MOTOROLA
Semiconductor Products Inc.

AN-853
Application Note

M146805 CMOS FAMILY EMULATORS

Prepared by
Rex Davis
Microprocessor Applications Engineering
Austin, Texas

INTRODUCTION

Emulators are used in place of single-chip microcomputers (MCU) during the debug stage of product development. An MCU is a self-contained system generally consisting of mask ROM for program storage, RAM for temporary storage, a timer/counter, and various amounts of I/O. Because the system software is likely to change during debugging and since mask ROM is expensive in low volume, a substitute for the actual MCU must be used. The substitute must duplicate as many of the MCU features as possible so that the target user system may be debugged, as thoroughly and easily as possible, before the mask ROM is finalized. Also, for this reason, the emulator should appear as transparent to the target user system as possible. Obviously then, an MCU emulator I/O, memory, and pinout should duplicate the MCU I/O, memory, and pinout as closely as possible. However, most importantly, the MCU emulator must allow quick and easy alteration and verification of program memory.

There are two common types of emulation. The first replaces the mask ROM in the MCU with EPROM such as is done with the MC68701 and MC68705s. After the EPROM is programmed, the EPROM MCU version can then be evaluated in the target user system. If errors are found or changes made, the EPROM is then erased and reprogrammed with the new system software. The EPROM method is cost effective and does allow for exact duplication of all MCU features; however, its ability to follow program flow is somewhat limited.

A second method of emulation, more costly but more versatile, employs a processor that can execute the same code as the MCU and can be interfaced with different external memory and peripheral configurations. MCU program memory accesses now occur externally and can be monitored by the user. Valuable debugging aids such as single-stepping and breakpoints can be added to allow instruction-by-instruction or even bus-cycle by cycle-bus examination.

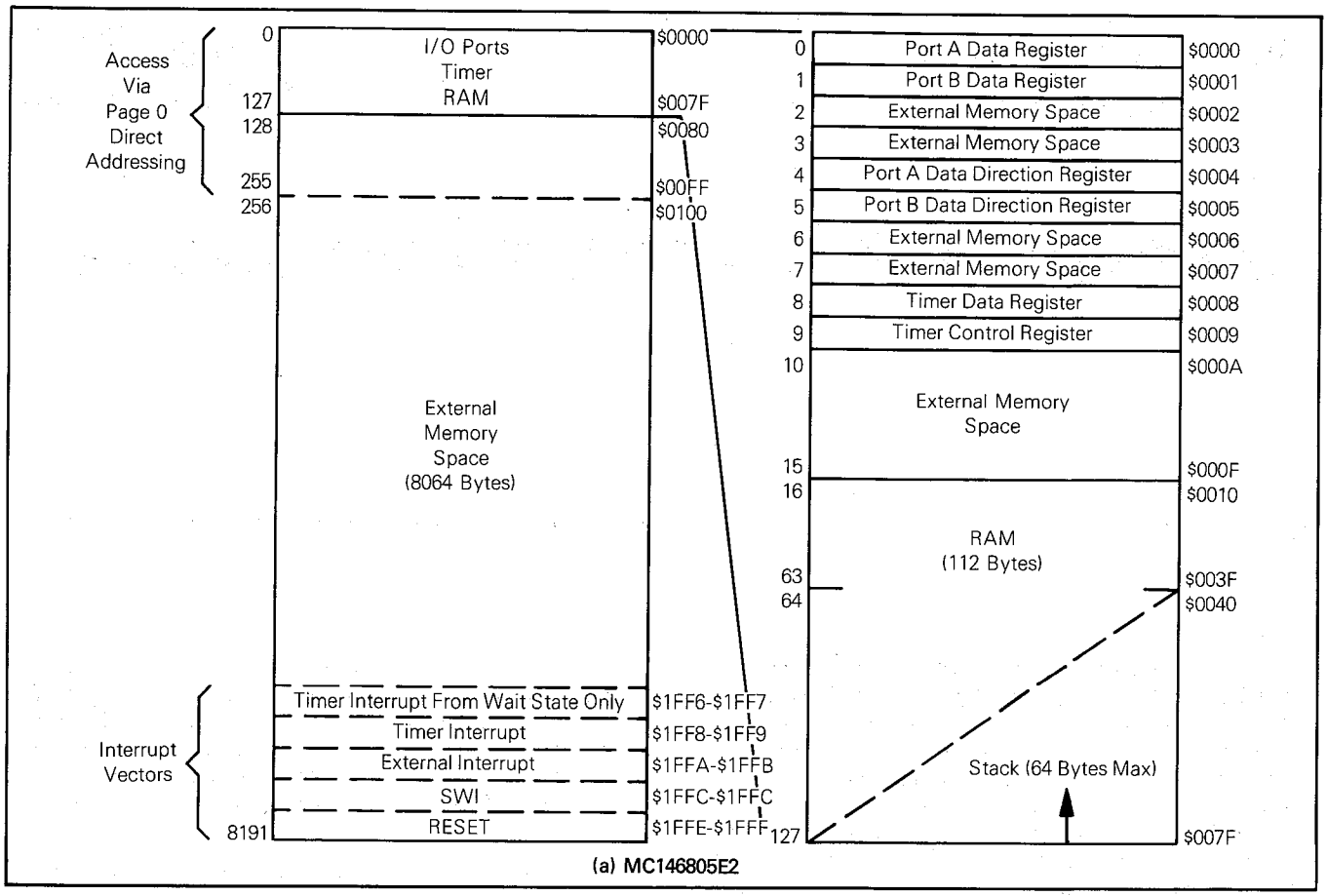
This second emulation method includes systems such as the User System Emulator (USE) modules. These USE modules are available from Motorola Microsystems and contain extensive debug and development capabilities. The schematic diagram for two simple, inexpensive, and powerful emulators are shown in Figures 3 and 5. Descriptions of the design criteria and emulator examples for the MC146805F2 and MC146805G2 are contained in the following text.

