Dynamic RAM Board Design Made Easy

National Semiconductor
Memory Application
Stephen Calebotta*
November 1977

INTRODUCTION

Many new memory system designs are being done with dynamic RAMs. This is especially true as the new 16k RAM chips become more readily available. This application note is aimed at those engineers who are doing their first dynamic RAM designs. Its intent is to give some direction in how to design a RAM board so that it will give the greatest production yield with the least amount of difficulty.

We shall not talk about specific RAMs or interface to a specific processor. Most engineers can design control and interface logic for RAMs and most board designers can lay out a PC board for that design. However, the quality of that design will become apparent only after the RAM board is built and phased into production. Does it come up easily? Does it go through testing with little or no RAM chip fallout? Is it reliable in the field?

The key to success in a dynamic RAM system, or any other system for that matter, is margin. A system designed to maximize power supply and timing margins will be reliable and easy to manufacture. One that doesn't will be a manufacturing and field service nightmare.

In this application note we shall discuss RAM chip characteristics, power supply and control signal distribution on PC boards, and control logic implementation suggestions. As successful examples, in the appendix we shall provide the schematics and foils for some memory boards that are in production.

RAM CHIP CHARACTERISTICS

For reference we shall compare dynamic and static RAMs at the chip level. Then we shall describe the unique characteristics of dynamic RAMs which must be considered in a memory system design.

Dynamic RAMs versus Static RAMs

The basic difference between dynamic and static RAMs is the way they store data. The static RAM uses a flip-flop to store a bit, while the dynamic RAM uses a capacitor to store a bit. (See figure 1.)

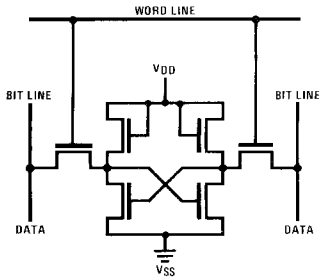
It is their respective cell designs that give each RAM its advantages over the other. Let's compare the RAMs for ease of use, power dissipation, die size, and price.

Ease of Use: The static RAM is easier to use because no refresh logic is required. In addition, static RAM control signals tend to be easier to generate because cycling is usually unnecessary.

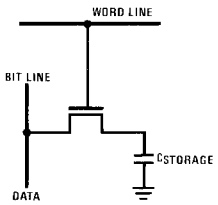
Power Dissipation: The dynamic RAM draws less power. The static RAM draws power continuously to sustain its flip-flops, while the dynamic RAM draws minimal power (1 to 2 mA) between cycles. With continuous cycling, the dynamic RAM draws about as much power as the static. However, in a large memory system, dynamic RAMs save total system power since only one bank of RAMs is ever accessed during a memory cycle. All other banks draw minimal current except during refresh cycles. The duty cycle for refresh is approximately 1½ to 3%.

Die Size: Dynamic RAMs tend to be smaller. Due to the difference in cell designs, the die size of the dynamic RAM is often at least 20% smaller than that of a comparable static RAM from the same manufacturer.

Price: Because of smaller die sizes and much larger production runs, dynamic RAMs should always remain considerably cheaper than comparable static RAMs. In addition, dynamic RAMs save money in larger systems. Less chip power means smaller and cheaper power supplies. Smaller supplies mean a further saving in reduced cooling requirements. In general, the larger the memory system, the greater the savings by using dynamics.



Static RAM Cell



Dynamic RAM Cell

Figure 1.

Dynamic RAMs

Refresh: Since charge leaks off the storage capacitors, it must be replenished periodically in order for a dynamic RAM chip to retain its data. The charge in any one cell is replenished, or refreshed, every time that cell is accessed for a read or a write. At the same time, all the other cells in the same row are also refreshed. For that reason the entire RAM chip can be refreshed by doing only 64 cycles (for 4k RAM; a 16k RAM needs 128

*Refer to Introduction.

cycles) in 2 ms while sequencing through all the row addresses. The bit pattern presented to the column addresses does not matter. However, the setup and hold times must still be met. Unstable column addresses during refresh will cause data loss.

