DESIGN CONSIDERATIONS

TMS4164A and TMS4416 Input Diode Protection

The 64K DRAM family from Texas Instruments has departed from conventional input schemes for DRAMs to provide the user with improved ESD protection and input clamping diodes for negative undershoot. Both enhancements of device capability are possible due to the use by TI of a grounded epitaxial substrate in the manufacture of its 64K DRAMs. While the input circuit technique has provided the user with additional protection and ease of use, it may cause anomalous testing results for the unwary test engineer who tries to drive the inputs to large negative voltages.

Figure 1 shows an equivalent circuit for the input circuitry of the TMS4164A and the TMS4416.

The diode and transistor clamping circuit is the essential element in protecting the device from electrostatic discharge (ESD) damage. Figure 2 shows the physical layout of this circuit.

The essential element of the circuit is the input diode (A), which is surrounded by a diffused guard ring (B) connected to V_{SS}. This circuit can be viewed as a combination of a lateral NPN transistor, a bipolar diode, and a thick field transistor — all occupying the same area and connected to the input pad. The P – /P + substrate is both the base of the transistor and the anode of the diode. Both are connected to V_{SS} through the resistance of the substrate from the surface of the chip to the backside. During an ESD with positive voltage, the input diffused area goes into reverse-bias breakdown, which turns on the bipolar transistor, thus clamping the input voltage. The action of the transistor is identical to second breakdown observed in conventional bipolar transistors. Once the transistor turns on, it can sink a large amount of transient current which is evenly distributed over the area of the input diffusion (collector of the lateral transistor). This avoids localized heating from the energy in the ESD. Localized heating could destroy the integrity of the input diode. For ESD with negative voltage, the diode and the transistor act to clamp the input voltage. When the input voltage drops below −0.7 V, the input diffusion appears as a cathode for a diode tied through the substrate resistance to ground. It also acts as an emitter for the lateral NPN transistor. Both elements turn on and tend to uniformly source the current in the input diffusion.

The polysilicon resistor included in the input circuit serves to limit the amount of voltage that reaches the thin oxide associated with the address buffers and clock inputs.

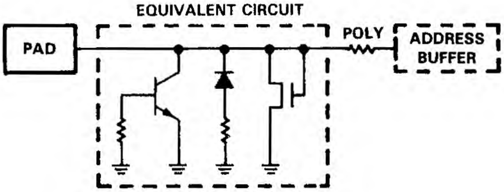
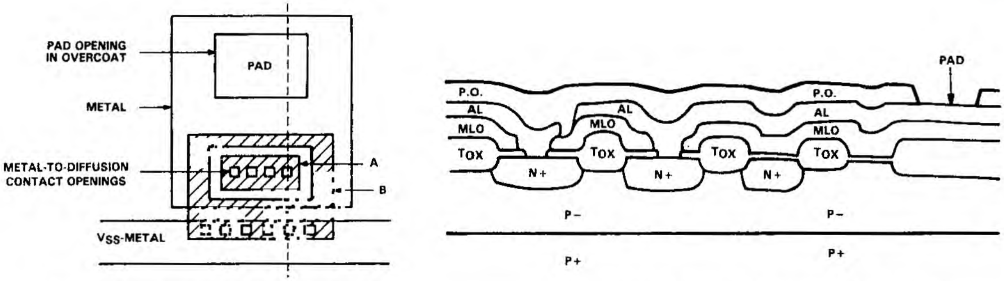


Figure 1. Equivalent Input Circuitry



(A) DIODE AND TRANSISTOR CLAMPING CIRCUIT

(B) CROSS SECTION OF CLAMPING CIRCUIT

Figure 2.

The dynamic impedance of the input clamping circuit is considerably lower than the resistance of the polysilicon resistor.

The input circuit also offers the advantage of clamping negative undershoots on the inputs during normal operation. While this provides advantage to the board and system designer, it can cause confusion for the test engineer unless he fully understands the limits of his tester. DRAMs have historically been specified with negative dc input voltages of − 1 V. In addition, they are often tested/characterized to − 3 V. This testing has been done to ensure that the devices will operate correctly with a negative input undershoot, which is transient. Such testing was required due to the inability of a MOSFET of reasonable size on the chip to clamp the negative-going input and due to the susceptibility of address input buffers on some MOS RAMs to negative input undershoots. The input clamping mechanism, provided on the TMS4164A and the TMS4416, can supply sufficient current to clamp the input transient.

Difficulty in testing the device with negative dc input voltages can occur due to the tester's output driver devices going into saturation when forward biasing the input diode. Also, most testers are unable to supply the large transient current requirement during reversal of bias on the input diode and transistor. Both effects will result in distortion of the tester's waveforms. What may appear to be poor setup and hold time margins of the device may actually be a tester's inability to supply the correct waveforms to the device at the proper time.

The improvement in both ESD protection and signal undershoot on system boards offered by the input circuit may be overlooked if erroneous conclusions are drawn from incoming testing with negative dc input voltages below −0.7 V.

TESTER LIMITATIONS WITH PROGRAMMED INPUT LOW LEVELS OF LESS THAN −0.7 V

Driver distortion occurs when input low levels are programmed for values below −0.7 V. The input diode/lateral NPN transistor shares a common PN junction which becomes forward biased at −0.7 V and below. The transistor collector, which is tied to V_{SS}, carries most of the forward-bias current (see Figure 3).

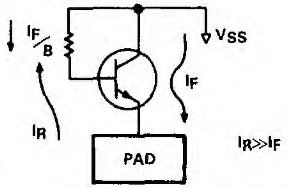
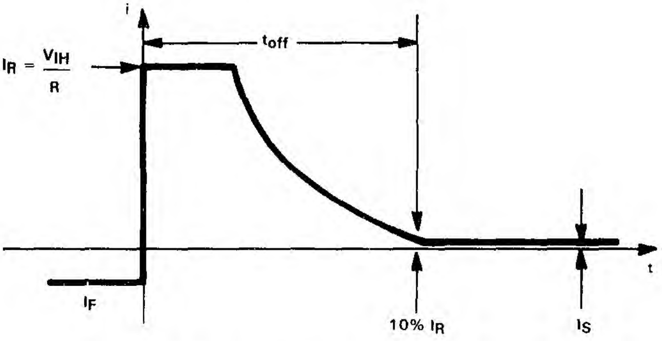


Figure 3. Forward- to Reverse-Bias Current

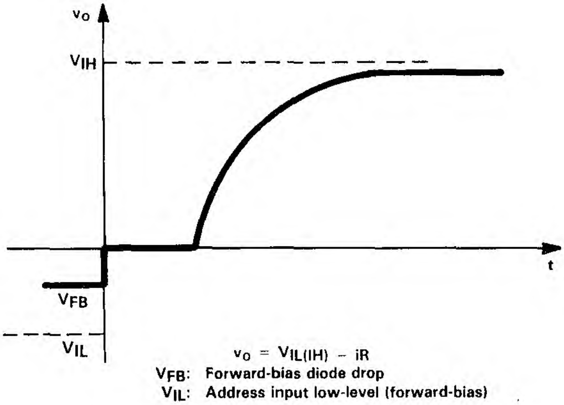
The diode exhibits a classical forward- to reverse-bias recovery delay due to a momentarily large reverse current of amplitude limited only by the programmed reverse voltage V_{IH} and driver output impedance R, i.e., the input approximates a momentary short circuit. An initially large short-circuit current plateau ($I_R = V_{IH}/R$) subsequently relaxes to the normal dc reverse-bias current I_S (see Figure 4). The time from the positive transition edge, corresponding to t = 0 in Figure 4, to the point where the reverse current surge relaxes to 10% of the plateau value is the diode recovery time (t_{off}).

Figure 5 shows how the driver output waveform is altered. During forward bias, the transistor clips the negative level at V_{FB} (in the range −0.7 V to − 1.4 V) with the V_{IL}



R: Driver impedance
V_{IH}: Address input high-level (reverse-bias)

Figure 4. Forward- to Reverse-Bias Recovery Current



$v_O = V_{IL(IH)} - iR$
V_{FB}: Forward-bias diode drop
V_{IL}: Address input low-level (forward-bias)

Figure 5. Address Driver Output Voltage

level programmed over the range of −0.7 V to −6 V. During the initial reverse bias, the driver output voltage v_O is given by the programmed V_{IH} level minus the iR drop across the driver output impedance, i.e.,

$$v_O = V_{IH} - iR$$

During the current plateau, the v_O value is essentially 0 V. The driver output voltage then recovers to the programmed value as the diode reverse current relaxes to the dc reverse-bias level I_S. In effect, the t_{off} value is a measure of the time that the output waveform is distorted if the unwary engineer programs V_{IL} values significantly below −0.7 V.

The following equation derived from diode switching theory serves to estimate the magnitude of the t_{off} values for input levels below −1 V.

$$t_{off} = 40\ [r - [r/(r + 1)]^2]\ \text{ns}$$
$$\text{where } r = -(V_{IL} + 1)/V_{IH}$$
$$\text{and } V_{IL} < -1\ \text{V}$$

The coefficient 40 ns is a characteristic of the diode structure and physical parameters of the material. For example, $V_{IL} = -3$ V and $V_{IH} = +3$ V give $t_{off} = 20$ ns. This estimate is not accurate at very small forward-bias values, because it ignores the rise time of the driver's positive transition edge. As long as the predicted value is greater than or equal to the edge transition time, the estimate is good. It is assumed that driver output impedance has the same value R at both upper and lower levels, V_{IH} and V_{IL}.

The distortion in driver waveform, shown in Figure 5, increases as the driver input low level is progressively driven more negative. Depending on driver output impedance, only slight distortion is observed in the positive transition for input levels near −0.7 V to −1 V. This irregularity corresponds to the onset of the recovery phenomenon short-circuiting the output of the driver. Significant distortion occurs at large negative values of V_{IL}, and the test engineer must be aware of this phenomenon to prevent erroneous conclusions as to the performance of the device.

The input transistor provides great advantage to device use in a system environment. In a system, the negative undershoot of an address line is caused by transient transmission-line reflections (undershoot of negative-edge transitions). Here the input transistor clips much of the swing below −0.7 V on the address line. Positive-edge transition from a settled negative address low level, which gives rise to the forward-to reverse-bias recovery delay, does not occur in the typical system environment.

TMS4164 and TMS4416 Interlock Clock

The TMS4164 (64K × 1) and TMS4416 (16K × 4) dynamic RAMs use a novel interlocked clock to yield enhanced immunity to process variations, temperature, and voltage induced parametric changes. The basic concept of an interlock clock structure is to provide a synchronous timing operation that eliminates race conditions. As an aid to understanding the interlock clock, an overview of the memory control structure and its functions will be presented first.

The TMS4164 (Figure 1) and TMS4416 (Figure 2) need a minimum of 16 address bits to address all of their 64K memory locations (2^{16} = 65,536). Instead of physically using 16 address pins, the DRAMs only use 8 address pins and receive the addresses in two parts of 8 bits each (8 (row) and 6 (column) in the case of the TMS4416). The first 8 addresses are called the row addresses; once stable on the address pins, they are latched by the low going edge of the row address strobe ($\overline{RAS}$) input. The 8 column address bits are then set up on the 8 address pins and latched by the low going edge of the column address strobe ($\overline{CAS}$) input. The TMS4416 only uses 6 of the column address lines, disregarding A0 and A7 (These will be utilized in next generation parts providing an address for 64K × 4 memories). This sharing of address lines is known as multiplexing which keeps the number of pins on a package to a minimum.

The TMS4164 and TMS4416 use a square array of memory cells consisting of 256 rows and 256 columns which is divided into an upper and lower half. A word line (which corresponds to a row) is connected to the transfer gates of 256 cells that comprise a row of memory. The transfer gates control access to the data stored on the memory cell capacitor. The bit line (which corresponds to a half of a column) has for each half of the array 128 memory cells and 1 dummy cell connected to it via the transfer gates. Located physically between the two halves of the memory array are 256 sense amps whose inputs connect to the bit lines from each half of the array. The dummy cell provides the reference (V_{REF}) to a sense amp to determine the state of the memory cell.

On an access cycle, the row decoders drive the selected word line high turning on all 256 transfer gates in the selected row and connect 1 memory cell to each bit line. Concurrently, dummy enable (DE) decodes and drives the transfer gates of one of the rows of dummy cells, and connects 1 dummy cell to each bit line on the opposite side of the sense amps. The dummy selection uses RA7 so that the row of dummy cells selected is on the opposite side of the sense amp from the selected row of memory cells. Connecting the memory and dummy cells to their respective bit lines causes a differential voltage to be established at the inputs of the sense amps. This differential voltage is then detected by the sense amps whose outputs will change to reflect the detected state of the memory cells. After sensing is completed, the output of the sense amp is driven back onto the bit lines to refresh the memory cells. Signal restoration is necessary because an access results in a destructive read (the memory cells no longer contain valid data after the access). This is due to the large bit line capacitance (≈600 fF) and the relatively small capacitance of the cell (50 fF). Connecting the cell to the bit line depletes the cell charge, and makes refresh necessary to ensure valid data retention. This restoration is transparent to the user but should not be confused with providing external refresh. After sensing is completed, the data on the bit lines can now be selected by the column decoders. The column decoders select 4 of the 256 sense amps using A0-A5 (TMS4164) and A1-A6 (TMS4416) for the selection. On the TMS4164, these 4 bits are further decoded by a 1 of 4 decoder using A6 and A7 (the 4 bit output of the TMS4416 eliminates the need for the 1 of 4 decoder). This 1 of 4 selector acts as a bidirectional switch for data transfer to or from the sense amps. Now tha the basic blocks and functions of a DRAM have been described, a detailed look at the interlock scheme will be presented.

A simplified logic representation of the clock structure is shown in Figures 1 and 2. The clock interlock points are shown as inverting input NAND gates. The inputs represent timing events that must be complete before the output of the inverting input NAND gate can trigger a third event; this system provides interlocking. Approximately 60-100 clock signals are generated in a DRAM to control the various functions (address latching, decode timing, sensing, data transfers within the device, etc.); approximately 15 of these have been represented. The following discussion briefly shows the operation of the TMS4164 and TMS4416 DRAMs.

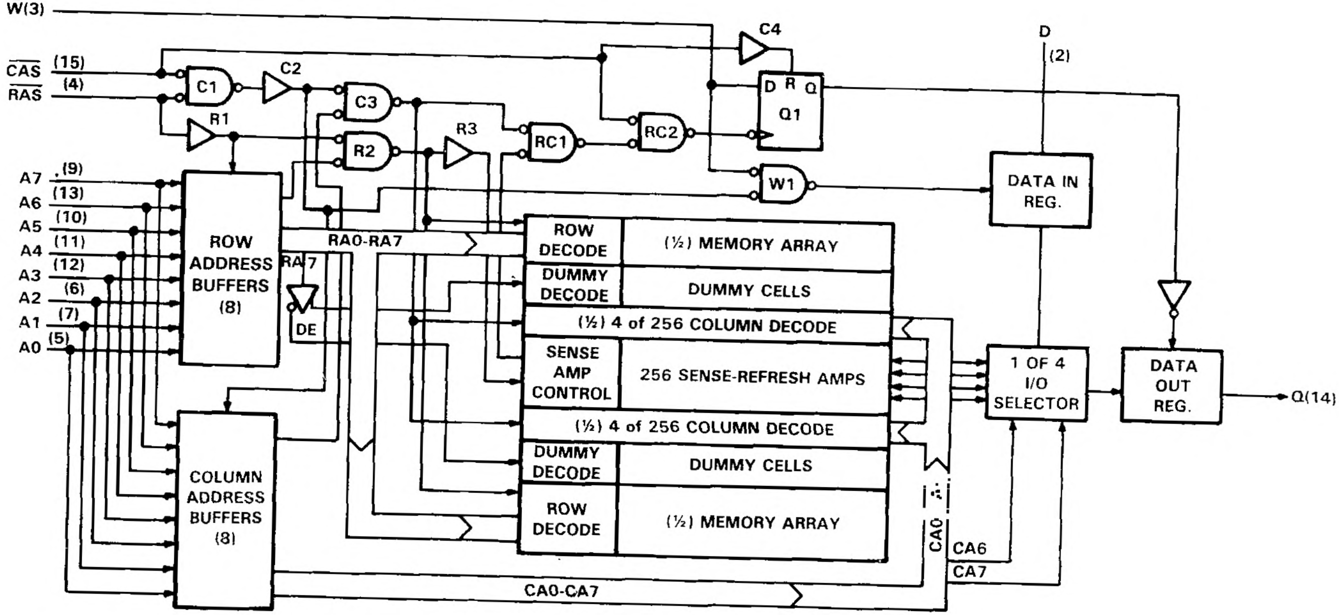


Figure 1. TMS4164 Block Diagram

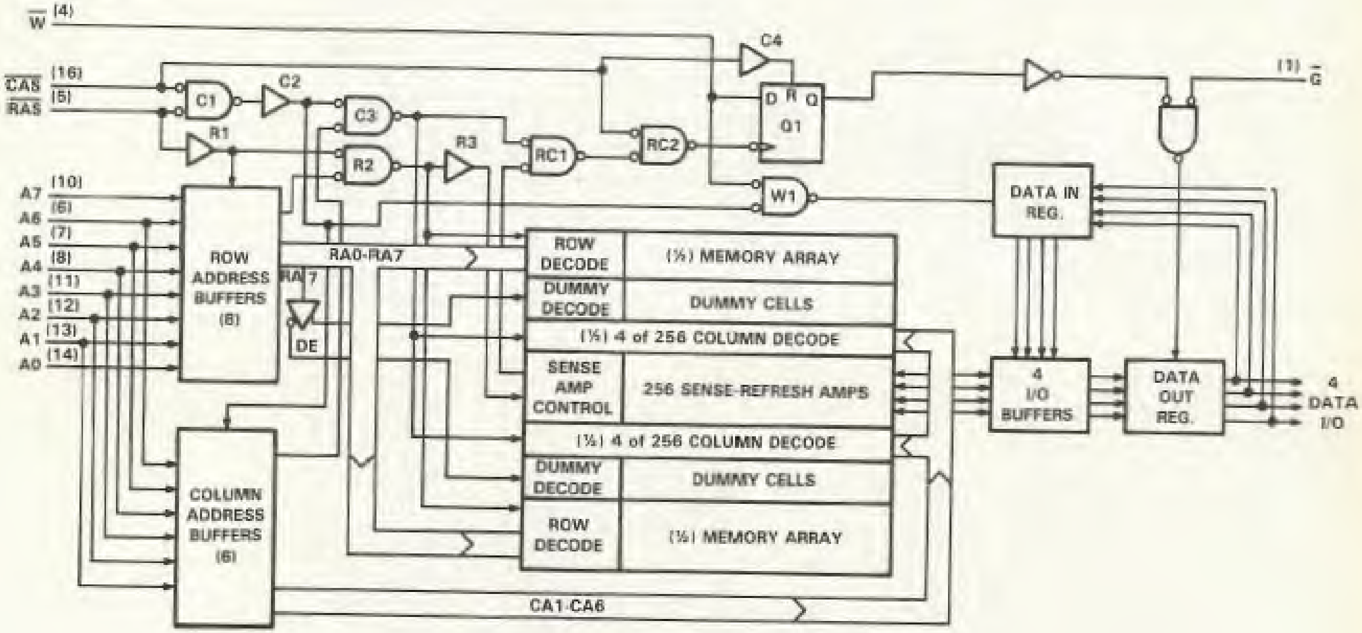


Figure 2. TMS4416 Block Diagram

The falling edge of $\overline{\text{RAS}}$ causes R1 to latch the row addresses into the row address buffers and enables interlock point R2. The row addresses are then amplified and drive the row decoders for row selection. When RA0-RA7 are valid, the row address buffers output a signal to interlock point R2. A delay stage within R2 allows the row decoders time to complete their decoding before the output of R2 goes low. R2 going low enables the row decoders to drive the selected word line high. Interlock R2 ensures two things: the row addresses are valid, and decoding is complete before the selected word line is activated. Address RA7 causes dummy enable (DE) to select the row of dummy cells on the opposite side of the array from the selected row of memory cells. After row and dummy selection is completed, the decoders then drive the appropriate word lines high, connecting the memory and dummy cells to their corresponding bit lines. The differential voltage at the inputs of the sense amp is sensed, amplified, and driven back onto the bit lines; this refreshes the memory cells in the selected row. The sense amp control then outputs a signal to interlock RC1 that indicates sensing is complete.

A high logic level on $\overline{\text{CAS}}$ holds the reset on Q1 active and forces the Q output of the data out buffer into a high-impedance state. A logic low level on $\overline{\text{CAS}}$ removes the reset to allow clocking.

The falling edge of $\overline{\text{CAS}}$ causes interlock C1 to go low (assuming $\overline{\text{RAS}}$ low) driving C2 low to latch the column addresses into the column address buffers. Interlock C1 ensures that the $\overline{\text{CAS}}$ cycle is inactive until $\overline{\text{RAS}}$ is low. The column addresses are then amplified and drive the column decoders for column selection. With CA0-CA7 valid, the column address buffers output a signal to interlock point C3. A delay stage within C3 allows the column decoders time to complete their decoding before the output of C3 goes low. C3 going low enables the column decoders to access the selected columns (4). Interlock C3 ensures two things: the column addresses are valid, and decoding is complete before the selected columns are accessed. After selection is completed, data can now either be input or output depending on the $\overline{\text{W}}$ signal timing. Interlocks RC1 and RC2 ensure that the sense amps are active and the proper column is selected before a read or write can take place.

In the case of a read or read-modify-write cycle, the high logic level on the write line ($\overline{\text{W}}$) prevents any transfer into the data in register by keeping the output of W1 high. The presence of $\overline{\text{CAS}}$ low and the output of RC1 low allows RC2's output to go low; this clocks the level of $\overline{\text{W}}$ into register Q1. Only in the case of an early write ($\overline{\text{W}}$ low prior to $\overline{\text{CAS}}$ low), when the output of Q1 is not clocked to a logic one, will the data out register be maintained in the high-impedance state. In any read cycle, the output of Q1 is a logic one and the data out register is enabled although data will not be valid until $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ access times are both satisfied.

In a write cycle, the low logic level on $\overline{\text{W}}$ allows the output of W1 to go low which latches the data present at D (thus the latter of either $\overline{\text{CAS}}$ or $\overline{\text{W}}$ going low latches the data). The logic level at the output of the data out register will remain until $\overline{\text{CAS}}$ returns to a high level. (When $\overline{\text{CAS}}$ is high, the output will go to a high-impedance state.) Data out reflects the data read from the cell rather than the new data that is written for read-modify-write cycles.

The $\overline{\text{RAS}}$ low time following sensing complete, is used to restore data to the memory cells currently selected by the word line (restoration after the destructive read). Any data that is changed by a write cycle causes alterations of the sense amplifier which then stores the new data in the memory cells. When $\overline{\text{RAS}}$ goes high, the word line is turned off and the cells now hold the data restored from the sense amps. $\overline{\text{RAS}}$ going high initializes a precharge state used to equalize the bit lines by charging them to full V_{DD} potential. Precharge is necessary to ensure the charge on the bit lines is equal on both sides of the sense amp. Another access cycle may begin once the precharge time has been met.

The representation used in Figures 1 and 2 is a simplified logic diagram which does not depict all points of signal interlocking. It does however demonstrate the principle of an interlocked clock scheme. The signal generation and timing becomes very critical as device delays decrease. In many dynamic RAMs there are over 100 timing signals used to control internal operations, and these timing signals are generated using delay chains without interlocking. The signal skew resulting from non-interlocked timing increases device sensitivity to operating conditions and process variations. Although the interlock clock is transparent to the user, its incorporation on the TMS4164 and TMS4416 offers greater component reliability and avoids timing race conditions inherent in previous generation DRAMs.

Introduction to Surface Mounting

ABSTRACT

The demand for high-density, cost-effective printed circuit boards has prompted the electronics industry to seek alternative methods to traditional plated-through-hole technology. One such alternative is surface mounting, a technology traditionally used in hybrid fabrication. The advantages of surface mounting are numerous but the bottom line is that it is cost effective and will begin to displace plated-through-hole technology as the availability of surface-mount components increases.

Texas Instruments is fully supporting the growth of the surface-mount industry with its line of plastic leaded chip carriers. An introduction to the surface-mount technology will be given in this application report.

INTRODUCTION

The post molded leaded chip carrier (PLCC) was developed by Texas Instruments in 1980 to improve the packing density of ICs on printed circuit (PC) boards and overcome some of the size constraints normally caused by dual-in-line (DIP) packages. The PLCC was also designed to be used under the same environmental conditions as the DIP without any reliability degradation. The PLCC occupies approximately 40% to 60% of the PC board area of an equivalent DIP, and requires no through holes (surface mount), therefore, it lowers the cost on PC boards. Unlike some surface-mounted packages, TI's PLCC requires no special PC board material considerations. The design of the lead provides compliance allowing the use of any commercial substrate. Digital, Linear, Gate Array, and MOS devices will be offered in 18-, 20-, 28-, 44-, 52-, 68-, and 84-pin packages through TI.

Package Outline

The mechanical data for the PLCCs is given in Figures 1 and 2; their thermal properties are listed in Table I. The following general statements apply to the packages:

1. Each of the chip carrier packages consists of a circuit mounted on a lead frame and encapsulated within an electrically nonconductive plastic compound. The compound withstands soldering temperatures with no deformation, and circuit performance characteristics remain stable when the devices are operated in high humidity conditions.
2. These packages are intended for surface mounting on solder pads with 1,27-mm (0.050-inch) centers. The leads require no additional cleaning or processing when used in soldered assembly.
3. All dimensions shown are metric units (millimeters), with English units (inches) shown parenthetically. Inch dimensions govern.
4. Lead spacing shall be measured within the zones specified.
5. Tolerances are noncumulative.
6. Lead material CD-155. T60 (Copper Alloy).
7. Dimple in top of package denotes pin 1.

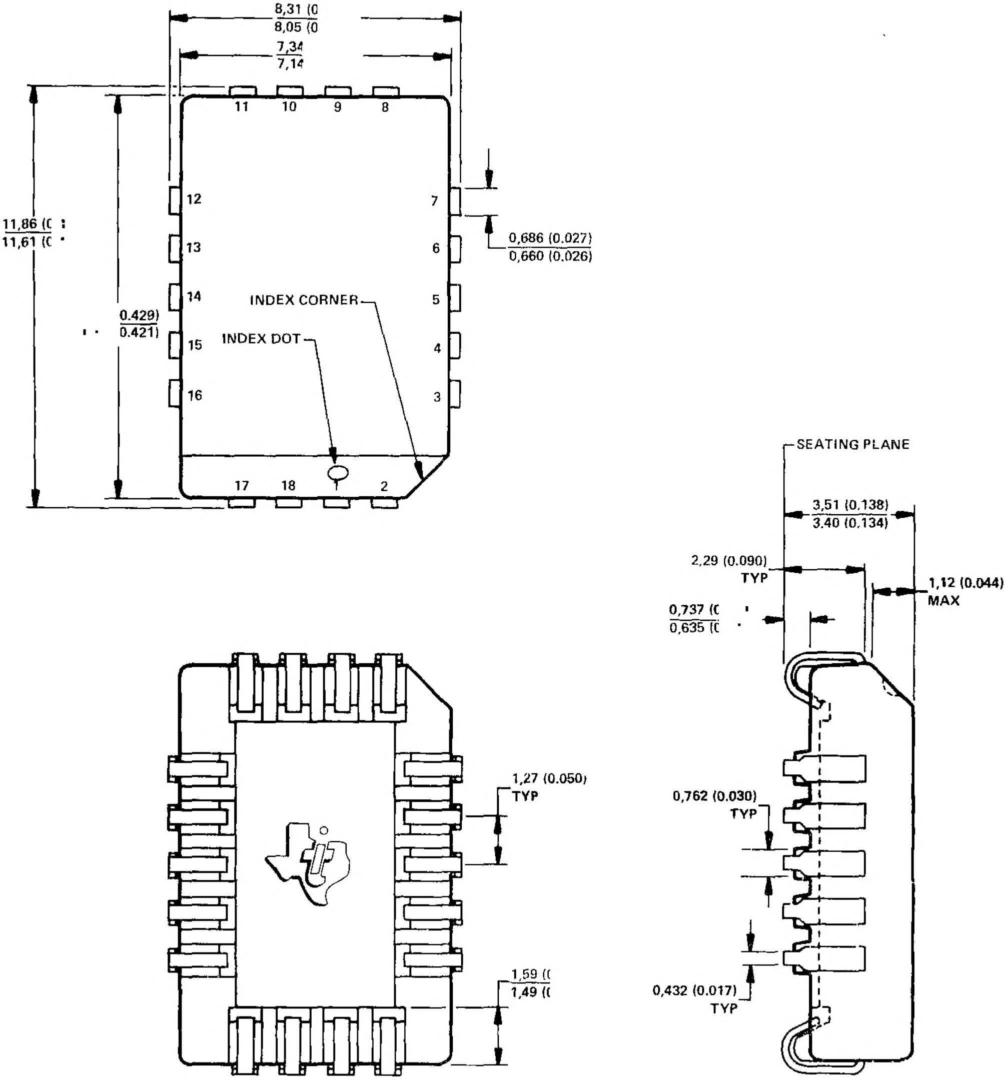
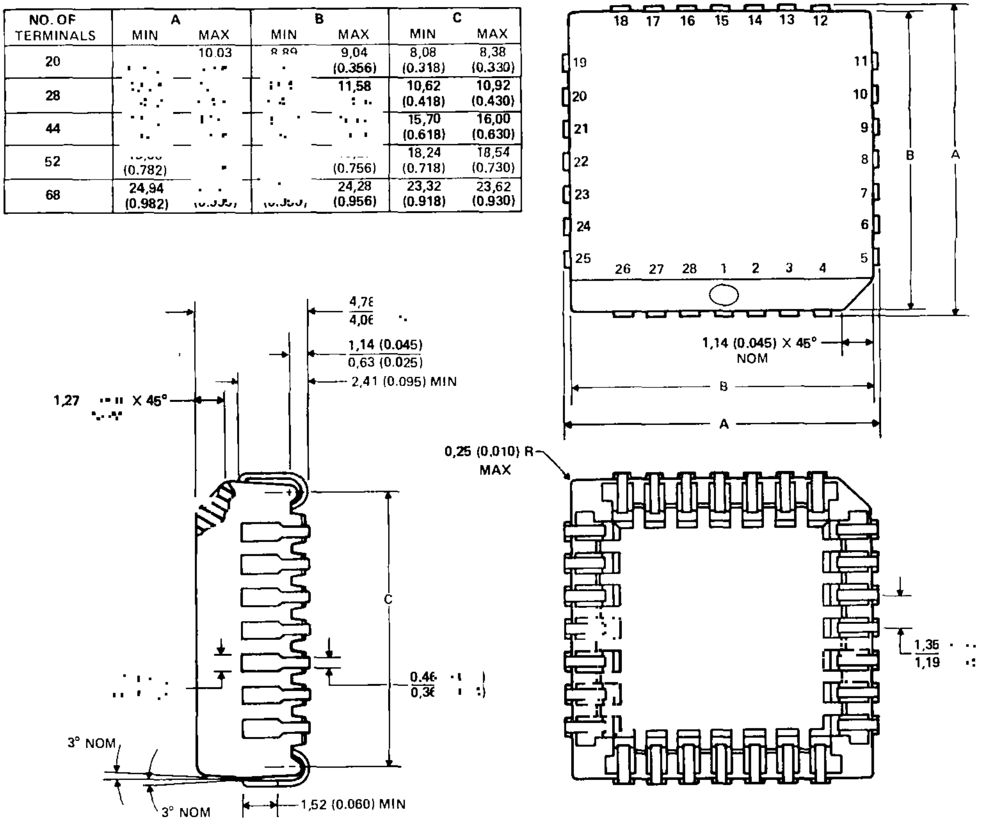


Figure 1. Plastic Chip Carrier Package (FP Suffix)



NO. OF TERMINALS	A MIN	A MAX	B MIN	B MAX	C MIN	C MAX
20	[illegible]	10.03 [illegible]	8,89 [illegible]	9,04 (0.356)	8,08 (0.318)	8,38 (0.330)
28	[illegible]	[illegible]	[illegible]	11,58 [illegible]	10,62 (0.418)	10,92 (0.430)
44	[illegible]	[illegible]	[illegible]	[illegible]	15,70 (0.618)	16,00 (0.630)
52	[illegible] (0.782)	[illegible]	[illegible]	[illegible] (0.756)	18,24 (0.718)	18,54 (0.730)
68	24,94 (0.982)	[illegible]	[illegible]	24,28 (0.956)	23,32 (0.918)	23,62 (0.930)

ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES.

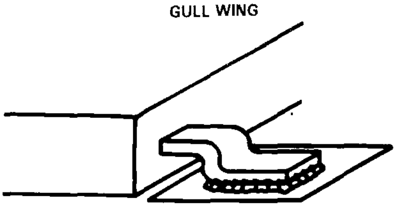
Figure 2. FN Plastic Chip Carrier Package

Table I. Thermal Properties, of Plastic Chip Carriers

NO. OF LEADS	PACKAGE DESIGNATION	θJA (° C/N)	θJC (° C/W)
18	FP	85.4	13.8
20	FN	113.6	37.1
28	FN	76.8	32.2
44	FN	68.0	20.3
68	FN	45.7	11.4

J-Lead Advantage

Texas Instruments PLCC packages are constructed with the J-lead structure due to its superior performance when mounted on a wide spectrum of substrates ranging from ceramic to epoxy-glass. This is possible due to the compliancy of the J-lead which compensates for the possible thermal mismatch between plastic packages and mounting substrates. More care must be taken when using ceramic leadless chip carriers mounted on nonceramic substrates in order to prevent solder joint fracturing under thermal cycling. The J-lead also offers advantages over plastic surface-mount packages using different lead structures. Figure 3 gives a comparison of the J-lead used on the PLCC to the "gull wing" commonly used on small-outline integrated circuits (SOICs) and "quad packs."



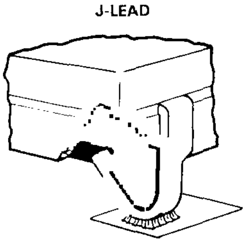
DEVICE AREA (16 L-PIN SOIC) = 111,6 mm^2
(0.173 in^2)

ADVANTAGES

- PROVEN PROCESS
- POSITIVE SOLDER "WITNESS"
- EASY AUTO-POSITIONING
- NESTED STACKING (PERIPHERAL)

DISADVANTAGES

- EXTENDS X-Y SIZE
- LEADS SUBJECT TO DAMAGE
- HIGH PIN COUNT PACKAGES IMPRACTICAL



DEVICE AREA (18-PIN PLCC) = 98.6 mm^2
(0.153 in^2)

ADVANTAGES

- PROVEN PROCESS
- LEADS ARE COMPLIANT, USEABLE WITH PC BOARD AND CERAMIC SUBSTRATES
- MINIMUM X-Y SIZE, MAXIMUM BOARD DENSITY
- EASY AUTO POSITIONING
- LEADS WELL PROTECTED
- EASY REPLACEMENT
- SOCKETING EASY
- JEDEC STANDARDS EXIST
- STAND-OFF FROM THE BOARD ALLOWS EASY CLEANING
- LARGEST LINE OF AVAILABLE PACKAGES: FROM 18 TO 68 LEADS. HIGHER PIN COUNTS UNDER DEVELOPMENT

DISADVANTAGES

- TOTAL PACKAGE-HEIGHT THICKER THAN SOIC
- INFRARED (IR) REFLOW DIFFICULT

Figure 3. Gull Wing Vs. J-Lead

Area Savings with PLCC

The PC board area savings that can be realized with the PLCC is best demonstrated by a comparison of two Texas Instruments one megabyte memory boards (see Figure 4). The DIP board is eight layers, measuring 279,4 mm (11 inches) by 355,6 mm (14 inches) with 226 ICs. The PLCC board is four layers, measuring 165,1 mm (6.5 inches) by 243,84 mm (9.6 inches) also with 226 ICs. The savings that can be realized with the PLCC board amounts to 60% less board area at an overall cost savings of approximately 55%. This illustrates the viability of surface mount as a low cost means of improving circuit board density while reducing PC layout complexity.