M146805 EMULATOR DESIGN CONSIDERATIONS

The MC146805E2 CMOS external, multiplexed address/data bus microprocessor may be used as a core for M146805 CMOS family emulators. Both the MC146805F2 and MC146805G2 MCUs can be emulated using the MC146805E2 MPU. The MC146805G2 contains 2K of mask ROM and 32 I/O lines. The MC146805F2 contains 1K of mask ROM and 16 I/O and four input lines. Table 1 lists a features chart for all current M146805 CMOS family members. Note that since EPROM MCU versions are not yet available, the MC146805E2 can be used for emulation. All M146805 CMOS family members have a similar architecture which is illustrated in the address maps of each family member as shown in Figure 1. Note that the MC146805E2 memory and I/O locations are identical to those of the MC146805G2 and MC146805F2. In order for the MC146805E2 to emulate either the MC146805G2 or MC146805F2, the differences in Table 1 must be resolved.

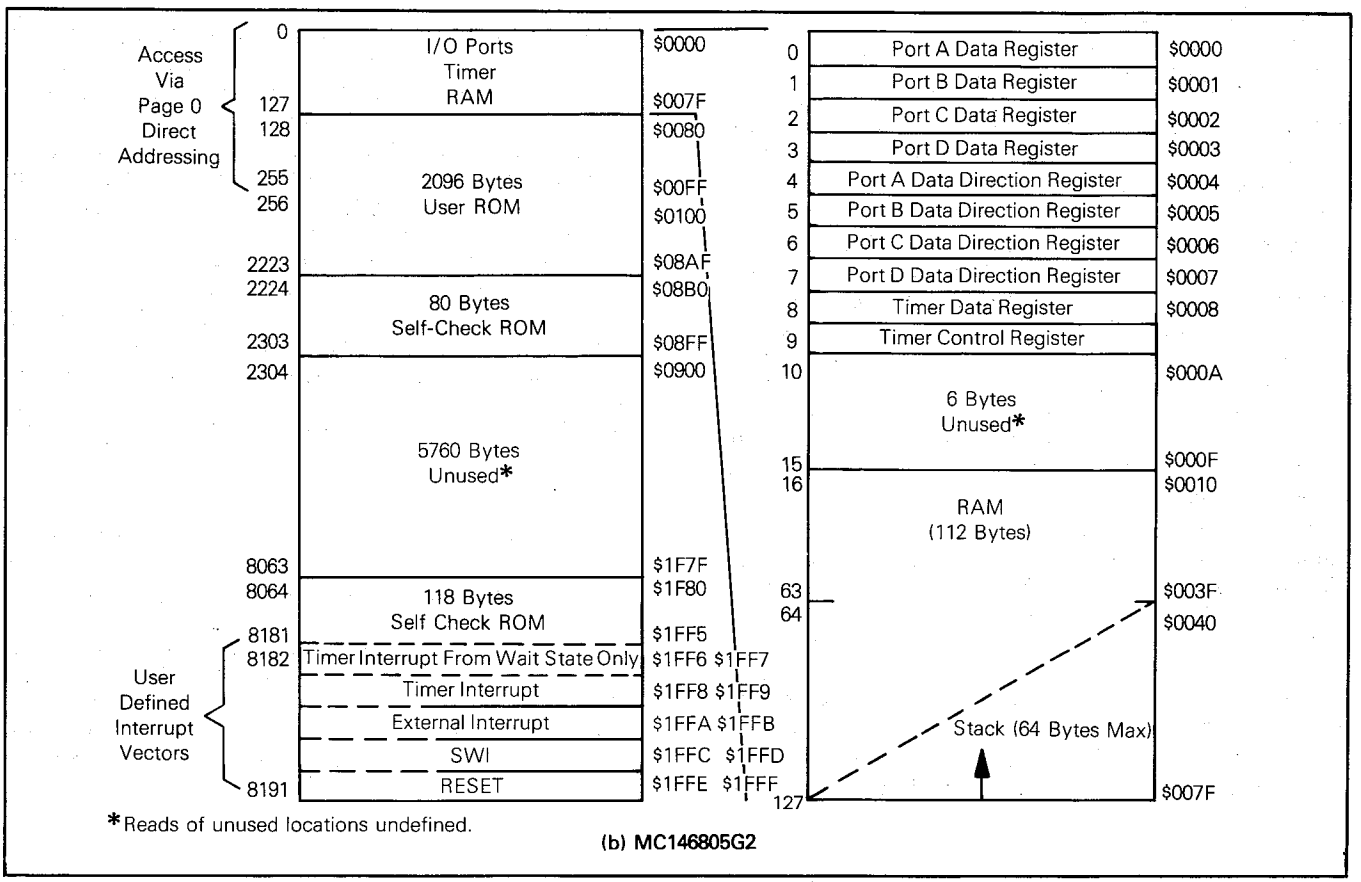
ROM Emulation

Mask ROM can be emulated using either EPROM or RAM. The EPROM version offers a nonvolatile copy that can be erased and reprogrammed; however, the erase and program sequence cannot be done very quickly. The RAM version can be used to allow quick alterations to be made if the RAM is shared with another controller. The RAM offers a trade-off between debug capability and circuit complexity; however, whatever memory type is chosen as a mask ROM substitute, it should reside at the same location as the MCU memory.