The hardware required for refresh amounts to a 6- or 7-bit counter for the refresh addresses, some way to multiplex the counter onto the RAM row address lines, a timer to signal when a refresh should be done, and the miscellaneous gating needed to couple into the usual read/write logic.

In some systems no extra refresh logic is needed. For example, in CRT systems normal operation sequences through all the row addresses in less than a 2 ms refresh period. This will be true only if the row address bits on the RAM chip are driven from the least significant address bits of the system. As a rule, this is good practice in all systems. By placing the most active system address bits on the RAM row addresses, normal system operation will automatically refresh the bulk of the RAM.

Cycling: One of the key functional differences between static and dynamic RAMs is the fact that dynamic RAMs must run through a cycle in order to read or write. Aborting the cycle by removing the chip enable too early or by trying to start a second cycle too soon after the first will probably cause data loss. Minimum chip enable on and off times must be observed.

Summary

Static RAMs are easier to use. Dynamic RAMs are cheaper, use less power, must be refreshed, and must be cycled.

MEMORY SUBSYSTEM DESIGN CONSIDERATIONS

Some memory board designs are easy to manufacture, while others, functionally identical, have low manufacturing yields seemingly due to the many "bad" chips. The difference between them is usually the amount of margin designed into each system. Power supply and timing margins are both critical, and as the margins go to zero or negative, the amount of "soft" errors goes up. (A chip has a "hard" error if a location consistently cannot be written and read back properly. It has a "soft" error if it only occasionally fails.)

On careful analysis, "soft" errors usually occur during a memory cycle in which some system parameter has gone out of spec. Since the RAM chips themselves have variations in their margins, replacing the offending RAM with one that has a greater margin in the out-of-spec parameter seems to cure the problem. This results in a large pile of "bad" RAMs. However, the real solution to this type of problem is in a careful system design and board layout in the beginning.

Power Distribution

By far the single most important aspect of a successful RAM system is good power distribution consisting of carefully designed decoupling and power gridding. The importance of good power distribution cannot be overemphasized.

Let's examine the problem. All dynamic RAMs have at least two supplies (V_{DD} and V_{BB}; V_{SS} is the RAM internal ground). Most also have a third called V_{CC}. We will discuss only V_{DD} since the other supplies have similar characteristics. Figure 2 shows the I_{DD} current waveform for a typical dynamic RAM chip during a memory cycle.

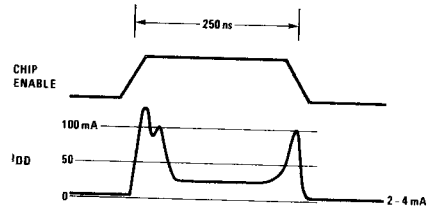


Figure 2.

At the beginning and end of chip enable, each RAM chip draws 50 to 100 mA current spikes with rise times of 20 ns. In addition, each RAM package draws a 20 to 40 mA DC current lasting for the duration of chip enable. The power distribution system must supply these currents while the voltages at the RAMs remain constant. Figure 3 is a schematic of the V_{DD} supply for a row of eight RAMs. The inductors are due to PC trace inductance which is about 10 nH per inch for a 13 mil trace. If the RAMs are on ½-inch centers there is 10 nH total between RAMs.

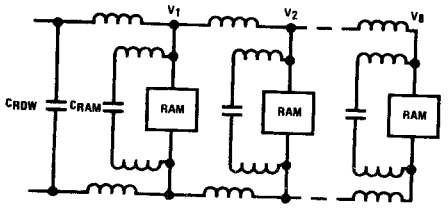


Figure 3.