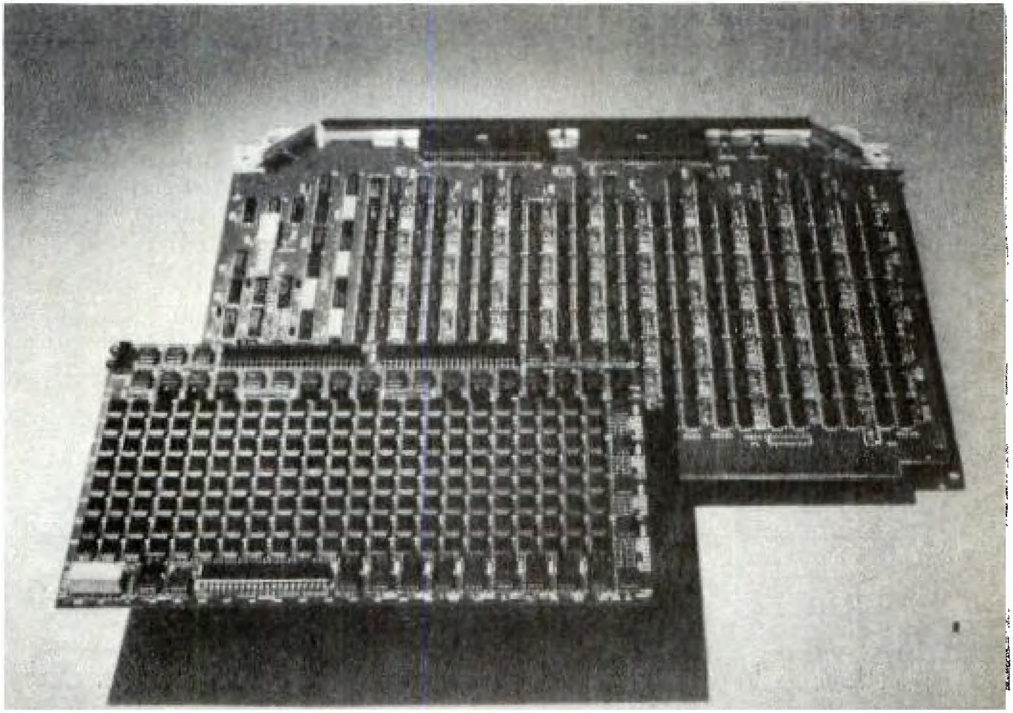


Figure 4. PLCC and DIP One Megabyte Memory Expansion Boards

Surface Mount Component Availability

Most IC manufacturers are presently producing surface-mount components for a large part of their product line. The devices available range from the sophisticated VLSI to the discrete transistor. Non-integrated circuit components ranging from chip resistors and capacitors to surface-mount connectors are also being produced in volume by major manufacturers. As the demand for surface-mount components increases, most products now produced for standard through-hole technology will also be available in surface mount. As of the printing of this application report TI produces over 700 ICs in surface mount packages.

Surface mounting consists of five basic steps:

1. PC board design
2. Solder paste application
3. Component mounting
4. Oven drying (optional)
5. Solder reflow

A brief description of each step will be given; detailed descriptions of the various steps can be obtained by component and equipment suppliers and from numerous technical articles on surface mounting.

PC Board Design

To produce reliable surface-mount PC boards the designer has to pay particular attention to IC solder pad (also termed footprint) layout. Not providing adequate footprint area and proper orientation will generally yield poor solder joints and lack of self-centering during reflow. Figure 5 shows the recommended footprints for the 18-pin PLCC. When laying out the IC footprints, as a general rule the footprint should extend approximately 10-15 mils past the outer edge of the PLCC lead. This provides a good solder fillet that will extend up the outer edge of the PLCC lead to yield a reliable solder joint that is easily inspected. The 70-80 mil length of the footprint should be a minimum, however a longer footprint can be used. It is recommended that the dimensions A and B never be less than the minimum width or length of the IC.

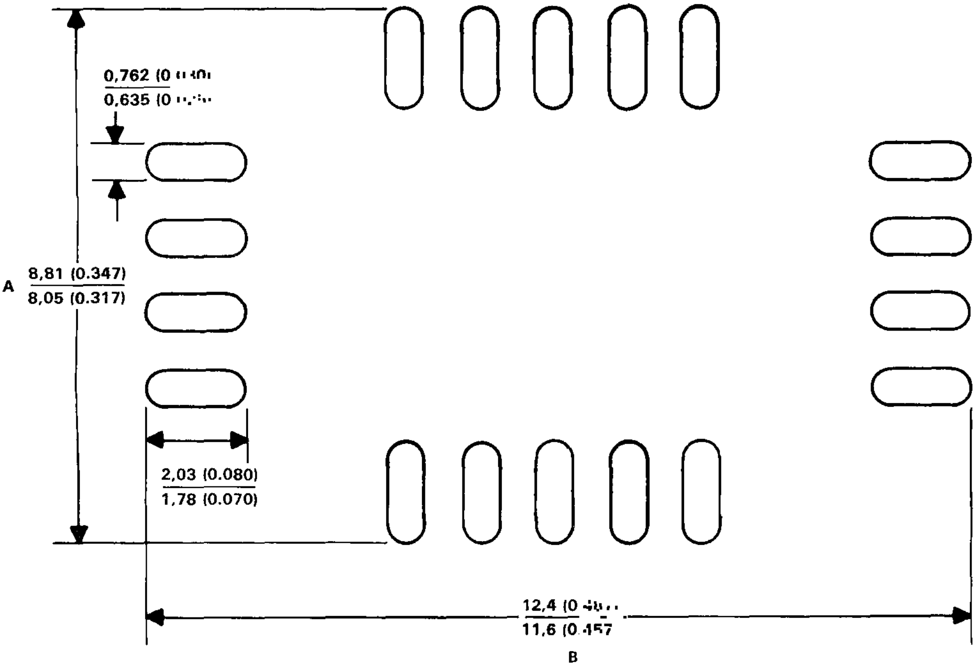


Figure 5. 18-Pin PLCC Footprint

Solder Paste Application

Solder paste can be applied using several methods: screening through a stencil or stainless steel mesh, pneumatic dispensing or by hand application with a syringe. Texas Instruments recommends screening through a stainless steel screen. The screen mesh must be chosen in accordance with the mesh of the solder paste to provide an adequate emulsion of the solder paste and to prevent screen clogging. In general an 80-100 mesh screen should be used with 200 mesh or finer solder paste particle. There are a number of factors that need to be considered when selecting a solder paste, a few key factors are as follows:

1. Particle size
2. Particle shape
3. Percentage of metal content
4. Temperature range

Component Placement

The components can be placed via several different modes into the still moist solder paste. In a production environment, the components are most efficiently placed with an automatic pick-and-place machine to achieve both speed and accuracy. Presently, pick-and-place machines can place between 600 and 600,000 components per hour and are priced accordingly. In a research and development environment, hand placement can be adequate due to the forgiving nature of surface mounting. When a component is placed off center it will tend to self-align during reflow due to the surface tension of the molten solder. Naturally there are limits to the amount of misalignment that can be corrected. Two important aspects of self-alignment are provision of adequate solder pad area, and proper placement of the solder pads with respect to the component.

Oven Drying

As solder pastes have evolved over the past several years, the drying process following component placement is not always necessary. In the past, drying was necessary to drive out the solvents in the solder paste. If the solvents were not driven out, the formation of solder balls was frequent due to the out gassing of the solvents prior to reflow. Today manufacturers of solder pastes report that drying is no longer necessary when using many of the new solder paste formulations. As a wide variety of solder pastes exists, it is necessary to consult the manufacturer before determining if drying is necessary in your process.

Solder Reflow

While several methods of solder reflow are available, vapor phase soldering has been the most successful and is becoming the industry standard.

Two types of vapor phase systems are the batch and the in-line.The batch system is a two-vapor system that uses a fluoroinert liquid such as FC-70 for the primary vapor and a clorofluorocarbon such as trichlorotrifluoroethane (R113) as the secondary liquid (see Figure 6). The secondary liquid has a lower boiling point (47.6°C) than the primary liquid (215°C) thus acting as a blanket to prevent loss of the expensive primary liquid. The in-line system (see Figure 7) is a single-vapor system using only a primary vapor (such as FC-70). The batch system is the forerunner of the in-line and is more suited to development and small production where the in-line is tailored to a mass production atmosphere requiring good throughput and minimal operating expense. Although the two systems are targeted to different markets their basic operation is the same. Both are capable of single and double sided surface mount.

Batch System Operation

The PC board complete with components is placed on an elevator and lowered into the secondary vapor. The elevator ascent-descent rate and dwell in the two vapor zones can be preset via the vapor phase machine front panel. The descent rate and hold time in the secondary zone should be set so as not to unnecessarily disrupt the secondary vapor blanket or cause defluxing of the solder paste. Lowering the board into the 215°C primary zone causes the solder to reflow. A dwell time of 10-30 seconds in the primary zone is generally sufficient for most PC boards. The dwell time in the primary zone is a function of the PC board mass. Once the solder is reflowed, the PC board is raised back into the secondary zone where the molten solder is allowed to solidify. In the batch system it is necessary to pay particular attention to the ascent-descent rate of the elevator as the disruption of the two-vapor zones will cause unnecessary loss of the expensive primary liquid.

In-Line System Operation

The operation of the in-line system is similar to that of the batch system except that there are no secondary vapor or dwell times with which to contend. The PC board is placed on a conveyor belt that transports it through the system at a constant speed. Passing through the vapor zone the solder becomes molten and solidifies as it moves toward the systems exit. Where the ascent-descent rate and dwell time are critical to the batch system, the conveyor speed is critical to the in-line system. The speed at which the conveyor should be set is also a function of the PC board mass.

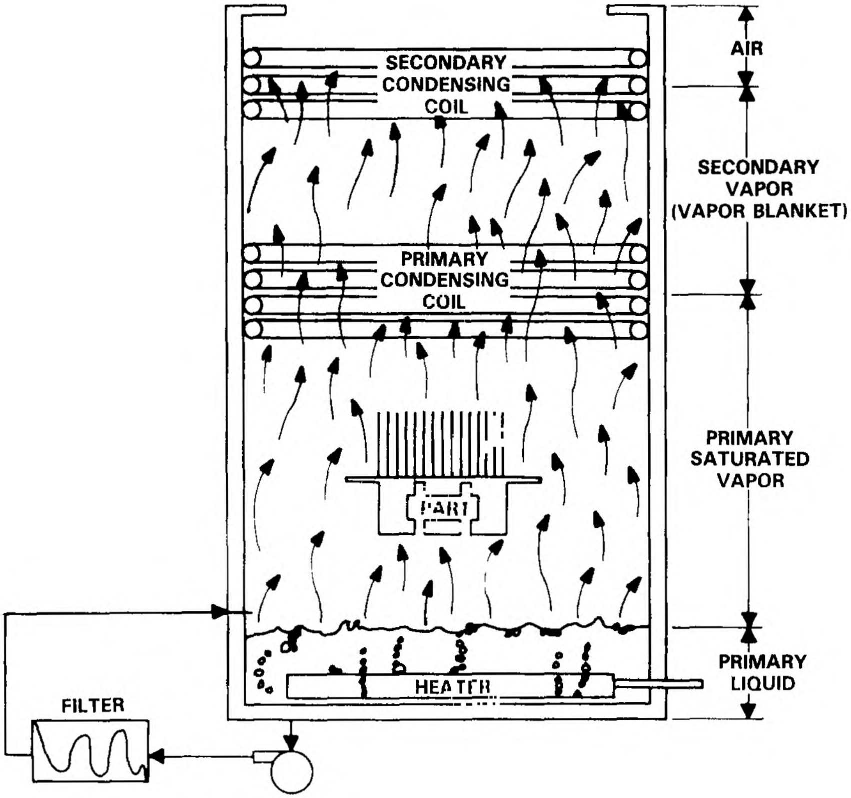


Figure 6. Vapor Phase Reflow System with Secondary Vapor Blanket

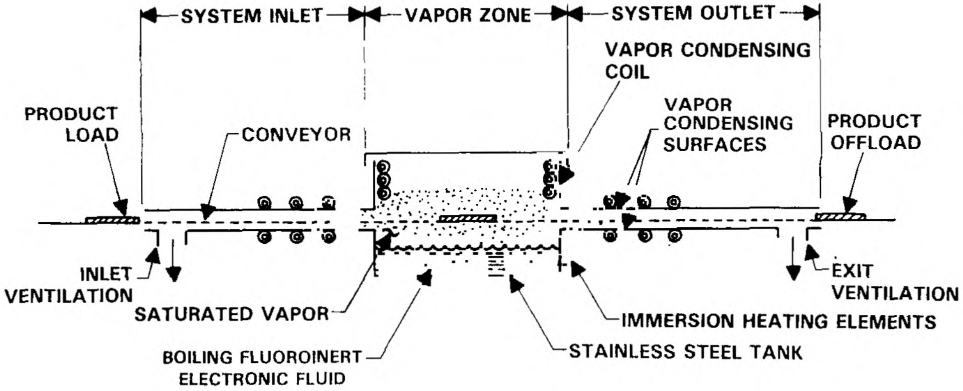


Figure 7. In-Line Single Vapor Heating System Schematic

Cleanup

Following the soldering process, it is necessary to remove the flux residues. These residues can be removed by traditional cleanup methods if the components have approximately 5 mils clearance to the PC board. One benefit of the PLCC with its J-leads is that it provides approximately 29 mils of clearance. Special soaking, agitation, or other methods will be necessary to provide adequate cleanup for components with less that 5 mils of clearance.

CONCLUSION

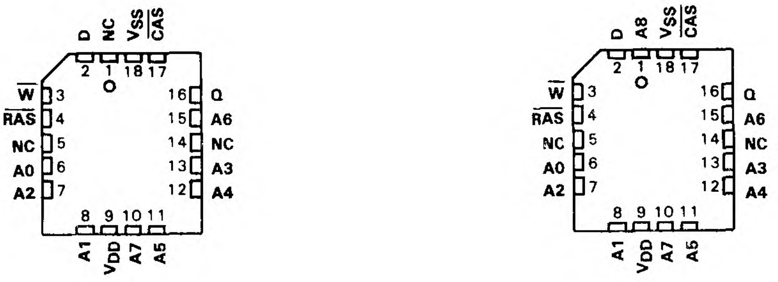
A brief overview of surface-mount technology has been given showing its advantages over standard plated-through-hole technology. Surface mounting is a cost-effective, sensible solution to the ever increasing demand for denser circuit boards. Detailed information about surface mounting is available from most major component and equipment manufacturers and through numerous technical articles on the subject. As the electronics industry strives to implement more functions in a given area, Texas Instruments believes surface mounting will become the predominant mounting technology of the future.

MOS Memory
Applications Engineering

64K-256K Plastic-Leaded Chip Carrier Compatibility

Designing memory arrays compatible with Texas Instruments 64K and 256K DRAMs in plastic leaded chip carrier (PLCC) is easily accomplished through proper PC board design. Unlike the 64K and 256K DIP packages which are identical, the PLCC packages have definite physical differences. This Application Report will cover those differences and how to design for compatibility along with a few general rules on PLCC footprint layout.

To design a 64K-256K compatible PLCC memory array, pin and package outline compatibility need to be considered. Figures 1 and 2 show both devices with their respective pin assignments and package outlines. The pin assignments for both devices are identical with the exception of pin 1. On the 64K package, pin 1 is a no connect (NC) and on the 256K package, pin 1 is address eight (A8). This presents no problem as the 64K device will ignore this input. The major difference between the two packages is their physical size. The 64K device is packaged in the JEDEC standard 290 mil X 425 mil (nominal) PLCC while the 256K will be packaged in the 290 mil X 490 mil PLCC. The increase in package size for the 256K device is due to its larger chip size.



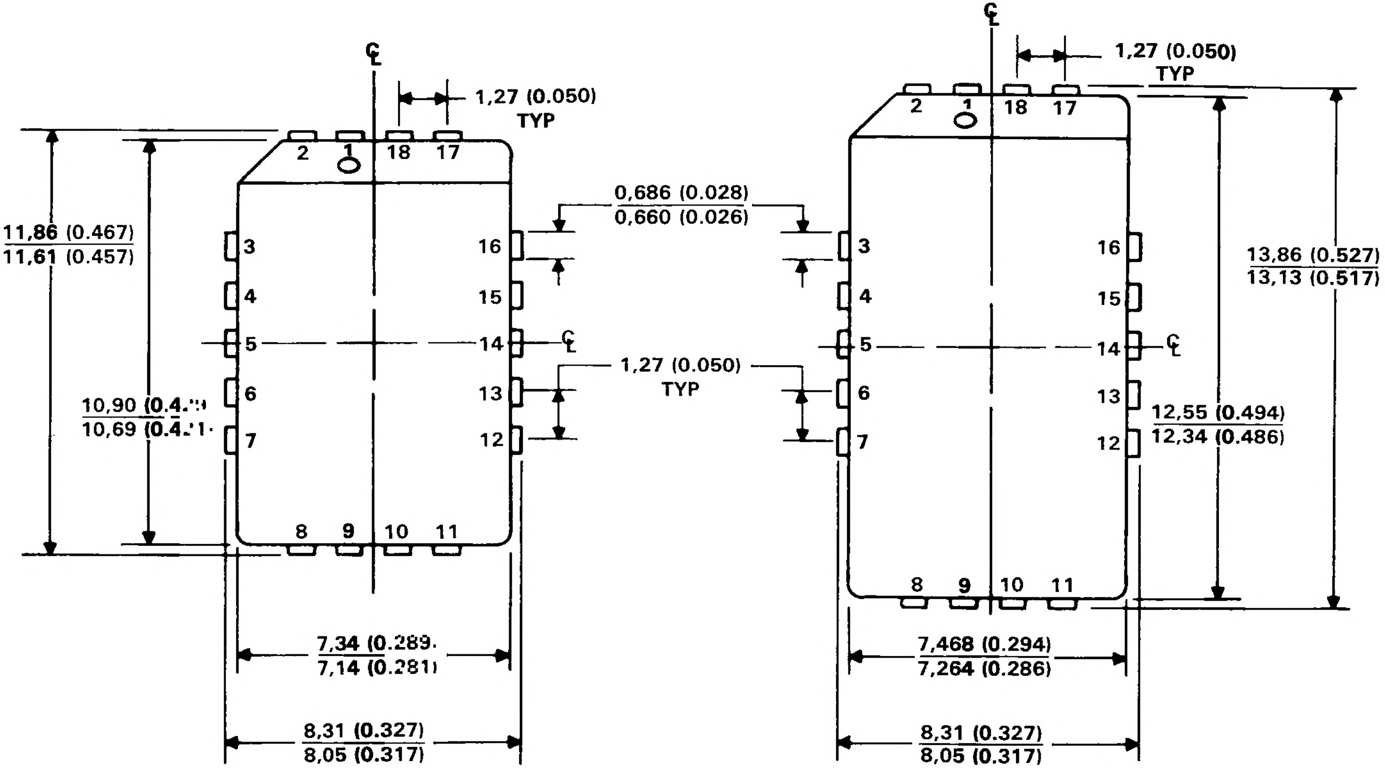
a. 64K X 1

b. 256K X 1

c. 64K X 1/256K X 1

PIN NOMENCLATURE	
A0-A8	Address Inputs
$\overline{\text{CAS}}$	Column Address Strobe
D	Data In
NC	No Connection
Q	Data Out
$\overline{\text{RAS}}$	Row Address Strobe
V_{DD}	+ 5-V Supply
V_{SS}	Ground
$\overline{\text{W}}$	Write Enable

Figure 1. PLCC Pin Assignments



a. 64K X 1

b. 256K X 1

Figure 2. 64K and 256K PLCC Mechanical Data

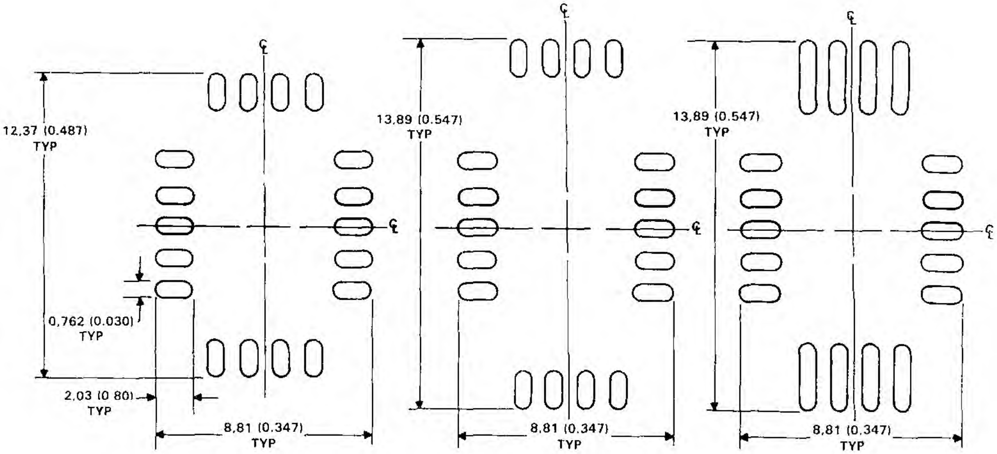


Figure 3. PLCC Footprints

Designing a PLCC memory array to accommodate both package sizes is accomplished through appropriate PLCC footprint layout. The 64K, 256K and 64K-256K PLCC footprints with recommended mechanical dimensions are given in Figure 3. The only difference between the 64K and 256K footprints are the location of the solder pads across the top and bottom of the footprints (necessary to accommodate the differences in package length). By overlaying the two footprints a 64K-256K compatible footprint can be obtained. Notice that the solder pads along both sides are unchanged from the 64K and 256K footprints while the solder pads across the top and bottom have been stretched. The only drawback to the 64K-256K compatible footprint is that when using 64K devices some memory density is sacrificed due to the extra pad length at the top and bottom of the footprint. However, this may not be a serious constraint compared to the cost of laying out and producing two separate memory boards.

To arrive at the PLCC footprint dimensions in Figure 2, several general design rules were used. Adhering to these design rules helps to ensure good solder joint integrity between the PC board and the PLCC.

1. Solder pad length should be 70-80 mils; shorter pads may cause poor solder joints, longer pads can be used but require more board area.
2. Solder pad width should be 25-30 mils.
3. The solder pads should extend a minimum of 10-15 mils past the outer edge of the PLCC leads. This provides a good solder fillet that will extend up the outer edge of the PLCC lead to yield a reliable solder joint that is easily inspected.
4. Footprints must be symmetrical. Solder pads on opposing sides should be of equal length and width. Solder pads on adjacent sides do not have to be of equal length and width.

This Application Report has illustrated that the 64K and 256K PLCC are compatible through proper footprint layout. Also several general footprint design rules were presented to help ensure solder joint integrity.

MOS Memory
Applications Engineering

High-Performance Memory Access with the TMS4161

The dual-port concept of the TMS4161 Multiport Video RAM eliminates the large bottleneck caused by display refresh overhead by dedicating a high-speed serial port to that requirement. This enhances performance by eliminating the need for processor access to share the random-access port with display refresh. The random-access port can also be configured with a system to provide enhanced system performance over a standard memory access cycle. This involves interleaving the display memory such that multiple CPU memory access cycles coincide with a single VRAM memory access cycle. This application report presents a method in which system timing generation supports the access of interleaved banks of display VRAMs.

Figure 1 exemplifies the memory configuration for a 1024 x 1024 resolution, four-plane graphics system. Memory mapping of the display requires 1024 x 1024 or 1M-bits of memory per plane or a total of 64 TMS4161 devices (16 in each of the four planes). Four pixel planes are utilized to enable 16 colors to be displayed on the screen simultaneously.

Four 16-bit parallel-to-serial shift registers are added externally to correspond to the four pixel planes. The serial port of the 16 TMS4161's feed into the 16-bit external shift register which will require a reload of the on-chip 256-bit shift register every 16 × 256 or every 4096 pixels (four scan lines). A TMS4161 shift register reload operation involves the transfer of a designated row of memory to the on-chip 256-bit shift register. This transfer operation is completed within a single memory cycle. In addition, the 256-bit shift register of the TMS4161 can be segmented into cascaded 64-bit increments by virtue of a selectable tap point. An alternate scheme to consider is to set the tap points on the internal 256-bit shift register of the 4161 to 64-bit increments. This allows an internal TMS4161 memory-to-shift-register transfer to occur on every scan line. With every scan line transfer, the tap point on the internal shift register will be set to the next sequential 64-bit tap point, and that section will be shifted out. Tap point utilization may serve to simplify graphics system design since memory transfer cycles will automatically occur after displaying every scan line.

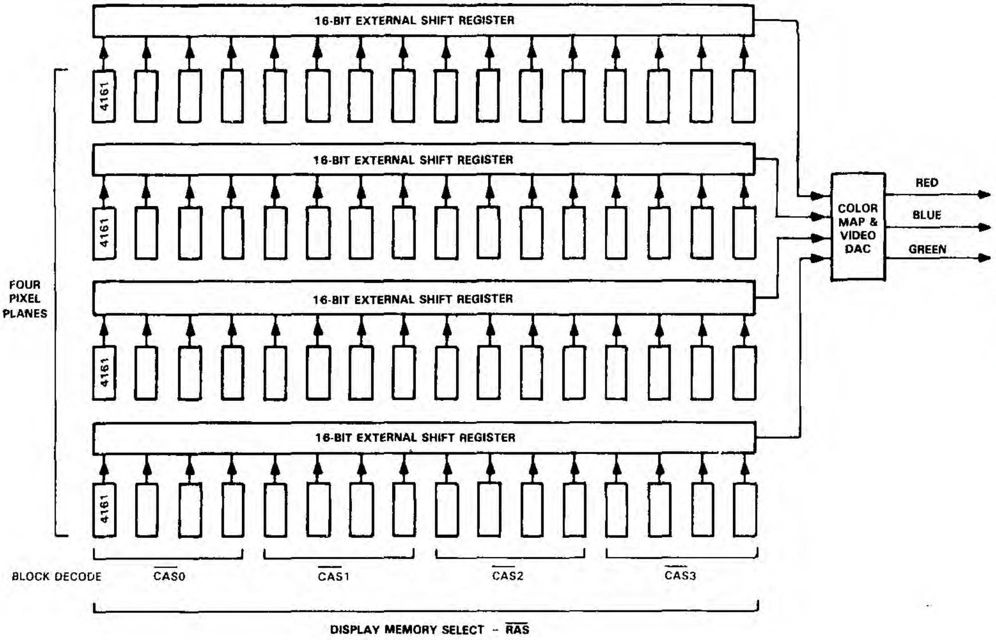


Figure 1. High Performance Display Memory Block Diagram

When a memory access is attempted by the processor, all 64 memory devices will be accessed simultaneously due to the single $\overline{\text{RAS}}$ decode signal (refer to Figure 1). The $\overline{\text{CAS}}$ signals are then used to interleave memory access so that 16-bit (four bits from four planes corresponding to four pixels) data increments coincide with individual processor memory accesses. Four $\overline{\text{CAS}}$ signals are generated from the control circuitry and correspond to the four banks of memory in Figure 1 ($\overline{\text{CAS0}}$-$\overline{\text{CAS3}}$). As a result, four consecutive memory access cycles are required to obtain all 64 data bits from all 64 TMS4161 devices. The initial processor access ($\overline{\text{CAS0}}$) will have to accommodate a wait state to realize a full DRAM access period (assuming a high-performance processor in the 125 to 150 ns cycle time range). Subsequent memory accesses ($\overline{\text{CAS1}}$-$\overline{\text{CAS3}}$), however, will approach the $\overline{\text{CAS}}$ access time of the device since $\overline{\text{RAS}}$ remains low and the minimum $\overline{\text{RAS}}$-to-$\overline{\text{CAS}}$ delay has already been satisfied. The only contention problem that may arise is between a fast $\overline{\text{CAS}}$ access of a successive access with the turnoff period (t_{OFF}) of the data from the preceding $\overline{\text{CAS}}$ access. The possibility of this contention is eliminated due to t_{CAS} (min) > 35 ns which is greater than t_{OFF} (max) < 20 ns. This system performance criteria allows the use of a graphics processor capable of a 125 to 150 ns cycle time because of the successive nature of the memory accesses. As mentioned previously, the first processor memory access will have to incorporate a wait state; however, subsequent accesses can operate at the processor cycle time. Figures 2 and 3 are the control signal schematic and control signal timing diagram for this implementation. In Figure 2, the first S175 quad D-type flip-flop enables the memory only with the first memory request from the processor. The subsequent three processor memory requests (generated from ALE) are used to latch data on the data bus as determined from the appropriate $\overline{\text{CAS}}$ bank enable. Circuitry and timings in Figures 2 and 3 show only the VRAM memory access timings for clarification purposes; the final system will have to arbitrate between memory access and memory transfer operations. Note that the actual cycle time of the memories, including $\overline{\text{RAS}}$ precharge periods, will be 640 ns (see Figure 3 where sixteen 40-ns clock cycles correspond to a complete memory cycle). The memory speed requirement corresponds to only the access time and not the cycle time since the memory banks are interleaved. A more practical implementation would allow for the number of memory access cycles to range from one to four to accommodate those

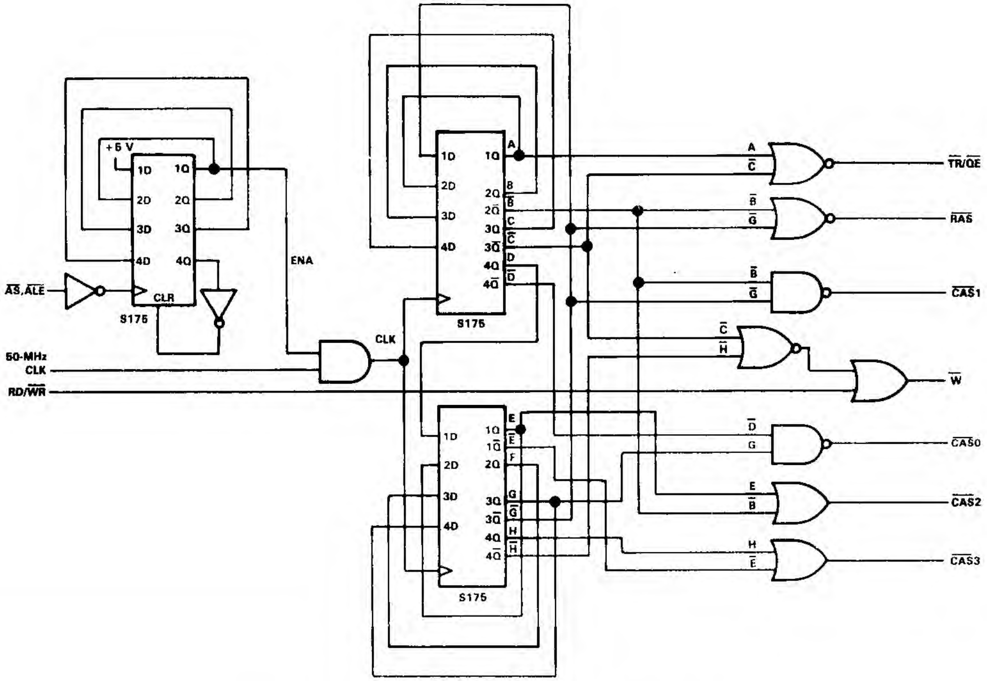


Figure 2. Control Logic Implementation of TMS4161 Banks

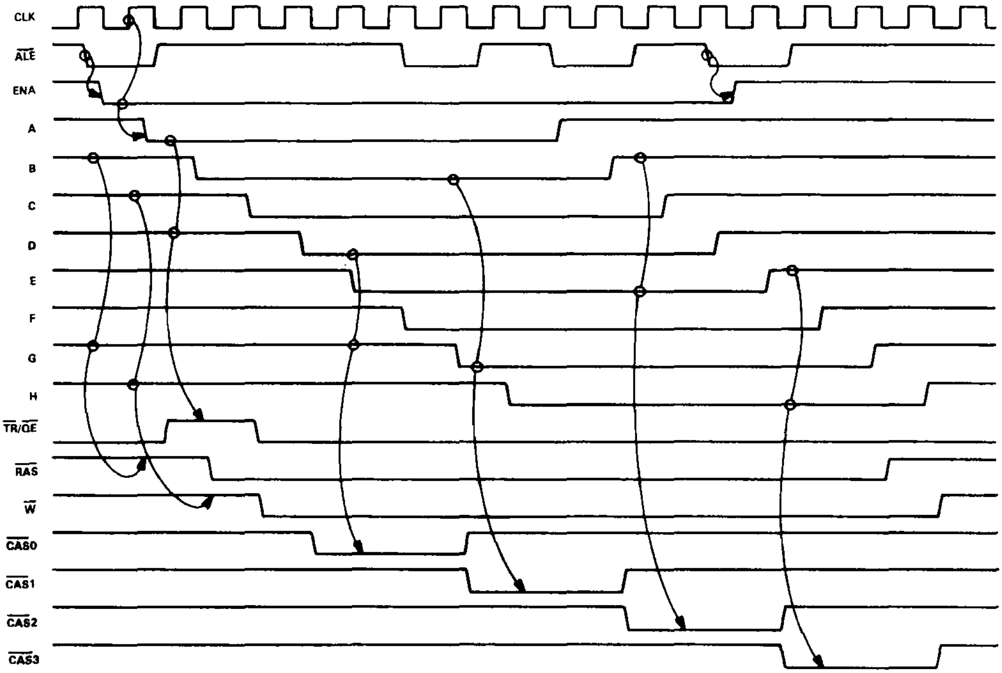


Figure 3. TMS4161 Control Timing Diagram

memory accesses which change only four pixel groups (software would be used to determine which one of four pixels to change). Peak performance will be achieved when memory is accessed in four processor memory cycle increments where average memory cycle time approaches the $\overline{\text{CAS}}$ access time. The utilization of delay lines or more optimized control signal generation could increase system performance at a greater implementation cost. Optimized control circuitry would allow timing edges to more closely align with actual memory and processor specifications to maximize performance.

The interleaved architecture offers a high performance memory access solution for those graphics system requiring:

1. The high-performance requirement of a high-resolution graphics or engineering work station (processor cycle time of 100-150 ns).
2. A high enough display resolution to require a large amount of memory. Specifically, the memory size must be large enough to establish a memory data path at least twofold the processor data bus width (i.e., 32-bit memory data path to a 16-bit processor).

High-performance microprocessor-based systems can take advantage of this scheme since the access time of the DRAM approaches that of the $\overline{\text{CAS}}$ access time instead of the total DRAM cycle time. In these applications, the TMS4161 VRAM devices can be configured to yield enhanced access performance through the random-access port while simultaneously shifting data out of the serial port.

Dual-Port Memory with High-Speed Serial Access

Ray Pinkham
Video Design Manager
MOS Memory Activity
Texas Instruments Incorporated
P. O. Box 1443
Houston, Texas 77001

SUMMARY

The demand for a friendlier user interface, versatility in displayed information, and color have made raster scanned bit-mapped graphics a viable and attractive approach to graphics. Traditionally, however, high resolution bit-mapped graphics systems have been cost prohibitive, owing greatly to the high cost of semiconductor memory and the system level cost of the support electronics needed to achieve the required video memory bandwidth. This paper describes a 64K × 1 dynamic RAM with an onchip 256-bit shift register and fast bulk data clear capability developed specifically for video applications. The RAM, designated the TMS4161 Multiport Memory, is intended to provide relief to the graphics system designer wishing to design a bit-mapped graphics system at a reasonable cost. It also provides attractive solutions for many non-video applications.

INTRODUCTION

Recently, there has been steady growth in the mid-to-high end arena of raster refreshed bit-mapped graphics displays, due largely to the growing demand for professional computers and work stations. Even the highest resolution displays found in CAD/CAM systems and flight simulators have turned to bit-mapped architectures. In bit-mapped displays, each pixel on the display can be controlled individually and independently, providing the ultimate in versatility. The growing appeal of bit-mapped systems is due largely to demand for:

- Friendly user interface
- Mixing of text and graphics on a single screen at will
- Flexibility in the types of information which can be displayed such as windowing, priority planing, and variable text fonts.