(a) MC146805E2

FIGURE 1 — Address Maps



*Reads of unused locations undefined.

(b) MC146805G2

FIGURE 1 — Address Maps (Continued)

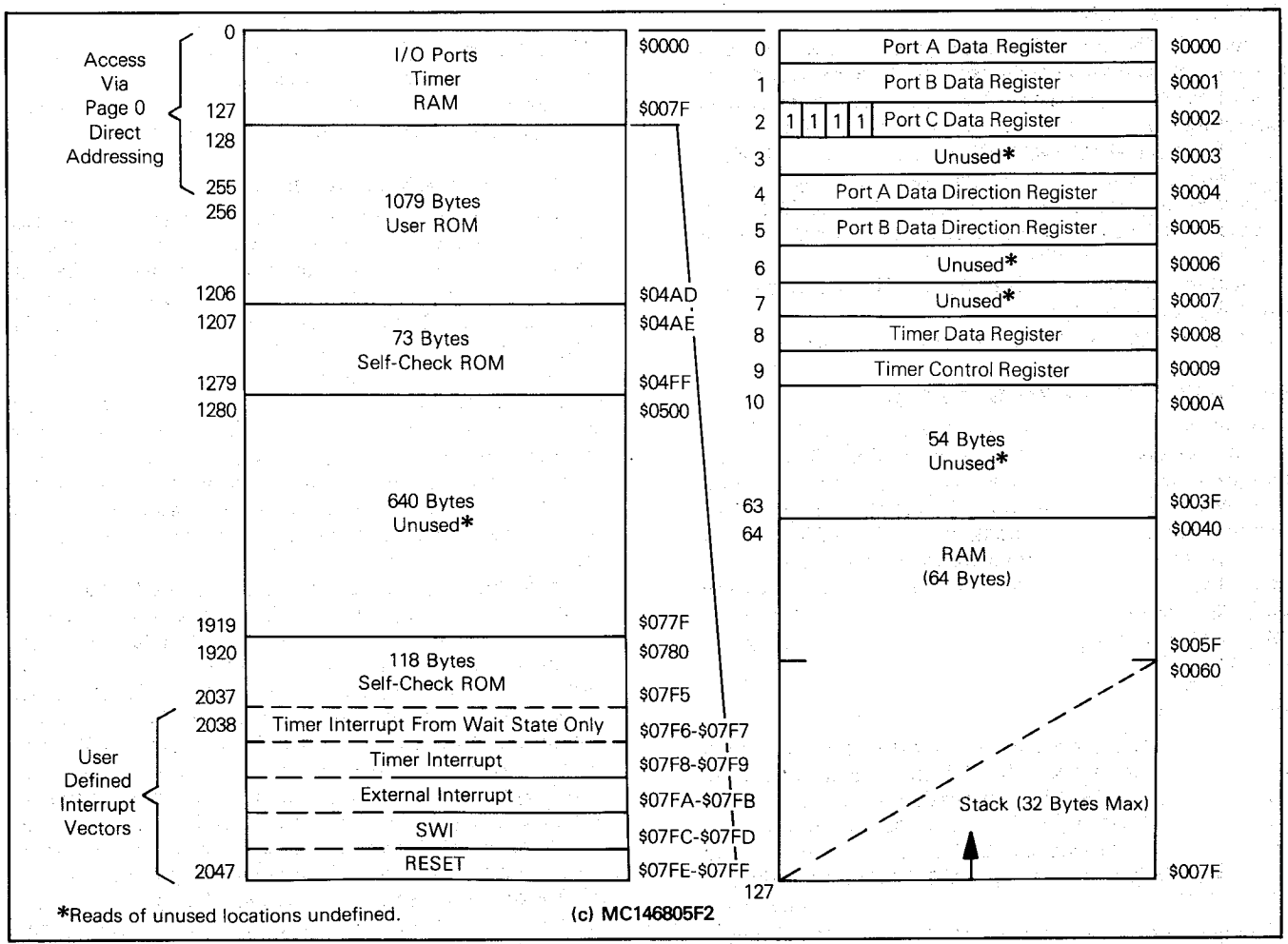


FIGURE 1 — Address Maps (Concluded)

TABLE 1 — M146805 CMOS Family Comparison

	MC146805E2	MC146805G2	MC146805F2
Pins	40	40	28
ROM (Bytes)	External Bus	2K	1K
RAM (Bytes)	112	112	64
I/O Lines	16	32	16 I/O, 4 Input
I/O Drive	TTL	TTL, LED	TTL
Interrupt:			
Edge-Sensitive	Yes	Yes	Yes
Level-Sensitive	Yes	Optional	Optional
Oscillator	÷5	÷4 or ÷2	÷4 or ÷2

I/O Emulation

The MC146805E2, with 16 I/O lines, requires 16 additional I/O lines to emulate the MC146805G2 and four additional input lines to emulate the MC146805F2. As shown in Figure 1, the register locations for these additional lines are available as external address space on the MC146805E2. By mapping a PIA or high-impedance buffer into those address spaces, the additional I/O can be accessed exactly as the duplicated MCU I/O.

The MC146805E2 I/O lines are all configured to drive one LSTTL load; however, the MC146805G2 has additional output drive on half of its 32 I/O lines. The additional drive, if necessary, can be duplicated by adding drivers on the desired lines.

