Using the 54/74LS610-13 memory mapping units

Microprocessors with 16-bit address buses present the most cost effective solution for many systems requirements, but are constrained by their 64 kbyte memory access limit. **Thomas J Tyson, Deene Ogden, Jim Gallia and Dennis Frailey** show how memory mapping units can be used to expand addressing capability by a factor of 256

The 8-bit microprocessor has not been rendered obsolete by newer and more powerful 16- and 32-bit microprocessors. Many applications are more than adequately served by 8-bit microprocessors, with their lower-cost address and data buses. However, there is a 'grey area' in which 8-bit chips are not sufficient but 16-bit chips represent an uneconomic 'overkill'. A significant limitation of 8-bit microprocessors with 16-bit address buses is their inability to access more than 64 kbyte of data. This application shows how the SN54/74LS610-3 series of memory mapping units enables an 8-bit microprocessor to access memories much larger than 64 kbyte without significantly increasing the chip count of the system. Note that a memory mapping unit cannot simply be engineered into a microprocessor system without adding the appropriate software, as the 8-bit microprocessor can still address only 64 kbyte of logical address space at any instant. Software is usually included in the operating system to map 64 kbyte of the physical memory onto the logical address space. A.C.

microsystems microprocessors memory addressing peripheral devices

With the advent of high-density semiconductor memories, microprocessors are being used more and more in systems featuring memory structures larger than 64 kbyte. The majority of the microprocessors in use or available today have a 16-bit address bus, with a maximum addressing capability of 64k words. Due to this limitation, some sort of memory mapping is necessary to adapt the microprocessors to applications where large memory structures are required.

The SN54LS10-13 and SN74LS610-13 memory mappers from Texas Instruments were designed to alleviate this addressing limitation. The devices employ a paged memory mapping technique to expand the system memory address bus by 8 bit, thus effectively increasing the system addressing capability by a factor of 2^8, i.e. 256. For microprocessors with a 16-bit address bus (such as the Z80, the 8085 and the 6800), this results in an increase in the maximum addressing capability from 64 kbyte to 16 Mbyte, and for the TMS9900 (which has a 15-bit address bus) the result is an increase from 32k words to 8M words (where 1 word = 2 byte).

In the mapping operation, the four most significant bits (MSBs) of the microprocessor address word are used to access one of the sixteen 12-bit registers of the memory mapper's 16 × 12-bit RAM array. Each mapper register is capable of holding a 12-bit address which will be termed the page address and will be used as the 12 MSBs of the memory address bus. The remaining 12 bits (11 in the case of the TMS9900) of the microprocessor address bus will be transferred directly to memory from the microprocessor and will be used to address the memory locations within each page (see Figure 1).

The memory will be organized into 2^x pages (where x is the number of bits of the page address) with 2^{n-4} words or bytes (where n is the bit length of the microprocessor address bus) per page. Once loaded, the mapper can access only 16 pages or 64 kbyte (32k words in the TMS9900 case). To access more pages, the memory mapper RAM array must be reloaded with 16 new page addresses. This is done by the microprocessor via the data bus with the mapper in the write mode. (A more detailed description of the modes of operation is given below.)

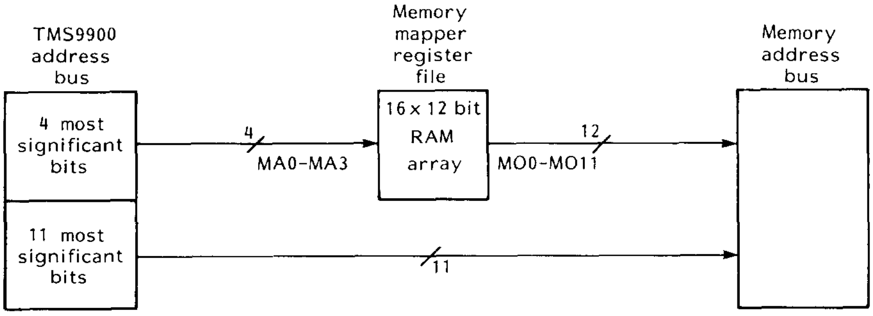

Figure 1. Mapping operation

Low-Power Schottky Applications Group, Texas Instruments Inc., 13500 North Central Expressway, PO Box 655012, Dallas, TX 75265, USA