Design Challenges for Bit-Mapped Displays

These attractive features inherent with a bit-mapped display have historically been achievable only at high cost. The amount of memory needed to describe each pixel on the display usually dominates the system costs. Furthermore, the memory must not only be fast enough to deliver the pixel information to the display with no screen flicker, but the CPU must have enough access to the display memory to update it and prevent very long screen redraw times. Figure 1 shows the square law relationship between screen resolution, memory requirements and video pixel scan rate necessary to maintain a flickerless display operating at 60 Hz in non-interlaced mode. It is clear that high resolution bit-mapped

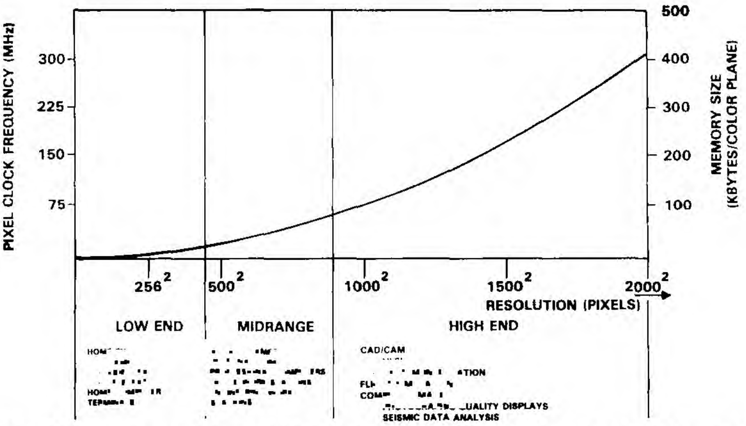
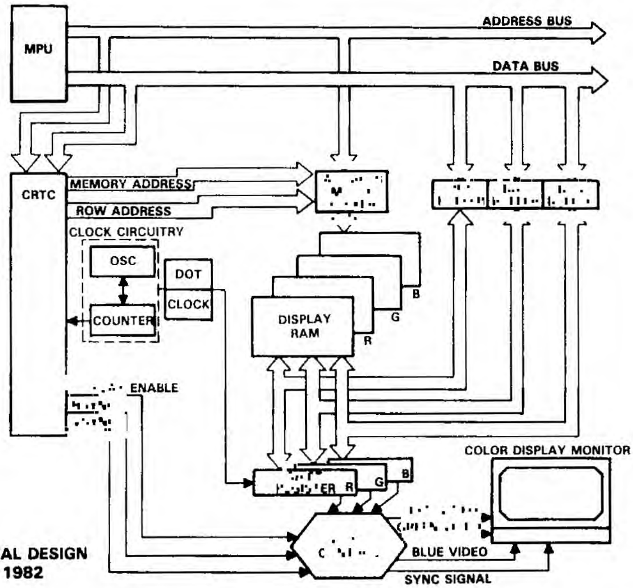


Figure 1. Video Memory Size and Bandwidth Requirements Vs. Resolution Showing Three Basic Categories.



SOURCE: DIGITAL DESIGN
JULY 1982

Figure 2. Typical Color Graphics System Illustrating Basic Blocks. Crucial to Such a Design are the CPU to Memory and Memory to Video Bandwidth

graphics systems are very memory intensive. In very high resolution systems (greater than 1K × 1K) incorporating color, the memory system alone, including shift registers, FIFOs multiplexers, and buffers, can represent 50% of the total graphics system cost. Previously, the high cost of the semiconductor memory chips themselves has limited the use of bit-mapped architectures. Furthermore, video memory bandwidth is often a limiting factor that the display system designer must consider. Obtaining memory chips with low cost-per-bit and high data rates has proven to be a burdensome task. Specifically, the architectures of high density dynamic RAMs have not until now been optimized for video applications.

Figure 2 illustrates a typical graphics system implementation. The key bandwidth requirements are twofold. First, the video display information contained in the bit-map memory must be transferred to the display on every vertical scan. Using available dynamic RAMs (typically 300 ns cycle time), this requires very wide (60 to 120 bit) data buses loaded in parallel into high speed TTL or ECL shift registers operating at the dot clock rate. Second, the graphics processor must be able to access the display memory often enough to keep it updated and not be locked out by the video memory controller which must keep the display refreshed.

Figure 3 shows a block diagram for a 1K × 1K black and white display system. To maintain a 60 Hz refresh rate requires a 12 ns dot clock rate assuming horizontal and vertical retrace times of 3.6 μs and 0.4 ms respectively. To achieve this data rate using conventional DRAMs with 300 ns cycle times requires a data bus of 32 bits or 32 16K × 1 dynamic RAMs, assuming that the processor is completely locked out during active scan. Typically, the processor will share access to the memory with the video controller on a fifty-fifty basis. That is, the video will have access on one cycle, the processor on the next, and so on. This doubles the necessary data bus width which means that all 64 chips necessary to hold the 1K × 1K display (64 × 16K = 1K × 1K = 1 megabit) will be brought out in parallel and loaded into a 64-bit shift register operating at the dot clock rate. In addition, arbitration logic must be included to determine whether the CPU or the video refresh has control of the display memory at any given time. Tristate buffers under the control of the arbiter connect the address and data buses to either the CPU or the video. Lastly, the wide video data bus creates a mismatch to the (typical) 16-bit data bus from the CPU, necessitating a bank decode of 16 bits from the CPU selected onto 16 of the 64 video data lines. Use of 16K × 4 dynamic RAMs which have become available recently[1] provides some relief to the chip count problem. They reduce the DRAM chip count from 64 to only 16. They do not, however, provide any improvement with respect to reducing bus widths, eliminating support circuits, or

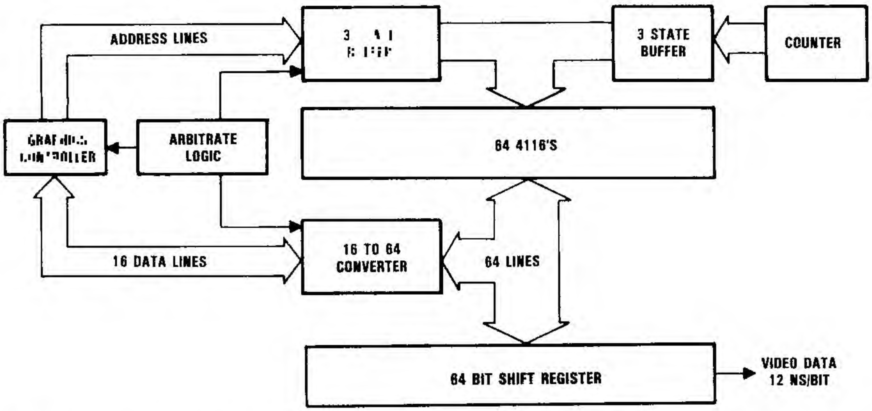


Figure 3. A 1K × 1K Black and White Graphics System Implemented with 16K × 1 Dynamic RAMs.

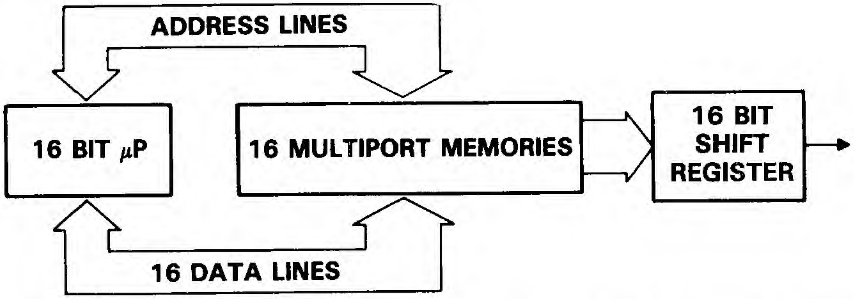


Figure 4. A 1K × 1K Black and White Graphics System Implemented with Multiport Memories.

increasing the available bandwidth between the memory and the graphics processor.

Clearly, a graphics system such as this one could be vastly improved in terms of cost and performance by: 1) decoupling the video data bus from the CPU side of the video memory, allowing simultaneous access to memory by the video and the CPU, and 2) integrating the high-speed support circuits onto the memory chips themselves.

Overview and Description of the TMS4161

The display refresh function in raster scanned bit-mapped systems is very regular. Therefore, rather than simply trying to build a faster DRAM, a new architecture concept for memory devices has been developed. The Multiport Memory interfaces a high-speed 64K DRAM to an internal 256-bit dynamic shift register. This gives the device, the TMS4161, a unique dual-port-like architecture that eliminates memory contention problems and reduces the hardware complexity of bit-mapped raster scanned displays.

To lend some perspective as to what the Multiport Memory can provide, Figure 4 illustrates the same 1K × 1K black and white graphics display system implemented using TMS4161 memory chips. The arbitration logic has been significantly reduced since the CPU has total access to the video memory during active scan. This also eliminates the need for large buffer/driver circuits. Also, since the serial ports of the memory chips can operate at speeds up to 40 ns without sharing access to the memory with the CPU, the 64-bit data bus can be reduced to 16 bits and still provide the necessary bandwidth to the video. Thus the CPU data bus width has been matched to the video bus and no bank decode is necessary. Lastly, the refresh counter can, in many applications, be eliminated or at least combined into the refresh circuits needed to refresh non-video system DRAM used to store graphics commands, display lists, etc. The TMS4161 would typically boast a reduction of 72 chips (48 memory and 24 logic) versus the system implemented with 16K × 1 DRAMs. Versus the system implemented with the 16K × 4 chips, the main chip reduction comes from the 24 logic chips.

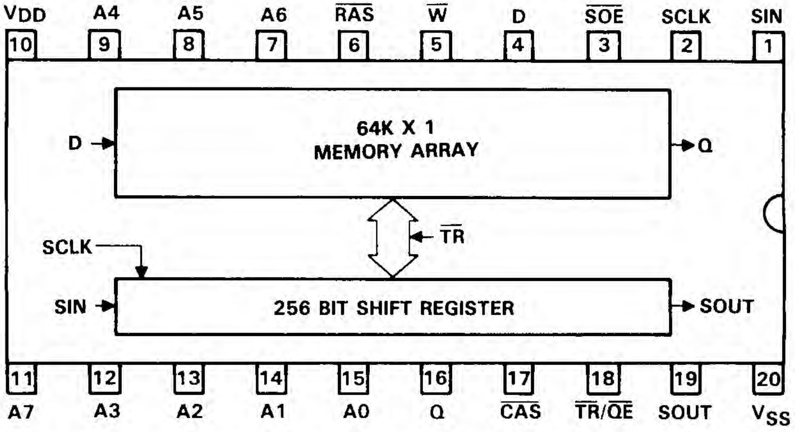
Figure 5 illustrates a block diagram of the TMS4161 showing all address, data, and control signals used for both normal DRAM (random) operation and sequential operation. The $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and R/$\overline{\text{W}}$ control signals; the 8-bit multiplexed row and column address inputs; and the random data-in (D) and data-out (Q) pins provide the same interface to the DRAM portion of the Multiport Memory as for normal 64K × 1 DRAMs. In addition, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, R/$\overline{\text{W}}$, and the address inputs provide information to the Multiport Memory during transfer operations in which 256 bits of information contained in one row of the memory array is transferred in parallel into the 256-bit shift register in one 260 ns RAS cycle time.

Five additional pin functions have been added to control the serial ports. The serial input (SIN) and serial output (SOUT) deliver data to and receive data from the on chip register respectively. The serial output enable ($\overline{\text{SOE}}$) pin is useful when multiplexing more than one video source into the same video circuitry. When $\overline{\text{SOE}}$ is low, the serial output is in a low-impedance state and data can be shifted out of the register. When $\overline{\text{SOE}}$ is high, the serial output is placed into a high-impedance state. The serial clock signal (SCLK) is the clock needed to shift data along the shift register. It is analogous to the clock input to 256 D flip-flops cascaded D to Q. The $\overline{\text{TR}}$/$\overline{\text{QE}}$ pin serves two functions. First, it controls whether or not data is to be transferred between the shift register and the memory array during $\overline{\text{RAS}}$ low. Second, it serves as an output enable of the normal random output after $\overline{\text{CAS}}$ goes low during random mode memory accesses. This makes it possible to multiplex address and data information onto the same bus.

The TMS4161 has three basic modes of operation:

1. Read from or write to the memory array via the Q and D ports.
2. Read from or write to the shift register via SOUT and SIN.
3. Transfer 256 bits of data between the shift register and the memory array.

Items 1 and 2, previously mentioned, can be performed simultaneously in the TMS4161. Since the shift register is disconnected from the memory array except when $\overline{\text{TR}}$/$\overline{\text{QE}}$ is latched low on $\overline{\text{RAS}}$ falling, the data in the register can be shifted in and out under control of the SCLK pin while normal DRAM reads and writes are taking place. The transfer of 256 bits between the register and the array can take place in either direction. That is, the memory array can be written to by transferring 256 bits into one of the memory rows or the memory array can be read from by transferring the contents of one of the memory rows into the shift register. This feature makes it possible to write a fixed row pattern to the entire memory array in just 256 cycles. This is particularly useful in double frame buffered systems in which



- DUAL PORT MEMORY — ONE PARALLEL (A LA 4164), ONE SERIAL (25 MHz)
- MODES OF OPERATION
  - — READ/WRITE THE 64K X 1 ARRAY VIA THE D AND Q PORT (150 ns ACCESS)
  - — READ/WRITE THE 256-BIT SHIFT REGISTER VIA SERIAL PORTS (25 MHz/40 ns BIT RATE)
  - — TRANSFER 256 BITS FROM ARRAY TO SHIFT REGISTER AND VICE VERSA

Figure 5. Block Diagram of the TMS4161 Multiport Memory

the contents of one frame buffer must be cleared while the other is being displayed. This feature also allows re-ordering of the memory rows which is useful for scrolling operations.

Figure 6 shows the internal architecture of the TMS4161. The 256-bit shift register is divided into two sections. The top section contains the even numbered shift register locations and the bottom section contains the odd numbered shift register locations. The two sections are interfaced to interleaved buffer circuits at the SIN and SOUT ports to deliver the register data in the proper sequence. This has been made necessary since the shift register bit, which contains six transistors, cannot be laid out in the same area as the normal dynamic RAM storage cell which contains one transistor and one capacitor. Thus, the register bit consumes the pitch of two DRAM cells.

A key feature of the TMS4161 is its segmented register architecture. The 256-bit shift register has been optimized to provide for interlaced as well an non-interlaced displays by segmenting the register into four cascaded sections. The shift register is organized as a single 256-bit shift register that can be selectively tapped every 64 bits. A two-bit code, entered onto A6 and A7 instead of a column address during a shift register transfer cycle, is used to select the one-of-four tap points. For example, if the two bits are binary 00, the entire shift register of 256 bits can be shifted out in order. If the two bits are binary 01, then 192 bits starting at bit 64 can be shifted out in order. If the two bits are binary 10, then 128 bits starting at bit 128 can be shifted out in order. A binary 11 allows 64 bits starting at bit 192 to be shifted out. All bits are shifted out least-significant bit first, most-significant bit last with respect to the random access column address.

Consider a 1K × 1K graphics system similar to Figure 4 but using an interlaced display. In each horizontal row, 1024 pixels must be scanned. Using 16 Multiport Memory chips, this means that each chip will contain a bit of data for 64 pixels on each line. Furthermore, each time that the shift registers of the TMS4161's are loaded, the registers will contain 64 bits of information corresponding to pixels on four consecutive (adjacent) scan lines, with segment 00 being the topmost of the four scan lines and segment 11 being the bottommost of the four lines (CRT scans top to bottom). In interlaced systems, the CRT first traces out the even numbered scan lines (even field) and then the odd numbered scan lines (odd field) for every vertical scan. To start the display at scan line 0 of the even field, the register is loaded and A6 and A7 are set to 00 on the falling edge of $\overline{\text{CAS}}$, selecting the 00 tap point and the 64 bits in segment 00 are shifted out to the display. Since scan line 2 is needed next, the same row of 256 bits is reloaded into the shift register and segment (tap point) 10 is selected, skipping over the bits in segment 01 representing line 1. All other scan lines in the even field are put on the screen using the same procedure. After vertical retrace, the odd numbered lines are filled in by setting the tap point to 01 and then 11 during the register loads and shifting the data out.

Once the tap point is selected during a register load operation, that tap point will remain selected until $\overline{\text{CAS}}$ goes low during the next register load operation. Thus, for non-interlaced systems, the code 00 can be applied to A6 and

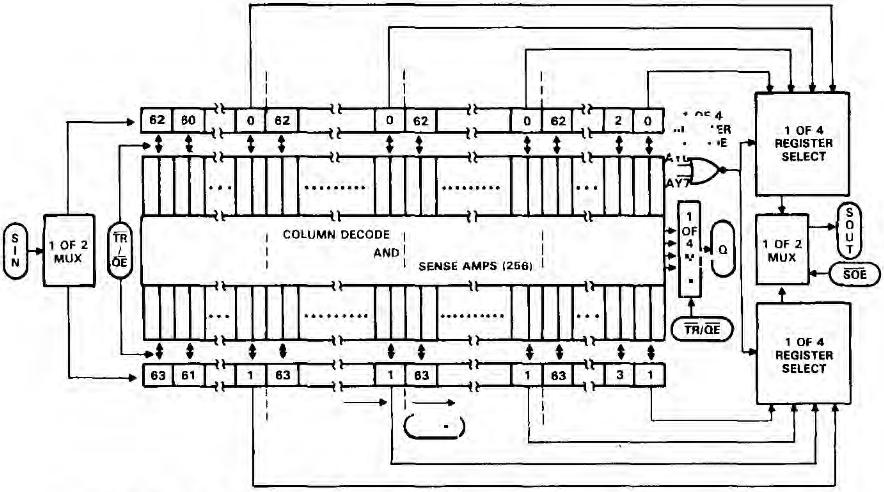


Figure 6. Block Diagram of the TMS4161 Showing Segmented Register Architecture.

A7 during the first transfer operation. Subsequently $\overline{\mathrm{CAS}}$ can remain high during all future transfer operations and the 00 tap point will remain selected. The tap-point code is stored in static latches and will remain valid as long as power is supplied to the chip.

Figure 7 shows a timing diagram for a memory-to-register load operation followed by simultaneous and asynchronous operation of a random mode write cycle (e.g., CPU writing to video memory) and serial shift out of the register (to the video circuitry). The first bit out of the shift register is triggered off the rising edge of $\overline{\mathrm{RAS}}$. All subsequent 255 bits are propagated through the register off the rising edge of SCLK. To simplify designs in which non-video DRAM is contained in the same address space as the video DRAM, the TMS4161 Multiport Memory has been designed to be spec compatible with industry standard 64K × 1 dynamic RAMs.

Graphics Applications of the TMS4161

Figure 8 shows a basic block diagram of a 1K × 1K black and white graphics system similar to that described in Figure 4. The 16-bit CPU operates on 16 pixels at once and the pixels have a regular one-to-one correspondence with the physical memory locations. The 16-bit data bus feeds directly into the D and Q ports of the *Multiport* Memories. The serial output ports of the 16 memory chips feed in parallel to an external 16-bit shift register which operates at the 12 ns dot clock rate determined earlier. The TMS4161 internal shift registers need only operate at one-sixteenth of the frequency of the video dot rate, or 192 ns, well within their minimum clock period of 40 ns.

The design simplicity inherent with the TMS4161 is illustrated in Figure 9, which shows the same 16 Multiport Memory chips configured into a 512 × 512 pixel display system with four bits per pixel giving 16 possible colors. The memory is organized into four color planes, each 512 × 512. Four memory chips are used for each color plane. Here the wiring of the CPU data bus onto the memory D and Q ports is slightly different. In this application the pixel information is read across the four color planes. That is, the CPU operates on four pixels at once and the computer word contains all four bits of information describing each of the four pixels as shown. The external 16-bit shift register is divided into four 4-bit registers each of which will supply one bit of information (corresponding to a single pixel) to a color look-up table on each dot clock cycle. It is interesting

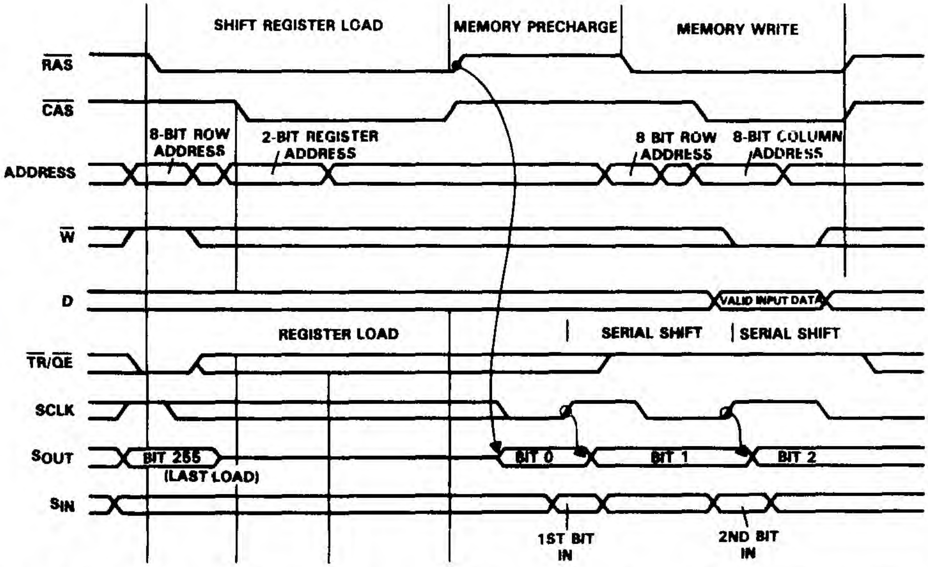


Figure 7. Timing Diagram of TMS4161 Register Load Followed by Simultaneous Serial and Random Mode Access.

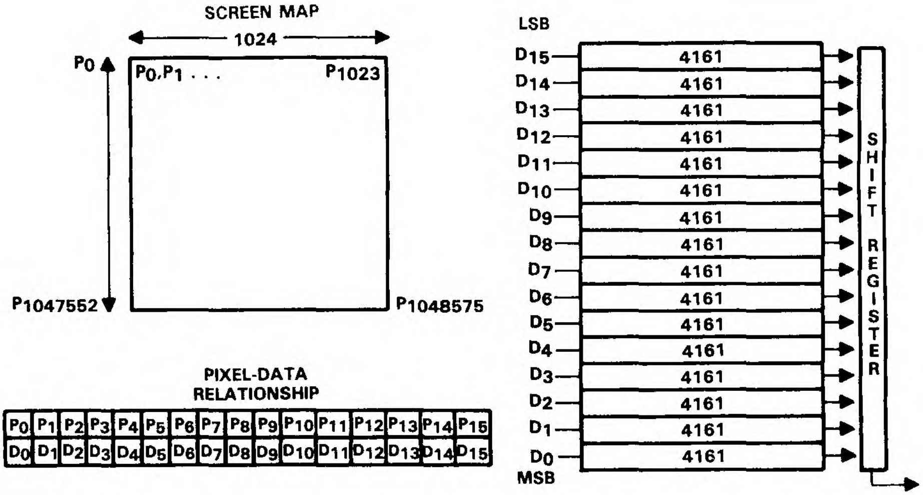


Figure 8. Memory Configuration of a 1K × 1K Black and White Graphics Showing Pixel-Data Relationship and Data Bus Configuration.

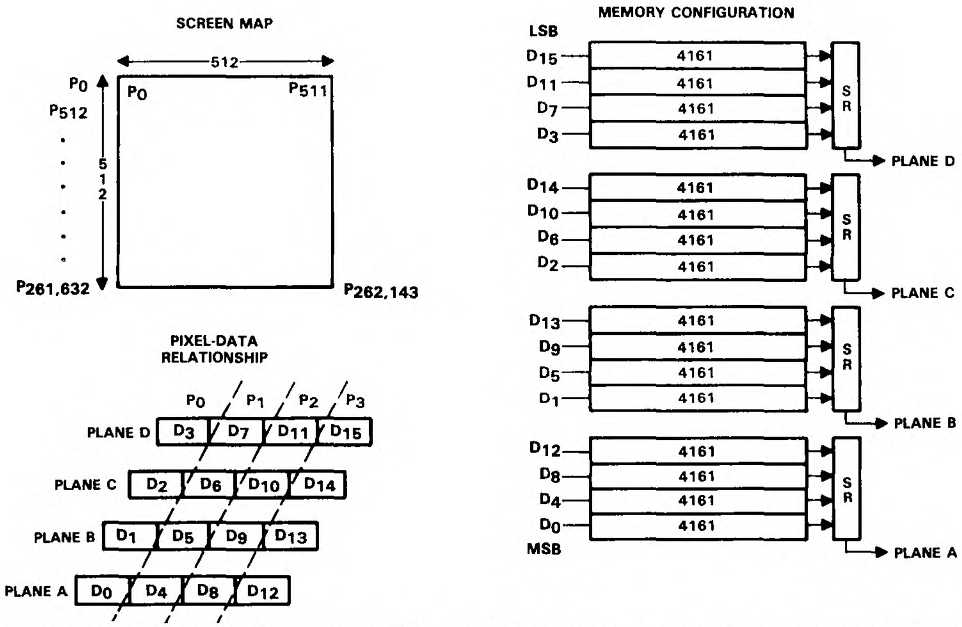


Figure 9. Memory Configuration of a 512 × 512 Graphics Featuring Four Bits Per Pixel. The Memory Timing of the TMS4161 Serial Ports is the same as the Black and White System in Figure 8.

to note that even though the internal shift registers of the TMS4161's must now operate at one-fourth the frequency of the external registers rather than one-sixteenth the frequency as in the previous case, there are only one-fourth as many pixels to display so the shift rate of the Multiport Memory shift registers remains 192 ns, exactly like the previous case. In fact, regardless of screen resolution and number of color planes, the shift rate required of the TMS4161 shift registers remains relatively constant. This points to a key plus in utilizing the TMS4161: design flexibility.

Non-Video Applications of the TMS4161

The serial input port of the TMS4161 significantly enhances the chip's usefulness in both video and non-video applications. For example, it can be used to load in a video data stream in an image processor application such as charge coupled device (CCD) imagers. Non-video applications which could benefit from such a device include main memory in high-speed memory systems whereby the dual-access nature of the device facilitates disc to main memory and main memory to cache memory bulk data transfers. Tag-bit processing is greatly simplified with a substantial increase in performance. The tag bits of a data word in a computer system are read from TMS4161 memory chips and thus can be quickly set and cleared *sequentially* as well as be completely cleared using the fast clear feature described earlier. This can be useful in virtual memory systems when keeping track of available physical memory space. Network communications is also an attractive application area for the TMS4161. Whether collision detection or token ring based, 10 MHz serial data transmissions can be handled easily using the serial ports of the TMS4161 while its memory array is being accessed.

CONCLUSION

The decreasing cost-per-bit of dynamic RAM coupled with an advanced memory architecture will provide both a cost reduction and a performance enhancement to todays' bit-mapped graphics systems. By providing asynchronous random and serial ports and integrating the necessary support electronics, the TMS4161 promises to provide unprecedented design simplicity in todays systems. The TMS4161 is also well suited for a variety of non-video applications, providing performance enhancements and design simplicity here as well.

REFERENCES

1. Peyton M. Cole, David W. Gulley, and Lionel White, "Wide-Word Memory Chips Spur New μP Applications", Electronic Design, Hayden: 1981.

This paper was initially presented at WESCON '83 and is reprinted with their permission.

Topological Structure of the TMS4161 Multiport Video RAM

In many applications, there will be a need to analyze the architecture of the TMS4161 Multiport Video RAM to confirm proper operation. The purpose of this application report is to define the internal architecture and function of the TMS4161 as they relate to its operation. In this manner, the memory array and shift register of the TMS4161 can be tested properly to guarantee device adherence to application requirements.

INTERNAL ADDRESS WEIGHTINGS

Sixteen address bits are needed to decode one of 65,536 memory locations in the memory array. The multiplexed addressing scheme provides for $\overline{\text{RAS}}$ (Row Address Strobe) to latch 8 row addresses, and $\overline{\text{CAS}}$ (Column Address Strobe) to latch 8 column addresses. A 256-bit shift register is integrated onto the TMS4161 to add an additional port for data access in a serial mode. The external pin address names correspond directly to the internal address weights of the memory array (i.e., RA7 has addressing weight of 2^7) as indicated by Table I.

Table I. Internal Address Weightings

Desired Row or	Weight	TMS4161	
Column Address		Pin Name	Pin Number
(MSB) RA7,CA7	2^7	A7	11
RA6,CA6	2^6	A6	7
RA5,CA5	2^5	A5	8
RA4,CA4	2^4	A4	9
RA3,CA3	2^3	A3	12
RA2,CA2	2^2	A2	13
RA1,CA1	21	A1	14
(LSB) RA0,CA0	2^0	A0	15

ARRAY TOPOLOGY

The same memory array layout is utilized with the TMS4161 as with the 64K × 1, TMS4164 DRAM (See Application Note MM4164A9 for further information concerning the TMS4164 topology). Translation from the pinout of the TMS4161 to the memory array topology is represented by Figures 1(a) through 1(d). The algorithms for determining near and nearest neighbors to a monitored memory cell are summarized in Tables II and III with (R,C) representing the cell location where R = row address and C = column address.

Table II. Near and Nearest Neighbors if Row and Column Addresses are Either Both Even or Both Odd

Row Address	Nearest Neighbors	Near Neighbors
≤7F	R−2, C+1 R+0, C+1	R−2, C+0 R+2, C+0 R−1, C+2
≥80	R−2, C−1 R+0, C−1	R−2, C+0 R+2, C+0 R−1, C−2

Table III. Near and Nearest Neighbors if Row and Column Addresses are Neither Both Even or Both Odd

Row Address	Nearest Neighbors	Near Neighbors
≤7F	R+0, C−1 R+2, C−1	R−2, C+0 R+2, C+0 R+1, C−2
≥80	R+0, C+1 R+2, C+1	R−2, C+0 R+2, C+0 R+1, C+2

INTERNAL DATA INVERSION

Note that the algorithm changes for each half of the array due to the fact that the top half is laid out as the mirror image of the bottom half. Data in the top half of the array is stored in inverted form while data in the lower half is stored in true form. This internal data inversion is transparent to the user; however, for generation of specific patterns, this data inversion must be taken into account. Row address bit seven selects between the upper and lower memory arrays, thus the circuit shown in Figure 2 may be used to compensate for this internal data inversion. This allows data written in and read out to be the same polarity as the data stored in the addressed memory cell.

MEMORY ARRAY ACCESS

During an access cycle, 256 column locations are enabled by $\overline{\text{RAS}}$ (corresponding to the selected row). Four of these 256 column locations are then decoded from CA0-CA5 after the addresses are latched by $\overline{\text{CAS}}$. The four column locations are then decoded by the least significant column address bits (CA0 and CA1) selecting a single bit to be read

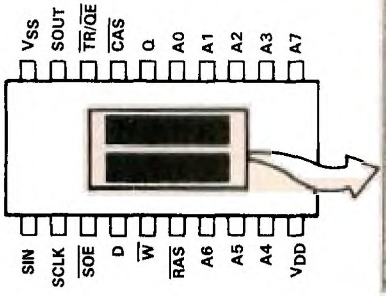


Figure 1a. TMS4161 Pinout

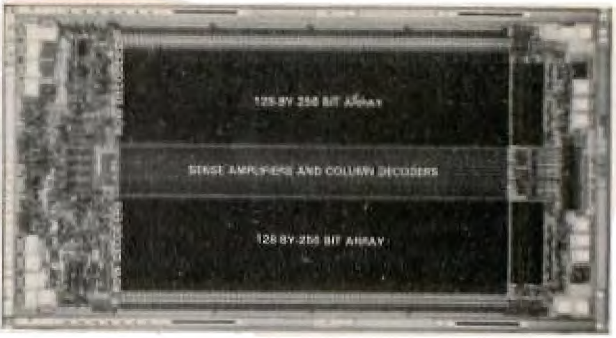


Figure 1b. Array Organization

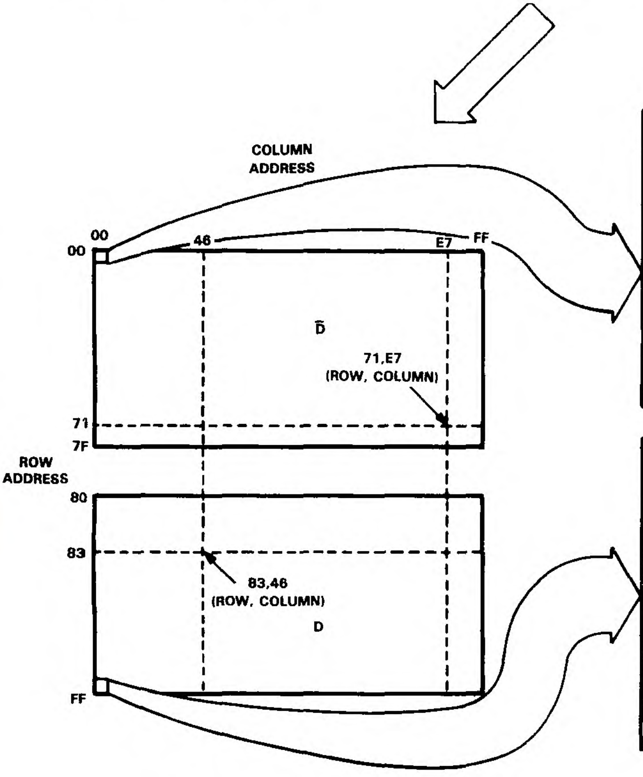


Figure 1c. TMS4161 Array Bit Map

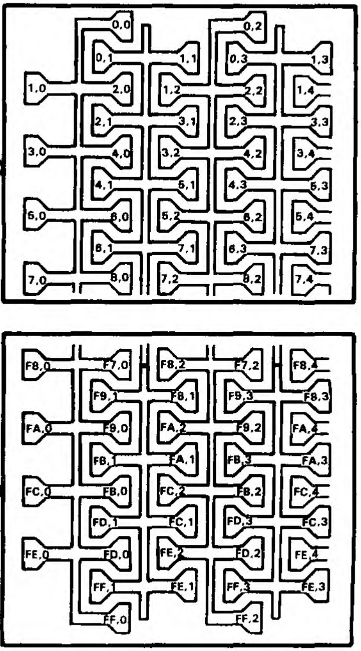


Figure 1d. Upper and Lower Array Cell Topology

Figure 1(a) shows the chip pinout, Figure 1(b) is a closeup of the array, Figure 1(c) shows the bit map for the rows and columns, and Figure 1(d) is a closeup of the cell topology in the array.

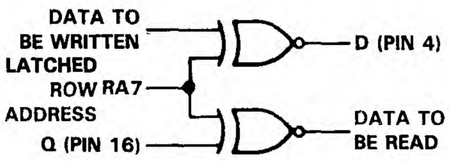


Figure 2. Circuit for Compensation of Internal Data Inversion

or written through the one-of-four data input/output selector (see Figure 3). As a result, the four internally accessed memory locations are obtained from four adjacent columns of a single accessed row. An algorithm can be developed which will calculate the other three accessed memory locations based on the position of the intended memory location in the array. This algorithm is shown in Table IV assuming the memory location (R,C) as the base location.

Table IV. Algorithm for Determination of Additional Accessed Memory Locations

Column Address		Additional Accessed Memory Locations		
CA0	CA1			
0	0	R+0, C+1	R+0, C+2	R+0, C+3
0	1	R+0, C−1	R+0, C+1	R+0, C+2
1	0	R+0, C−2	R+0, C−1	R+0, C+1
1	1	R+0, C−3	R+0, C−2	R+0, C−1

The algorithms can be useful in verifying that the one-of-four data input/output select function is operating correctly on the device.

SHIFT REGISTER FUNCTION

The significant difference between the TMS4164, 64K × 1 DRAM, and the TMS4161, Multiport Video RAM, is the addition of the 256-bit shift register on the TMS4161. Important tests to evaluate are the transfer of data from a

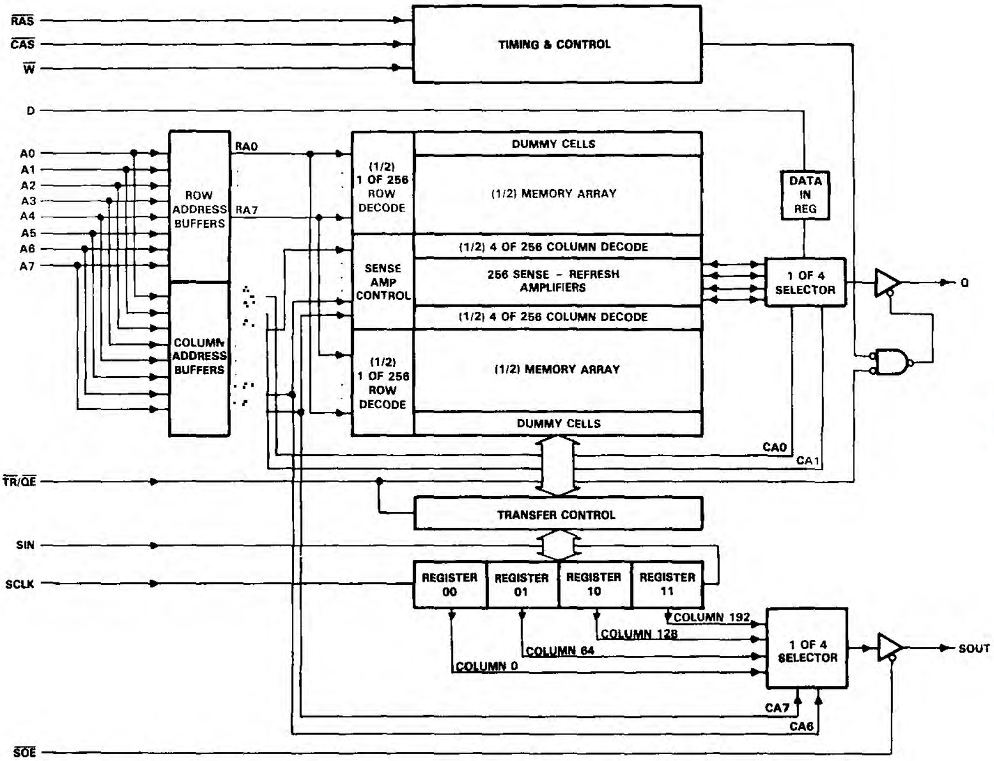


Figure 3. TMS4161 Functional Block Diagram

memory row to the shift register and from the shift register to a memory row. These transfers are easily verified since the column address of an individual data bit in the memory row to be transferred corresponds to the same position number of the shift register. For example, the data stored in column address 127 of the row being transferred will be written to position 127 of the shift register. As a result, the data obtained from the serial stream of the shift register will correspond, one for one, to the row of memory that was transferred. In a similar fashion, serial data can be input to the shift register position 255 by the SIN input. Subsequent shift clocks to the shift register will move the data from the most significant shift register bit to the least significant bit. When fully shifted (256 clocks), the data in the shift register can be transferred to the array and will correspond to that row of memory as if that row had been sequentially written in random access mode. For additional information concerning the testing of the TMS4161, refer to the "Testing Philosophy for the TMS4161" Application Report.