Interrupt Emulation

External interrupts provided on the MC146805E2 are both edge-sensitive and level-sensitive. The external interrupt provided on the MC146805G2 and MC146805F2 are mask programmable as: (1) either edge-sensitive only or (2) both edge-sensitive or level-sensitive. If the interrupt line is to be configured as edge-sensitive only, then the circuit in Figure 2 must be used.

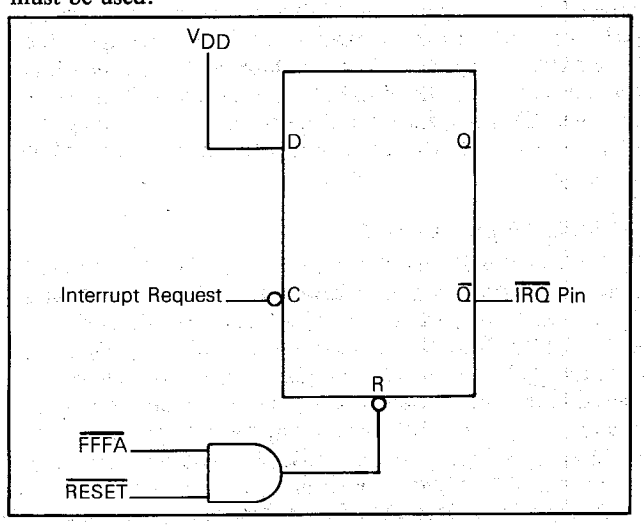


FIGURE 2 — Edge-Sensitive Interrupt Circuit, Schematic Diagram

Oscillator Emulation

The MC146805E2 oscillator produces instruction cycles that are a divide-by-five of the oscillator frequency. A divide-by-five oscillator was chosen because of its convenience in generating the MC146805E2 multiplexed bus signals. The MC146805G2 and MC146805F2 use either a divide-by-four or a divide-by-two of the oscillator frequency to generate its instruction cycle time. Therefore, the emulator oscillator frequency must be run as either 1.25 or 2.5 times the desired MCU target system oscillator frequency.

Example: Target system is to use 1 MHz oscillator with a divide-by-four mask option.

(1 MHz) × (1.25) = 1.25 MHz emulator oscillator frequency

MC146805G2 EMULATOR EXAMPLE

A very simple yet useful emulator for the MC146805G2 is shown in Figure 3. In this example, an MCM2716 EPROM is used for program storage and a 6522 (PIC) is used for the additional 16 I/O lines; therefore, a 5 V supply is required. The example assumes that the additional MC146805G2 output drive is not necessary, the oscillator for the target system is to be configured as divide-by-four, and the interrupts are to be both edge-sensitive and level-sensitive. Such a system can be built quickly and inexpensively and allows debugging using a logic analyzer.

ROM Emulation

Three MCM2716s are used for program storage. Note in Figure 4 that the first MCM2716 EPROM is mapped from $0080-$07FF. The first 128 locations $0000-$007F are excluded since only RAM and I/O reside at those locations in the MC146805G2. The second MCM2716 is located from $0800-$0FFF. Only locations $0800-$08AF are available for MC146805G2 program storage. Locations $08B0-$08FF are reserved for the MC146805G2 self-check routines and should not be used. Locations $0900-$0FFF, although available on the emulator, are not available on the MC146805G2. The third MCM2716 is mapped from $1800-$1FFF. Addresses $1800-$1F7F are not available on the MC146805G2. Locations $1F80-$1FF5 are reserved for self-check and locations $1FF6-$1FFF contain the MC146805G2 vector addresses.

I/O Emulation

The 6522 PIA contains data and data direction registers that are functionally identical to those in the MC146805G2 except for output drive. The 6522 PIA registers can be mapped into the same locations as the corresponding MC146805G2 registers. The MC146805E2 then provides ports A and B and the 6522 PIA provides ports C and D. A complete CMOS system could be formed by replacing the 6522 with the MC146823 CMOS PIA adding and CMOS memory.

MC146805F2 EMULATOR

The MC146805F2 emulator is similar to the MC146805G2 emulator. The differences include: (1) the MC146805F2 has a 2K byte address space instead of 8K as in the MC146805E2 and MC146805G2; (2) the example, as shown in Figure 5, decodes the MC146805E2 addresses to allow the vectors to appear in the same locations as for the MC146805F2; and (3) in addition to the 16 I/O lines on the MC146805E2, four input lines are required for MC146805F2 emulation. The four additional MC146805F2 inputs are not latched and can be read on the MC146805E2 bus via a three-state buffer such as the 74LS244 shown in Figure 5. The unused 74LS244 inputs (bits 4-7: pins 11, 13, 15, 17) should be tied high since that is how the MC146805F2 functions. The user should exercise caution when using RAM since the emulator has more user and stack RAM available than the MC146805F2.

ADDITIONAL AIDS AND HINTS

There are few ways in which the simple stand-alone emulator circuits for the MC146805G2 and MC146805F2 could be expanded and improved. As mentioned earlier, all of the differences between the MC146805E2 and the MCUs could be resolved. Additional drive could be added to the outputs. The interrupt circuit of Figure 2 could be added for the edge-sensitive only option.

As discussed above, RAM could be used in place of the MCU program storage and shared with a separate processor to allow quick downloading of programs. The second processor could also perform additional debug duties such as are found in most debug-type monitors. Utilities such as memory and register examine/change, breakpoints, and single-stepping could be included.