0141-9331/88/05286-06 $03.00 © 1988 Butterworth & Co. (Publishers) Ltd

FUNCTIONAL DESCRIPTION

A functional block diagram of the SN54/74LS610-13 series memory mapper is shown in Figure 2. This consists mainly of a 4-bit two-to-one multiplexer, a 16 × 12-bit RAM array, a 12-bit two-to-one multiplexer, 24 tristate buffers, control logic and, in the case of the 'LS610 and 'LS611, a 12-bit transparent latch. Table 1 lists the functional differences between the 'LS610, 'LS611, 'LS612 and 'LS613 devices. Table 2 lists the function of each pin.

Depending on the state of the input control signals (i.e. CS*, R/W*, STROBE*, MM* and ME*), the mapper can be operated in three basic modes of operation: I/O (read or write), map and pass. An explanation of each mode of operation and the control signals necessary to achieve that mode is given below and in Table 3.

I/O mode

In this mode a page address can either be loaded into a mapper register or can be read from a memory mapper register depending on the state of the R/W* (read/write) input. This input signal controls either the read or the write function of the I/O mode. Read and write modes are illustrated in Figure 3.

Write mode

One of the sixteen 12-bit registers is loaded with a page address via the D0-D11 I/O ports from the microprocessor. The address of the selected register is input via the RS0-RS3 inputs, and usually comprises the four least significant bits (LSBs) of the microprocessor address word. The chip select (CS*), strobe (STROBE*) and R/W* controls should all be low.

Read mode

The contents of one of the sixteen 12-bit registers is read from the mapper via the D0-D11 I/O ports. As in the write mode, the mapper register is selected by the address on the RS0-RS3 inputs. Again, chip select (CS*) should be low, while R/W* should be kept high.

Table 1. Device comparison

Device	Map outputs latched	Map output type
SN54/74LS610	Yes	Tristate
SN54/74LS611	Yes	Open collector
SN54/74LS612	No	Tristate
SN54/74LS613	No	Open collector

Table 2. Pin functions

Pin	Pin name	Functional description
7-12 29-34	D0-D11	I/O connections to data and control bus used for reading from and writing to the map register selected by RS0-RS3 when CS* is low. Mode controlled by R/W*. (D0 corresponds to MO0 and is the most significant bit)
36, 38, 1, 3	RS0-RS3	Register select inputs for I/O operations. (RS3 is the least significant bit)
6	R/W*	Read or write control used in I/O operations to select the condition of the data bus. When high, the data bus outputs are active for reading the map register. When low, the data bus is used to write into the register.
5	STROBE*	Strobe input used to enter data into the selected map register during I/O operations
4	CS*	Chip select input. A low input level selects the memory mapper (assuming more than one used) for an I/O operation
35, 37, 39, 2	MA0-MA3	Map address inputs to select one of 16 map registers when in map mode (MM* low and CS* high). (MA3 is the least significant bit)
14-19 22-27	MO0-MO11	Map outputs. Present the map register contents to the system memory address bus in the map mode. In the pass mode, these outputs provide the map address data on MO8-MO11 and low levels on MO0-MO7 (MO11 is the least significant bit)
13	MM*	Map mode input. When low, 12 bit of data is transferred from the selected map register to the map outputs. When high (pass mode), the four bits present on the map address inputs are passed to the map outputs
21	ME*	Map enable for the map outputs. A low level allows the outputs to be active while a high input level puts the outputs at high impedance
28	C	Latch enable input for the 'LS610 and 'LS611 (no internal connection for 'LS612 and 'LS613). A high level will pass data transparently to the map outputs. A low level will latch the outputs
40, 20	V_{CC}, GND	Power supply (5 V) and network ground (substrate) pins

Map mode

Map mode is illustrated in Figure 4. The contents of one of the sixteen 12-bit memory mapper registers is output to the system address bus via the MO0-MO11 outputs. The address on MA0-MA3 selects the mapper register, and usually comprises the four MSBs of the microprocessor address word. CS* must be inactive (high), the map mode (MM*) control and the map enable (ME*) must both be active (low). The $(n - 4)$ LSBs, where n is the microprocessor address bit length, of the microprocessor address bus will be transferred directly to memory from the microprocessor, while the remaining 12 MSBs of the system address bus will be driven onto the bus by the memory mapper.