SHIFT REGISTER ARCHITECTURE

Figure 3 is a functional block diagram of the TMS4161, Multiport Video RAM (VRAM), and Figure 4 is a block diagram of the shift register and control circuitry of the VRAM. The shift register is actually subdivided into two sections; the sections correspond to the odd and even column loctions, respectively, of the memory row to be transferred. As shown in Figure 4, the even shift register is physically located adjacent to the top half of the array, while the odd shift register is adjacent to the bottom half of the array. The two shift register sections are interleaved to yield the sequential output of the memory row; the interleaving is done synchronously with the register shift clock and automatically selects the even shift register bit for the initial output of the shift register. This is accomplished transparent to the external device operation. For those operations where the shift register is to be loaded serially from the SIN input, a dummy transfer must first be made for initialization purposes (either to or from the shift register). This serves to properly

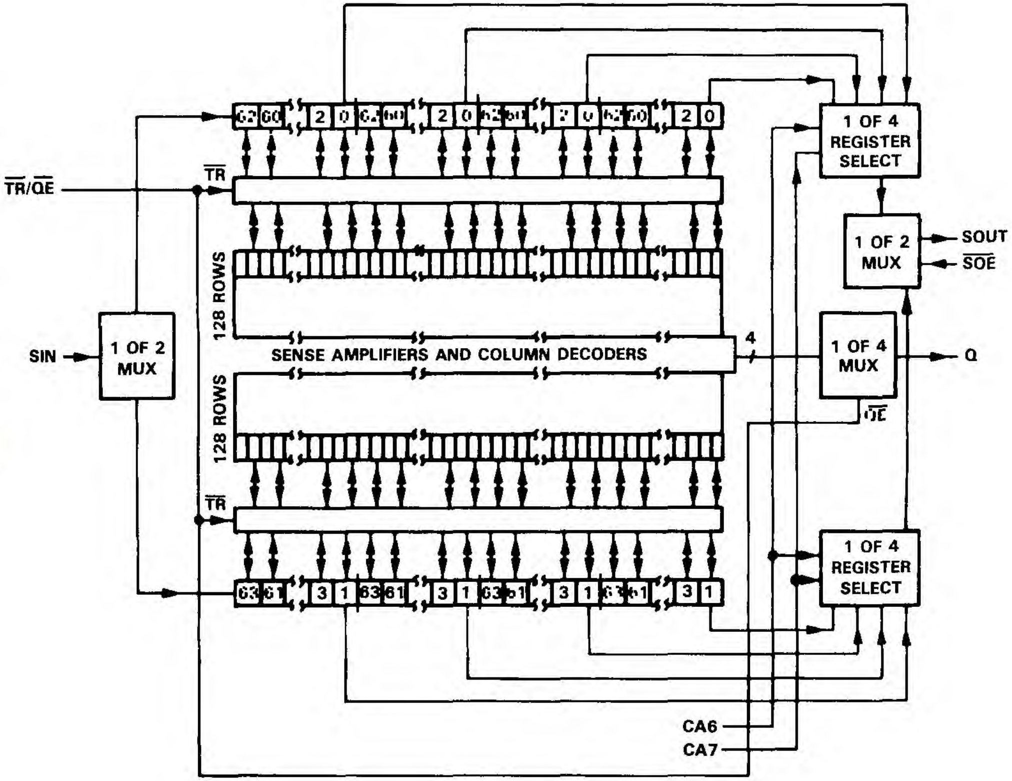


Figure 4. TMS4161 Shift Register and Control Block Diagram

set up the serial input sequence from the input multiplexer so that the even bit shift register will be loaded first. Subsequent serial input (from SIN) will switch between the two shift registers to achieve the proper input sequence.

SHIFT REGISTER TAP POINTS

The shift register is divided into four cascaded 64-bit shift registers segments based on the select function which is decoded by the two most significant column address bits. Tap points are set up along the 256-bit shift register at 64-bit increments. This allows the shift register to be tapped at less than 256 bits for those applications that may demand register addressability down to 64 bits. Table V illustrates the column addresses needed to select the desired tap point in the shift register.

Table V. Shift Register Tap Points

Column Address		Shift Register Position	Total Available S.R. Bits	Corresponding Column Location
CA7	CA6			
0	0	0	256	C0-C255
0	1	64	192	C64-C255
1	0	128	128	C128-C255
1	1	192	64	C192-C255

These tap points for the shift register are applicable only to the output of the shift register (SOUT); data into the shift register (SIN) is always shifted into position 255 (column bit 255) and shifts toward position 0 (column bit 0).

SUMMARY

With multiprocessor and dual-ported applications becoming very commonplace, the added features of the TMS4161 Multiport Video RAM make it an excellent choice to eliminate the bottlenecks that have previously occurred in accessing system memory. To ensure proper system performance of the TMS4161, the previous discussion has described the topological structure of the device and how it corresponds with device functionality. As a result, test algorithms can be configured which will comprehend a worst case system condition and ensure proper device operation. Texas Instruments is committed to the production of only the most reliable components, and will continue to supply the engineer with the necessary tools and support so that the components that are used will conform to his or her system and design requirements.

An Introduction to the TMS4256 and TMS4257

MOS Memory
Applications Engineering

INTRODUCTION

As applications utilizing dynamic RAM's have increased, the need for a flexible DRAM has also increased. The TMS4256 and TMS4257 expand this flexibility with the advent of the latest generation, 256K dynamic RAM devices. The TMS4256 is the page-mode version while the TMS4257 is the nibble-mode version of the 256K × 1 dynamic RAM's offered by Texas Instruments. The purpose of this application report is to guide the prospective DRAM user in the operation of the TMS4256 and TMS4257 to take full advantage of the device potential. Many of the same design and production techniques from the 64K DRAM generation are utilized with the 256K generation; however, there are important differences.

ARCHITECTURE

A primary consideration in the design and development of the 256K architecture was to retain the same refresh scheme as the previous 64K generation. Refresh for the 256K DRAM is accomplished by supplying 256 refresh cycles in a 4 ms-period. The memory array can be visualized as a matrix of 256 rows by 1024 columns which has been split into two arrays of 256 rows by 512 columns for the upper and lower halves of memory – the two halves actually mirror each other. A single refresh cycle causes a row in both upper and lower arrays to be refreshed. When a memory read or write is performed on the memory array, there are actually four memory locations that will be enabled. External to the memory array is a one-of-four selection function where one out of the four memory locations enabled from the memory array will be selected by decoding row address bit 8 and column address bit 8. As a result, a single data bit is placed at the Q output during a memory read, or a single memory location is enabled for writing the data bit D during a memory write. This architecture lends itself very easily to either nibble mode or ×4 arrangements.

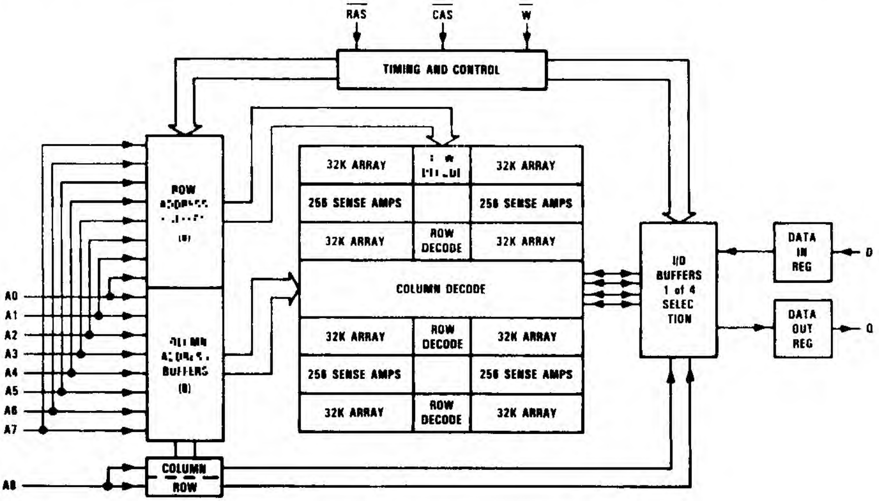


Figure 1. TMS4256/TMS4257 Functional Block Diagram

DEVICE ENHANCEMENTS

Several product improvements have been integrated into the TMS4256 and TMS4257 to enhance both performance of the device and user flexibility.

SUBSTRATE PUMPS

Internally, substrate pumps have been designed to minimize the effect of control signal undershoot. Compensation circuitry is utilized to minimize fluctuations in the substrate voltage. From the device point of view, the overall result of the substrate pump is the reduction of the charge injection of minority carriers into memory cell storage capacitors and periphery circuitry. From the system point of view, it provides the designer extra margin in minimizing the effects of excessive undershoot in the system and improving the overall performance of the system.

FOLDED METAL BIT LINES

Two important changes were implemented with the 256K device bit lines which improve both device noise immunity and performance. Aluminum (metal) bit lines are utilized in the TMS4256 and TMS4257 where diffused bit lines were used in previous generation devices. The capacitive loading of a metal bit line is significantly less than for a diffused bit line. The folded bit line approach makes use of excellent common mode noise rejection to minimize the effects of noise on the sensing margins. Approximately the same noise levels will be experienced by two bit lines that are input to the differential sense amplifier because they are physically close and parallel to one another. Since a differential sensing scheme is used, any common noise has little effect on the sense amplifier.

POLYCIDE WORDS LINES

The TMS4256 and TMS4257 utilize polycide film for the word lines for enhanced device speed. Polycide is a polysilicon material which has been sputtered with a metal for the purpose of signal propagation enhancement. The relative resistance of the polycide material is a factor of ten less than that of polysilicon. As a result, reduced signal delay is achieved through the word lines without the complexity and cost of a double-metal process (metal bit line and metal word line). Architectural analysis for the 256K revealed a higher performance device would be achieved with the utilization of metal bit lines instead of metal word lines. The use of polycide word lines with its metal-like properties, became the obvious approach to achieve the fast access times necessary to accommodate required system performance.

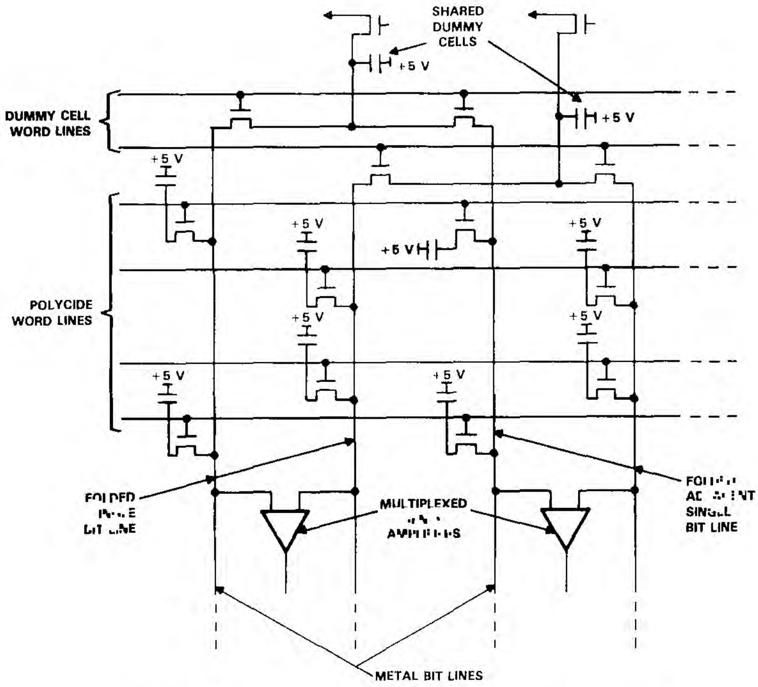


Figure 2. Folded Bit Line and Shared Dummy Cell Concept

SHARED FULL-SIZE DUMMY CELLS

A dummy cell is used as one input to the differential sense amplifier with the accessed memory cell being the other input. Typically, the charge from the dummy cell will create a differential voltage across the sense amplifier to distinguish between logic "1" and logic "0" of the memory cell. With the use of dummy cells that were only half the normal memory cell size (previous generation DRAM's), slight process deviations or marginalities would cause a larger performance degradation on the dummy cell than the memory cells (the net area reduction due to process variation depends directly on the perimeter of the cell). Figure 3 shows the relative effect of a 0.25 micron process deviation on both a full and half size cell. The TMS4256 and TMS4257 employ a full size dummy cell which is shared by the two columns enabled during a normal memory access cycle (see Figure 2). Recall that two rows and two columns are actually enabled in the memory array when memory read or write cycles are implemented. The overall charge distribution with respect to each bit line will be the same as with the half size dummy cell; however, process deviation effects on sense amplifier performance will be minimized. The relative voltage differential between a dummy and memory cell will remain the same despite minor process deviations.

REDUNDANCY

Redundancy is the technique of replacing initially defective memory locations with spare rows or columns. Redundancy has been implemented by adding two additional rows for each 64K section of the 256K DRAM (8 total), and four total additional columns. When defective memory locations are encountered on a chip during probe, one of the additional rows or columns will be physically enabled to replace the defective memory location(s) – this is accomplished by laser repair technology. This will be totally transparent to the device user, thus he will not notice a change in either performance or reliability of the part. Use of redundancy for the 256K generation enhances probe yield, especially in the early production life; this translates to lower device cost to the DRAM user.

MULITPLEXED SENSE AMPLIFIERS

An important parameter to examine in judging a dynamic RAM's sensing capability is the capacitance ratio of the bit lines to the memory cell. Ideally, the full storage voltage of a memory cell should appear on the sense amplifier input; however, this is not practical since the input to the sense amplifier is composed of a capacitive network. The objective then is to minimize the ratio of the bit line capacitance to the memory cell capacitance in order to minimize the voltage loss and time delay of a signal enroute from a memory cell to the sense amplifier input. The multiplexed scheme allows the sense amplifier to be placed in the middle of the bit line instead of at either end, allowing selection to occur in only half of the bit line as determined by row address bit 7. Only the half of the bit line selected at any given time will contribute to the capacitive loading on the selected cell. Consequently, the voltage loss and time delay of a memory cell signal will be half as much with the multiplexed sense amplifier architecture. The bit line to memory cell capacitance ratio for 256K generation devices is approximately 8:1.

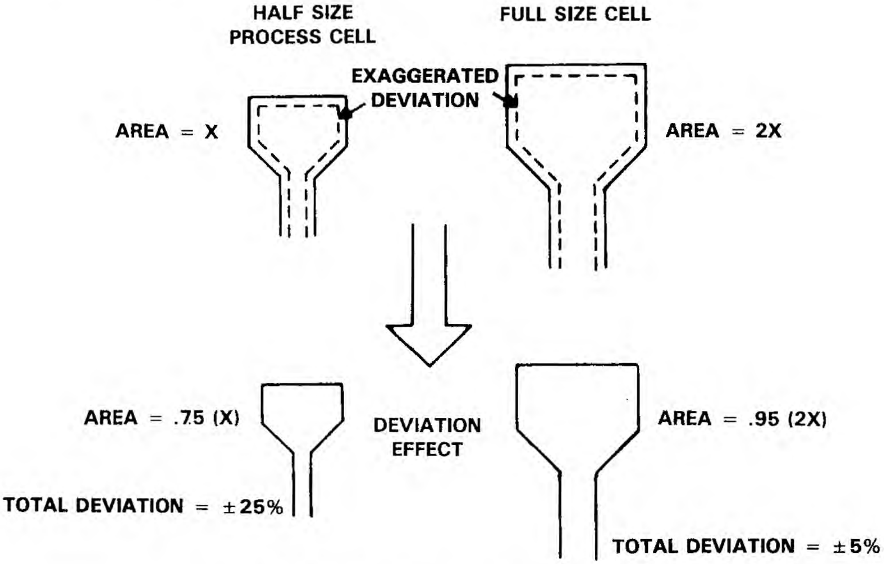


Figure 3. Process Deviation Effects on Dummy Cells

SHARED COLUMN DECODERS

In order to conserve silicon real estate, multiplexed column decoders are utilized with the 256K devices. Figure 1 shows how the column decoders are located down the middle of the device. Each column decoder corresponds to two sense amplifiers in both the upper and lower half of the memory array, which necessitates only 256 column decoders. *Eight* column addresses are needed to decode one of 256 column decoders. Each decoder selects four data bits from four columns to be sent to the periphery where row and column bit 8 is used to select one of the four bits. The resultant four columns are decoded outside of the memory array (see paragraph on Architecture).

CONTROL SIGNALS USED

In order to take advantage of the relative functional and space saving attributes of dynamic RAM's, necessary control signal implementation must be considered. These control signals follow along the line of standard DRAM control signal implementation.

ADDRESSES (A0-A8), $\overline{\mathrm{RAS}}$, $\overline{\mathrm{CAS}}$

Eighteen address bits are necessary to decode one of 262,144 different memory locations. Since minimizing the pin count on a dynamic RAM is essential, the eighteen address lines are multiplexed. The architecture of the TMS4256 and TMS4257 can be visualized externally as a 512 row × 512 column memory cell matrix. Operating together with the row and column addresses will be the row address strobe ($\overline{\mathrm{RAS}}$) and column address strobe ($\overline{\mathrm{CAS}}$) signals, respectively. Initially, $\overline{\mathrm{RAS}}$ will select one of 512 rows from the nine row addresses presented at the DRAM address inputs. A multiplex (MUX) signal must be generated within the system in addition to $\overline{\mathrm{RAS}}$ and $\overline{\mathrm{CAS}}$ in order to cause multiplexing circuitry to switch from row to column addresses at the DRAM address inputs. A typical system will employ a shift register or delay line in generating $\overline{\mathrm{RAS}}$, MUX, and $\overline{\mathrm{CAS}}$ which will be synchronized by a system clock. It is important that generation of these control signals be synchronized with the host system to optimize the memory performance within the system.

WRITE ENABLE ($\overline{\mathrm{W}}$)

A read or write function is determined by a high or low state, respectively, of the write enable signal ($\overline{\mathrm{W}}$). $\overline{\mathrm{W}}$ will be generated by the host controller directly or through some decoding circuitry to the DRAM memory. The way in which it will appear at the DRAM $\overline{\mathrm{W}}$ input will depend on the type of memory cycle to be utilized with the dynamic memory. It is necessary that the generation of $\overline{\mathrm{W}}$ be coordinated with other control signal generation to accommodate the TMS4256 and TMS4257 memory cycle specifications.

Various combinations of these basic operational signals give the TMS4256 and TMS4257 the flexibility to adapt to different modes of operation.

AVAILABLE MEMORY CYCLE TYPES

READ CYCLE (REFERENCE FIGURE 4)

A read cycle will be utilized at a point in a host controller instruction sequence where external memory is required. The condition which dictates a read cycle is to have $\overline{\mathrm{W}}$ high for a specified setup time before the falling edge of $\overline{\mathrm{CAS}}$ (t_{RCS}). Valid data will be available at the DRAM Q output after a specified maximum period of time following $\overline{\mathrm{CAS}}$ low (t_{CAC}) and $\overline{\mathrm{RAS}}$ low (t_{RAC}). $\overline{\mathrm{W}}$ must be kept high until after both $\overline{\mathrm{CAS}}$ and $\overline{\mathrm{RAS}}$ return to high states. Data at output Q will remain valid for a maximum period of t_{OFF} following the rising edge of $\overline{\mathrm{CAS}}$ before Q will go into a high-impedance condition.

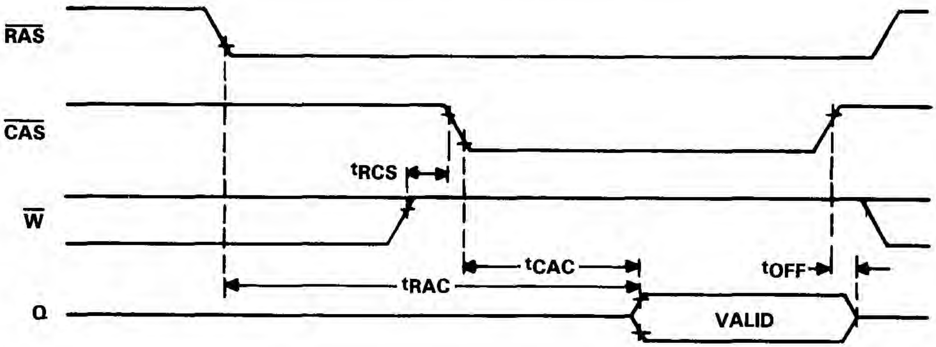


Figure 4. Read Cycle Timing

EARLY WRITE CYCLE (REFERENCE FIGURE 5)

Early write describes a write cycle where $\overline{W}$ is taken low a minimum time period (t_{WCS}) before $\overline{CAS}$ goes low. Valid data must be available before the falling edge of $\overline{CAS}$ (t_{DS}), and is latched by $\overline{CAS}$ at the D input. An important attribute of an early write cycle is that a high-impedance Q output is guaranteed. This will allow the user to connect both D and Q together without the utilization of three-state buffers for output control. A possible complication may arise in using a controller with a multiplexed address/data bus. Valid data will not be present on the address and data bus for a specified delay period following address latching; so $\overline{CAS}$ going low will be forced to wait for valid data before latching it. The $\overline{CAS}$ low period will still have to be satisfied which may cause the early write cycle to be longer than the corresponding implementation of a write cycle.

WRITE CYCLE (REFERENCE FIGURE 6)

A write cycle will be instituted at a point in the instruction sequence of a host controller when data is to be stored in external memory (DRAM in this case). Invalid data will be present on the DRAM Q output since the pre-condition of output three-state (t_{WCS} in the early write cycle) was not met. However, the Q output will not be utilized during a write cycle so valid data is not necessary at Q. A problem arises with the implementation of a bidirectional data bus where contention will occur between invalid data present at Q and valid data available at D. If bidirectional data lines are required, an external set of three-state buffers will be necessary to prohibit bus contention between input and *output lines of* the DRAM during a write cycle. A standard write cycle will have $\overline{W}$ going low following $\overline{CAS}$ – the data must be valid before $\overline{W}$ goes low. In addition, the minimum specified $\overline{CAS}$ low to $\overline{W}$ high hold time (t_{WCH}) is important in order to guarantee sufficient time to write data to the addressed memory location. Closely related to this are the requirements for minimum $\overline{W}$ low to $\overline{RAS}$ high (t_{RWL}) and $\overline{W}$ low to $\overline{CAS}$ high (t_{CWL}). A write cycle will typically be used with a system with multiplexed address and data lines where data will become valid for some delay period following address latching. Also, the system may require $\overline{CAS}$ to go low as soon as possible in order to comprehend the minimum

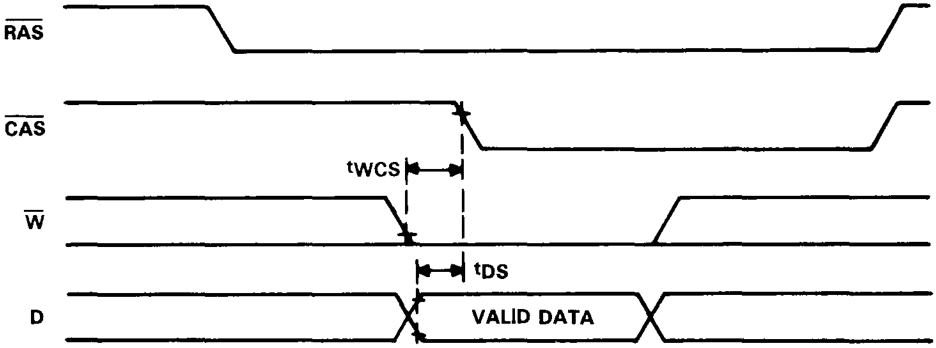


Figure 5. Early Write Cycle Timing

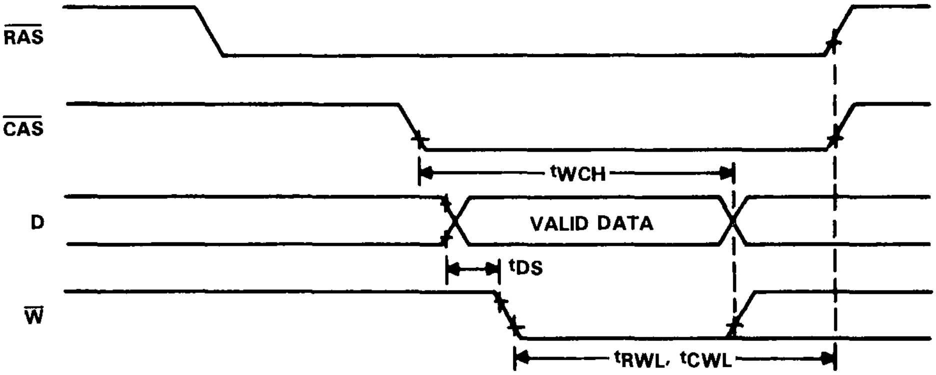


Figure 6. Write Cycle Timing

specified $\overline{CAS}$ low time – data may not be set up by this point which will prohibit an early write cycle.

READ-MODIFY-WRITE CYCLE (REFERENCE FIGURE 7)

For those applications which require a specific memory location to be read and written to in the same memory cycle, a read-modify-write cycle will be the most efficient cycle. $\overline{W}$ must be high a specified setup time before the falling edge of $\overline{CAS}$ to ensure that the memory location will be read. Recall that this is the same procedure as with a standard read cycle; so valid data will appear at the output Q a maximum t_{CAC} time from the falling edge of $\overline{CAS}$. A read-modify-write cycle can be instituted by bringing $\overline{W}$ low after a minimum time from the falling edge of $\overline{CAS}$ (t_{CWD}). This minimum time will guarantee sufficient time for the data located at the accessed memory location to be enabled and sensed during the read part of the cycle. The low edge of W will latch valid input data on the data bus to the D input of the DRAM. Subsequently, satisfying minimum periods for t_{CWL} and t_{RWL} will ensure that valid data at the D input will properly be written to the specified memory location. If a bidirectional data bus is to be utilized, three-state buffers will be necessary to ensure no contention on the data bus between input and output data of the DRAM. Any type of application where memory must be read and manipulated quickly may utilize read-modify-write cycles.

PAGE MODE (REFERENCE FIGURE 8)

Page mode is an optional mode, designated by TMS4256, which allows the user to generate numerous memory accesses with only a single $\overline{RAS}$ low edge. It is important to note that these accesses will occur only along the same row and will require subsequent $\overline{CAS}$ pulses to coincide with new column addresses. After a particular row is latched by the $\overline{RAS}$ low signal, $\overline{CAS}$ will latch individual memory locations as it accesses various column locations within the latched row. It is necessary to provide a different column address for every memory location that is desired. In this manner, page mode can be utilized in an accelerated access of nonsuccessive memory locations if the application demands it. The $\overline{RAS}$ low maximum specification (10.0 μs) is the limiting factor here and will allow approximately 64 read or write cycles, or 42 read-modify-write cycles before $\overline{RAS}$ must be taken high to satisfy DRAM precharge requirements (– 150 ns devices). Page mode memory access

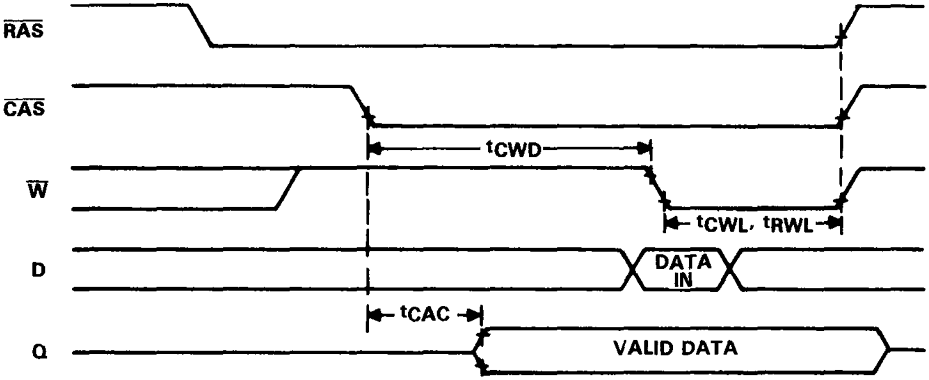


Figure 7. Read-Modify-Write Cycle Timing

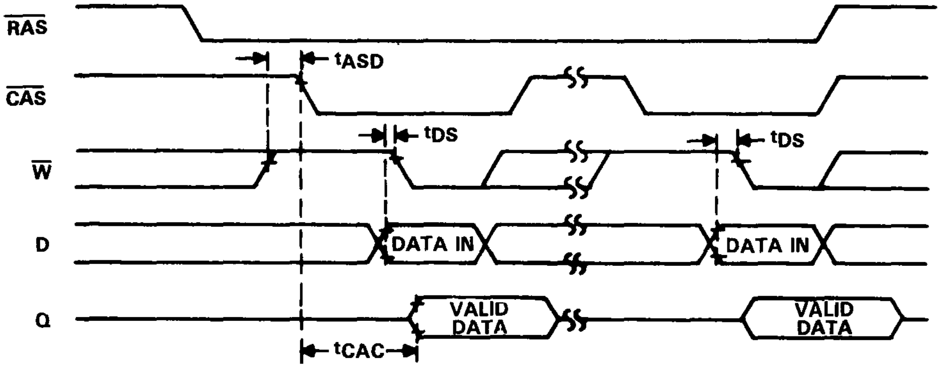


Figure 8. Page Mode R-M-W Cycle Timing

can be instituted for memory read, write, or read-modify-write cycles.

NIBBLE MODE (REFERENCE FIGURE 9)

A nibble mode cycle is a cycle where up to four successive bits are read from or written to memory in an accelerated manner over conventional cycles. The TMS4257 incorporates circuitry providing nibble mode operation. All preconditions for read, write, and read-modify-write cycles hold true with nibble mode as well. Only the initial memory address needs to be presented to the DRAM for proper nibble mode operation. $\overline{RAS}$ will stay low for the complete memory operation, while $\overline{CAS}$ is toggled up to four times to drive the desired memory cycle functions. If $\overline{CAS}$ continues to be toggled past four times, the same four bits will be recycled in a modular fashion. Note that only the initial column address is needed since the other three bits are linked to the first one.

NIBBLE MODE VS. PAGE MODE COMPARISON

Following examination of differences between page mode and nibble mode operation, the design engineer must decide which option will best suit his application. Basically, three factors need to be analyzed:

1. The relative performance improvement of one option over the other,
2. The required additional hardware to implement one option over the other, and
3. The interval at which the processor can be interrupted to reaccess the DRAM.

Examination of the specification for particular speed ranges reveals, on the average, a 25-30% performance improvement of nibble mode over page mode. Practical implementation, however, shows approximately a 10% performance improvement. This practical implementation involves aligning important signal edges with an appropriate clock according to device specification guidelines. Hardware implementation for both the TMS4256 and TMS4257 will be approximately the same. The major difference with respect to hardware is that nibble mode requires only one row and one column address for four output data bits, while page mode requires one row and a separate column address for every data output; however, if the column address are sequential, they could be generated by an external counter. Nibble mode requires a $\overline{RAS}$ precharge period and a new row and column address to continue past four output bits, while the $\overline{RAS}$ low period maximum dictates the maximum number of page mode cycles that can occur before a precharge period must be inserted. As a result, those applications which require the host controller to not be interrupted while reading or writing a long continuous data stream may warrant the use of the page mode option. If the host controller can afford to be interrupted every four bits (from the data stream perspective) or if only short data bursts (< four bits) are necessary, the nibble mode option may be preferred. It is important to note that nibble mode will continue to outperform page mode for long data streams; but this is at the expense of host controller interruption every four bits. In addition, page mode has the capability to manuever randomly through an accessed row if the application demands it.

REFRESH OPTION

Inherent with the design of a dynamic RAM memory subsystem is the need to refresh the dynamic RAM's. Refresh is necessary because charge in the individual memory cells is stored dynamically. Every row address cycle will regenerate the stored data in all memory locations of the addressed row. Thus the full memory may be refreshed in 256 cycles. Since refresh is an important consideration in the design of dynamic RAM systems, various refresh alternatives are important to retain system design flexibility. The TMS4256 and TMS4257 employ three separate ways in which dynamic RAM refresh can be implemented:

1. $\overline{RAS}$-only refresh,
2. $\overline{CAS}$-before-$\overline{RAS}$ refresh, and
3. Hidden refresh.

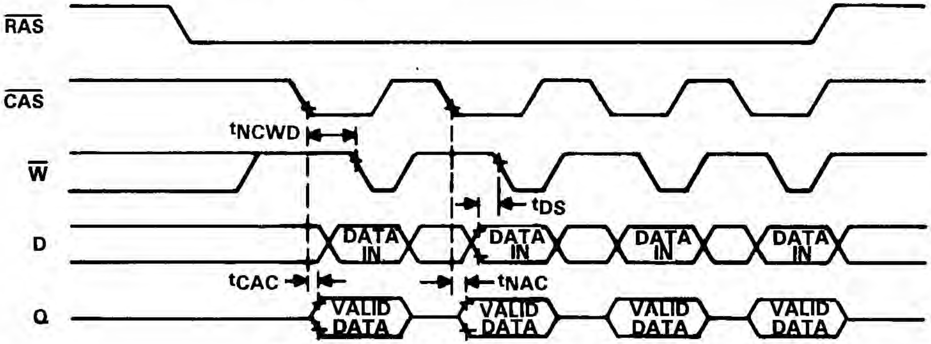


Figure 9. Nibble Mode R-M-W Cycle Timing

RAS-ONLY REFRESH

This is the historical approach to refreshing dynamic RAM's. An external chip timer will enable an external counter at the appropriate refresh period to output the current row to be refreshed – the counter will be automatically incremented for the next refresh cycle. From the DRAM point of view, RAS is to be brought low a delay period following CAS going high (Figure 10). It is assumed that the row address has been generated by the external counter and it is valid a specified setup period before RAS goes low. Subsequent row refreshes can be done by bringing RAS high following the RAS low period, satisfying RAS high time (precharge), and again bringing RAS low (it is assumed again that the subsequent row address is valid and is setup a specified period before RAS goes low). There is no restriction as to the number of refresh cycles that can be accommodated sequentially. This is referred to as a "burst" refresh when several refresh cycles are done together. The periodic interjection of refresh cycles, one row at a time, is usually referred to as distributed refresh. Note that row address 8 is not necessary to accommodate the 256 cycle refresh.

CAS-BEFORE-RAS REFRESH

This feature will be available on the TMS4256 and TMS4257 dynamic RAM's. This feature includes an on-chip refresh counter to eliminate the requirement for an external refresh counter. Examination of the cycle (Figure 11) reveals that CAS must be low a specified setup period before RAS goes low to enable a CAS-before-RAS refresh cycle. For successive refresh cycles, CAS can remain low while RAS is taken high to satisfy precharge requirements and obtain the next refresh address. From a system point of view, the CAS-before-RAS option allows a level of address multiplexing (for refresh) and refresh counter circuitry to be eliminated from system dynamic RAM design. The on-chip counter will enable the present refresh address when the CAS-before-RAS refresh preconditions are met. It is possible that the use of an available DRAM controller device, already provided with multiplexing and counter features, may not effectively utilize the CAS-before-RAS refresh feature.