If there is only one capacitor per row, and if that row capacitor, C_{row}, were infinite, the voltage spikes at the first, second, and last RAMs would be:

$$V_1 \text{ spike} = L \times 8\frac{di}{dt} = 10\,nH \times 8 \times \frac{100\,mA}{20\,ns}$$
$$= 400\,mV$$

$$V_2 \text{ spike} = L \times 15\frac{di}{dt} = 750\,mV$$

$$V_8 \text{ spike} = L \times 36\frac{di}{dt} = 1800\,mV$$

These spikes, especially the last two, are unacceptable.

If each RAM had its own decoupling capacitor, C_{RAM}, in series with 10 nH of trace inductance, the voltage spikes would be:

$$V_{spike} = L\frac{di}{dt} = 10\,nH \times \frac{100\,mA}{20\,ns} = 50\,mV$$

which is very good. Local decoupling should be used to overcome the spiking problem.

The ability of the power distribution system to supply the 20 to 40 mA per chip during a RAM cycle is also a function of the series inductance. For example, see figure 4.

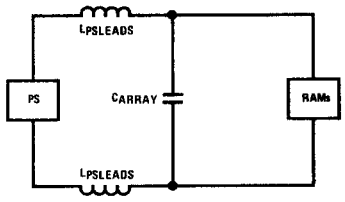


Figure 4.

Let the total power supply lead inductance be 50 nH. (We are ignoring the PC trace inductance within the RAM array itself.) What would the voltage step have to be at the RAMs to get 40 mA x 8 flowing in 20 ns? The equation is:

$$E = L\frac{di}{dt} = 50\,nH \times 8 \times \frac{40\,mA}{20\,ns} = 800\,mV$$

which is a large step which should not occur with adequately designed decoupling. Therefore, the bulk of the DC current for cycling the RAMs comes from decoupling caps themselves, and therefore they should be large enough to supply these currents with little droop. They should also be close enough to the RAMs to minimize the effect of the spikes. If we use a 0.1 μF cap, the droop for a 250 ns cycle would be:

$$V_{droop} = V = \frac{I}{C}t = \frac{40\,mA}{0.1\,\mu F} \times 250\,ns = 100\,mV$$

which is acceptable. Obviously, a larger capacitor will do an even better job of handling the droop. In addition, to help keep the droop to a minimum, the board should have about 50 to 200 μF of bulk decoupling on the +12 V supply. The other supplies can have less. Half should be placed near the point where the supplies enter the board. The other half should be placed at the far side of the RAMs so that the array lies between the bulk decoupling capacitors.

Alternatively, half of the bulk capacitors can be spread throughout the array. If this is done, use approximately 5 to 10 μF per eight RAM chips for V_{DD} and V_{BB}. For V_{CC}, 5 to 10 μF for 32 chips should be adequate.

Intuitively, this second approach seems better. However, the first technique works fine and is probably more cost effective.

The choice of capacitor types is very important. In all of the above decoupling calculations we have ignored the effective series resistance (ESR) of the capacitors. The effect of the ESR of real capacitors is probably at least as large as the effect of PC trace inductance and is a function of the capacitor type.

For best results, use ceramic capacitors for the local decoupling. The Memory Systems Group at National Semiconductor has had good results with ceramic capacitors of Z5U material from AVX and Sprague. To illustrate how important the Memory Systems Group feels these capacitors are to good memory board performance, every lot is subjected to an incoming inspection which includes, among other things, a transient response test.

For bulk decoupling, solid tantalum capacitors are recommended. They have better transient response than most other large value capacitors and they put a lot of capacitance into a small package which simplifies board layout.

A word about power gridding. If there are a number of rows of RAMs, all power supply traces to all RAMs should be run both vertically and horizontally throughout the array. Providing multiple paths through the array reduces the effective inductance of the power distribution system.