Table 3. Modes of operation

Mapper inputs	I/O		Map	Pass
	Write (load)	Read (verify)		
CS*	Active (low)	Active (low)	Inactive (high)	Inactive (high)
STROBE*	Active (low)	Don't care	Don't care	Don't care
R/W*	Low	High	Don't care	Don't care
MM*	Don't care	Don't care	Active (low)	Inactive (high)
ME*	Inactive (high)	Inactive (high)	Active	Active
RS0-RS3	Address of selected register	Address of selected register	Don't care	Don't care
MA0-MA3	Don't care	Don't care	Address of selected register	Address of selected register
MO0-MO11	High impedance	High impedance	Valid address	Valid address
D0-D11	Register contents to be loaded (input)	Register contents to be read (output)	Input mode	Input mode

Pass mode

Pass mode is illustrated in Figure 4. The four LSBs (MO8-MO11) of the memory mapper address bus (MO0-MO11) will be the same as the address on the MA0-MA3 input bus, while the remaining eight MSBs of the memory mapper address bus will all be low. CS* and MM* should both be inactive, and ME* should be active. In other words, the address on the system address bus will be the same as the address output by the microprocessor, and the memory mapper becomes transparent to the system.

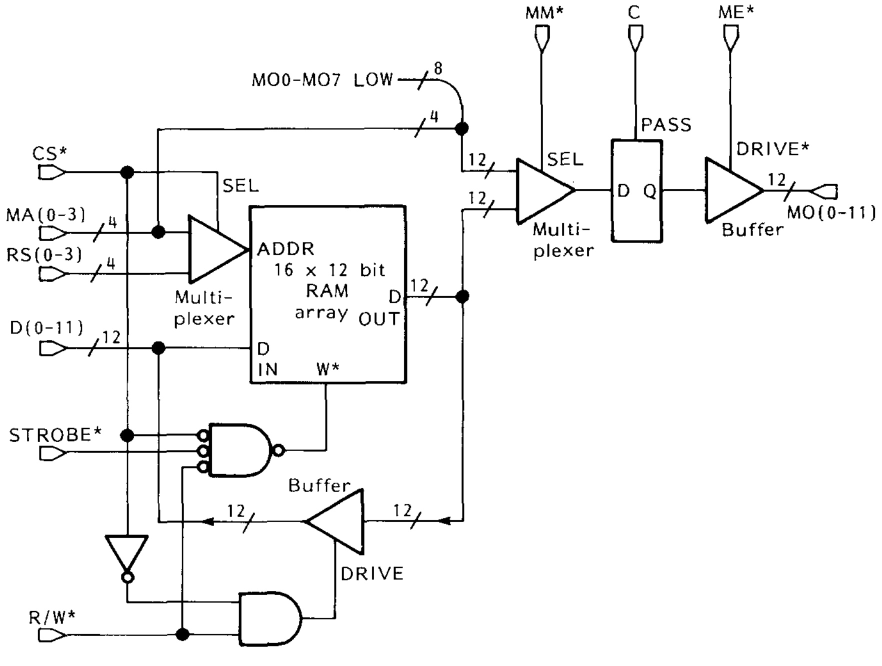

Figure 2. Logic diagram of the 'LS610 memory mapper

SYSTEMS INTEGRATION

The flexibility of the memory mapper is such that it can be used with microprocessors that have either an 8-bit or a 16-bit data bus. To use the memory mapper to its fullest potential with an 8-bit microprocessor (i.e. to expand the address bus by 8 bit), the 12-bit page address must be multiplexed into the mapper via the 8-bit data bus. This means that the time it normally takes to load or read the memory mapper will be at least doubled and extra external circuitry will be necessary. If the requirement of the system is such that the address bus needs to be increased by only 4 bit then there is no need for multiplexing in the page address. This means that the address bus is expanded to only 20 bit, resulting in a 1 Mbyte addressing capability. As examples, we will consider two 8-bit systems using the 'LS612 memory mapper. After this a 16-bit system is considered and multimapper systems are discussed.