HIDDEN REFRESH

Hidden Refresh is a form of CAS-before-RAS refresh which involves attaching a refresh cycle (or multiple refresh

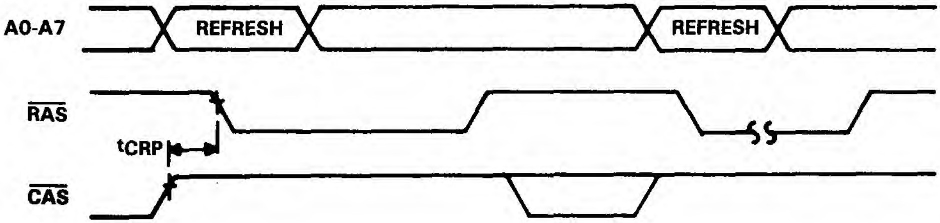


Figure 10. RAS-Only Refresh

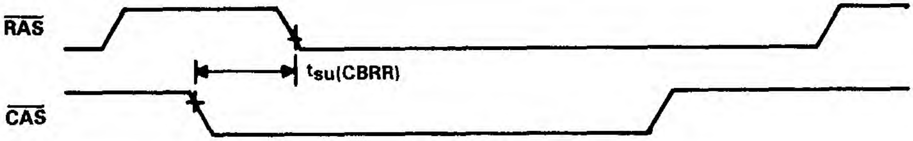


Figure 11. CAS-Before-RAS Refresh

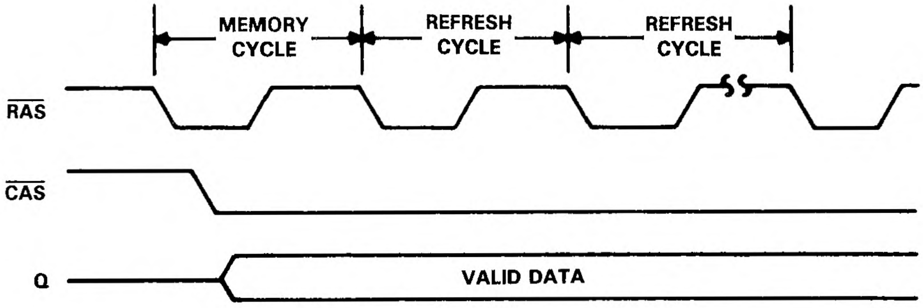


Figure 12. Hidden Refresh

cycles) directly behind a normal memory cycle (Figure 12). Instead of bringing $\overline{\text{CAS}}$ high at the end of the memory cycle as is normal, $\overline{\text{CAS}}$ remains low indicating hidden refresh cycles are to be implemented. $\overline{\text{RAS}}$ must be brought high before hidden refresh begins in order to satisfy $\overline{\text{RAS}}$ precharge requirements. This will allow sufficient time for the current refresh row address (internally generated) to be presented at the DRAM address inputs and select the appropriate row to be refreshed. Data that was presented at Q during the memory access cycle remains valid throughout the following refresh cycle(s). Multiple refresh cycles can be implemented simply by continuing to keep $\overline{\text{CAS}}$ low, bringing $\overline{\text{RAS}}$ high again to satisfy precharge, and providing the next refresh address at the appropriate time.

CONCLUSION

The purpose of this application note has been to introduce the potential 256K DRAM designer with 256K DRAM features and flexibility. Such features as nibble mode, page mode, read-modify-write, early write (etc.) allow the designer to use the memory to fit his particular application. Refresh, historically a problem in DRAM design, has been minimized as a problem with several options. The memory depth advantages of dynamic RAM's can be easily taken advantage of with the TMS4256 and TMS4257 with minimal interface problems.

TMS4256/TMS4257 Topology

INTRODUCTION

Effective dynamic RAM testing requires a thorough knowledge of the exact topology of the memory array. The most critical test routines test the pattern sensitivity of selected memory cells by performing memory accesses on surrounding cells, and monitoring the selected cells. Changes in the data stored in the monitored cell will reveal any sensitivities that the cells may have.

A total of eighteen address bits are needed to decode the one of 262,144 (256K) memory locations. The addresses are multiplexed as nine row addresses (latched by the falling edge of $\overline{\text{RAS}}$) and nine column addresses (latched by the falling edge of $\overline{\text{CAS}}$); which decode one of 512 rows or one of 512 columns, respectively. The nine addresses (A0-A8) appear on the pinout of the 256K in Figure 2(a), and are compatible with previous generation 64K DRAM devices. The only difference is the addition of the ninth address on pin 1 for the 256K device which was a no connect on 64K devices.

INTERNAL ADDRESS WEIGHTING

A closer examination of the internal addressing scheme reveals that the external pin address names do not correspond with internal address weightings. This is not relevant to system operations since the address translation will be transparent; however, testing for memory cell pattern sensitivity depends on true address weighting. This is necessary so that actual adjacent and diagonal memory cells will be exercised when monitoring a selected memory cell for pattern sensitivities. Tables I and II indicate the true row and column addresses as a function of the external address names. It is important to note that the respective row and column internal addresses do not correspond to the same external address pin names.

Table I. Row Address Weightings

Desired Row Address	Weight	TMS4256	
		Pin Name	Pin Number
RA8	2^8	A8	1
RA7	2^7	A7	9
RA6	2^6	A0	5
RA5	2^5	A2	6
RA4	2^4	A1	7
RA3	2^3	A5	10
RA2	2^2	A4	11
RA1	2^1	A3	12
RA0	2^0	A6	13

Table II. Column Address Weightings

Desired Column Address	Weight	TMS4256	
		Pin Name	Pin Number
CA8	2^8	A7	9
CA7	2^7	A0	5
CA6	2^6	A2	6
CA5	2^5	A1	7
CA4	2^4	A5	10
CA3	2^3	A4	11
CA2	2^2	A3	12
CA1	2^1	A6	13
CA0	2^0	A8	1

Since the external address names (A0-A8) correspond to different relative row and column address weights, the following multiplex configuration will simplify address decoding for the 256K DRAMs (Figure 1).

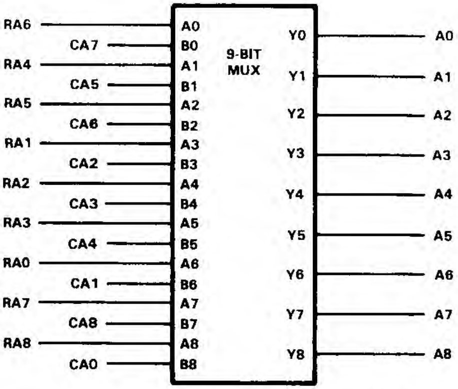


Figure 1. 256K DRAM Row and Column Multiplexer

CELL NEIGHBOR DEFINITION

With the internal address weights having been determined, the actual memory array layout must be examined to determine which memory cells are near and nearest neighbors to a given monitored memory cell. Figure 2 represents the steps of transition from the device package to the actual memory cell topology. Figure 2(a) is a representation of the TMS4256 or TMS4257 within the package, and Figure 2(b) is a photograph showing the major functional blocks of the 256K DRAM. The bit map of the memory array is shown in Figure 2(c), while the actual cell topology for a portion of the upper and lower memory arrays are shown in Figure 2(d). From the topology (Figure 2d), there are four possible cell orientations which are dependent upon the row location of the monitored memory cell. The four memory cells outlined in Figure 2(d) indicate the four possible orientations that can occur (they are located in four separate rows). This fundamental set of cell orientations is repeated sequentially throughout the memory array. Using Figures 3(a) — 3(d) it can seen that there will be a total of 3 nearest neighbors and 9 near neighbors for each of the four orientations. In the figures, the cell marked with an X represents the cell of interest, the dark shaded cells are the nearest neighbors, and the light shaded cells represent the near neighbors.

Let (R,C) represent any cell location where R = row and C = column. The following simple routine can be utilized to develop an algorithm which calculates the nearest and near neighbors to the monitored memory cell. Simply take the row number of the monitored memory cell and divide it by four (R/4); the remainder from this calculation can be used to qualify four sets of algorithms pertaining to four sets of rows. Alternately, RA0 and RA1 may be used to determine the neighbors as indicated in the following routine.

Row =	0,4,8,...1FC	1,5,9,...1FD	2,6,A,...1FE	3,7,B,...1FF
	RA0 = 0	RA0 = 1	RA0 = 0	RA0 = 1
	RA1 = 0	RA1 = 0	RA1 = 1	RA1 = 1
Remainder of R/4 =	0	1	2	3
NEAREST NEIGHBOR				
	R+1,C−1	R−2,C+0	R−2,C+0	R−1,C+0
	R+1,C+0	R−1,C+0	R+1,C+0	R−1,C+1
	R+2,C+0	R−1,C+1	R+1,C−1	R+2,C+0
NEAR NEIGHBOR				
	R−2,C+0	R−3,C+0	R−2,C−1	R−3,C+0
	R−1,C−1	R−3,C+1	R−2,C+1	R−3,C+1
	R−1,C+0	R−2,C−1	R−1,C−1	R−2,C+0
	R+0,C−1	R−2,C+1	R−1,C+0	R+0,C−1
	R+0,C+1	R+0,C−1	R+0,C−1	R+0,C+1
	R+2,C−1	R+0,C+1	R+0,C+1	R+1,C+0
	R+2,C+1	R+1,C+0	R+2,C+0	R+1,C+1
	R+3,C−1	R+1,C+1	R+3,C−1	R+2,C−1
	R+3,C+0	R+2,C+0	R+3,C+0	R+2,C+1

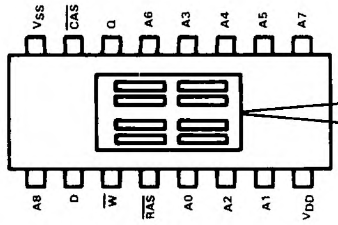


Figure 2(a). 256K DRAM Pinout

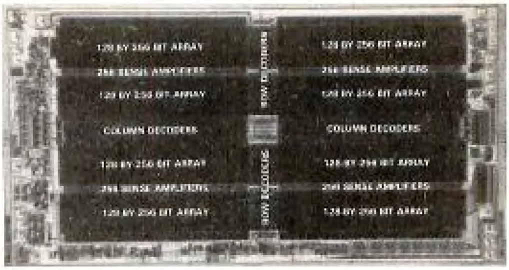


Figure 2(b). Functional Organization

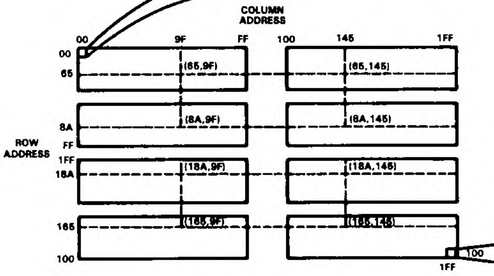


Figure 2(c). Memory Array Bit Map

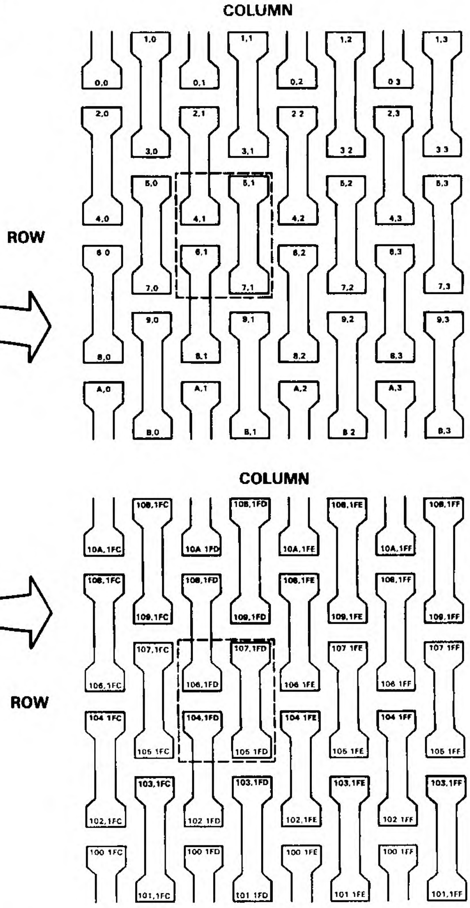
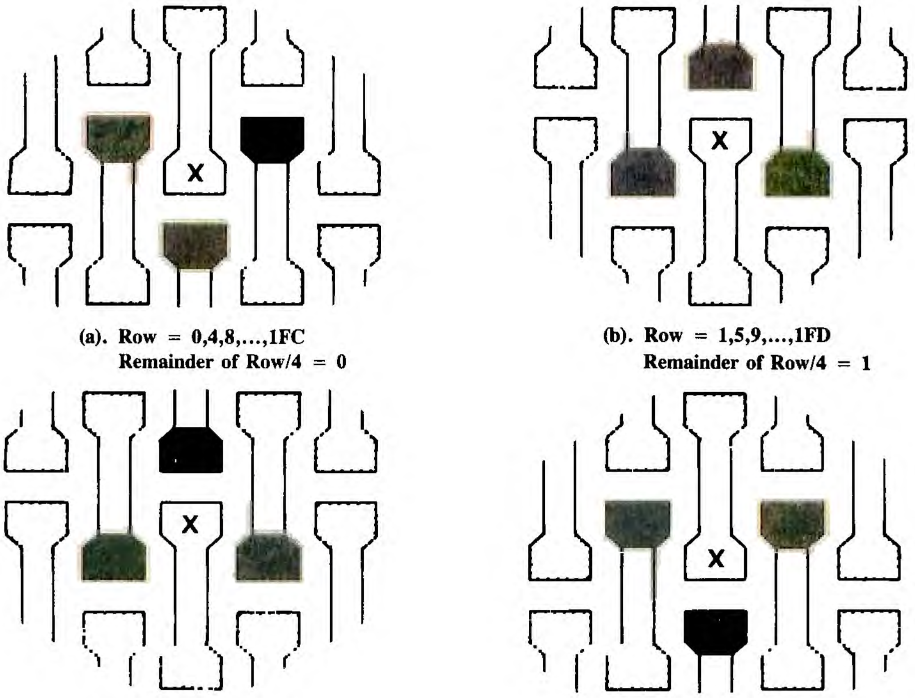


Figure 2(d). Upper and Lower Array Cell Topology



(a). Row = 0,4,8,...,1FC
Remainder of Row/4 = 0

(b). Row = 1,5,9,...,1FD
Remainder of Row/4 = 1

(c). Row = 2,6,A,...,1FE
Remainder of Row/4 = 2

(d). Row = 3,7,B,...,1FF
Remainder of Row/4 = 3

Figure 3. Near and Nearest Neighbors as a Function of Row

The algorithm for determination of nearest and near neighbors is dependent upon which of the four possible groups of rows the monitored cell is located (there is no column dependence). The lower half of the memory array mirrors that of the upper half; notice that the positive sequence of the row count proceeds from the bottom to the top in the lower half of the memory array. Consequently, the memory cell lay-out as shown in Figure 2(d) uses the above algorithms for both halves of the memory array.

INTERNAL DATA INVERSION

The data in the memory array is stored such that half of the cells are complemented with respect to the data input. The cells that store data in this inverted form are found in the odd rows; however, the internal data inversion is transparent to the device user as it is restored to a true state when read. As a result, the least significant row address (RA0) selects between true and complement data. The circuit shown in Figure 4 will provide the necessary data conversion if it is desired to compensate for the internal data inversion within the memory array.

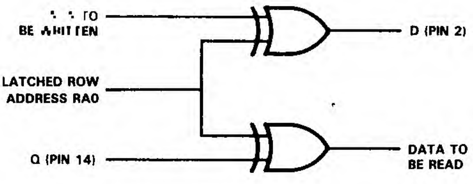


Figure 4. Circuit to Compensate for Internal Data Inversion

COMMON BIT SENSITIVITY

In order to allow the same basic memory design to implement nibble mode or x4 organization, there are actually four memory locations accessed during a single memory access cycle. The four data bits accessed are decoded by the most significant row and least significant column addresses (RA8 and CA0). RA8 selects between the upper and lower halves of the memory array, while CA0 selects between adjacent bit lines. As a result, an algorithm can be developed which calculates the other three memory locations accessed based on the position of a given memory location. This algorithm is as follows assuming the memory location (R,C) as the base location:

CA0 =	0	1	0	1
RA8 =	0	0	1	1
		MEMORY ALGORITHMS		
	R+0,C+1	R+0,C−1	R−256,C+0	R−256,C−1
	R+256,C+0	R+256,C−1	R−256,C+1	R−256,C+0
	R+256,C+1	R+256,C+0	R+0,C+1	R+0,C−1

WORD LINE SENSITIVITY

Further examination of the layout of the 256K (TMS4256) reveals that internally the word lines are in non-sequential order (0,1,3,2,...). To properly test for any word line to word line sensitivities that may occur, the testing routine must account for the non-sequential nature of the word lines. The same formula introduced earlier can be utilized to develop the algorithms for the adjacent word lines to the monitored word line (R = row). These algorithms will again be a function of the two least significant row addresses (RA0 and RA1) or the remainder of the current row divided by four.

Row =	0,4,8,...,1FC
	RA0=0
	RA1=0
		1,5,9,...,1FD
		RA0=1
		RA1=0
			2,6,A,...,1FE
			RA0=0
			RA1=1
				3,7,B,...,1FF
				RA0=1
				RA1=1

Remainder of R/4 =	0	1	2	3
Adjacent Rows	R−2 R+1	R−1 R+2	R+1 R+2	R−2 R−1

TMS4464 Topology

INTRODUCTION

In order to effectively test the interaction between individual cells in the TMS4464, it is necessary to have a knowledge of the memory array organization and cell topology. Cell sensitivity can be tested by accessing surrounding cells and monitoring the selected cell for changes in the stored data.

Sixteen address bits are needed to decode 1 of 65,536 memory locations. Eight row address bits are set up on pins A0 through A7 and latched by the falling edge of $\overline{RAS}$. Then eight column address bits are set up on pins A0 through A7 and latched by the falling edge of $\overline{CAS}$. Data to or from memory is presented in 4-bit wide words, which must be taken into consideration when developing algorithms for cell sensitivity tests.

This report presents the pinout of the TMS4464, a bit map of the array showing the cell topology, formulas for finding ''near'' and ''nearest'' neighbor cells, a circuit for compensating for internal data inversion, and formulas for testing word line sensitivity.*

Table 1 shows the true address bit significance for both row and column addressing. This information along with Figure 1(c) can be used to write various data patterns to the array.

CELL NEIGHBOR DEFINITION

Figure 1 depicts a step-by-step magnification of the TMS4464 from a view of the package to a closeup of the array topology. Figure 1(a) shows the chip pinout and Figure 1(b) is a photograph showing the major functional blocks of the device. The bit map of the memory array is shown in Figure 1(c) while the actual cell topology for three portions of the array are shown in Figure 1(d). Note that the address of cells labeled in Figure 1(c) show (R,CD), where R is the internal row address and CD is the internal column/databit address of the cell.† Figure 1(d) best illustrates the location of each of the four databits.

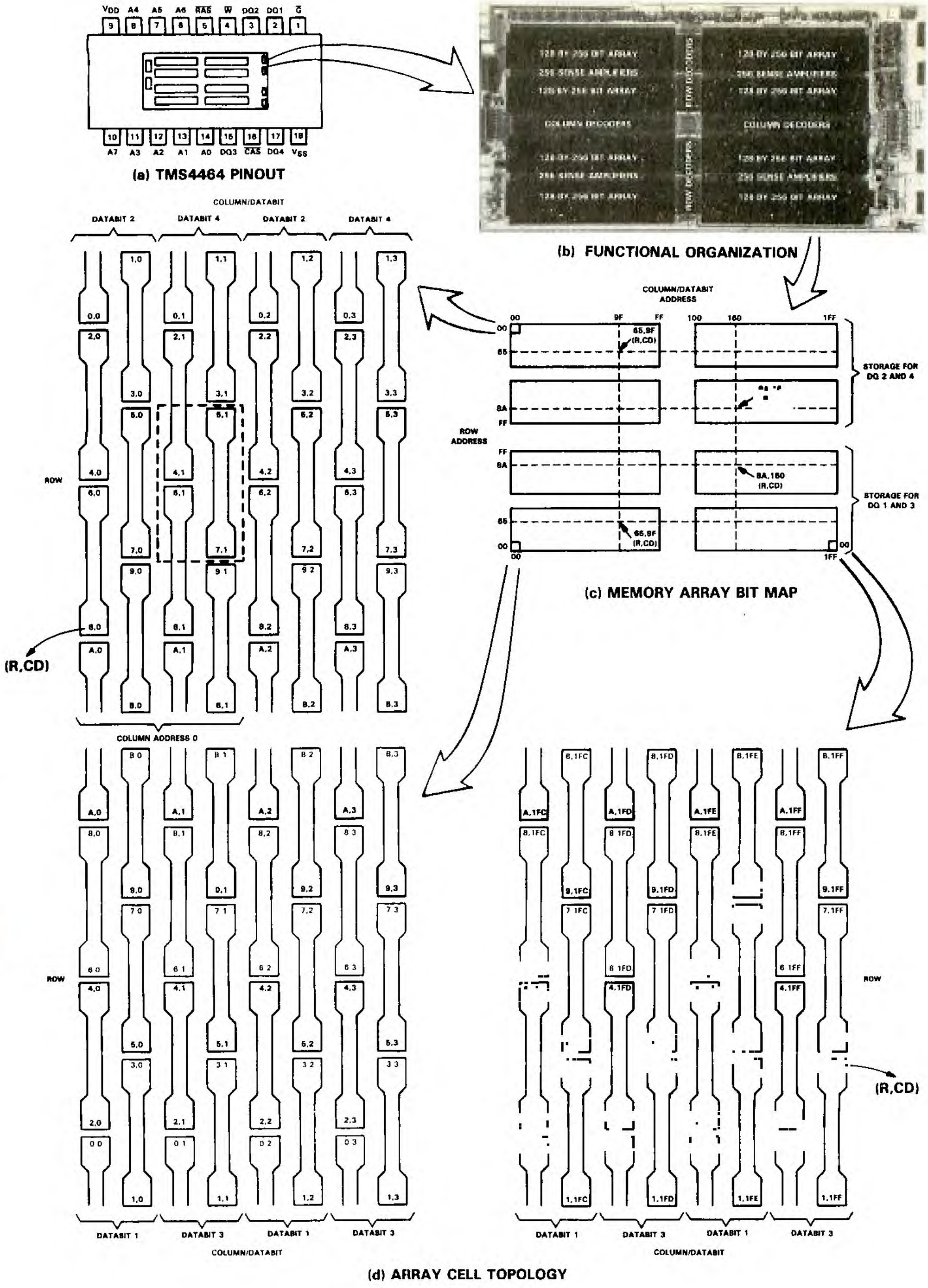
Due to the folded bit line approach used in this memory array, cells shown to be horizontally adjacent in Figure 1(d) are actually paired so as to map to the same column/databit (CD). For example, cell (0,0) is shown in Figure 1(d) to be horizontally adjacent to cell (1,0). However, both are addressed in column/databit 0 (CD 0; column 0, databit 2). Likewise, cells (0,1) and (1,1) are both in column/databit 1 (CD 1; column 0, databit 4). All four of these cells, along with the mirrored cells in the lower half of the array corresponding to databits 1 and 3, are contained in column address 0. (Note that column address 0 addresses all cells referenced by CD 0 and CD 1.) In summary, a column address selects four databits and each databit is comprised of two horizontally adjacent cell arrays. Based on this addressing sequence, it can be seen that four unique cell orientations exist for each column/databit (CD) corresponding to four separate rows. The four orientations are shown by the outlined box in Figure 1(d). In this particular case the four rows are rows 4, 5, 6, and 7; however, this fundamental cell orientation is repeated throughout the memory array and is related, as shown in Figure 2, to row addresses 0 and 1 (RA0 and RA1).

* Throughout the text of this report, the terms DQ1-DQ4 and databit 1-databit 4 are used synonymously. In actuality, DQ1-DQ4 refer to the external pins of the TMS4464 and databit 1-databit 4 refer to the internal bits accessed.

† The column/databit addresses are not the same as the column addresses but rather increment twice as fast. To convert from column/databit to column address, divide the column/databit by two. The resulting integer is the actual column address. In order to determine the actual databit, take the remainder of the division and multiply it by two. This result is scaled depending on the location of the cell. If it is located in the upper half of the array, add two to the result; if it is found in the lower half of the array, add one to the result. This final sum is the databit in question.

Table 1. Address Weightings

Desired Column Address	Desired Row Address	Weight	TMS4464	
			Pin Name	Pin Number
CA7	RA7	2^7	A7	10
CA6	RA6	2^6	A6	6
CA5	RA5	2^5	A5	7
CA4	RA4	2^4	A4	8
CA3	RA3	2^3	A3	11
CA2	RA2	2^2	A2	12
CA1	RA1	2^1	A1	13
CA0	RA0	2^0	A0	14



(a) TMS4464 PINOUT

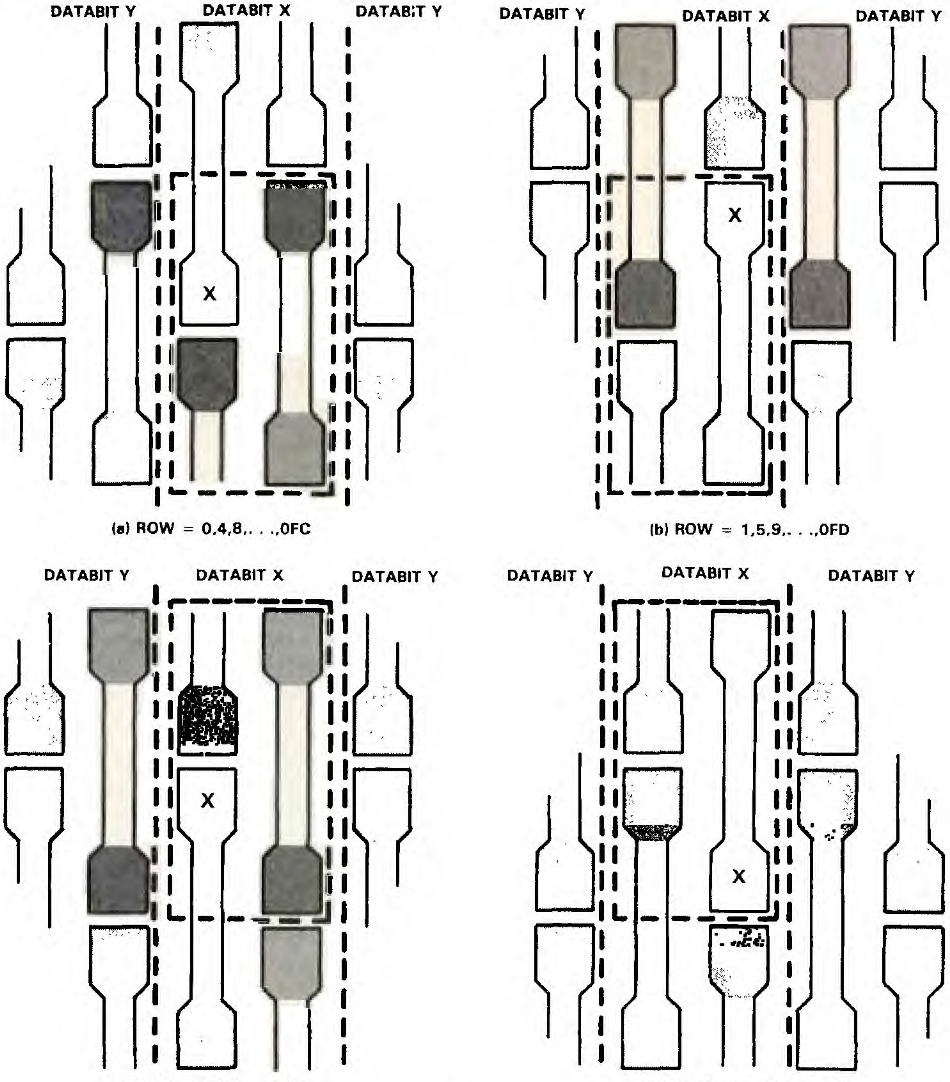
(b) FUNCTIONAL ORGANIZATION

(c) MEMORY ARRAY BIT MAP

(d) ARRAY CELL TOPOLOGY

Figure 1.

Figure 2 depicts the four possible cell orientations based on the row location of a selected cell. The boxes better show the selected cell orientation as it refers to the boxed area in Figure 1(d). Cells that surround any one given cell are called neighboring cells or neighbors, and are considered here for their degree of influence. As shown, there will be a total of 3 "nearest" neighbors and 9 "near" neighbors for each of the four orientations. In the figure, the cell marked with the X represents the monitored cell, the dark shaded cells are the nearest neighbors, and the light shaded cells represent the near neighbors. Near neighbors have a lesser degree of influence on the selected cell than do nearest neighbors. It should be noted here that one of the three nearest neighbors to a selected cell will always fall in an adjacent databit. If the selected cell is in databit 1 (DQ1), the adjacent databit will be DQ3 and a nearest neighbor will be located there. If the selected cell resides in DQ2, then one nearest neighbor will be in DQ4. A monitored cell in DQ3 will find a nearest neighbor in DQ1, and a monitored cell in DQ4 will have a nearest neighbor in DQ2. Further examples are shown in Appendix A.



(a) ROW = 0,4,8,. . .,0FC

(b) ROW = 1,5,9,. . .,0FD

(c) ROW = 2,6,A,. . .,0FE

(d) ROW = 3,7,B,. . .,0FF

Figure 2. Near and Nearest Neighbors as a Function of Row

The formulas for finding the near and nearest neighbors are given below. The row address of the selected cell is divided by four (R/4) and the remainder of this calculation is used to qualify four sets of equations pertaining to the four rows in each orientation. An alternate method of using RA0 and RA1 to qualify the set is also shown.

Let (R,CD) represent any cell location where R = ROW ADDRESS and CD = COLUMN/DATABIT ADDRESS.

As can be seen by Figure 1(c), the lower half of the memory array mirrors that of the upper half. Note the positive sequence of the row count from bottom to top in the lower half of the array. Figure 1(d) illustrates the location of the four databits accessed during a read or write operation, two in the upper array and two in the mirrored lower array.

Row=	0,4,8,...,0FC	1,5,9,...,0FD	2,6,A,...,0FE	3,7,B,...,0FF
	RA0=0	RA0=1	RA0=0	RA0=1
	RA1=0	RA1=0	RA1=1	RA1=1
Remainder of R/4=	0	1	2	3
		NEAREST NEIGHBOR		
	R+1,CD−1	R−2,CD+0	R−2,CD+0	R−1,CD+0
	R+1,CD+0	R−1,CD+0	R+1,CD+0	R−1,CD+1
	R+2,CD+0	R−1,CD+1	R+1,CD−1	R+2,CD+0
		NEAR NEIGHBOR		
	R−2,CD+0	R−3,CD+0	R−2,CD−1	R−3,CD+0
	R−1,CD−1	R−3,CD+1	R−2,CD+1	R−3,CD+1
	R−1,CD+0	R−2,CD−1	R−1,CD−1	R−2,CD+0
	R+0,CD−1	R−2,CD+1	R−1,CD+0	R+0,CD−1
	R+0,CD+1	R+0,CD−1	R+0,CD−1	R+0,CD+1
	R+2,CD−1	R+0,CD+1	R+0,CD+1	R+1,CD+0
	R+2,CD+1	R+1,CD+0	R+2,CD+0	R+1,CD+1
	R+3,CD−1	R+1,CD+1	R+3,CD−1	R+2,CD−1
	R+3,CD+0	R+2,CD+0	R+3,CD+0	R+2,CD+1

WORD LINE SENSITIVITY

As can be deduced from the topology, the word lines of the TMS4464 are in nonsequential order (0, 1, 3, 2,...). In order to test for word line to word line sensitivity, this fact must be taken into consideration. The formulas shown earlier in the description of near and nearest neighbors can be utilized here to find the adjacent word lines to a monitored word line. Each equation is a function of the two least significant row addresses (RA0 and RA1) or the remainder of the current row divided by four (R=row).

Row=	0,4,8,...,0FC	1,5,9,...,0FD	2,6,A,...,0FE	3,7,B,...,0FF
	RA0=0	RA0=1	RA0=0	RA0=1
	RA1=0	RA1=0	RA1=1	RA1=1
Remainder of R/4=	0	1	2	3
Adjacent Rows	R−2	R−1	R+1	R−2
	R+1	R+2	R+2	R−1

INTERNAL DATA INVERSION

Data is stored in the memory array such that half the cells are complemented with respect to the input data. The odd rows contain inverted data, while the even rows store the data in its true form. The inverted data is restored to a true state when read, making the inversion transparent to the user. The least significant row address selects between the true and complemented forms. Figure 3 shows a circuit to compensate for the internal data inversion within the memory array.

When row address 0 is low, the true data form is accessed and data is passed without inversion. When row address 0 is high, the inverted form is accessed and data is inverted as it is written to or read from the memory. Also, the 74LS241 remains ready to write data to the TMS4464 until $\overline{G}$ goes low. When this occurs, data is transferred from the TMS4464 to the system databus for read operations.

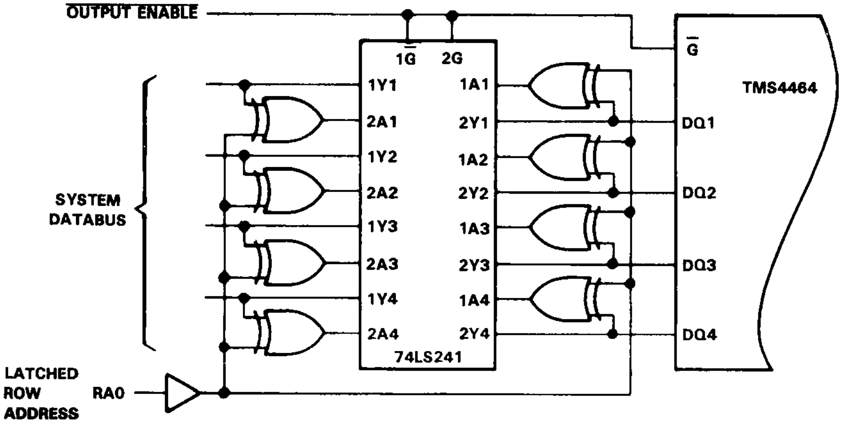


Figure 3. Circuit to Compensate for Internal Data Inversion

APPENDIX A

The two examples below use the formulas given in this report to find the nine near and three nearest neighbors to a selected cell. In order to illustrate the difference between column addresses and column/databit addresses, locations for the neighboring cells are stated using both notations.

The set of equations used will be determined by the selected row address (refer back to the listed equations on page 4 of the text). The equations to convert from column/databit (CD) to column address (Col) and from column address to column/databit are shown below.

CD=(Col*2)+constant1
where constant1=0 for DQ=1, DQ=2
constant1=1 for DQ=3, DQ=4

Col=INT[CD/2]
DQ=(MOD[CD/2]*2)+constant2
where constant2=1 for cell in lower half of array
constant2=2 for cell in upper half of array

In the examples, R=Row; C=Column,Col; CD=Column/Databit; DQ=Databit.

Example 1:

Row	5A		Row	5A
Col	F2	is equivalent to	Column/Databit(CD)	1E5
DQ	4			

(DQ4 indicates that cell is located in upper half of array)

NEAREST NEIGHBORS

Basic Formula	R,CD Format	R,C,DQ Format
R−2, CD+0	(R=58, CD=1E5)	(R=58, C=F2, DQ=4)
R+1, CD+0	(R=5B, CD=1E5)	(R=5B, C=F2, DQ=4)
R−1, CD+1	(R=59, CD=1E6)	(R=59, C=F3, DQ=2)

NEAR NEIGHBORS

R−2, CD−1	(R=58, CD=1E4)	(R=58, C=F2, DQ=2)
R−2, CD+1	(R=58, CD=1E6)	(R=58, C=F3, DQ=2)
R−1, CD−1	(R=59, CD=1E4)	(R=59, C=F2, DQ=2)
R−1, CD+0	(R=59, CD=1E5)	(R=59, C=F2, DQ=4)
R+0, CD−1	(R=5A, CD=1E4)	(R=5A, C=F2, DQ=2)
R+0, CD+1	(R=5A, CD=1E6)	(R=5A, C=F3, DQ=2)
R+2, CD+0	(R=5C, CD=1E5)	(R=5C, C=F2, DQ=4)
R+3, CD−1	(R=5D, CD=1E4)	(R=5D, C=F2, DQ=2)
R+3, CD+0	(R=5D, CD=1E5)	(R=5D, C=F2, DQ=4)

Example 2:

Row	A0		Row	A0
Col	46	is equivalent to	Column/Databit(CD)	8C
DQ	1			

(DQ1 indicates that cell is located in lower half of array)

NEAREST NEIGHBORS

Basic Formula	R,CD Format	R,C,DQ Format
R+1, CD−1	(R=A1, CD=8B)	(R=A1, C=45, DQ=3)
R+1, CD+0	(R=A1, CD=8C)	(R=A1, C=46, DQ=1)
R+2, CD+0	(R=A2, CD=8C)	(R=A2, C=46, DQ=1)

NEAR NEIGHBORS

R−2, CD+0	(R=9E, CD=8C)	(R=9E, C=46, DQ=1)
R−1, CD−1	(R=9F, CD=8B)	(R=9F, C=45, DQ=3)
R−1, CD+0	(R=9F, CD=8C)	(R=9F, C=46, DQ=1)
R+0, CD−1	(R=A0, CD=8B)	(R=A0, C=45, DQ=3)
R+0, CD+1	(R=A0, CD=8D)	(R=A0, C=46, DQ=3)
R+2, CD−1	(R=A2, CD=8B)	(R=A2, C=45, DQ=3)
R+2, CD+1	(R=A2, CD=8D)	(R=A2, C=46, DQ=3)
R+3, CD−1	(R=A3, CD=8B)	(R=A3, C=45, DQ=3)
R+3, CD+0	(R=A3, CD=8C)	(R=A3, C=46, DQ=1)

Latchup Immunity of the HVCMOS EPROM Family

CMOS technology provides N-Channel and P-channel MOS transistors as well as both NPN and PNP parasitic bipolar transitors. Figure 1 shows the HVCMOS process cross section illustrating the above mentioned devices.