To summarize power distribution, we can say the following:

1. It is the single most important aspect of a good RAM board layout.
2. Use plenty of decoupling. The decoupling caps not only reduce voltage spikes, but also provide most of the RAM power during the cycling. Lay out the board for a 0.1 μF capacitor per power supply per RAM chip (up to three capacitors per chip). As production history accumulates, it may be possible to omit half the capacitors. However, lay out the board for one per supply per chip. Use 50 to 200 μF of bulk decoupling on +12 V. On +5 V and −5 V use 25 to 100 μF.
3. The decoupling capacitors should have the shortest possible traces back to their respective RAM power supply and ground pins. To reduce inductance further, these traces should be as wide as room will allow.
4. Traces running the power supply voltages throughout the array should be as wide as possible. However, with good decoupling design, even minimum trace widths will probably be acceptable. If some power supply traces can be wider than others, make V_{SS} (Ground) wider first, V_{DD} next, V_{BB} next, and finally V_{CC}. Ground is the key. Grid the supplies even if the traces are heavy in one direction and light in the other.
5. We have purposely omitted any discussion of multilayer boards. They tend to simplify power distribution problems, but the types of problems that must be solved are the same. Only the magnitudes have been somewhat reduced. Almost everything that has been said up to now is still applicable to multilayer boards.

Data and Control Signal Distribution

The second most important aspect of the successful RAM system is address, data, and control signal distribution.

Let's discuss the chip enables first. This is the most important signal to the RAM and all timing is referenced to it. There are two types of chip enables in common use today: 12 volt and TTL level swings. Running chip enable lines through an array tends to be less of a problem than one would think. There are only two things to keep in mind. First, place the actual driver chip near the RAM array it is driving, making the chip enable run short and direct. Second, put a damping resistor near the driver. Do this for either TTL level or 12 volt chip enables. Select the value of this resistor to

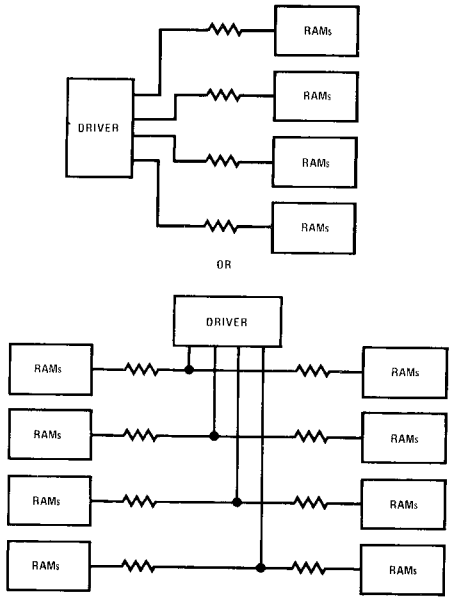


Figure 5.

give the best clock waveform at the RAM chips. Its value will probably be between 10 and 51 Ω. Figure 5 shows the commonly used arrangements.

The reasons for these two recommendations stem from the fact that, at the frequencies encountered here, the clock lines are, in fact, transmission lines. The impedance of the line is determined by:

$$Z_O = \sqrt{L/C}$$

where the impedance of the clock line within the array of chips is in the range of 10 to 15 Ω while the unloaded line between the clock driver and the chip array is in the range of 30 to 50 Ω.

In order to drive the clock line cleanly, some attempt must be made to match the clock driver's output impedance to that of the line. The actual output impedance of most monolithic clock drivers varies as much as 3 to 1 and so we choose a fast clock driver with low output impedance and put a damping resistor in series to empirically set the effective output impedance to match the line (with only about 10% variation).

Long clock lines or long lengths of unloaded clock lines can cause problems. In the case of the long clock line, the open circuit at the far end of the line causes the reflection from the end of the line to return to the driver after the end of the rise time, resulting in ringing. In the case of the long unloaded length of line, the reflection from the junction of the unloaded and loaded sections of the line (due to the mismatch) causes glitches in the clock transitions.