TMS9995-based system

Figure 5 shows a TMS9995-based system using the 'LS612 to expand the address bus by 4 bit. The TMS9995 is an 8-bit microprocessor

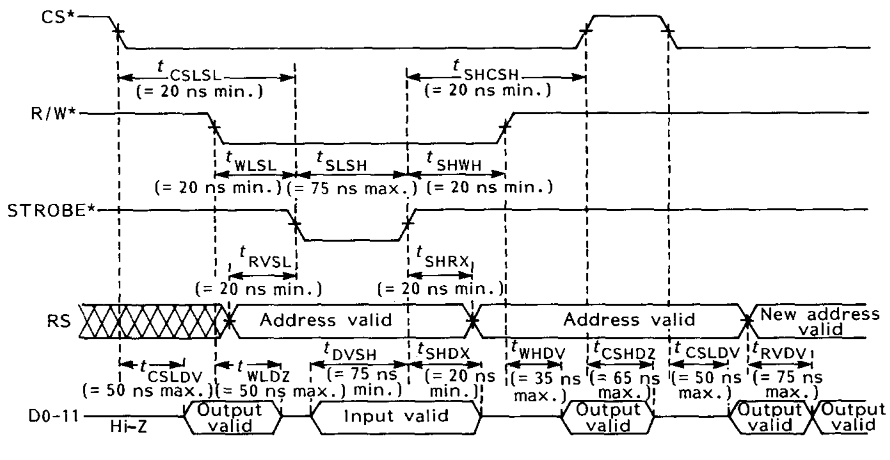
CS*
R/W*
STROBE*
RS
D0-11
t_{CSLSL} (= 20 ns min.)
t_{SHCSH} (= 20 ns min.)
t_{WLSL} (= 20 ns min.)
t_{SLSH} (= 75 ns max.)
t_{SHWH} (= 20 ns min.)
t_{RVSL} (= 20 ns min.)
t_{SHRX} (= 20 ns min.)
Address valid
Address valid
New address valid
t_{CSLDV} (= 50 ns max.)
t_{WLDZ} (= 50 ns max.)
t_{DVSH} (= 75 ns min.)
t_{SHDX} (= 20 ns min.)
t_{WHDV} (= 35 ns max.)
t_{CSHDZ} (= 65 ns max.)
t_{CSLDV} (= 50 ns max.)
t_{RVDV} (= 75 ns max.)
Hi-Z
Output valid
Input valid
Output valid
Output valid
Output valid

Figure 3. Write and read modes

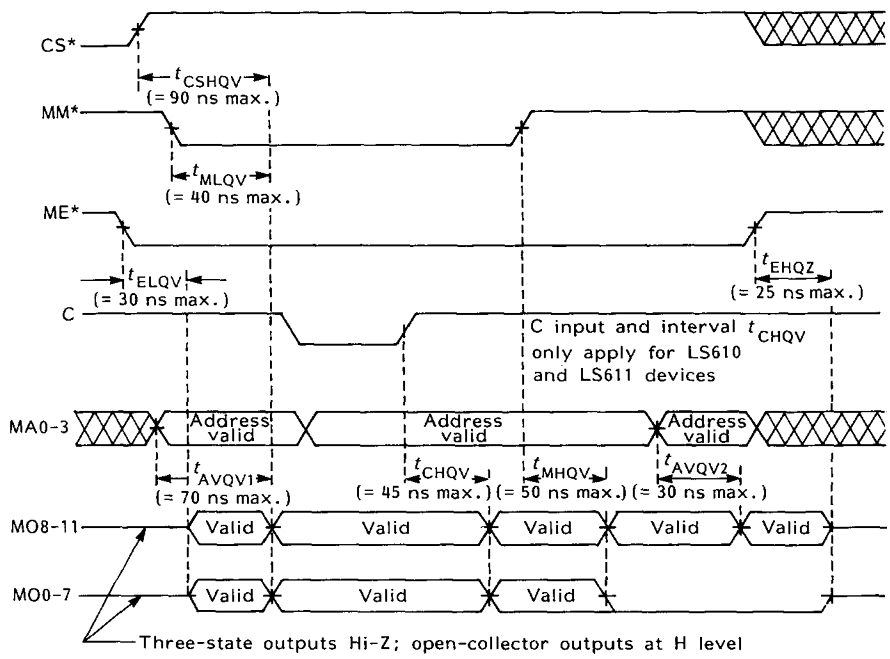
CS*
MM*
ME*
C
MA0-3
MO8-11
MO0-7
t_{CSHQV} (= 90 ns max.)
t_{MLQV} (= 40 ns max.)
t_{ELQV} (= 30 ns max.)
t_{EHQZ} (= 25 ns max.)
C input and interval t_{CHQV} only apply for LS610 and LS611 devices
Address valid
Address valid
Address valid
t_{AVQV1} (= 70 ns max.)
t_{CHQV} (= 45 ns max.)
t_{MHQV} (= 50 ns max.)
t_{AVQV2} (= 30 ns max.)
Valid
Valid
Valid
Valid
Valid
Valid
Valid
Valid
Three-state outputs Hi-Z; open-collector outputs at H level