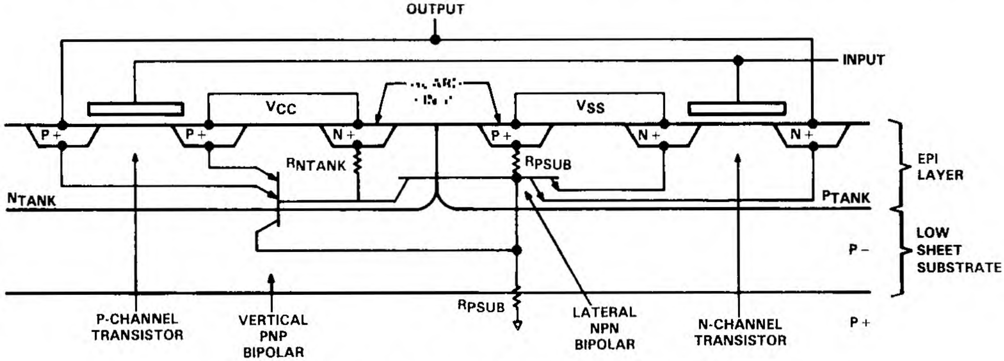


FIGURE 1. HVCMOS PROCESS CROSS SECTION

TI's HVCMOS family will drive full CMOS output levels due to the complimentary output driver configuration (N-channel pull down device and P-channel pull up device). This means that the system has direct access to the emitters of both parasitic bipolars via the output pins and access to the NPN parasitic bipolar emitter via the input pins. Figure 2 shows the equivalent parasitic SCR of the input pin. Figure 3 illustrates the output pin.

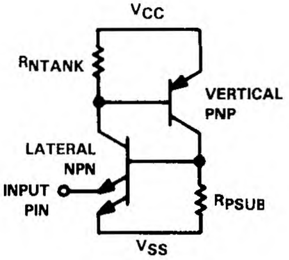


FIGURE 2. INPUT PIN PARASITIC SCR

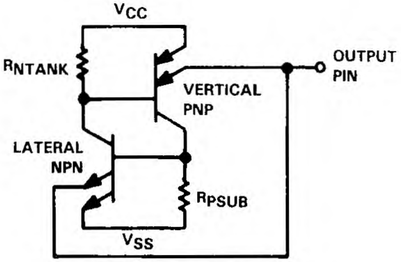


FIGURE 3. INPUT PIN PARASITIC SCR

The HVCMOS process and circuit design combine to offer significant improvements in system immunity:

- —The EPI substrate material lowers R_{PSUB} and is primarily responsible for holding off the lateral NPN device.
- —Full V_{CC} tank and V_{SS} substrate guardrings on inputs and outputs lower the critical base resistors R_{NTANK} and R_{PSUB}. This helps hold off the parasitic PNP and NPN respectively.
- —Maximum horizontal spacing from the emitter of the NPN to the vertical PNP (NPN base width) on all input and output pins minimizes the gain of the lateral NPN.

System latchup immunity on the TI HVCMOS EPROM family is a minimum of 250 mA on all input and output pins. This provides latchup immunity well beyond any potential current/voltage transients at the P.C. board level when the EPROM is interfaced to industry standard TTL or MOS logic devices.

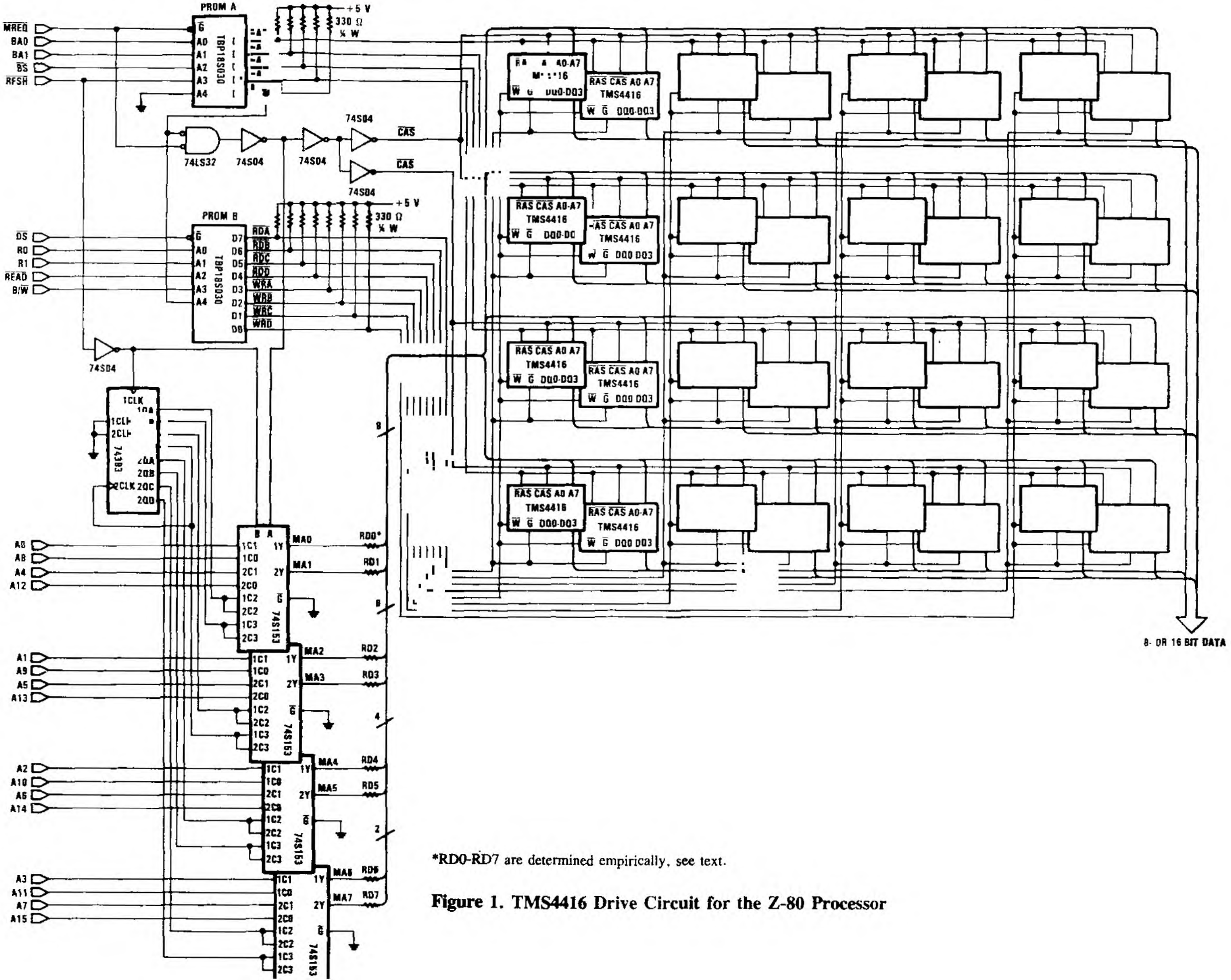
TTL Drivers for the TMS4416-15

Some form of driver circuitry is needed when DRAMs are used with processors, such as the Z-80 or Z-8000. One possible solution involves the use of a precision delay line; however, a more cost-effective and efficient approach uses TTL devices as drivers. Two versions of TTL driver circuits are shown in Figures 1 and 2. The first figure shows the drive circuit for a memory array using TMS4416-15 DRAMs and the Z-80 processor; Figure 2 shows the same array configured for use with the Z-8000 processor. Both circuits are designed to drive 256K bytes of memory arranged in either 8- or 16-bit words. They provide all DRAM control signals, address multiplexing, and refresh address generation. The circuits shown for the Z-80 and the Z-8000 use the hidden refresh provided by these devices so that refresh/access arbitration is not necessary. Time delays were selected to provide maximum performance from the TMS4416-15 with off-the-shelf components. (Enhanced operation could be obtained by hand selecting components for single applications.) A comparison of the two circuits will reveal the differences between the two. The following description applies to both circuits.

The memory array is arranged as 4 banks of 8 TMS4416s. Two TBP18S030 PROMs decode and generate the control signals for the drive circuit. BA0 and BA1 are used to select which bank of memory will be accessed. $\overline{\text{MREQ}}$ and $\overline{\text{ACCESS}}$ are NORed and then delayed by 3 inverters to provide a $\overline{\text{CAS}}$ signal. The MUX signal that is used to switch the 74S153 multiplexers and propagate the column address to the memories is taken from the output of the first inverter in the $\overline{\text{CAS}}$ delay. $\overline{\text{CAS}}$ is connected to all the devices in the array. (Since $\overline{\text{RAS}}$ acts as a chip enable, $\overline{\text{CAS}}$ will only activate the memories in the bank that has $\overline{\text{RAS}}$ active; this keeps the power consumption of the array lower than using $\overline{\text{CAS}}$ as select logic.) Two $\overline{\text{CAS}}$ drivers are used to reduce the effects of the capacitive load of the DRAM $\overline{\text{CAS}}$ inputs. (This also improves drive characteristics and reduces noise.) Series damping resistors have been added to reduce ringing on the address lines. These resistors should be between 15 and 68 ohms, depending on the circuit board layout, and can be determined by examining the address waveforms with an oscilloscope and selecting a value that produces the clearest signal. The desired 8- or 16-bit data word from the active bank is selected using R0, R1, and the $\overline{\text{READ}}$ line. R0 and R1 can be address lines from the Z-8001 or they can be generated from memory mapping logic. If the $\overline{\text{READ}}$ input is low during an access cycle, the output enable of the TMS4416 will be activated ($\overline{\text{RDA}}$-$\overline{\text{RDD}}$); a high input to $\overline{\text{READ}}$ will select a write output ($\overline{\text{WRA}}$-$\overline{\text{WRD}}$). Using this matrix, the memory can be divided into sixteen 16K × 8 or eight 16K × 16 blocks. The desired word width of the data output will be dependent on the microprocessor being used. For an 8-bit data bus the two data busses shown in the diagram would be connected in parallel. Since the Z-80 only directly accesses 64K of memory, bank select logic must be included in this memory system to provide higher order address lines. The design of the bank select circuitry has been left up to the user, but might include memory mapping or other logic.

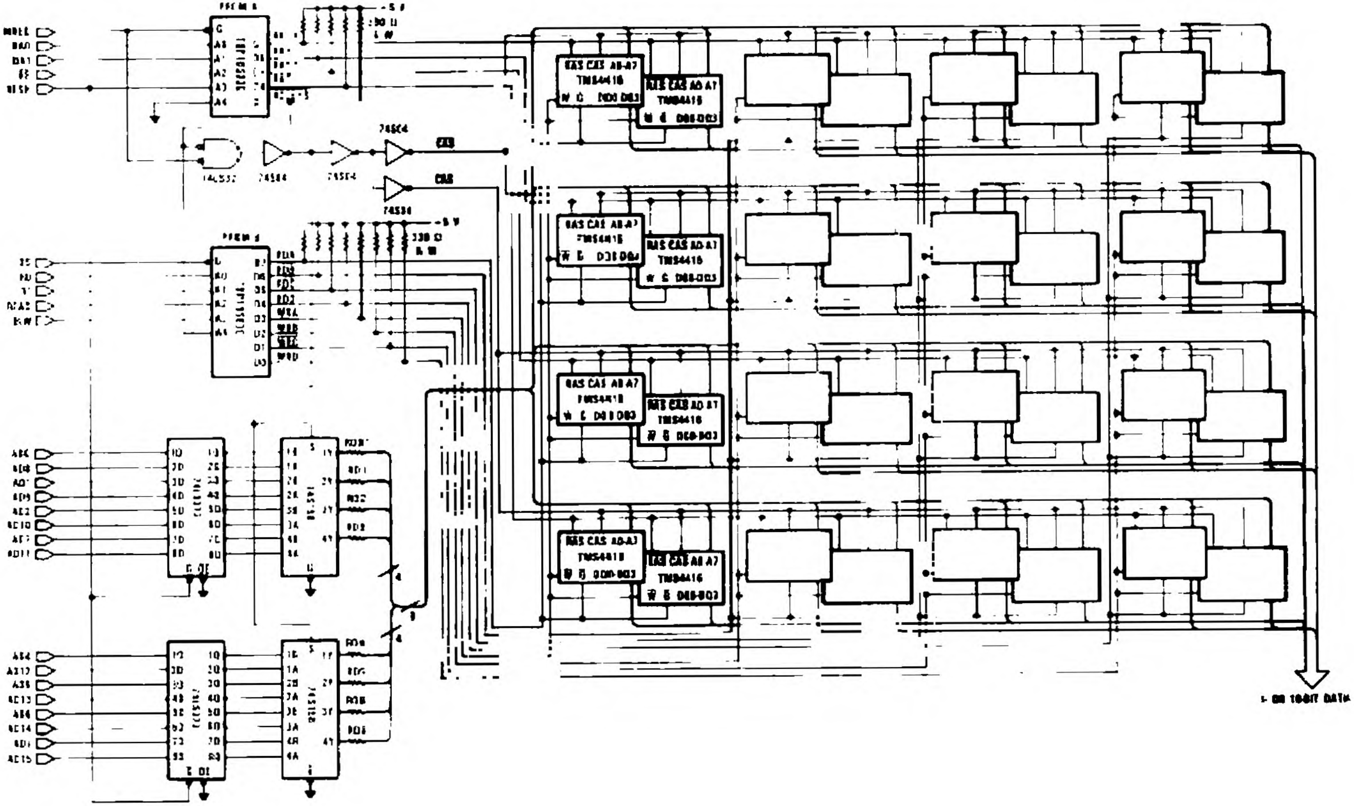
An external refresh counter has been added to the drive circuit for the Z-80 since the Z-80 internal refresh counter does not support 256 cycle refresh. (Application Brief DR-7 shows a circuit to add the extra refresh address bits similar to the implementation used here.) As the Z-8000 provides 9-refresh-address bits, its internal refresh counter was used.

A description of the signals used in both circuits previously illustrated is given in Table 1. Due to slight differences in the signals available from the Z-80 and Z-8000 processors, a slight modification of the interface between the processor and the TTL drive circuits shown will be required. The differences in the interface are shown in Figure 3. The $\overline{\text{DS}}$ signal is generated from the Z-80's $\overline{\text{RD}}$ and $\overline{\text{WR}}$ lines. The B/$\overline{\text{W}}$ input should be tied to a 5-10 kilohm pullup resistor. The $\overline{\text{RFSH}}$ signal can be decoded from the status lines of the Z-8000 as shown with the 74S138; however, it could also be done with other types of logic if desired. The address of the Z-8000A is only guaranteed valid for 35 ns so the address latches are necessary when using DRAMs with this microprocessor. $\overline{\text{MREQ}}$ is used to enable the 74LS373 transparent latches for both memory accesses and refresh cycles.



*RD0-RD7 are determined empirically, see text.

Figure 1. TMS4416 Drive Circuit for the Z-80 Processor



*RD0-RD7 are determined empirically—see text

Figure 2. TMS4416 Drive Circuit for the Z-8000

Table 1. Signal Description

SIGNAL NAME	DESCRIPTION
[illegible]	From the Z-80 or Z-8000, indicates address valid
BA0, BA1	Address for $\overline{RAS}$ selection, decoded from high order addresses
$\overline{BS}$	Board select to designate DRAM access, decoded from high order addresses
$\overline{RFSH}$	Z-80 output or decoded from the Z-8000 status outputs. Signals a refresh cycle
$\overline{DS}$	Z-8000 output indicating data valid on the multiplexed address/data lines
R0, R1	Address for read or write selection, decoded from high order addresses
$\overline{READ}$	If low, indicates a memory read and if high, indicates a memory write
B/$\overline{W}$	Indicates if the Z-8000 is doing a 8- or 16-bit memory access
A0-A15	Z-80 address outputs
AD0-AD15	Z-8000 multiplexed address/data lines

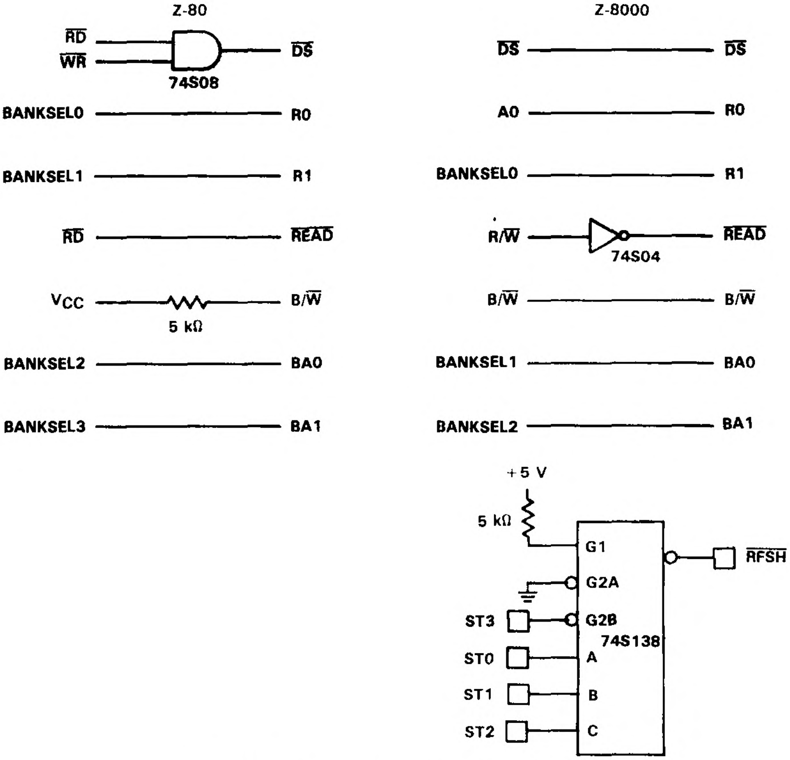


Figure 3. Interface Circuitry Differences

Tables 2 and 3 list the data for the PROMs to provide the control signals. Both the binary and hexadecimal programming data have been supplied.

The TTL drive circuits previously described allow the TMS4416-15 to operate at maximum speed. Although there are many ways to provide the necessary control signals for DRAMs, the drive circuits described will provide insight into the control logic that is necessary to use dynamic RAMs. TTL circuitry was selected in order to avoid the cost of a precision delay line.

Table 2. PROM A Program

PIN NAME	A3	A2	A1	A0	D7	D6	D5	D4	D3	HEX
FIGURE NAME	$\overline{RFSH}$	$\overline{BS}$	BA1	BA0	$\overline{RAS0}$	$\overline{RAS1}$	$\overline{RAS2}$	$\overline{RAS3}$	ACCESS	
	0	X	X	X	0	0	0	0	1	0F
	1	0	0	0	0	1	1	1	0	77
	1	0	0	1	1	0	1		0	B7
	1	0	1	0	1	1	0	1	0	D7
	1	0	1	1	1	1	1	0	0	E7
	1	1	0	0	1	1	1	1	1	FF
	1	1	0	1	1	1	1	1	1	FF
	1	1	1	0	1	1	1	1	1	FF
	1	1	1	1	1	1	1	1	1	FF

Table 3. PROM B Program

PIN NAME	$\overline{G}$	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	HEX
FIGURE NAME	$\overline{DS}$	B/$\overline{W}$	$\overline{READ}$	R1	R0	$\overline{RDA}$	$\overline{RDB}$	$\overline{RDC}$	$\overline{RDD}$	$\overline{WRA}$	$\overline{WRB}$	$\overline{WRC}$	$\overline{WRD}$	
	0	0	0	0	0	0	0	1	1	1	1	1	1	3F
	0	0	0	0	1	0	0	1	1	1	1	1	1	3F
	0	0	0	1	0	1	1	0	0	1	1	1	1	CF
	0	0	0	1	1	1	1	0	0	1	1	1	1	CF
	0	0	1	0	0	1	1	1	1	0	0	1	1	F3
	0	0	1	0	1	1	1	1	1	0	0	1	1	F3
	0	0	1	1	0	1	1	1	1	1	1	0	0	FC
	0	0	1	1	1	1	1	1	1	1	1	0	0	FC
	0	1	0	0	0	0	1	1	1	1	1	1	1	7F
	0	1	0	0	1	1	0	1	1	1	1	1	1	BF
	0	1	0	1	0	1	1	0	1	1	1	1	1	DF
	0	1	0	1	1	1	1	1	0	1	1	1	1	EF
	0	1	1	0	0	1	1	1	1	0	1	1	1	F7
	0	1	1	0	1	1	1	1	1	1	0	1	1	FB
	0	1	1	1	0	1	1	1	1	1	1	0	1	FD
	0	1	1	1	1	1	1	1	1	1	1	1	0	FE
	1	X	X	X	X	1	1	1	1	1	1	1	1	FF

TM4164EC4 Provides High-Density Memory Array

MOS Memory
Applications Engineering

This Application Report illustrates the use of the TM4164EC4 (64K × 4) Memory Module with the TMS4500A DRAM Controller (see Figure 1). The description of a memory board using both devices will be given along with full schematics, edge connector pinout, and signal description. An interface to the Intel 8086 microprocessor is also provided as a typical application.

BOARD DESCRIPTION

Designed by Tẹxas Instruments to demonstrate the TM4164EC4 in a system environment, the board provides a flexible, high-density memory array which is adaptable to most applications.

The board uses one TMS4500A and eight TM4164EC4s for 256K bytes of dynamic RAM memory on a 3.25 inch × 4.5 inch card. The TMS4500A gives the board a static appearance in the system, providing many of the necessary timing and control signals to the DRAM array. Each TM4164EC4 is comprised of four TMS4164 plasitic-leaded-chip-carriers and two ceramic chip capacitors, that are surface mounted on a PC substrate to form a single-in-line package (SIP).[1] The cost savings that can be realized with SIPs include reduced PC board size, fewer plated-through-holes, and the elimination of bypass capacitors on the motherboard.

The TM4164EC4 SIPs are mounted on 0.350 inch centers, and occupy 6.16 square inches of board area

[1] See Appendix A

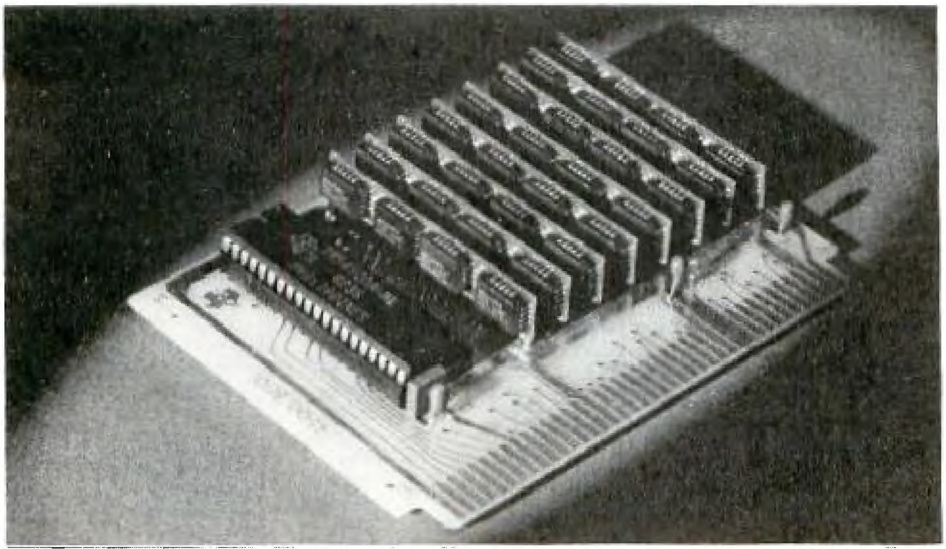


Figure 1. TM4164EC4/TMS4500A Memory Board

[8 × 0.350 × 2.2 inches (TM4164EC4 length)], for a density of greater than five memory devices per square inch. This is approximately a 2X density improvement with respect to DIPs. The TM4164EC4s can be mounted on centers as narrow as 0.200 inches if adequate cooling is provided. This would give a density of greater than nine memory devices per square inch or approximately a 3.5X improvement over *DIPs* for the above array.

The equivalent DIP implementation of the TM4164EC4 would require 68 plated-through-holes (four 16-pin packages and two, 2-lead capacitors) as opposed to the 22 required for a single TM4164EC4. The large number of plated-through-holes increase board cost and reduce the available PC board area for trace routing often requiring an increase in the number of board layers.

The on board capacitors eliminate the need for bypassing on the motherboard and offer superior performance over equivalent leaded capacitors due to the reduced lead inductance.

While the TMS4500A gives the board a static appearance and the TM4164EC4 provides a high-density memory array, the interconnect bus gives the board flexibility. All the signals necessary to provide for 8- or 16-bit operation, separate or common I/O, and internal or external memory refresh along with the address and control lines for the TMS4500A are brought to the board edge.

MEMORY ORGANIZATION

The memory is organized as two banks of 128K bytes, accessible in byte or word format (word = 16 bits). Each row is selected by $\overline{\text{RAS0}}$ or $\overline{\text{RAS1}}$ (see Figure 2) to provide 16 bits of available data. The 16 bits of data are then read or written in byte or word format by controlling the upper and lower $\overline{\text{CAS}}$ and $\overline{\text{WR}}$ signals ($\overline{\text{UCAS}}$, $\overline{\text{LCAS}}$, $\overline{\text{UWR}}$, and $\overline{\text{LWR}}$). The lower byte of data corresponds to D0-D7 and Q0-Q7, while upper data corresponds to 'D8-D15 and Q8-Q15. It is necessary to organize the memory as such to provide operation with 16-bit microprocessors that do both byte and word accesses. For microprocessors with 8-bit data busses, D0-D7 and Q0-Q7 are tied to D8-D15 and Q8-Q15, respectively. The $\overline{\text{CAS}}$ and $\overline{\text{WR}}$ signals are then used to multiplex and demultiplex the data onto and from the microprocessor data bus.

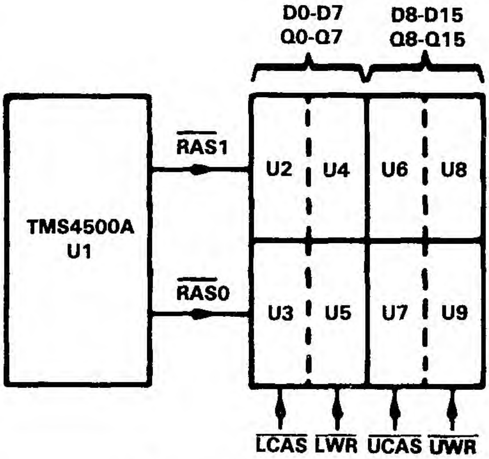


Figure 2. Block Diagram

Table 1. Pin Nomenclature

Name	Description
A0-A15	Address Inputs
$\overline{\text{ACR}}$	Access Control, Read
$\overline{\text{ACW}}$	Access Control, Write
ALE	Address Latch Enable
$\overline{\text{BRDEN}}$	Board Enable
$\overline{\text{CAS}}$	Column Address Strobe
CLK	Clock Input
D0-D15	Data In
FS0	Frequency Select 0
FS1	Frequency Select 1
GND	Ground
$\overline{\text{LCAS}}$	Lower $\overline{\text{CAS}}$
$\overline{\text{LWR}}$	Lower Write
$\overline{\text{REFREQ}}$	Refresh Request
REN1	$\overline{\text{RAS}}$ Enable 1
RDY	Ready
Q0-Q15	Data Out
TWST	Timing/Wait Strap
$\overline{\text{UCAS}}$	Upper $\overline{\text{CAS}}$
$\overline{\text{UWR}}$	Upper Write
+5	+5 Volts

BOARD OPERATION

As mentioned earlier the board provides a 16-bit data bus with separate data-in and data-out connections (see Figure 3). This allows the board to be configured for common (D0-D15 tied to Q0-Q15) or separate input/output (I/O). Notice that corresponding D and Q lines are located across from each other for easy interconnection (see Table II). Common I/O operation requires the memory array to be accessed in the early-write mode ($\overline{WR}$ low prior to $\overline{CAS}$). The board provides flexible control of the memory array by bringing $\overline{CAS}$ from the TMS4500A to the board edge allowing it to be combined with external logic to derive the $\overline{UCAS}$ and $\overline{LCAS}$ signals. If desired the $\overline{CAS}$ signal from the TMS4500A can directly drive the memory array by connecting $\overline{CAS}$ to $\overline{UCAS}$ and $\overline{LCAS}$ via jumpers J1 and J2. This type of configuration necessitates the use of $\overline{LWR}$ and $\overline{UWR}$ to control access to the memory array on write cycles. Also, all 16 bits of data will be active on a read cycle for both byte and word accesses. All the necessary signals needed to interface to the TMS4500A have been brought to the board edge. Notice that the binary weighting on the memory address outputs of the TMS4500A do not correspond to that of the TM4164EC4 memory addresses. This does not in any way affect the operation of the board as the TM4164EC4s are random-access devices. This configuration was chosen to simplify the board layout. Table III gives the relationship between the TMS4500A and TM4164EC4 addresses.

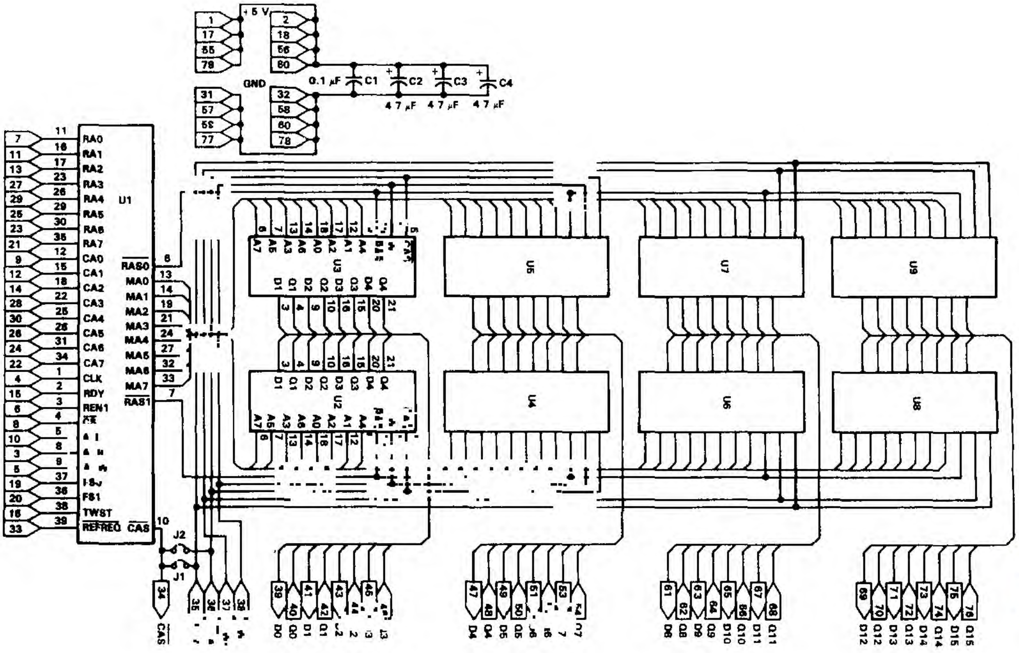


Figure 3. Board Schematic

Table II. Buss Format

Pin	Signal	Pin	Signal
1	+5	2	+5
3	$\overline{\text{ACR}}$	4	CLK
5	$\overline{\text{ACW}}$	6	REN1
7	A0	8	BRDEN
9	A8	10	ALE
11	A1	12	A9
13	A2	14	A10
15	RDY	16	TWST
17	+5	18	+5
19	FS0	20	FS1
21	A7	22	A15
23	A6	24	A14
25	A5	26	A13
27	A3	28	A11
29	A4	30	A12
31	GND	32	GND
33	$\overline{\text{REFREQ}}$	34	$\overline{\text{CAS}}$
35	$\overline{\text{UCAS}}$	36	$\overline{\text{LCAS}}$
37	$\overline{\text{UWR}}$	38	$\overline{\text{LWR}}$
39	D0	40	Q0
41	D1	42	Q1
43	D2	44	Q2
45	D3	46	Q3
47	D4	48	Q4
49	D5	50	Q5
51	D6	52	Q6
53	D7	54	Q7
55	+5	56	+5
57	GND	58	GND
59	GND	60	GND
61	D8	62	Q8
63	D9	64	Q9
65	D10	66	Q10
67	D11	68	Q11
69	D12	70	Q12
71	D13	72	Q13
73	D14	74	Q14
75	D15	76	Q15
77	GND	78	GND
79	+5	80	+5

Table III. Address Relationship

TMS4500A	TM4161EC4
MA0	A4
MA1	A1
MA2	A2
MA3	A0
MA4	A6
MA5	A3
MA6	A5
MA7	A7

8086 INTERFACE

The circuit is designed to operate with a 5 MHz 8086 in the maximum mode configuration without memory wait states (see Figure 4). The memory interface is simplified by configuring the memory for early write operation, which allows corresponding D and Q lines to be tied together for common I/O operation (see Table II). The board select logic is derived from addresses A18 and A19 and mapped via a 74S139 at address locations 40000-7FFFF hex (256K bytes). Address A17 is connected to REN1 of the TMS4500A to defferentiate between the two banks of memory (REN1 = 0, selects $\overline{\text{RAS}}$0; REN1 = 1, selects $\overline{\text{RAS}}$1). To provide for byte accesses, AD0 and $\overline{\text{BHE}}$ are combined with other logic to yield the necessary upper and lower $\overline{\text{CAS}}$ signals ($\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$). The 8284 is strapped for asynchronous ready operation to provide sufficient $\overline{\text{CAS}}$ access time on access-grant cycles. See the *TMS4500A Users Manual* for details of the TMS4500A operation. The $\overline{\text{AMWC}}$ and $\overline{\text{MRDC}}$ signals from the 8288 are used to derive $\overline{\text{ALE}}$ and $\overline{\text{ACR}}$ which initiates memory-access cycles. $\overline{\text{AMWC}}$ and $\overline{\text{MRDC}}$ are used instead of ALE from the 8288 to allow sufficient row address setup time to the memory (the row addresses are delayed by two propagation delays, 74LS373, and TMS4500A). This signal is also fed into the input of a 74S74 to be synchronized with the rising edge of CLK (see Figure 5). The output of the 74S74 is then combined with AD0 and $\overline{\text{BHE}}$ and $\overline{\text{CAS}}$ to form the upper and lower $\overline{\text{CAS}}$ signals. Synchronizing ALE of the TMS4500A with CLK ensures data valid at the memory before the falling edge of $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ (necessary for early write operation). The $\overline{\text{UWR}}$ and $\overline{\text{LWR}}$ signals are driven by $\overline{\text{AMWC}}$ to guarantee them to be valid before $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ low. $\overline{\text{AMWC}}$ is buffered to drive the 32 DRAMs.

This Application Report has illustrated the use of the TMS4500A and TM4164EC4 for a flexible, high-density memory array. The TMS4500A gives the board a static appearance, while the TM4164EC4 provides a density of greater than five memory devices per square inch. Higher densities can be obtained with narrower SIP spacings requiring adequate cooling. The 8086 interface provides a typical application and demonstrates the flexibility of the board. As circuit board designers strive to reduce board space and implement more functions on a board, the use of SIPs such as the TM4164EC4 will provide a vehicle by which this goal can be achieved.