To minimize crosstalk from chip enable to other signals, try to run chip enable at 90° to other signals. This is usually hard to do in an actual layout. As an alternative, leave as much room as possible between chip enable and adjacent traces as it runs through the array. Typically, signals in the array are on 50 mil centers. Moving the two adjacent signals more than 50 mils away from chip enable will help some in reducing crosstalk. However, as stated earlier, there seem to be very few problems associated with chip enable. Neither CE itself nor crosstalk to other signals will be troublesome if the above guidelines are observed.

Address, data, and control signals such as read/write (or equivalent) should be run as directly as possible. Their layouts tend to be non-critical. The critical thing is timing. The control logic should be designed to maximize setup and hold times with respect to chip enable. Again, high production yield is related to margins. As an example, consider a RAM board that was built for an 8080 system. The chip used was MM5271 4k RAM which has a low true TTL level clock input. The signal that controls read, write, and refresh is called TSP. The MM5271 data sheet says that the setup time for TSP is zero ns with respect to the leading edge of chip enable. When doing a refresh, TSP must be low at the beginning of chip enable. The original timing brought TSP down at the same time as chip enable. The system seemed to work. However, it would make an error once every half hour or so. With an oscilloscope everything appeared to be within specification. When TSP and chip enable were superimposed on the scope, their leading edges were absolutely coincident in both time and waveshape. The TSP/chip enable relationship was examined very, very carefully and pronounced okay. Finally, in an attempt to cure the problem, the TSP timing was changed to give about 50 ns of setup time and the problem disappeared.

The point was that the original design was done to the limit of the memory data sheet even though there was no need to do so. The success of the operation depended on the shape of the two waveforms to keep the system in spec. Once margin was designed in, with no hardship at all in the design, the system operated flawlessly.

In high-speed systems where it is hard to design in extra margin, use damping resistors in address, data, and control lines to help control their waveshapes. A resistor in every address, data, and control line allows these waveforms to be optimized, which gives the system improved margin over an undamped design. Use damping resistors only where necessary. Leave them out of signals that have time to settle down before they are needed.

Summarizing the use of damping resistors: always put them in chip enable lines, whether they are TTL levels or 12 V levels. Use them as necessary in those address, data, and control lines whose timings are approaching the limits of the RAM chip data sheet. Design in margin first. Tune it in when it can't be designed in.

LOGIC CONSIDERATIONS

These also affect yield. For example, a RAM cycle must never be aborted before its normal completion. The control logic must be designed to never permit a shortened cycle. At this point we shall briefly discuss some techniques for timing and control.

Timing Generation

The actual phasing of control signals can be done a number of ways. Existing system level control signals can be used. This is easy in some 8080 and PACE systems. When the available system control signals aren't quite up to the job, another technique that works

quite well is to use a high frequency oscillator and a shift register connected as a Johnson counter. Any timing signal that is necessary can be generated from a Johnson counter using a 2 input gate. This technique has a minor drawback if the high frequency oscillator is asynchronous with respect to the main system timing. The RAM cycle timing will always have a finite uncertainty with respect to the system cycle timing. This uncertainty is equal to the clock period of the high frequency oscillator. To apply the same technique but to avoid the timing uncertainty, use a gated delay line oscillator instead of a crystal or RC oscillator. Delay line oscillators can be started and stopped reliably. Crystal and RC oscillators take a few cycles to settle down and therefore are not reliable in a start-stop mode. How about one-shots? Do not, under any circumstances, use one-shots in critical timing applications!

Refresh Timeouts

A counter and oscillator are best. An astable oscillator is acceptable but must be carefully designed for worst case minimum frequency with respect to temperature to ensure the RAM gets refreshed often enough. At the end of the timeout a flip-flop should be set. When the refresh cycle is finally completed, that flip-flop should be reset. The timing should be such that it doesn't matter when the RAM gets refreshed within the refresh timer period.

Transparent versus Non-Transparent Refresh

Most microprocessors have predictable periods of time when they will not access the RAM board. Usually it takes little effort to insert refresh cycles in these times, thereby making refresh transparent to the CPU. When the CPU is very fast and is using the bus almost continuously, the refresh will have to hold up the processor. Even then, some clever design will minimize the time spent doing non-transparent refresh.