Figure 4. Map and pass modes

Table 4. TMS9900-LS610 control signals

Memory mapper mode of operation	Control signals		
	P1	P0	AD0
Map	L	H	L
Pass	H	H	L
I/O	H	L	L

Table 5. Z80-LS610 control signals

Control signals			Memory mapper mode of operation
D0	D1	(AD0) IORQ*	
L	L	↑	Pass
H	L	↑	Map
L	H	↑	I/O

with a 16-bit address bus. This system employs a programmable system interface (TMS9901) to control the operation of the mapper. The control of the mapper is software programmable via the I/O ports of the TMS9901. Since the mapper registers are viewed as part of the logical memory space, an address decode (AD0) of the 12 MSBs is gated with a CRU bit to select the mapper for a read or write operation. The specific mapper register is then selected by the four LSBs of the microprocessor address bus (A15-A12) via the RS0-RS3 inputs of the mapper. Table 4 shows the state of the three control signals P0, P1 and AD0 and the corresponding mode of operation of the mapper. When placed in the I/O mode, the read or write operation is then controlled by memory signals from the microprocessor (i.e. WE*/CRUCLK*, MEMEN*, and DB IN*). On power up and reset, the I/O ports of the 9901 are put into the input mode. The pull-up resistors R_1 and R_2 ensure the mapper is placed in the pass mode during power up and reset. The resultant address bus is 20 bit wide, and SA19 is the LSB.

Z80-based system

Figure 6 shows another 8-bit (Z80-based) system using the TI memory mapper. In this case, control of the mapper is implemented by two flipflops feeding MM* and CS*. These flipflops are programmed by the Z80 and are addressed by the data bus, D0-D1. Table 5 shows the states of D0 and D1 necessary to set the mapper in its proper mode of operation. Again, during power up or reset the flipflops are both cleared by RST*, which is supplied by the system and which puts the mapper in the pass mode.

TMS9900-based system

One of the limitations of using an 8-bit microprocessor with the memory mapper, without multiplexing the page address, is that the address bus can only be expanded by 4 bit. In a 16-bit system, i.e. one based on a 16-bit microprocessor like the TMS9900, no extra circuitry is necessary to load the mapper with the full 12-bit address. Figure 7 shows a TMS9900-based system that uses an SN54/74LS612 for memory mapping. The control of the mapper is implemented in the same fashion as in the TMS9995-based system above. The resultant addressing capability is 8M words. Both of these microprocessors have address space set aside for reset, XOP and interrupt vectors, which are addressed when the microprocessor performs a context switch. During a context switch, the microprocessor must be able to address these locations, which are part of the logical address (i.e. locations that are capable of being addressed independently by the microprocessor). One method, besides placing the mapper in pass mode, is to load the memory mapper register whose 4-bit address is 0_H with the address of the first page of physical memory. This, like the pass mode, will make the memory mapper appear to be transparent.

Note that in all three of the above systems the ME* input was always connected to ground. This caused the mapper address buffers to be enabled during all modes of operation of the mapper. This is only a problem during the I/O mode where other memory locations are being written into while the mapper register is being loaded. The method used to avoid destroying data already in memory was to put the mapper into the pass mode during the I/O operation. This was accomplished simply by pulling MM* high, thus making the system address equal to the microprocessor address.