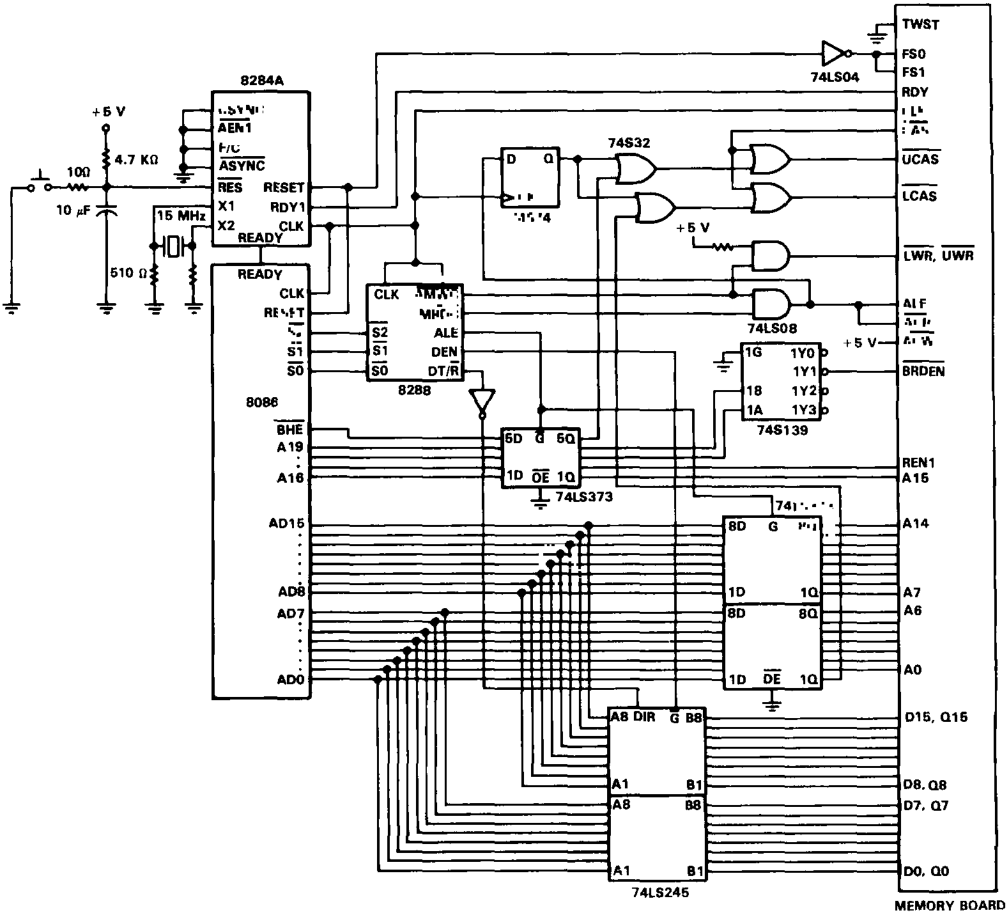


Figure 4. 8086 Interface

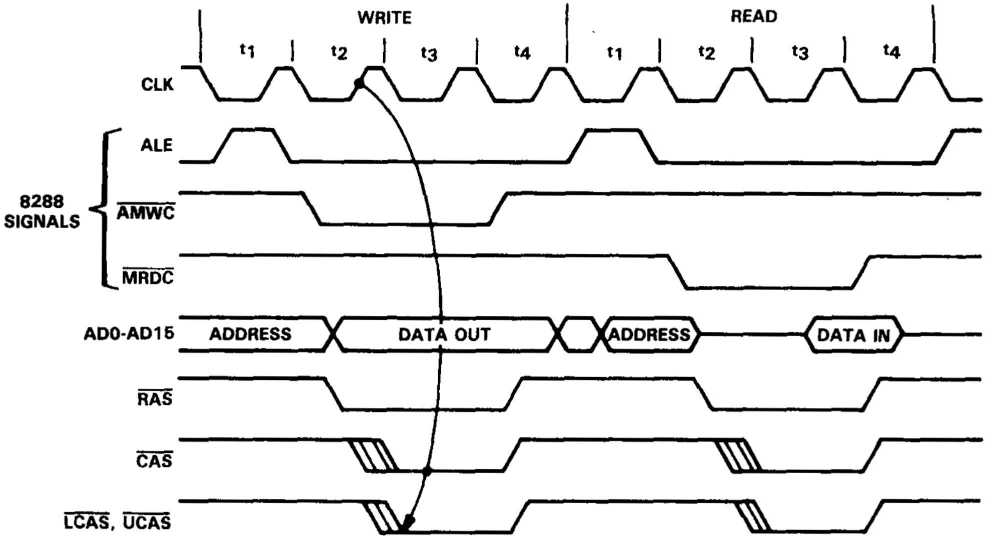


Figure 5. 8086 Interface Timing Diagram

APPENDIX A
TM4164EC4 PIN OUT AND FUNCTIONAL BLOCK DIAGRAM

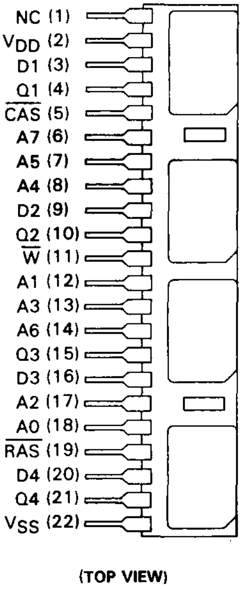


Figure A-1. Pin Out Drawing

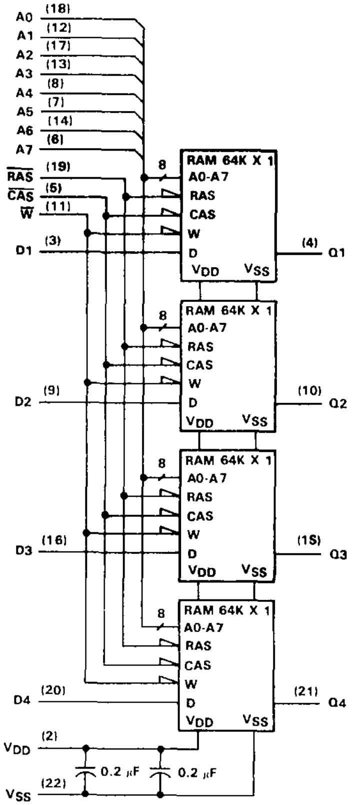


Figure A-2. Functional Block Diagram

JOINING TEXT AND GRAPHICS ENHANCES VIDEO PERFORMANCE

A dual-port RAM with a built-in shift register eliminates bottlenecks and speeds data transfers.

by David W. Gulley

Bit-mapped video graphics systems exemplify the need for higher density and higher performance semiconductor memories. Yet, all too often, these same memory devices are the bane of the system. The newest dynamic RAM devices, however, are allowing changes to the video graphics system organization. Thus, they are eliminating redundant support logic circuitry and providing a flexible system environment.

DRAMs, long associated with the frame buffer within the graphics section of a video system, provide the highest density and lowest cost storage for memory-intensive displays. High resolution graphics systems, such as those used in engineering workstations and computer aided design/computer aided manufacturing (CAD/CAM) terminals, require multiple memory planes to achieve the color capability necessary for a good user interface. In such a system, many parameters influence the available features while keeping the size and cost reasonable.

Often, the video display system designer is forced into "make-do" solutions when deciding on value-added features, especially where display memory is involved. Some features are common to many designs, and directly relate to the acceptance of a design in the market. Features considered high priority are the efficient integration of text and graphics, the time to redraw the screen image, the time to move objects onscreen, the amount of memory to map the display, and the support logic to use the memory effectively.

A typical video system contains separate text and graphics controllers (Fig 1). Thus, the system processor does not have to manipulate both the text display list and the graphics bit-mapped image. This system has evolved from the earliest text-only terminals, where there were no graphics requirements. In early systems, display memory consisted of perhaps 2 Kbytes for the display list RAM and 2 Kbytes for the character ROM. The need to place graphics images onscreen was first addressed using character graphics. By deepening the character ROM or adding a RAM to the character-generation circuit, user-defined characters could be produced.

To achieve more flexibility in image control, a bit-mapped memory is added into which the system can directly store images to be displayed. The mixed text

David W. Gulley is manager of MOS memory systems engineering at Texas Instruments, PO Box 1443, MS690, Houston, TX 77001. He holds a BS in electrical engineering from the Georgia Institute of Technology.

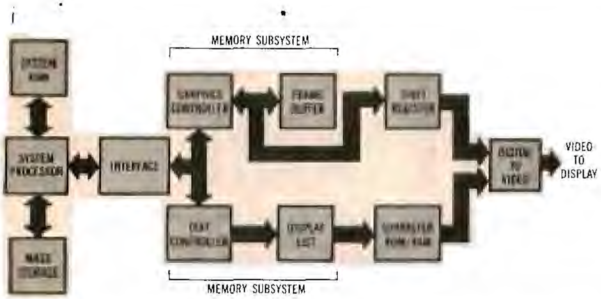


Fig 1 Mixed text and graphics systems, developed from simple text-only systems, have a lot of redundant circuitry.

and graphics solution is really a patch to add graphics capability to table-driven systems. However, future system design will treat text and graphics uniformly. New memory architectures are needed to make the transition to this type of system environment. The TMS4161 multiport video RAM is one device able to ensure this by providing a design path to the development of unified bit-mapped text and graphics systems (Fig 2).

Tracking the growth of video displays

Currently, uniformity is not in general use. Video display evolution has moved in another direction. As higher resolution and multiple gray-level or color planes were added, the screen refresh required higher data rates from the memory, giving less time to the system processor for data management in the frame buffer. As the resolution (pixels/in.^{2}) of the display increases by a factor of 2, the size of the display memory increases by 4, and the display interval for each pixel is reduced by a factor of 4.

The availability of dense, low cost DRAMs allowed expansion to higher resolutions (from a memory chip cost standpoint), but the DRAM architecture (1 bit wide) increased the data bus traffic needed to refresh the screen image. Graphics system controllers were added to the system to isolate the large bandwidth display bus from the system bus. If this isolation had not occurred, system processor throughput would have been seriously degraded. The data bus would be clogged with data passing from the frame buffer to the display.

The memory required for the frame buffer RAM is typically 10 (for black and white) to 40 (for 4 bits per pixel) times larger than the display list RAM in the mixed text system. Display data transferred to the screen loads the data bus so that there is considerably less time available to update the frame buffer memory than the display list RAM. Yet, since the nature of the data is single pixels, it requires more manipulating than display characters. More memory must then be accessed more often, and in less time. Hardware additions often implement many basic display functions, since there are not enough available memory accesses for sǒftware to optimally update the RAM.

The mixed video system consists of three memory subsystems, each containing a memory controller, memory logic, and glue logic. Glue logic also connects the controller to the system processor, and provides the required memory array drive. Each subsystem contains similar logic functions. Yet, the functions cannot be shared and still survive the data transfer bottleneck to the screen. Therefore, this is where DRAM features (actually, lack of features) have most influenced video system design. The many design approaches involving dedicated hardware control compensate for the limited accesses available to the memory. These approaches have partially relieved bus contention problems. But, the cost has

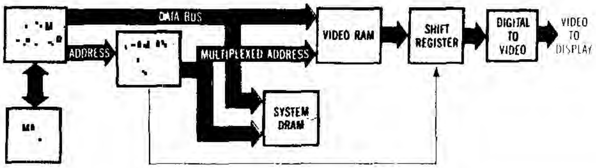


Fig 2 Unifying text and graphics eliminates redundancies. This results in faster and lower cost systems.

been a loss of system flexibility and compatibility for effective system upgrade. Dedicated controllers tend to lock the system into a set of fixed commands, character fonts, and data structures.

A high resolution (1024 x 1024 or 1280 x 1024) graphics display, as used in CAD systems, requires data from the refresh buffer at between 75 and 125 MHz from each plane, dependent upon the actual display device (monitor) specification. This is independent of the graphics controller's need to access the refresh buffer in order to update the image stored in memory. In the following analysis of system performance, a 1024 x 1024 noninterlaced display is used as a guide. Table 1 values describe the timings used in the analysis. Total frame time in Fig 3 consists of the active display interval, horizontal blanking interval (horizontal retrace), and the vertical blanking interval (vertical retrace).

TABLE 1
Display Parameters

Pixel clock frequency	88.00 MHz
Pixels per scan line	1380
Lines per frame	1063
Displayed pixels per scan line	1024
Displayed lines	1024
Horizontal blanking interval	4.05 μs
Vertical blanking interval	611 60 μs
Pixel time	11.36 ns

A typical video system design

The 88-MHz pixel data rate is in direct conflict with the need to update the memory quickly. The display refresh and the memory update must share the same data bus in the mixed text system. Updating the high resolution screen in a reasonable time frame requires some cycles to be available during the active display interval. A 1024 x 1024 display could be built using sixteen 64-Kbit DRAMs. But, even with the fastest parts, it is extremely difficult to get the video data rate required, and to be able to do useful screen image manipulations without reverting to a second (double) frame buffer.

The TMS4416 16-K x 4 RAM provides the large video bandwidth required in medium to high resolution video systems. Many systems that incorporate 16-K x 4 RAMs use the previous generation of 16-Kbit memories, and are using the x4 as a replacement for four 16-K parts. A wide-word architecture provides more data lines per depth of memory using standard DRAM access timing. Addressing four times as many bits per device simplifies the hardware needed to create the display frame buffer.

Wide-word devices used within the frame buffer provide the width needed to achieve the necessary bandwidth for display (Fig 4). This brute force design yields 64 data bits and requires a 64-bit shift register—all bits are loaded in parallel. The pixel clock is running at 88 MHz. S0 and S1 control the loading and shifting of the register. This approach contains the advantage of data access interleaving, first an interval for the processor access, and then an interval for the display access to the memory. It is more easily designed and manufactured than a similar approach using 16-K x 1 devices, and is much more reliable due to component and power reductions. The disadvantage is that there must be a way to buffer the data bus in order to convert from the 64-bit wide video section to the 16-bit wide system processor. In this design, a 64 to 16 multiplexer serves this function.

In the TMS4416 implementation of this circuit, there is one access available to the graphics controller for each display cycle. There are no highly critical access timings for the 16-K x 4, as the 64-bit shift register is loaded once each 727 ns (64 times 11.36 ns), and processor timing is assumed to be tightly coupled to the video shift rates. The storage cell refresh required by the DRAM is satisfied by reading across the memory chip rows for display accesses, and therefore does not require any additional logic or control. Even with all the data lines needed to connect the 64-bit shift register, this design runs at the top of its capability. If more flexible and higher performance systems are needed, the x4 RAM is not appropriate.

Many earlier high end video systems used the double buffer technique to avoid contention problems

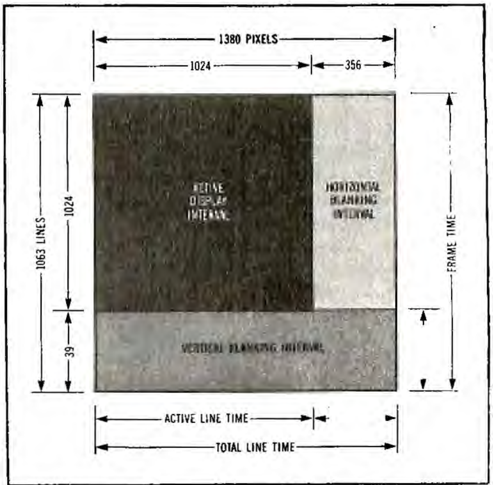


Fig 3 In a high resolution video display, the total frame time is the sum of active display time, and horizontal and vertical blanking intervals.

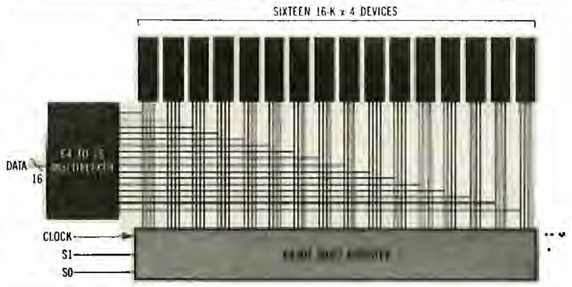


Fig 4 A brute force frame buffer design using wide-word memory devices provides interleaving of data accesses, and easier design and manufacturing compared with circuits using less dense memories.

between the graphics controller and the display refresh. In this scheme, two display frame buffers are used—one provides information for the display, the other is available to the graphics controller for updates. When the new drawing is complete, the system switches the function of the two buffers. Though this allows more interaction with the memory, it is at the expense of doubling the memory requirement. Also, when the buffers are switched, the graphics controller does not have a copy of the most recently available data image. In many systems, a form of DMA copies the data from one buffer to the other, effectively cutting the time available to the controller in half. Again, the system suffers from the lack of capability within DRAMs.

New systems are designed to be as functional to the end user as possible. The system must be flexible, tailored to individual needs, compatible with systems currently in use, and cost effective. Most new systems support multiple windows in order to display several simultaneous functions, and allow data manipulation within one window without affecting the contents of another window. But, there is a need to mix text and graphics information within a window.

Hardware control requires a large investment in design and components within the video system. New system architectures are needed to remove the display data-transfer bottleneck, eliminate redundant logic functions, and improve the system flexibility to conform to individual needs. Just as the industrial controller has progressed from a collection of SSI devices, to MSI, and now to single-chip processors, the video system control functions are moving from multiple subsystems to dedicated, optimized components.

Meeting the demands

The TMS4161 multiport video RAM remedies these problems by combining a standard 64-K x 1 DRAM with a 256-bit shift register, and the necessary controls to transfer data between the memory array and the shift register in a single package. By allowing simultaneous, asynchronous access to the two ports, the video RAM allows the system processor and the display refresh to work independently. Thus, the need for double buffering is removed, giving maximum time for the system processor to access the memory. The memory array access of the video RAM conforms to the signal and timing requirements of a standard DRAM. The onchip shift register supports high resolution data rates, and reduces video data shift logic and timing generation circuitry complexity. The shift register is configured as 4 linked 64-bit shift registers, able to provide shift lengths of 64, 128, 192 or 256 bits. These features help meet primary design criteria, and yield enhanced features for the video system.

The video RAM allows a more flexible approach to a video system design that eliminates mixed graphics approach patches. With the latter design, a bottleneck restricts data flow due to the single random access port on standard DRAM devices. Merging memory subsystems in a unified system substantially reduces design effort and cost. Redundant logic is eliminated by using the same functions for the video RAM as for system memory control. The logic required for the DRAM and video control section is currently implemented using several programmable logic arrays and MSI circuits (Fig 5). These could be placed in a gate array or other custom device.

The divider circuit is the only high speed device required, other than the external shift register, and provides the other logic with the appropriate timing signals. Not shown is the control to the external shift register, since it changes with implementation. A microprocessor or other controller can access the memory by issuing a MEMREQ/ with the appropriate read or write strobe. All other functions and timings are performed by the state sequencer.

A frame buffer using the video RAM could use a scan-line mapping architecture. This approach could also be used in the frame buffer of an existing design, although the full advantages of the dual-port would not be realized.

Scan-line mapping refers to positioning the memory devices to correspond to relative bit placements within a display scan line. Logic reduction in the frame buffer is evident, since there is only a 16-bit shift register, and no data bus buffer/separator requirement, as in the x4 example. In this particular example, each transfer from the memory array to the shift register moves a total of 4096 bits, which provide the data for four 1024-pixel scan lines. Onchip shift register data is loaded into the 16-bit shift register to accelerate the data to the required 88 MHz. The data in the memory's shift register is clocked at 5.5 MHz, well below the device's maximum clock frequency of 25 MHz. The timing for the video RAM is derived from the pixel clock to keep the system timings synchronous.

For this design, the row address strobe (RAS) cycle consists of 10 pixel clocks for the 114-ns precharge period, and 15 clocks for the 170-ns RAS low time, for a total period of 284 ns. All cycles (refresh, read, write, and transfer between arrays and shift register) use the same timings, with differences in the sequencing of the other control inputs to the video RAM (CAS/, W/, and TR/QE/). Each device holds every sixteenth pixel along the scan line of 1024 pixels. Scan-line data comes from 64 adjacent columns in each of the 16 devices. A 16-bit processor can directly access the memory array for image manipulation if it recognizes the appropriate addressing arrangement. Thus, the system processor can issue the address of the row and column for the desired pixel. The decoding of the active chip (when accessing via the DRAM port) may be done in hardware or as an internal operation of the processor.

The 256-bit register on the video RAM can be used by the video control logic to manipulate data as well as shift the data to the display. One way to employ this register is to clear (erase) the display quickly. The processor can write to the 256 locations corresponding to one row in the memory. This row can be transferred to the shift register. The shift register to memory transfer of the memory clears the remaining rows of the memory in 255 cycles. [Alternately, the serial input (SIN), could be grounded and SCLK clocked 256 times to load the shift register with all 0s.] Thus, the frame can be erased in a fraction of the vertical retrace interval of 612 μs, for improved performance in those applications requiring rapid screen clear.

Unlimited access

Since the video RAM shift register can be loaded from memory as little as once each four scan-line times for CRT refresh, the system processor has virtually unlimited access to the display memory. During a single 16.67-ms frame time, there would need to be 256 display access cycles (one of the video RAM's shift registers loads from memory for each four scan lines), and 1087 memory cell refresh cycles (a minimum of 256 refresh each 4 ms), which remove a small portion of the available time for updating the screen. The time for this overhead can therefore be calculated as:

MC x (#DIS + #REF)

284 x (256 + 1087) = 381 μs

Where MC is memory cycle time,

#DIS is the number of display cycles, and

#REF is the number of refresh cycles.

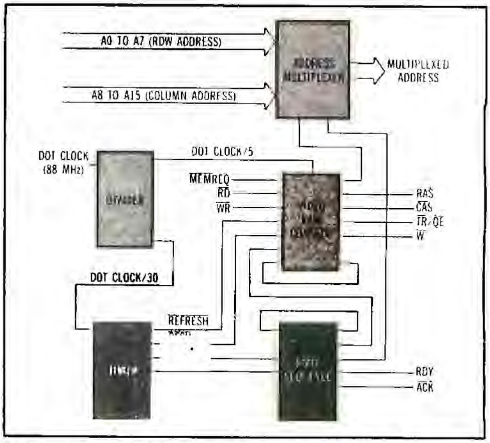


Fig 5 RAM and video control logic for a unified design can be implemented with programmable logic arrays and MSI circuits. A gate array or custom device could integrate the entire function in a high volume application.

So, in a single frame, all but 381 μs (about 2.3 percent) of the interval to be used by the system processor for display update are available. The remaining 97.7 percent of the time to scan a complete frame is available for access by the system processor. This allows memory accesses to follow logical, predictable patterns, and consistent timing sequences. These uniform cycles reduce the system hardware burden to fit memory update accesses into a narrow window or burst.

Modern screen imaging techniques indicate that hardware should not be used for read scrolling, to maintain maximum system flexibility. Designs usually call for moving data within defined regions of the frame buffer. However, for systems with hardware scrolling, the 256-bit register on the video RAM can be used by the controller to manipulate scan-line data in the displayed image. Data from one memory row can be transferred to the shift register, and then transferred back to another row (without shifting the data), which moves the pixel data from one displayed row to another. Several such transfers can be made, giving the effect of scrolling a full screen image vertically. This will scroll the entire width of the screen, so it may not be appropriate in a system with windows, where the scroll must be done in software.

TABLE 2
Maximum Accesses to Arbitrarily Located Region

Region Size	Scan Line	Symmetric	Improvement
32 x 32	96	81	15 percent
18 x 18	54	25	54 percent
16 x 16	32	25	22 percent
10 x 10	20	9	55 percent
8 x 8	16	9	44 percent
4 x 4	8	4	100 percent

In scan-line architecture, a move of one row to an adjacent row within the video RAM results in moving the displayed line four scan lines vertically. To scroll an entire screen of lines would take:

(MTS + TC) x #ROW

(284 + 284) x 256 = 145 μs

Where MTS is the time for a memory
  to shift register transfer,
TC is the shift register to
  memory cycle time, and
#ROW is the number of rows to be moved.

This scroll operation could wrap the image around the screen, or the processor could update the display memory with a new portion of the image. The ability to move rapidly the screen image vertically may have application for some realtime systems or forms of animation, since it gives the system processor more time to update the displayed image.

The scan-line technique is preferred because it is simple and logical, and offers direct processor to memory mapping. Although scan-line mapping is generally best, other memory chip to pixel mapping schemes can be advantageous. In such a system, the drawing hardware may be able to update multiple pixels at each memory access. Unfortunately, it is exceptional for multiple pixels to occur in horizontal lines, such that writes could occur parallel to (along) the scan line. Data manipulations of the display involve the equally probable writing of multiple pixels vertically, diagonally, and horizontally to create an image. Most data manipulations involve pixels within an arbitrary region occupying multiple scan lines. Using the scan-line mapping technique, these arbitrary regions will most likely not align with the work boundaries accessed by the graphics controller, thus requiring multiple accesses.

Opting for the symmetrical architecture

One method to reduce the number of accesses necessary to transfer an arbitrary block, and to minimize the access of unnecessary pixels, is to use a symmetrical architecture for the frame buffer memory array. The symmetrical architecture uses one 4-bit shift register per plane rather than the 16-bit shift register of the scan-line approach. It cascades video RAMs by connecting the serial output of one device to the serial input of another to move 1024-column data bits to the 4-bit shift register. As in the scan-line method, an array-to-shift register transfer occurs once each four scan lines, but the data is now shifted out of the video RAMs at 22 MHz. The mapping to the screen shows that when the system processor operates on the frame buffer, it accesses a 4 x 4 block of pixels. The manipulation of an arbitrary memory image will generally require fewer accesses, since the number of pixels operated on by the system processor will be maximized (included unnecessary pixels will be minimized).

Table 2 compares the maximum number of accesses required to read or write variously sized, arbitrarily located regions using the implementations described. For the smaller regions, symmetrical mapping yields the greatest improvement in required accesses. The pixels of no interest occur at the boundary edge of the region. In addition, accesses internal to a large region do not contain any unnecessary pixels.

The symmetrical mapping architecture causes the scan-line data to correspond to the 256 columns within the same row of four memory devices. Each device corresponds to every fourth pixel in the scan

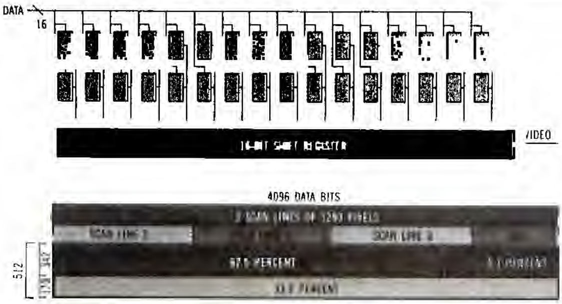


Fig 6 The video RAM organization offers the consolidation of memory, made possible by unified text and graphics design.

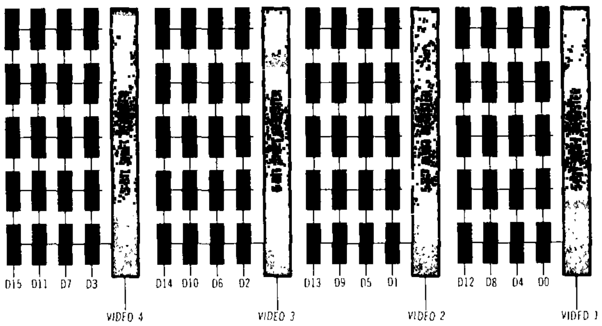


Fig 7 This frame buffer design provides four planes of display memory accessible to a 16-bit processor using 80 video RAMs.

line. A 16-bit processor can directly access the memory array for image manipulation, using the appropriate addressing arrangement. The system processor can directly issue the address of the row and column for the desired pixel. The decoding of the active chip may be done in hardware or as an internal operation of the processor.

To implement a 1280 x 1024 display (which is becoming somewhat standard), twenty 64-Kbit memory devices are required. There are apparent problems, however, with the use of a 16-bit processor with 20 memory devices. A bit-slice processor with 20 data bits could be used, but may not be practical for many systems. If a 16-bit processor is used, either the processor will access some or all of the memory as partial words (eg, 5 banks of 4 bits, or 1 bank of 16 bits and 1 bank of 4 bits), or extra memory is designated for use in video access. The use of partial words is possible. However, the added calculations to determine bit positions and increased number of accesses needed to update the display will cause some system performance degradation. This can be avoided by adding memory to fill out the data bus to a multiple of the processor width. This memory will not be wasted, since graphics systems typically require large regions of scratchpad memory to be used by the processors for placing text fonts, display lists, and for use in the calculations of drawing the displayed images.

Since more memory is required, the use of 32 video RAMs can simplify the task of matching the memory width to the processor width. If the memory is organized as shown in Fig 6, the transfer from array to shift register would place 4096 bits into the on-chip shift register. The data for 3 scan lines can be taken from these 4096 bits, leaving 256 unused bits. The display will use a total of 175,104 bytes (163,840 displayed and 11,264 left at the end of the rows) of the 262,144 bytes in the RAM. This noncontiguous memory amounts to about 4.3 percent of the total memory. The remaining 87,040 bytes consolidated within the second bank of video RAMs are available for use as system memory or scratchpad memory.

Lookup table eases calculations

To make the task of calculating the starting address of each scan line easier, a 1024-word table (2048 bytes), is set aside as a lookup table. Using a table to point to the start of the memory to be used for display allows rapid changes in portions of the screen image while not affecting other areas. When the same memory can be used for either display or system memory, the cost effectiveness and flexibility of the system is improved. The unified text and graphics design approach allows memory consolidation, especially in those systems where nonpower-of-two displays are used.

Fig 7 shows a possible implementation of a 1280 x 1024 frame buffer to provide four planes of display memory accessible to a 16-bit processor using 80 RAMs. The processor will access all four planes of data for each of four pixels, from what it considers as five banks of 16 memories. The data for display within each of the planes appears as four banks of five devices so that the array to shift register transfer will load four scan lines of data. The difference in this organization is the relative position of the four pixels accessed by the processor. The mapping separates the four pixels accessed by 1280 pixels into a vertical line. Depending on the address scrambling, the processor could map the memory sequentially in vertical rows rather than horizontal lines. The 5-bit shift registers allow the video dot rate to go up to 125 MHz before the data capacity of the RAMs is exceeded.

A 256K DRAM Organized for Applications Solutions

Gregory B. Clark
Systems Engineer
Texas Instruments Incorporated
Houston, Texas

Joseph M. O'Hare
256K Product Engineering Mgr.
Texas Instruments Incorporated
Houston, Texas

ABSTRACT

The need for high density dynamic random access memory (DRAM) will continue to increase with the increased requirements of applications software. A 256K DRAM which combines increased memory density and high performance to satisfy more sophisticated applications is described. The 256K DRAM has been organized to provide for both a 256K × 1 and a 64K × 4 architecture. System requirements will dictate which architecture is more effective in satisfying an application. In addition, the refresh scheme and device pinouts allow the 256K DRAM family to be the first truly upwards compatible generation of 5 volt DRAMs. A technique where memory system upgrade can be accomplished in modular increments is demonstrated.

INTRODUCTION

As applications software becomes more sophisticated, the need for high density dynamic random access memory (DRAM) will continue to increase. On the average, system memory size will increase three to four fold requiring larger boards, additional boards, sophisticated packaging techniques, or denser memory. Figure 1 is a projection of the average increase in dynamic memory per system that will be necessary to accommodate more sophisticated applications programs over the next five years. Coincident with the increased memory requirements, however, is the introduction of the next generation DRAM, the 256K. These latest generation devices will provide four times the amount of memory in the same board area as with 64K DRAM's. In addition, more device features will provide the flexibility to maximize utilization of 256K DRAMs in specific applications. This paper will describe a 256K DRAM, its technology and architecture, and how it simplifies the needs of expanding applications.

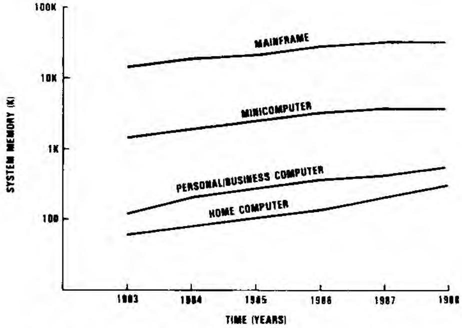


Figure 1. Projected DRAM Memory Increase Per System

FEATURES AND CHARACTERISTICS

The development of the 256K DRAM required new process technology [1] in addition to further scaling of the SMOS (Note 1) process.

With the announcement of the 256K DRAM's, it is apparent that the choice of architecture, silicide material, number of polysilicon/metal levels, and design techniques are numerous [2,3]. The TMS4256, 256K × 1 DRAM, is fabricated with a single metal, double level polysilicon (Note 2) process for performance and simplicity. The features of the technology are listed in Table I.

NOTES: 1. SMOS, scaled NMOS, is the proven technology of the TMS4164, 64K DRAM
2. Polysilicon and polysilicon/silicide

Table I. Technology Features

Bit Line	Aluminum
Word Line	Polycide
Tox (Periphery)	400 Angstrom
Tox (Cell)	200 Angstrom
Cell Capacitance	50 fF
C (bit line)/C (Cell)	8
Redundancy	Laser
Technology	Double Poly*
Design Rule	2 Micron
Metal Width (Min)	2 Micron
Chip Size	4.6mm × 8.7mm

*Polysilicon/polycide

Folded metal bit lines and polycide word lines provide an optimum signal for sensing and high speed performance. This signal is a result of low bit line capacitance and minimum word line delay. The double polysilicon approach required only the addition of a polycide process to an already proven technology.

The dimensions of the chip are 4.6 mm × 8.7 mm and can easily fit into a 300 mil plastic package. Over 72% of the die area is devoted to the memory array and decode circuitry as is illustrated by Figure 2, the chip photograph.

Four redundant columns and four redundant rows (Note 3) have been included to maximize yield in the early stages of production. Laser blowing of polycide fuses accomplishes the task of removing the defective row(s) or column(s) and replacing them with the proper redundant row(s) or column(s). Repaired memory has been characterized and no performance or reliability degradation was observed.

Typical speed and power characteristics are shown in Table II. The typical power dissipation at 3.8 MHz operation is 250 mW while in standby mode it is only 12.5 mW. Typical access speeds from row address strobe ($\overline{\text{RAS}}$) of 105 ns have been measured.

Table II. Typical Characteristics

Organization	256K × 1. 64K × 4
T (RAC)	105 ns T (RCD) = 25 ns
T (CAC)	62 ns
I_{DD} (operating)	50 mA T (RC) = 260 ns
I_{DD} (standby)	2.5 mA
Refresh	256 cycle, 4 ms
Package	16 Pin, 18 Pin 300 mil
Nibble Sequence	(0,0)→(0,1)→(1,0)→(1,1)
ESD	> 2 kV*

*MIL-STD-883B. Method 3015

NOTE: 3. Physically, eight rows on the chip

The voltage range of operation is shown in Figure 3. The specified supply voltage, V_{DD} is 5 volts ± 10%; the actual performance is shown to be from 3.5 volts to greater than 7.0 volts.

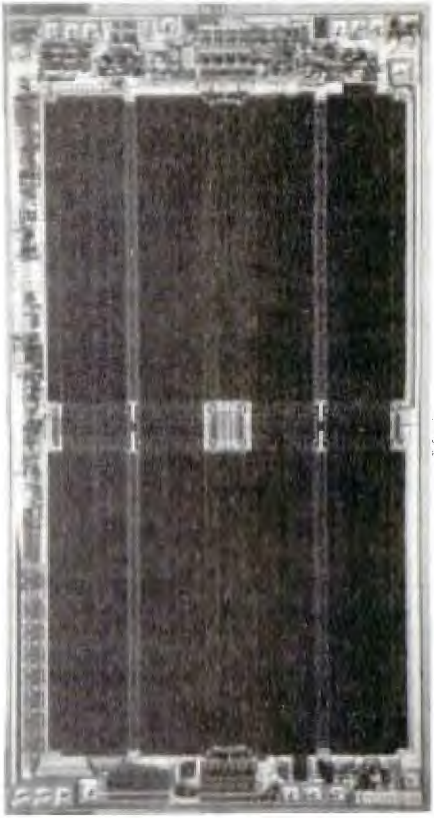


Figure 2. The 256K Chip Photograph

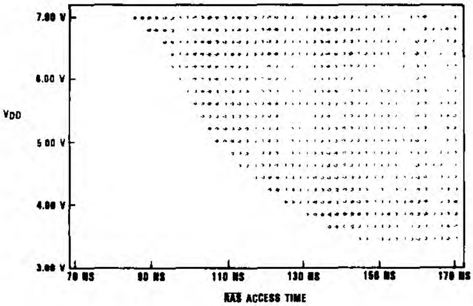


Figure 3. t_{RAC} Versus V_{DD}

ARRAY ARCHITECTURE

The 256K DRAM has been designed to provide optimum flexibility in terms of I/O's for system data bandwidth. The lead established by the 64K DRAM, with both 64K × 1 and 16K × 4 organizations, has been followed by the 256K DRAM. The memory architecture is designed for 256K × 1 and 64K × 4 I/O structures. One step further is the addition of a nibble mode option. Since its introduction [4], nibble mode has become an additional feature on many first generation 256K devices. The 256K offers the flexibility of utilizing either a nibble mode or page mode DRAM.