Single-stepping could be accomplished in different ways. First, a timer could generate an interrupt during the execution of an instruction which would cause the interrupt service routine to be called. This method allows an instruction-by-instruction type of single-stepping to be implemented. At other times it might be useful to examine what is occurring within an instruction (bus) cycle. The M146805 CMOS family members are all completely static and can be clocked as slowly as dc; therefore, the emulator clock can be stopped at any time so that the MC146805E2 bus state could be examined. Since the MC146805E2 uses a divide-by-five oscillator, instruction bus cycles would have to be stepped off in groups of five oscillator cycles. The MC146805E2 data sheet contains the relationship between the oscillator frequency and MC146805E2 bus signals.

The above debugging aids could be added to provide the user with a versatile emulator; however, if the user has an MC146805E2 USE model available, the simple stand-alone emulator examples could be used. By removing the EPROM from the emulator, the MC146805E2 USE module capabilities could be employed to replace the MC146805E2 in the emulator. Restrictions on the MC146805E2 would obviously apply to the MC146805E2 USE module. The MC146805E2 USE module would have to be patched to allow the locations required for the additional I/O to be read. The patch for DEBUG05 is listed in Table 2.

TABLE 2 — Patch for DEBUG05

a. Patch to 6800 Version of DEBUG05 Rev. 1.10 or Rev. 1.11				b. Patch to 6809 Version of DEBUG05 Rev. 1.10 or Rev. 1.11			
25A9	7E	2F	4D	25D1	73	2F	E4
2F4D	CD	00	00	2FE4	8E	00	00
2F50	FF	2A	30	2FE7	BF	2A	99
2F53	FF	29	BF	2FEA	BF	2A	1E
2F56	86	0D		2FED	86	0D	
2F58	B7	29	BD	2FEF	B7	2A	1C
2F5B	BD	29	71	2FF2	BD	29	C7
2F5E	CD	00	08	2FF5	8E	00	08
2F61	7E	25	AC	2FF8	7E	25	D4