Single Step

Some systems need the capability of single stepping through programs. Since dynamic RAMs must be refreshed continuously, the output data from RAM should be latched. This permits single stepping because every time the address changes, the RAM is read and the data is latched. Then the refresh proceeds behind the latches, never disturbing the data. The RAM appears static.

DMA

DMA should be little different from normal cycles. One thing that must be considered is how to handle refresh. Three techniques immediately come to mind. First, have the DMAing device permit refresh periodically. Second, limit the DMA frequency. Make the period between DMA cycles equal to a normal RAM cycle plus a refresh cycle. This way the refresh can be handled transparently to the DMA. The third technique would be to limit the DMA time to something under 2 ms and at the end of the DMA do a burst refresh of the entire memory. There are a number of other ways to handle refresh and DMA. Performance of the system will determine which technique is most appropriate.

From a system standpoint, the most important aspect of DMA and dynamic RAMs is the polarity of the system level control signals. In a TTL system where bus control can change hands, the control signals must be *low true.* The reason for this is that, as control transfers from the CPU to the DMA device and back again, there will be short periods of time when the control lines are floating since neither device is driving the lines. In a TTL system, floating lines look high. If control signals have been defined as high true, then as the lines momentarily float, devices such as our RAM board will think that a command has been issued and will start an unintended cycle. Then the problem gets compounded. During the unintended cycle comes the command for a real cycle. Either the unintended cycle will be aborted, which destroys some RAM contents, or the intended cycle will start too late, causing other problems in the external system. An example of a bus with this problem is the S100, or hobby standard bus. It mixes signal polarities and, therefore, makes dynamic RAM control logic unnecessarily complex. If there is only one controlling device, signal polarities are academic. But if control can transfer, make the control lines low true.

SUMMARY

Refresh requirements make dynamic RAMs slightly harder to use than static RAMs. However, they pay the designer back for his efforts by reducing overall system cost in three ways. First, dynamic RAMs tend to be cheaper than static RAMs of the same size. This is primarily due to smaller chip sizes and higher production volumes than comparable static RAMs. Second, dynamic RAMs use less power. When a dynamic RAM is not being accessed it draws much less current than a static RAM. During access, dynamic and static RAMs draw similar amounts of power. However, in a large array, only that bank being accessed draws full power. All others still draw standby currents so that the total system power is lower than for a comparable static system. Because of the reduced power requirements, power supplies are cheaper. And, third, due to lower power dissipation, cooling requirements are reduced, allowing a further saving.

There are three things the system designer can do to maximize RAM board yields during manufacture. First, design proper power supply decoupling. This is probably the single most important consideration for the designer. A good high frequency 0.1 μF capacitor per supply per memory chip is recommended. A capacitor per supply per two chips is probably okay, but the board should be laid out for one capacitor per supply per chip and then capacitors can be left out as yield data becomes available. For bulk decoupling use about 50 to 200 μF per board on +12 V, less on +5 V and −5 V.

Second, design in as much margin as possible in all control signal timing. Use damping resistors where necessary. If timing is designed right to the minimum specs, periodically the right combination of data pattern, power supply noise, temperature, cosmic radiation, etc., causes the system to fail. The combined worst case parameters push a signal beyond specification and the memory fails.

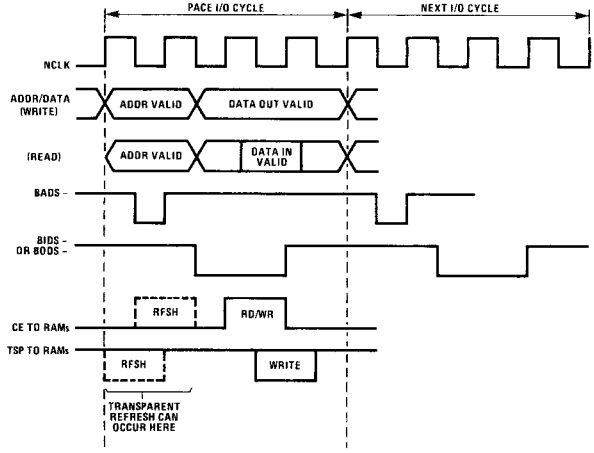
Third, never allow spurious, shortened memory cycles to occur. Shortened or aborted memory cycles are guaranteed to destroy data in the row that was addressed during the aborted cycle.