Furthermore, the memory array of the 256K DRAM is compatible with the following options:

1. 256K × 1 (TMS4256, Page Mode)
2. 256K × 1 (TMS4257, Nibble Mode)
3. 64K × 4 (TMS4464, Page Mode)

The functional block diagram of the 256K × 1 and 64K × 4 are shown in Figures 4 and 5. By examining these figures, two significant differences in the block diagrams are noted. First, on the 256K × 1 (Figure 4) the most significant address A8 performs a one of four selection to provide a single output. This address is not required for the 64K × 4 device as all four data bits are required, one for each DQ pin (Figure 5). Second, an additional control signal, $\overline{G}$ or output enable, is available on the 64K × 4 to provide additional HI-Z/enable flexibility on the DQ pins.

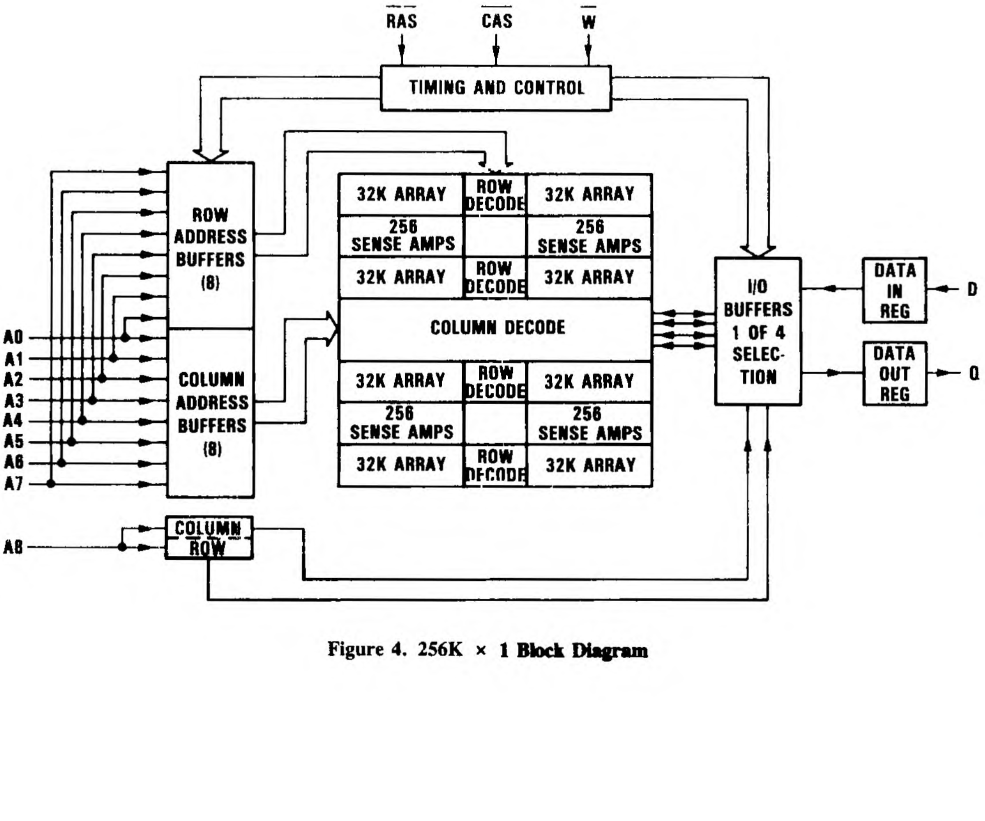


Figure 4. 256K × 1 Block Diagram

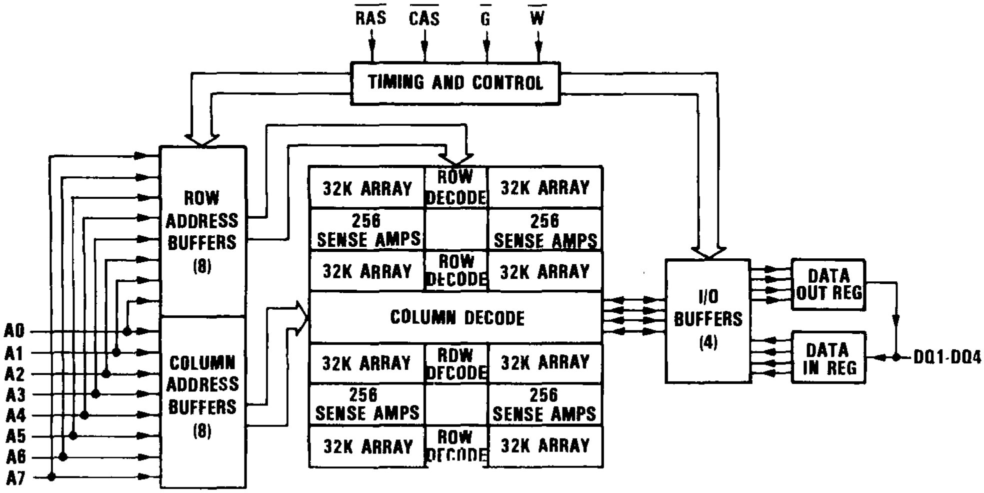


Figure 5. 64K × 4 Block Diagram

The unique array architecture can be appreciated by looking one level deeper into the functional diagram. Figure 6 illustrates how one memory array access generates four data bits internally. Essentially, the X-word lines for the top and bottom array are selected simultaneously (effectively one word line). One Y-decoder will select four sense amplifiers; as a result, four memory cells are accessed. The four bits can be: 1. decoded by RA8 and CA8 for the TMS4256; 2. shifted from the four intermediate output buffers using an A8 (AY8, AX8) sequence nibble operation for the TMS4257 (see Table II); or 3. loaded to four DQ buffers for the TMS4464.

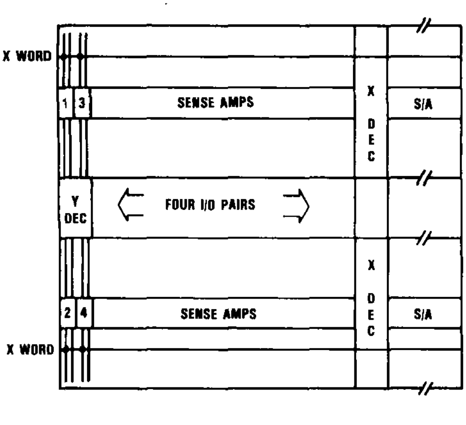


Figure 6. Functional Diagram

The array architecture also maintains the 256 cycle, four millisecond refresh that was standardized on the TMS4164 and TMS4416. This was accomplished by organizing the array as 256 rows and 1024 columns in four 64K blocks.

INCREASED REFRESH FLEXIBILITY

In addition to conventional refresh methods, the 256K DRAM family has been designed with expanded capabilities. Refreshing the device can be accomplished with any of the following techniques:

1. Normal Read/Write operation,
2. $\overline{\text{RAS}}$-only-refresh cycle,
3. $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh cycle (CBR), and
4. Hidden refresh cycle.

The 256K design includes an internal refresh address counter. This counter provides the row address to be refreshed in a CBR or Hidden refresh cycle. With this feature, an external refresh address does not have to be supplied by the user. The memory system design simplification is illustrated in Figures 7 and 8. Note that both an external refresh address counter and multiplexer have been eliminated with the utilization of $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh.

The timing diagrams for the various refresh cycles are shown in Figure 9. It is significant to observe from the timing diagrams that the address is in a "don't care" state during the $\overline{\text{RAS}}$ negative transition for the CBR or hidden refresh cycles.

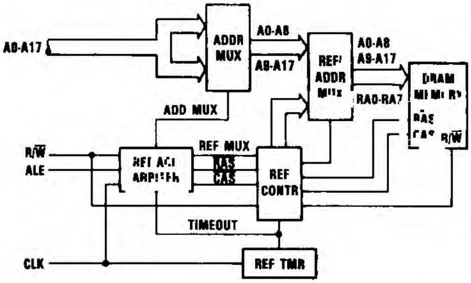


Figure 7. RAS-Only Refresh Implementation

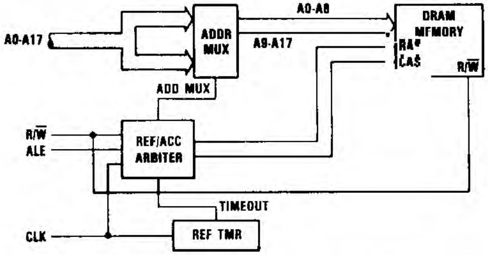


Figure 8. CAS-Before-RAS Refresh Implementation

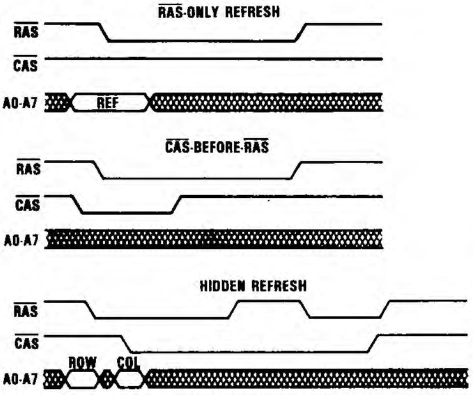


Figure 9. Refresh Cycle Alternatives

GENERATION TO GENERATION COMPATIBILITY

The pinout of the 256K × 1 and 64K × 1 devices (Figure 10) illustrates the pin-for-pin compatibility with the exception of pin 1 on the 256K × 1. This is designated as the ninth address pin since 18 addresses are necessary to decode one of 256K memory locations. From the board layout perspective, only an additional trace will be necessary to accommodate the ninth address pin. However, an additional multiplexer (2 to 1 MUX) is necessary since typically they are available in 4-bit increments only. If an external DRAM controller chip is to be utilized, a provision for the additional addressing bit may already be accommodated. The same refreshing scheme as used by the TMS4164, 64K × 1, generation devices will also be used by 256K × 1 devices; specifically, 256 cycles in a 4 ms period. The additional address pin is simply ignored by the refresh address generation circuitry when refresh occurs.

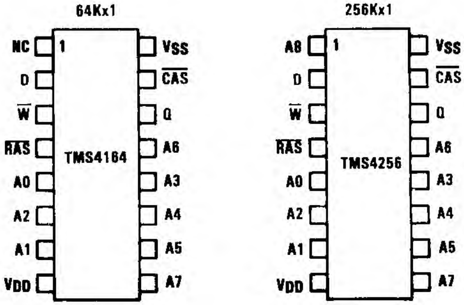


Figure 10. × 1 DRAM Pinouts

The 16K × 4 (TMS4416) and 64K × 4 (TMS4464) generation devices are the first entirely pin-for-pin compatible dynamic RAM generations (Figure 11). A system designed for TMS4416 devices will be able to immediately utilize a TMS4464 device. The main adaptation will be in the memory map circuitry where there is suddenly four times the available amount of memory. The TMS4416 devices employ an 8-row address, 6-column address decoding matrix to yield four bits from 65,536 possible memory locations. This decoding scheme provided the four bits all from the same row since the eight row address bits decode 1 of 256 rows, and the six column address bits decode 4 of 256 columns. Despite only 6 column bits being necessary, the trace layout included the eight address lines for row address decoding requirements. The 64K × 4 was designed to comprehend an

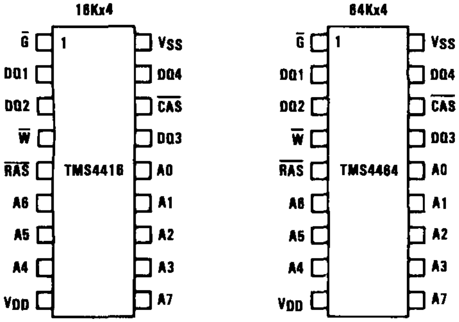


Figure 11. × 4 DRAM Pinouts

8 row address, 8 column address decoding matrix in order to maintain compatibility with the previous generation. Since the architecture is arranged as 256 rows by 1024 columns, all four data bits are selected on the same row. Eight traces have already been incorporated in the layout so no changes will be necessary for the TMS4464 board layout.

The generation-to-generation compatibility offers the ability to simply replace new generation devices for older generation devices with the appropriate board layout; thus, memory expansion can be taken advantage of with a minimum of board re-configuration. With the right support circuitry as illustrated by Figure 12, the memory size can be increased in a modular fashion to correspond with increased memory requirements from software. A TMS4416 to TMS4464 conversion system is a good example. Memory mapping for a TMS4164 to TMS4256 conversion system would be very similar except for the provision for a ninth address line. Consider an application that calls for a 16K word minimum memory requirement to be increased by minimal steps until a full 256K words are available by virtue of a fully populated 64K × 4 DRAM system (Figure 13.). This allows the user to increase memory to align it with his expanding software requirements without an immediate fourfold memory increase. In addition, devices will replace parts already existing on the board; so it is not necessary to purchase complete memory expansion boards. A 64K-word system will require 16 TMS4416 DRAM's arranged as four banks with four devices in each bank. Four TMS4416 DRAM's will provide the required word width.

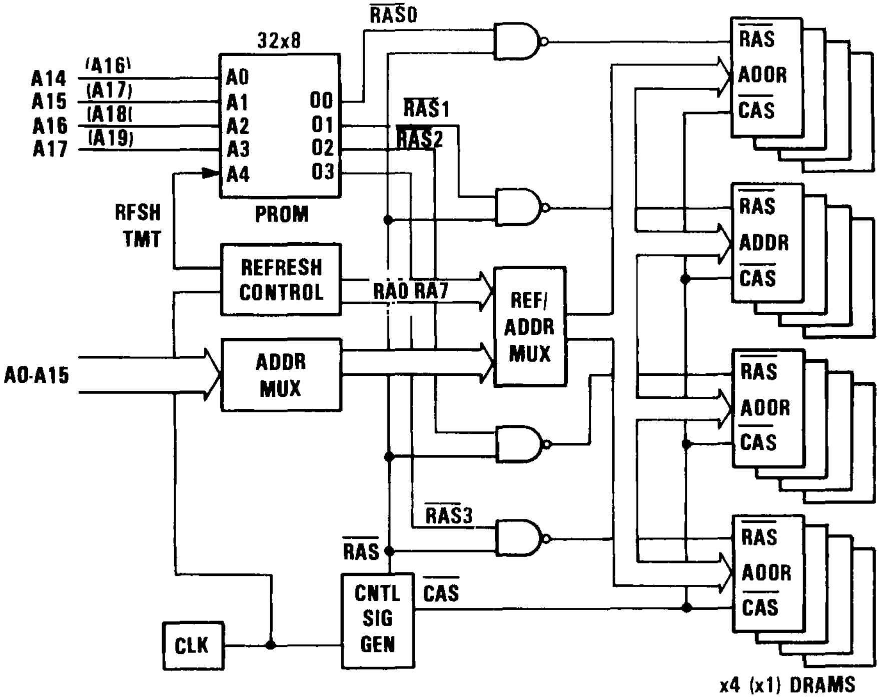


Figure 12. Memory Board Block Diagram

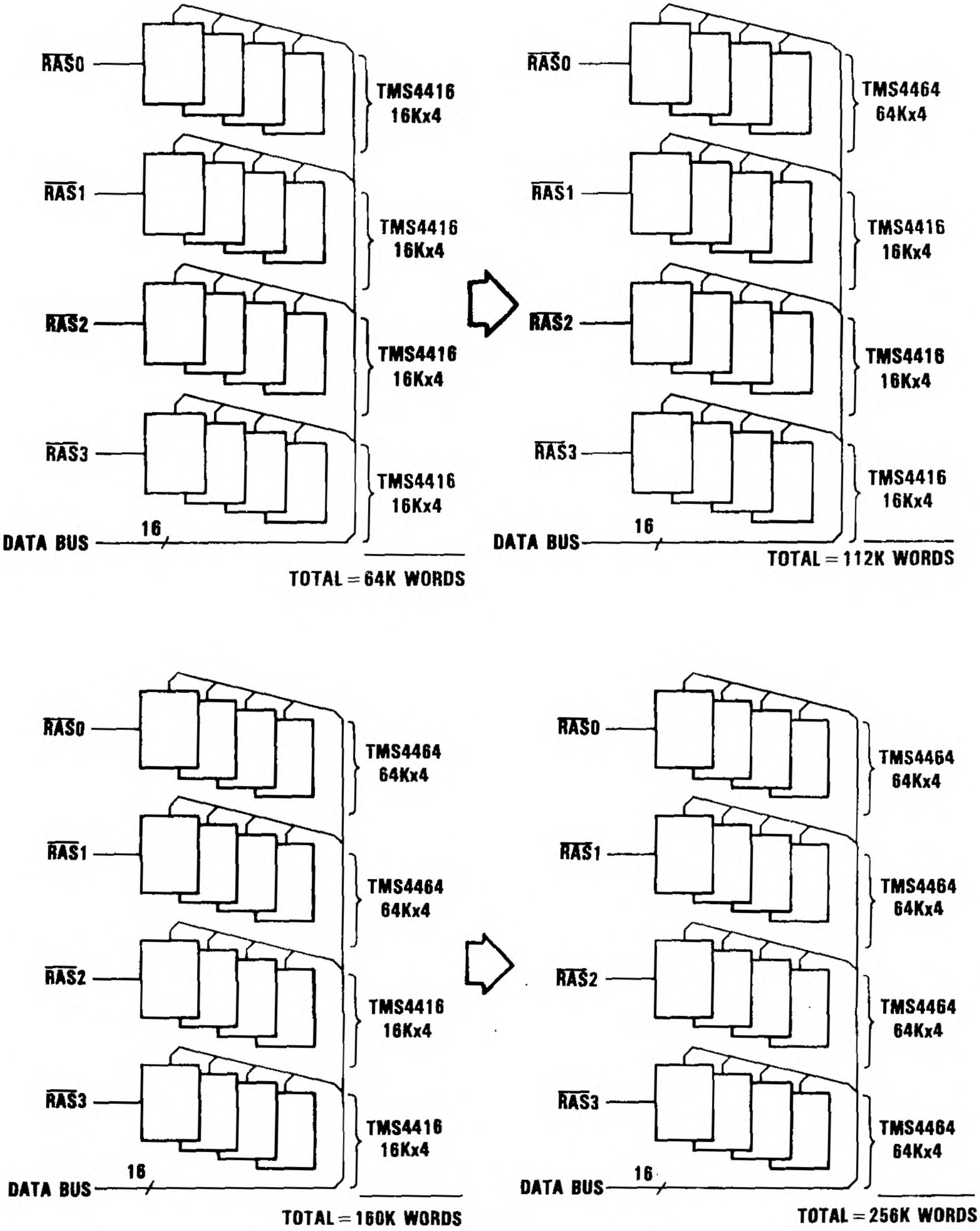


Figure 13. TMS4416 to TMS4464 Memory Expansion

The added consideration with this type of memory arrangement is mapping the memory from the onset to comprehend the memory increase. This can be accomplished by using addresses A14 to A17 to decode a 32 × 8 PROM to yield the four $\overline{RAS}$ bank select signals. Each × 8 location of the PROM (actually only four of the eight bits are used) corresponds to a 16K-word memory block. Table III is a truth table which indicates the decoding scheme for the four $\overline{RAS}$ bank selects. In the initial system of up to 64K words, memory will correspond to the first four memory locations (16K × 4 = 64K). As additional memory is added to the system, each additional 16K block of memory will correspond to subsequent PROM memory locations. Note that the addition of memory can happen in increments of one bank at a time (four devices). A fully populated TMS4464 DRAM system will absorb 16 out of the total 32 PROM memory locations. Thus the most significant PROM address bit, A4, is used as a refresh decode. All PROM memory locations (16, total) which correspond to A4 high activate all four $\overline{RAS}$ bank select signals. This allows refresh to occur on the current selected row for all system DRAM memory simultaneously. Remember that a TMS4464 DRAM has the same 256-cycle refresh scheme as a TMS4416 DRAM so a combination system will have no refresh constraints.

A TMS4256 conversion system (Figure 14) from a TMS4164 system will be very similar except that memory will be mapped in 64K-word blocks and would increase from a minimum of 64K-words to a maximum of 1M-words (a fully populated 256K DRAM system).

ORGANIZATION TRADEOFF

The previous compatibility examples illustrate significant differences in the implementation of either a × 1 or × 4 organized DRAM for a particular application. A complete evaluation of the system needs is necessary in order to decide on which organization DRAM to utilize. There are basically three categories to examine when evaluating your system:

1. Memory size requirements,
2. Memory speed requirements, and
3. Expandability requirements of the system.

Table III. PROM Truth Table

	PROM					OUTPUTS								SELECTED
	ADDRESSES									$\overline{RAS3}$	$\overline{RAS2}$	$\overline{RAS1}$	$\overline{RAS0}$	BANK
$\overline{CS}$	A4	A3	A2	A1	A0	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	OF MEMORY
0	0	0	0	0	0	0	0	0	0	0	0	0	1	Bank 0
0	0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0	2
0	0	0	0	1	1	0	0	0	0	1	0	0	0	3
0	0	0	1	0	0	0	0	0	0	0	0	0	1	Bank 0
0	0	0	1	0	1	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	1	0	0	0	0	0	0	1	0	Bank 1
0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
0	0	1	0	0	1	0	0	0	0	0	0	1	0	1
0	0	1	0	1	0	0	0	0	0	0	1	0	0	Bank 2
0	0	1	0	1	1	0	0	0	0	0	1	0	0	2
0	0	1	1	0	0	0	0	0	0	0	1	0	0	2
0	0	1	1	0	1	0	0	0	0	1	0	0	0	3
0	0	1	1	1	0	0	0	0	0	1	0	0	0	3
0	0	1	1	1	1	0	0	0	0	1	0	0	0	Bank 3
0	1	X	X	X	X	0	0	0	0	1	1	1	1	Refresh All Banks

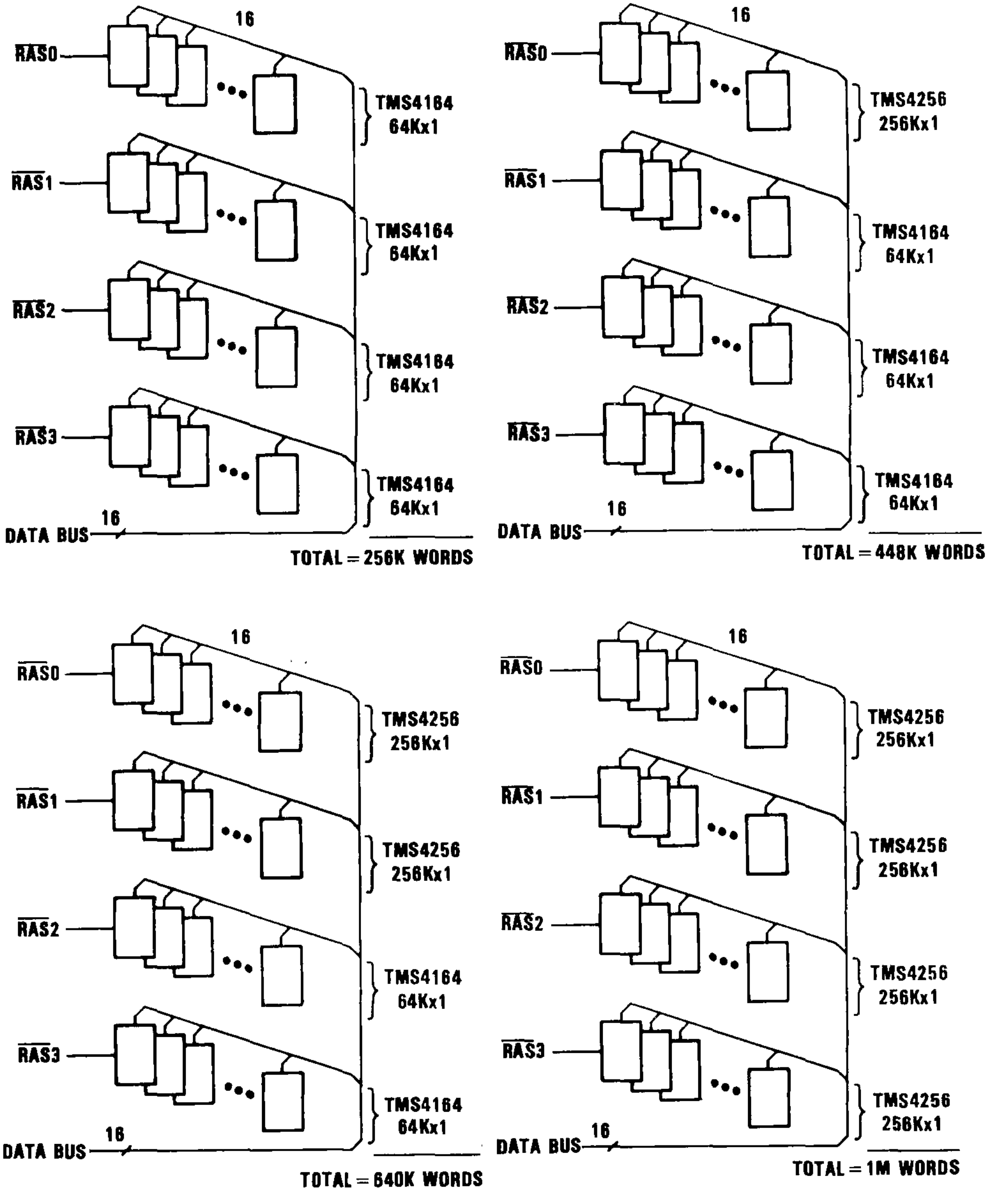


Figure 14. TMS4164 to TMS4256 Memory Expansion

Memory Size Requirements

The memory size needed to satisfy an application would be the first factor to evaluate in the consideration of either ×1 or a ×4 system implementation. A general rule to apply is to utilize the ×4 organization for memory requirements up to the ×1 memory size, or if N > 1, where N = ×1 memory size/memory needed. In the case of 256K DRAMs, for memory less than 256K (bytes, words, etc.) it would be advantageous to utilize 64K×4 devices. Figures 15 and 16 show a comparison of a system which uses ×1 DRAMs and ×4 DRAMs, respectively, to provide 256K bytes. The same number of memory devices are utilized in each case, but the ×1 device takes up somewhat less space because of the 16-pin package over the 18-pin package of the ×4 device.

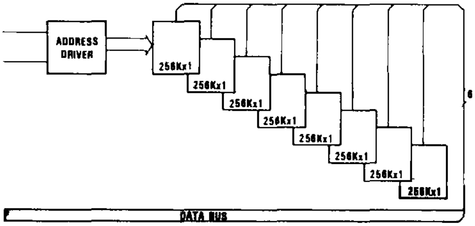


Figure 15. 256K Memory System Size (256K × 1 Implementation)

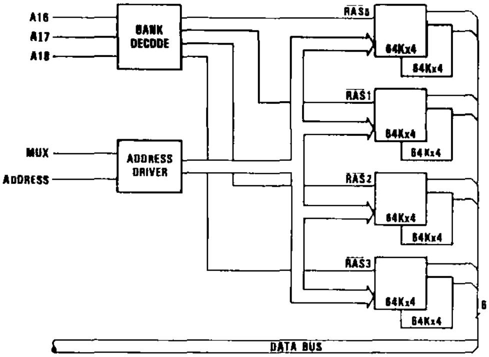


Figure 16. 256K Memory System Size (64K × 4 Implementation)

In addition, proper decoding circuitry will be necessary to decode one of the four memory banks of the ×4 implementation. The power savings of enabling only one bank at a time for a memory access will be offset by the power usage of the additional drive and decoding circuitry. The additional circuitry will also take up more board space, and require more signal routing to implement. A listing of relevant parameters is available in Table IV for both a ×1 and ×4 implementation of a 256K-byte system. Up to the 256K byte level, though, the bandwidth advantage of the ×4 devices allows better memory utilization and power savings by enabling one bank of DRAMs during any single memory access. Table V is a comparison of the same parameters for a TMS4464 and TMS4164 implementation of a 128K-byte system. The part count and board area savings of the TMS4464 implemented system is highlighted in Figures 17 and 18.

Table IV. 256K-Byte System

PARAMETERS	256K × 1 DRAM	64K × 4 DRAM
Component Count*	9	10
Board Area	3.34 sq."	4.0 sq."
Power	620 mA	375 mA

*Includes Support Circuitry

Table V. 128K-Byte System

PARAMETERS	64K × 1 DRAM	64K × 4 DRAM
Component Count*	18	6
Board Area	4.89 sq."	2.4 sq."
Power	590 mA	374 mA

*Includes Support Circuitry

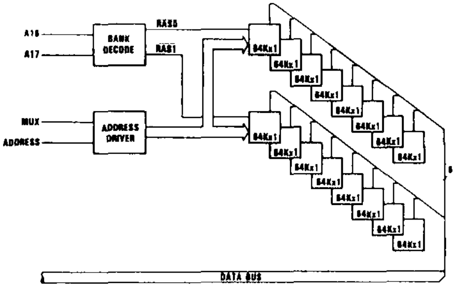


Figure 17. 128K Memory System Size (64K × 1 Implementation)

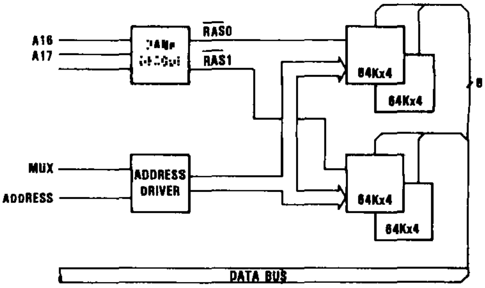


Figure 18. 128K Memory System Size (64K × 4 Implementation)

Memory Speed Requirement

The speed required to satisfy specifications for an application is another factor to consider when evaluating the use of ×1 or ×4 organized DRAMs. The minimum cycle time for a 150 ns dynamic RAM is 260 ns. With system decoding and buffer delays within the system, the realistic cycle time will be increased to 300 ns. Memory access becomes critical when the application demands a high performance microprocessor or bit slice controller. High performance microprocessor memory access periods are now reduced to under 200 ns in some cases, while a bit slice memory access period is under 100 ns. Direct interface of ×1 DRAMs would cause the fast processors to execute wait states while waiting for the memory. The example in Figure 19 shows a method in which the utilization of a ×4 device will decrease the average access time of each bit by a factor of up to four. The two least significant addresses decode one of the four latches which provide four times the data bus width worth of data. The processor will have to access the latches four times as often, as memory must be accessed to load the latches. The data from the first latch will be accessed in the normal DRAM access period (300 ns) as all four data latches will be filled. With a typical processor memory access period of 75 ns assumed, subsequent processor memory accesses will access memory from the other three latches allowing for memory access without wait states. This translates to an average cycle time of [300 ns+3*(75 ns)]/4 processor memory accesses or 131 ns. Since most processor instructions occur sequentially over short intervals and require multiple memory operations, the average memory access will be based upon enhanced access time. This compares with the previous average memory cycle time of 300 ns where the processor will be forced to wait for every memory access. Even further enhancement of apparent memory access time can be achieved in systems that allow memory access overlap or pipelined instruction execution. Such a system would have an apparent memory access time approaching 75 ns (or the cycle time of the processor). Obviously, memory speed and memory size are very interrelated when considering this trade-off.

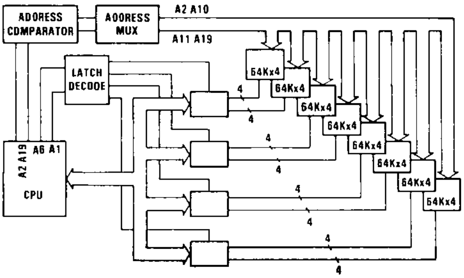


Figure 19. 64K × 4 Implemented System Performance Increase

Expandability

The expandability of a system can be sub-divided into two aspects: maximum memory size, and minimum memory increments. The compatibility examples are good illustrations of the granularity advantage of the ×4 devices over the ×1 devices where granularity is a measure of the smallest increment in which memory size can be increased. The increase of total memory size with the modular implementation of 64K ×4 DRAM's was only in 48K-byte increments; whereas the 256K ×1 system increased in units of 192K-bytes. By the same token, the maximum attainable memory size for the ×4 system will be 256K-words, whereas the ×1 system can be expanded to a maximum of 1M-words (if maintaining the same level of bank decode logic).

SYSTEM REFRESH CONSIDERATIONS

The use of dynamic RAMs in a system carries with it the responsibility of refreshing the DRAMs at regular intervals. Several refresh alternatives provide the designer the opportunity to adapt a particular refresh scheme to a particular application. The type of refresh to be implemented in a system depends directly on the type and speed of the processor being utilized, since these factors determine the length of time before memory access may be required. Typically in a memory sub-system design, refresh should be as transparent to the system operation as possible. This serves to reduce the interruption of processor access cycles to a minimum and thereby increase the performance of the system. For slow processors, hidden refresh provides the capability to complete a refresh cycle within a processor memory access cycle by latching the accessed data while the refresh is completed. As a result, separate hardware will provide for refresh cycle implementation, eliminating refresh responsibility from the processor. It is interesting to correlate refresh timing requirements of the DRAM memory with the memory access requirements of the processor. The refresh cycle time for a 150 ns DRAM is 260 ns. Hidden refresh implementation in a slow processor system allows enough time to complete both a memory access and a refresh cycle (Figure 20). There will be no affect on processor operation performance due to refresh requirements of the system. In a medium performance processor system, the period of time provided for memory access becomes critical when a refresh must occur. Wait cycles may have to be implemented during a processor memory access which occurs during a refresh cycle to accommodate the additional time that the refresh cycle adds to the memory access cycle. In a high performance system, the memory access cycle will cause the processor to be delayed to allow completion of the memory access. A refresh cycle will cause additional delay to be added to the memory access time. In systems with a processor that gives an indication that it will not be utilizing the memory, a refresh cycle may be inserted to eliminate the degradation of processor throughput because of refresh.

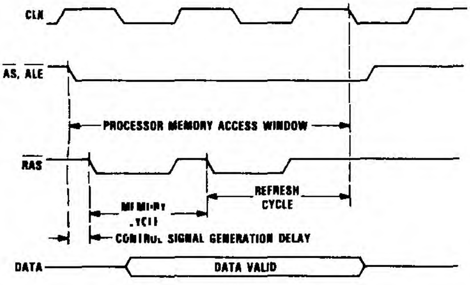

CLK
$\overline{AS}$, $\overline{ALE}$
PROCESSOR MEMORY ACCESS WINDOW
$\overline{RAS}$
REFRESH CYCLE
MEMORY CYCLE
CONTROL SIGNAL GENERATION DELAY
DATA
DATA VALID

Figure 20. Slow Processor Transparent Refresh

SUMMARY

Design and process technology advances combine to allow the manufacture of a 256K DRAM that is upwards compatible with previous generation 64K DRAMs. The memory array architecture is adaptable to both 256K × 1 and 64K × 4 organizations. Device performance is improved due to the use of polycide film, thin gate dielectric insulator, and folded metal bit lines.

Generation-to-generation compatibility provides additional memory for new applications or reduced component count for present applications. Furthermore, the TMS4416 and TMS4464 are pin-for-pin compatible and require no hardware modifications to upgrade.

Refresh requirements are identical between generations of × 1 and × 4 DRAMs. The 256K *DRAM is available* with an internal refresh address counter to accommodate hidden and $\overline{CAS}$ before $\overline{RAS}$ refresh schemes.

The performance and flexibility of the 256K DRAM family will meet the increasing system memory requirements without increasing system complexity. As a result, more sophisticated software can be accommodated.

ACKNOWLEDGEMENTS

The authors wish to thank David Gulley for his ideas and directions in the development of this paper. In addition, they would like to thank Frank Miu, and the 256K Design Team for their contributions.

REFERENCES

[1] M. **Smayling**, M. Maekawa. "256K dynamic RAM is more than just an upgrade", Electronics, Vol. 56, no. 17, pp. 135-137, Aug. 25, 1983.

[2] T. Fujii et al., "A 90 ns 256K × 1 bit DRAM with Double-level Al Technology". IEEE J. Solid State Circuits, Vol SC-18, no. 5, pp. 437-440, Oct. 1983.

[3] T. Nakano et al., "A sub-100 ns 256K DRAM with open bit line scheme", IEEE J. Solid-State Circuits, Vol, SC-18, no. 5, pp. 452-456, Oct. 1983.

[4] S.S. Eaton, et al., "A 100 ns 64K dynamic RAM using redundancy techniques". Dig. Tech. Papers, ISSCC, 1981, pp. 84-85.

This paper was presented at ELECTRO '84 and was reprinted with their permission.