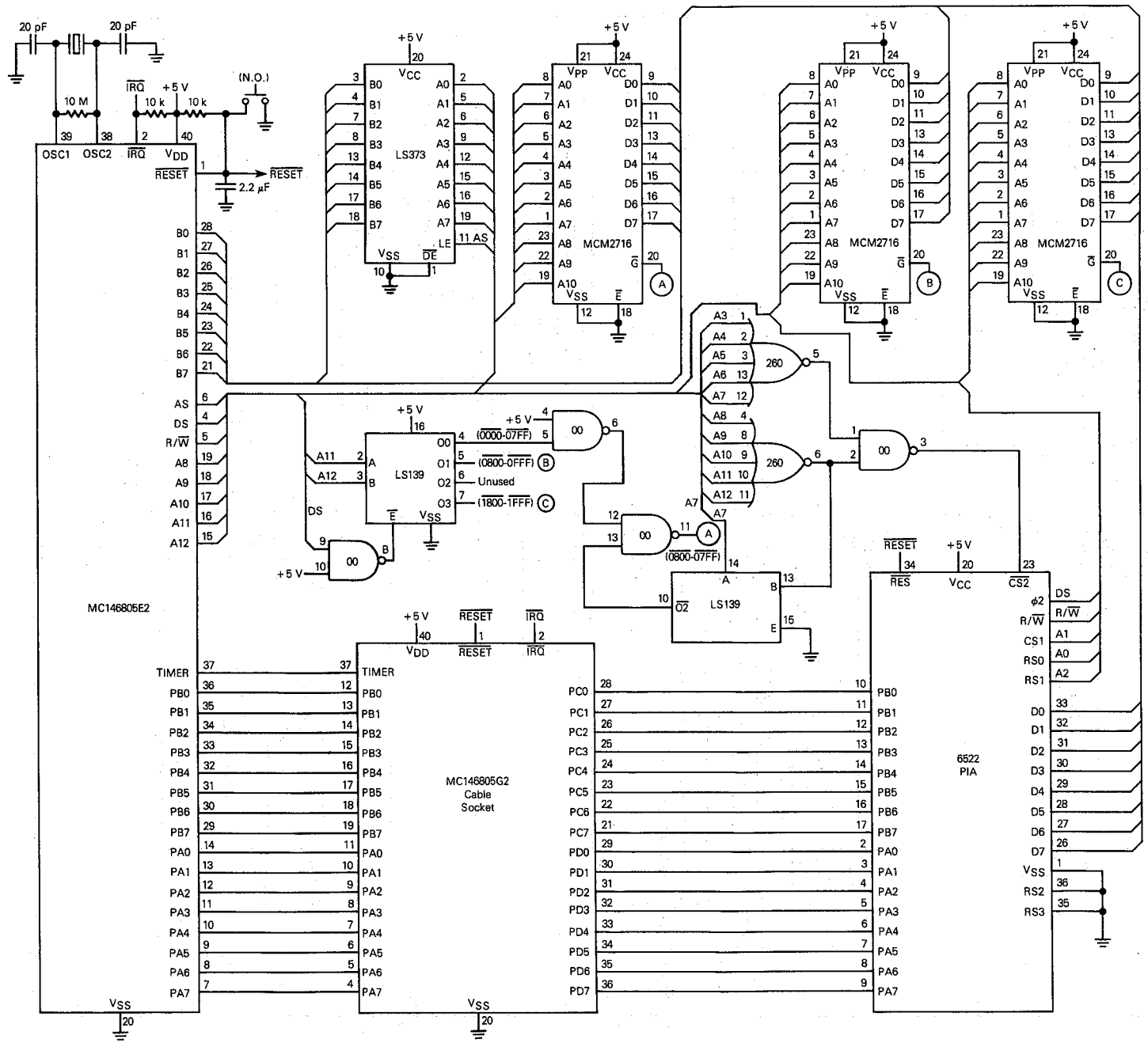
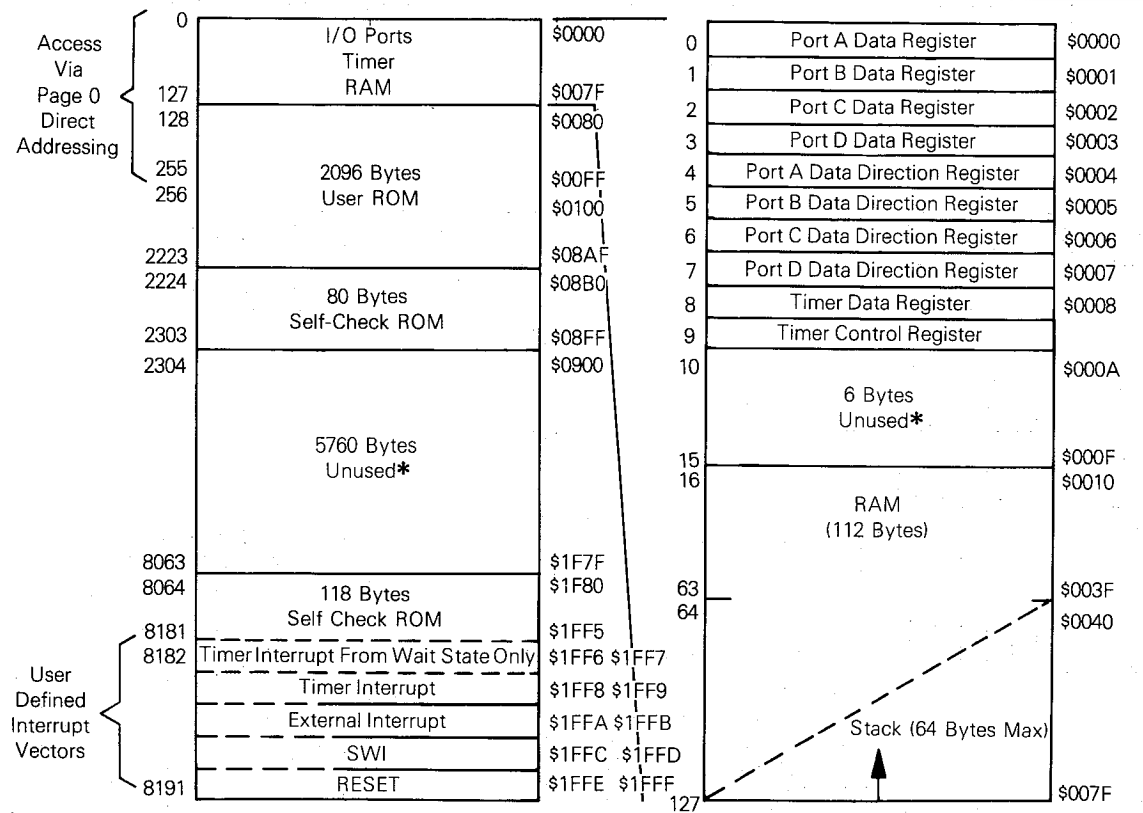
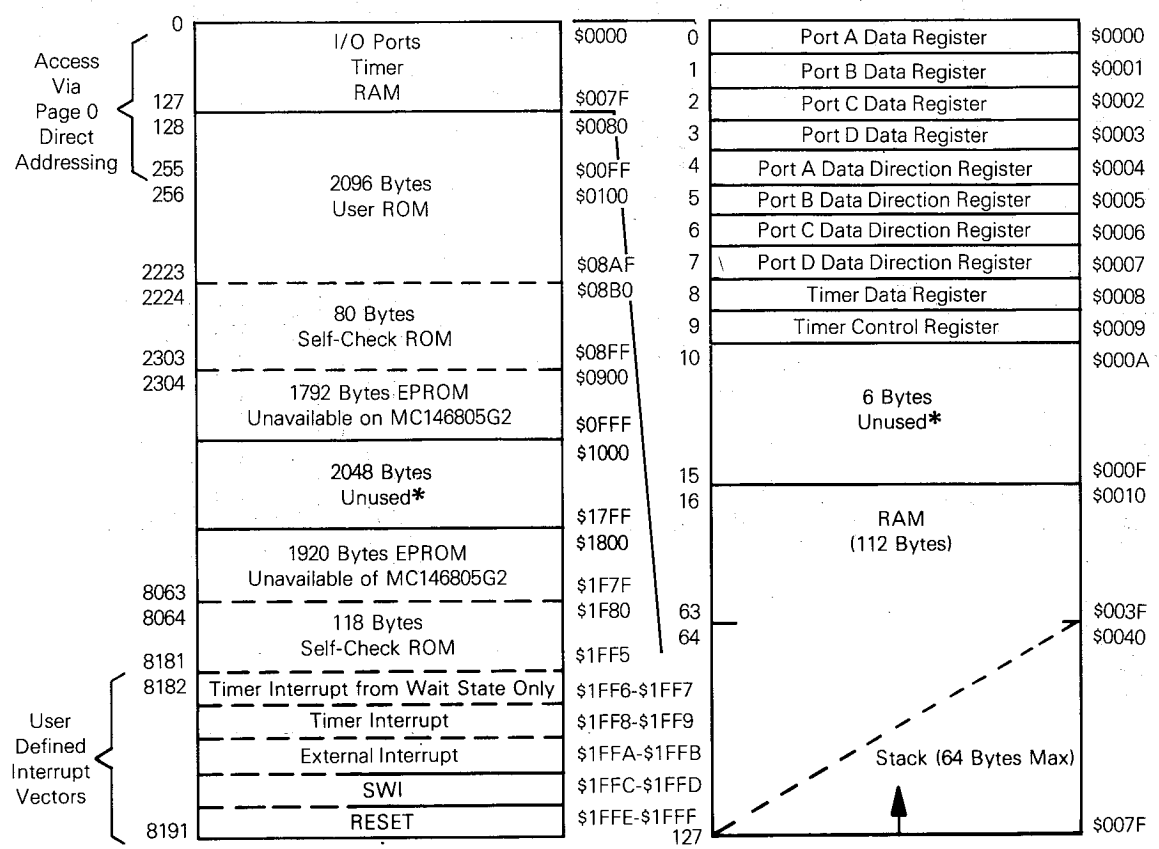