Multimapper systems

In a system employing a single memory mapper the maximum active addressing capability is only 16 pages; if increased addressing capabilities are needed, the mapper must be reloaded. To avoid this procedure, another mapper may be added to the system. This will not increase the overall addressing capability of the system, but it will double the amount of active pages and will also afford twice the active addressing capability. Although the control of two mappers is a little more detailed than the control of one, the same basic methods employed in the single-mapper systems can be used.

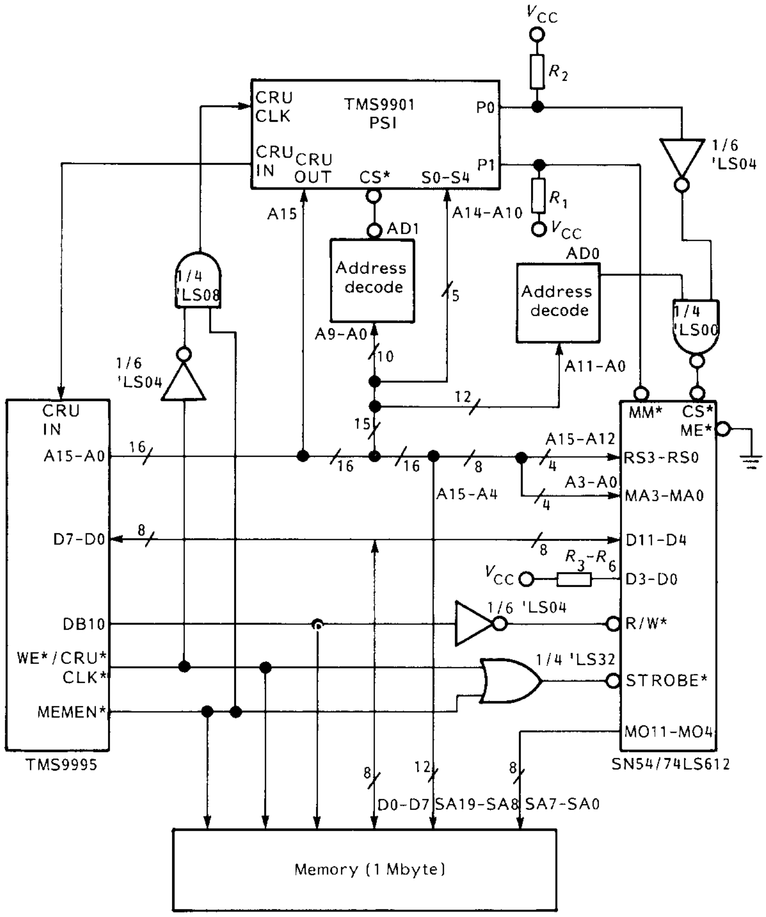

Figure 5. TMS9995-based system with memory mapper

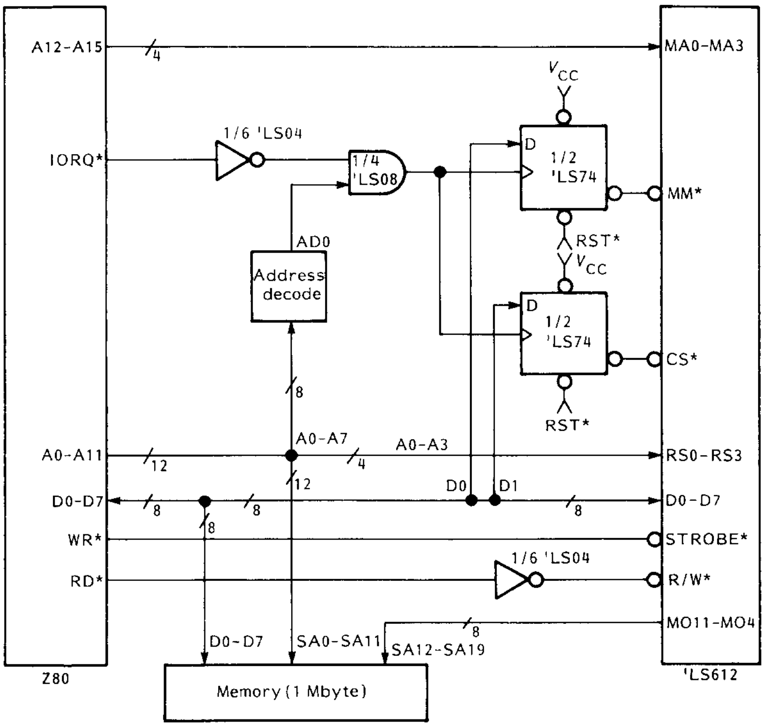

Figure 6. Z80-based system with memory mapper

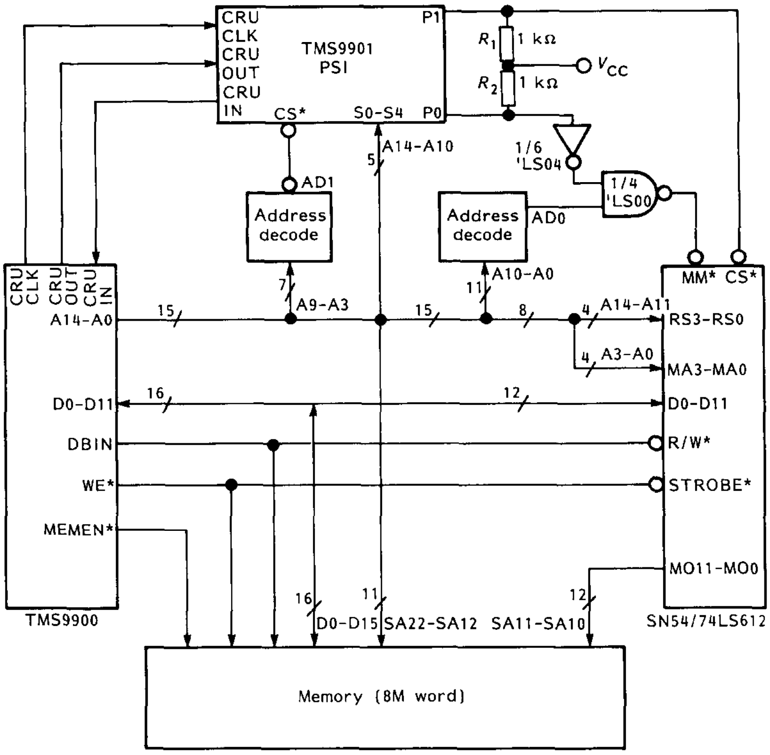

Figure 7. TMS9900-based system with memory mapper

TIMING

The questions of how the mapper affects the critical timing parameters of the memory read/write cycles and what changes, if any, are needed to accommodate the mapper have not yet been discussed here. First, looking at the I/O mode of operation, where the mapper registers are either loaded or read from, it is seen that the mapper registers can be regarded as standard common-I/O static RAMs with maximum access times (RS to valid MO, $T_A = 25°C$, $C_L = 50$ pF, $V_{CC} = 5$ V) of 75 ns. Once the I/O mode is set (CS* = low), the only two signals necessary to read or write into the mapper are STROBE* and R/W*. As shown in the previous system, these signals were supplied directly from the microprocessor with no wait states necessary to perform either function. This will be the case with most microprocessors.

In the map and pass modes, the main concern is the maximum access time (MA to MO). This access time is specified at a maximum of 70 ns which, depending on the timing of the microprocessor and the memory used, may or may not cause any problems. In the Z80-based system, no wait states were introduced by the mapper because the memory control signals become active 95 ns after the microprocessor address bus became valid. This gives the address bus sufficient time to settle down.

For most microprocessors and memories available at the time of writing, the operation of the mapper does not adversely affect the memory cycle timing and is flexible enough to be used with almost all microprocessors.

CONCLUSIONS

The possible uses of memory mappers and the various techniques that can be employed to control their operation are numerous, and only a few examples have been given here. Some of the other possible applications of the mapper include

- achieving system addressing capability greater than 16 Mbyte by reducing the number of mapper registers used by a factor of 2, thus increasing the size of each page by the same factor of 2 without affecting the total amount of pages
- use in systems employing DMA
- memory protection, accomplished by sacrificing one or two bits of the page address and gating these bits with the memory control signals.

Another technique that may be employed in controlling the modes of operation of the mapper is to use PROMs.