Any designer who uses reasonable care can successfully design dynamic memory systems which will be easy to manufacture and very reliable in the field.

APPENDIX A

Appendix A shows a simplified timing diagram and schematics for a 16k x 8 RAM board used by a PACE microprocessor for byte mode data storage.

RAMs are MM5270 4k RAMs with 12 V chip enables. All timing is generated from existing system signals. DMA is not now in use, but is possible in the future. Control signals from the bus are low true. Refresh is transparent, done at any time in the absence of any address, data in, or data out strobes, coincident with the rising edge of clock.



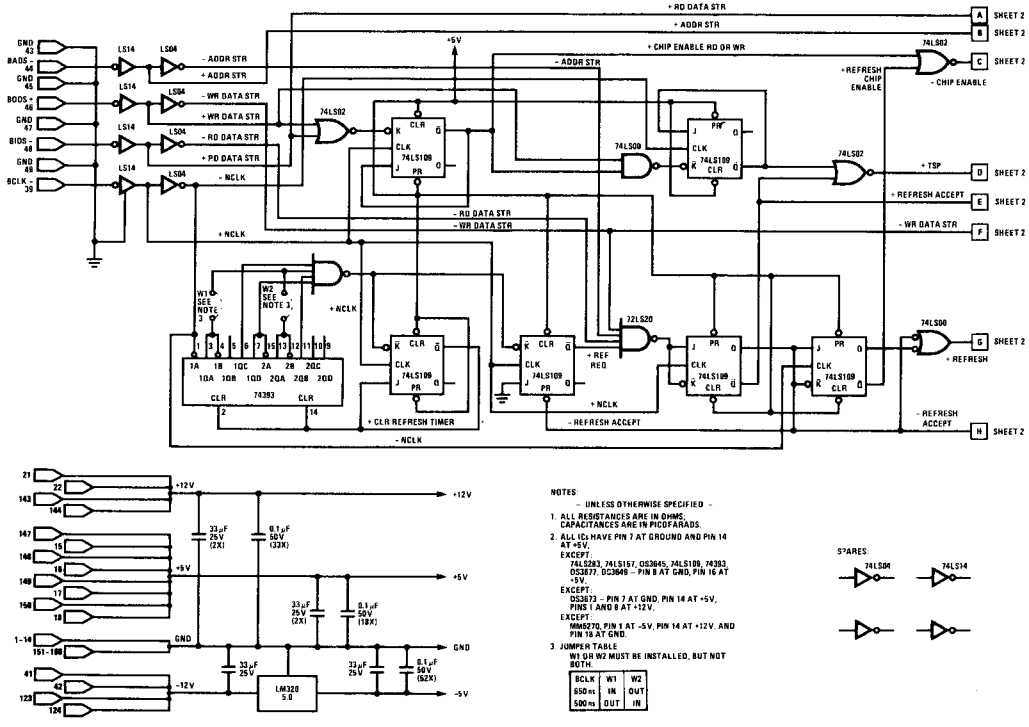


Figure 6. Control Logic, 16 x 8 R/W Memory Board

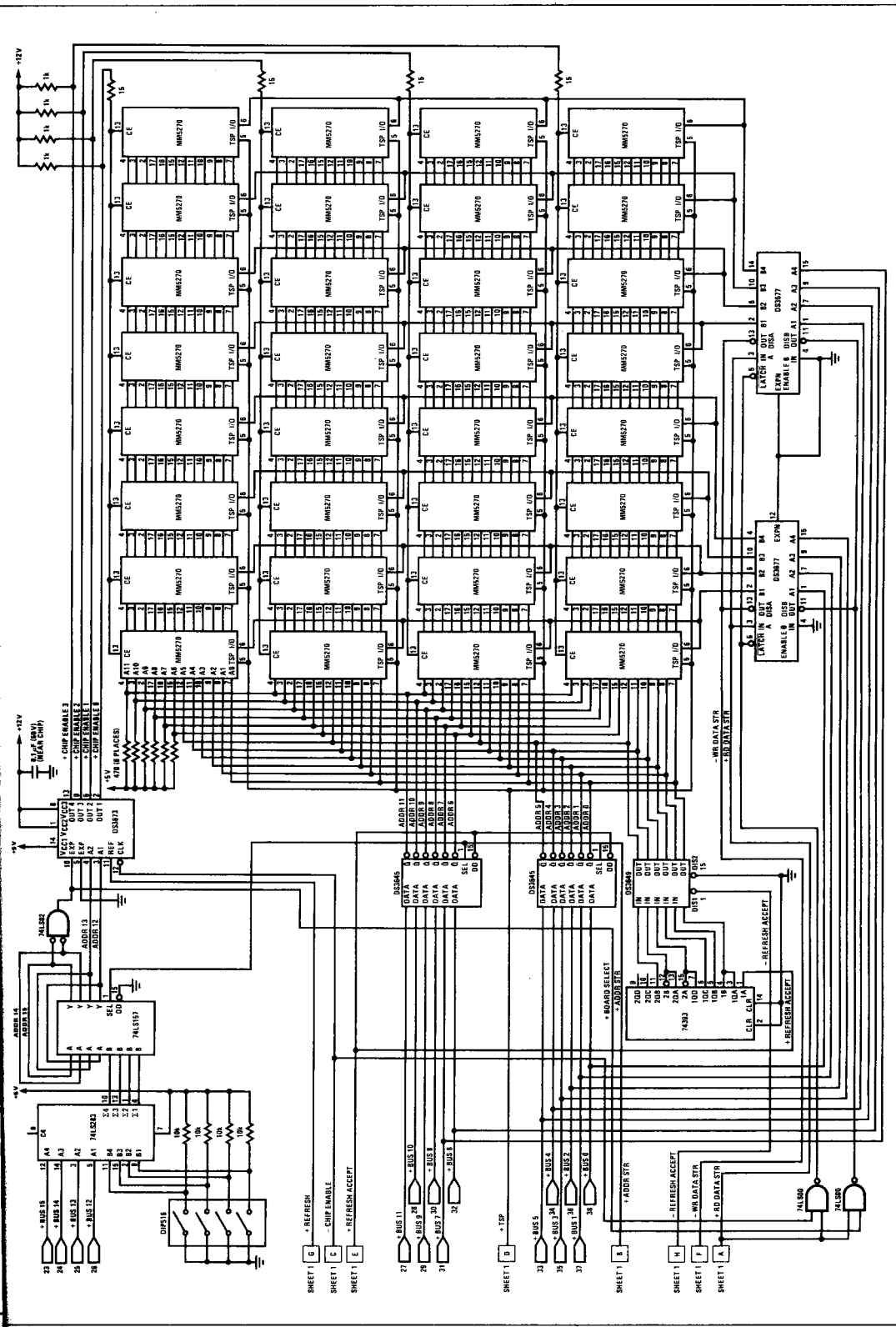


Figure 7. Memory Array, 16k x 8 R/W Memory Board

APPENDIX B

In Appendix B we show the printed circuit board layout for a 16k x 16 RAM board which uses MM5270 18-pin 4k RAMs. It is an excellent example of the proper way to grid and decouple the power supplies on a two-sided board. Also shown are the centrally located chip enable drivers with their series damping resistors.