FIGURE 3 — MC146805G2 Emulator Schematic Diagram



*Reads of unused locations undefined.

(a) MC146805G2



*Reads of unused locations undefined.

(b) MC146805G2 Emulator

FIGURE 4 — MC146805G2 and MC146805G2 Emulator Address Maps

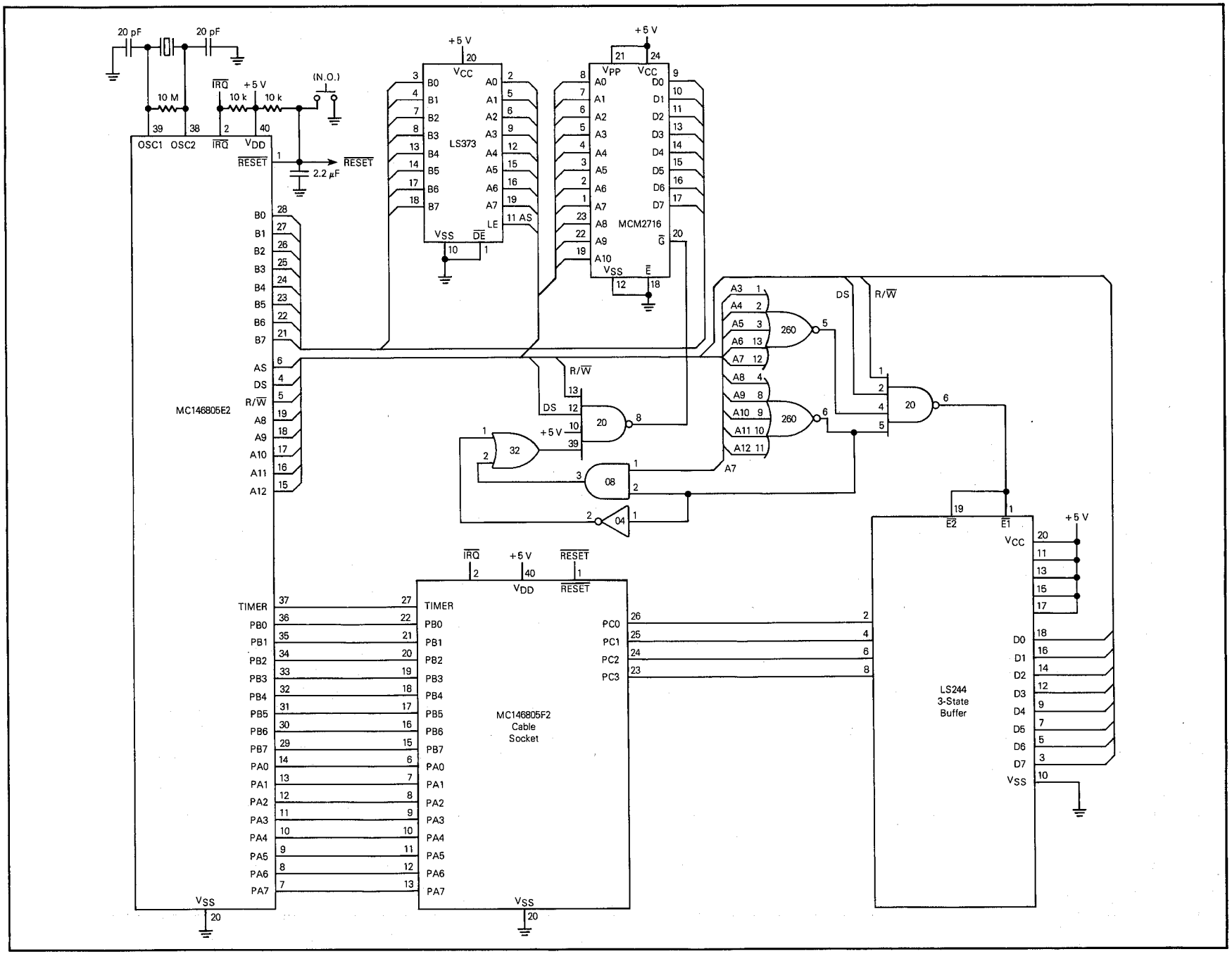


FIGURE 5 — MC146805F2 Emulator Schematic Diagram

MOTOROLA ***Semiconductor Products Inc.***

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.

