 15.6ms.
- Hence, the overhead = 0.8/15.6 ≈ 5.13%.

Refresh Overhead

- In a modern microprocessors such as a 3.0 GHz Pentium IV, the period of 15.6μs is a great deal of time!
- Since this Pentium IV executes an instruction in about one-third ns, the refreshment overhead is much less than 1%, as shown below:

$$\text{Overhead Time\%} = \frac{0.333\text{ ns}}{15.6\mu s} \times 100\% = \frac{111}{52000}\%$$

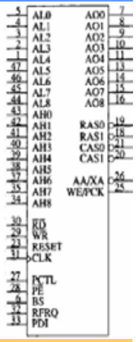
$$\text{Overhead Time\%} \approx 0.0021\%$$

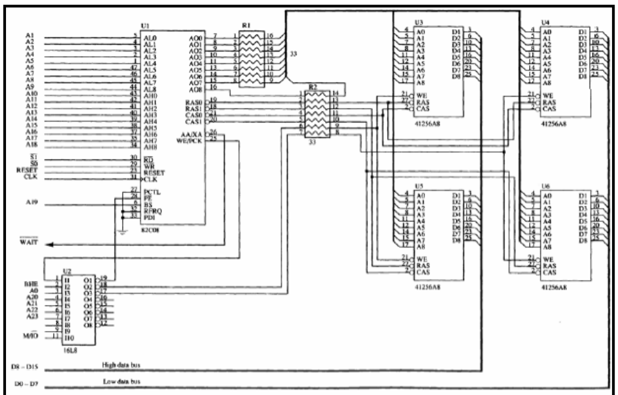
EDO Memory

- EDO (Extended data Output) RAM is a modified version of DRAM. In this type of memory, any memory access, including a refresh, stores the selected row of bits into latches. Thus, in most programs, which are sequentially executed, the row of latched data are available without any wait states.
- This slight modification to the internal architecture of the DRAM increases system performance by about 15 to 25%.
- Although EDO memory is no longer available, this technique is still employed in all modern DRAM chips (SDRAM and DDRAM).

The Intel 82C08 DRAM Controller

- The Intel 82C08 DRAM controller can control up to 2 banks of 256K×16 DRAM memory (i.e. 1M byte of memory).
- The 82C08 contains an address multiplexer that multiplexes an 18-bit address onto 9 address connections for 256K memory devices and circuitry that generates $\overline{RAS}$ and $\overline{CAS}$ signals for the DRAM.





A 1M-byte memory system using four 256K SIMM memory devices and the 82C08 DRAM controller. This section of memory is decoded at locations 000000H–0FFFFFH by the PAL 16L8.