14960 PRINTED IN USA 1-82 IMPERIAL LITHO C02531 18,000

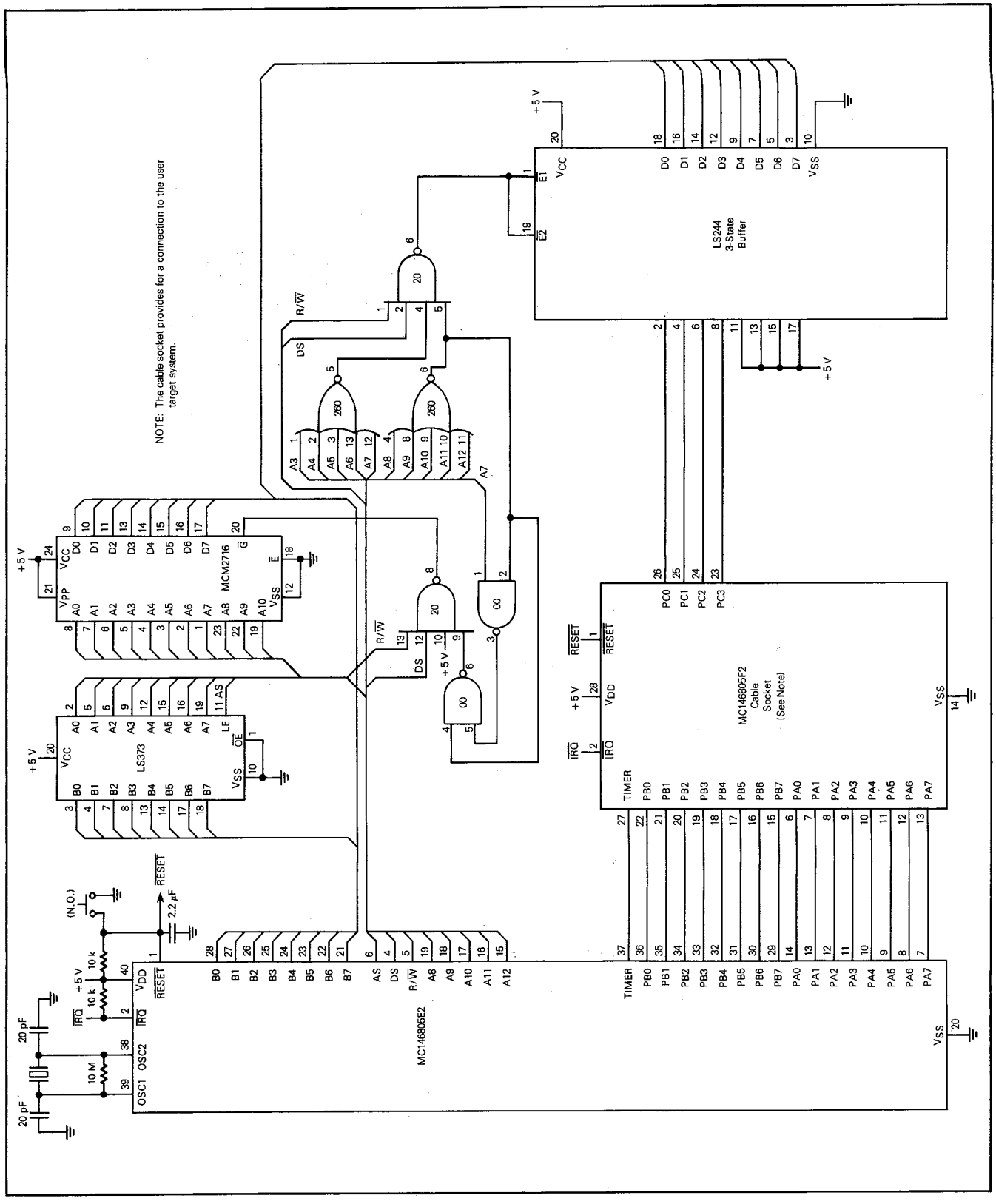
AN853

ERRATA TO APPLICATION NOTE AN-853 M146805 CMOS FAMILY EMULATORS

An error was discovered in the b portion of **TABLE 2—Patch for DEBUG05** of Application Note AN-853. A new Table 2 is provided below. Also, the address decode circuit of the MCM2716 in Figure 5 of AN-853 has been modified. A new MC146805F2 Emulator Schematic Diagram (for Figure 5) is shown on the reverse side of this sheet. In Figure 3 of AN-853, the decoded address at A is shown as (0800-07FF); however, it should be (0080-07FF).

TABLE 2 — Patch for DEBUG05

a. Patch to 6800 Version of DEBUG05 Rev. 1.10 or Rev. 1.11				b. Patch to 6809 Version of DEBUG05 Rev. 1.10 or Rev. 1.11			
25A9	7E	2F	4D	25D1	7E	2F	E4
2F4D	CD	00	00	2FE4	8E	00	00
2F50	FF	2A	30	2FE7	BF	2A	99
2F53	FF	29	BF	2FEA	BF	2A	1E
2F56	86	0D		2FED	86	0D	
2F58	B7	29	BD	2FEF	B7	2A	1C
2F5B	BD	29	71	2FF2	BD	29	C7
2F5E	CD	00	08	2FF5	8E	00	08
2F61	7E	25	AC	2FF8	7E	25	D4



MC146805F2 Emulator Schematic Diagram

MOTOROLA *Semiconductor Products Inc.*

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.

A14960-1 PRINTED IN USA 6/82 IMPERIAL LITHO C5980 15,000 AN-853