ABSTRACT

Many have felt that complementary metal-oxide silicon (CMOS) has not yet become a practical semiconductor technology for microprocessor-based systems. Recent progress has made that impression obsolete. A selection of CMOS microprocessors is available at speeds matching N-channel metal-oxide silicon microprocessor units (NMOS MPUs). CMOS memories have also become broadly available in the last few years. The needed peripheral circuits are now appearing. A CMOS parallel interface peripheral provides 24 interface pins and is bus-compatible with practically all the new-generation CMOS microprocessors. The last element needed to assemble practical all-CMOS microprocessor systems are the small-scale integration/medium-scale integration (SSI/MSI) logic functions. Gates, decoders, latches, and flip-flops are typically needed to operate a bus structure of a multichip system.

This report concentrates on the newest methods of achieving a full-performance all-CMOS microprocessor system. The focus is on the parallel interface peripheral and on using CMOS logic functions in practical bus connections.
PRACTICAL CMOS MICROPROCESSOR SYSTEMS

CMOS, as a semiconductor technology, has for years had a series of recognized benefits. Microprocessors have of course created whole arrays of new electronic uses as well as reconfiguring many conventional electronic products. But until now, combining the CMOS traits with the proliferation of microprocessors has occurred in only a small percentage of the applications. Most of the reasons for the slow acceptance of CMOS as a practical microprocessor technology have now dissipated.

CMOS AS A PRACTICAL MICROPROCESSOR TECHNOLOGY

Most of the attraction of CMOS is associated in some way with battery powering or power saving. There are other CMOS benefits—better noise immunity is a key one—but most CMOS microprocessor applications use batteries for primary or backup power. Some use other low-power energy sources, such as solar cells or very large capacitors.

There is a long-standing impression that CMOS is too slow for many microprocessor uses. The CMOS-is-too-slow image is no longer valid. Metal gate MOS, whether single-channel (NMOS or P-channel MOS [PMOS]) or complementary, is much slower than silicon gate MOS. Most of the high volume MOS processes today are silicon gate, which have the same throughput capability in N-channel (NMOS and high-density N-channel [HMOS]) as in CMOS, given the same device sizes. However, many CMOS users intentionally slow the system down to extend battery operating time.

Some prospective CMOS microprocessor users may have hesitated because of a narrow choice of processors. Until a year or so ago, only two processors were available. Some considered the architectures difficult to accept when compared to the many familiar 6800 and 8080 types of processors available in NMOS. Now, in addition to the traditional (such as the 1802), users have 8080 derivatives (NSC-800 and 80C35) and 6800 derivatives (the MC146805E2) to choose from. Higher-performance CMOS processors are rumored in both traditional NMOS camps. Hesitating to use CMOS microprocessors because of limited architectural choices is outmoded.

Another potential source of hesitation is the fear that CMOS microprocessors are produced only in low volume and thus will always be high-priced. It surprises some to learn that the 1802 is Number 5 in production microprocessor volume, according to Dataquest, behind only the 8080, 6800, Z80, and 6502. CMOS is attractive in certain volume automotive situations. Some of the newer CMOS microprocessors are part of a family of single-chip microcomputers. Read-only-memory (ROM)-based single-chips are built for dedicated volume applications. The ROM-less MPUs benefit from the volume-driven learning curve of the single-chips when they use the same processor and production process. There are volume applications in a number of fields for 8-bit CMOS single-chip microcomputers. Production volume allows costs to be lowered, which should reduce any hesitancy to consider CMOS a practical microprocessor technology.

PRACTICAL SYSTEM NEEDS

Assembling a multichip all-CMOS microprocessor system is now practical. The various elements of such a system are considered in turn, including the microprocessor, memory, peripherals, and interconnecting glue (SSI/MSI logic functions).

A typical mid-range CMOS microprocessor is the MC146805E2. The instruction set is a control-optimized derivative of the MC6800, including single-instruction bit modify and test and low-power stand-by instructions. The interface bus connects to external memory and peripherals using address-then-data multiplexing, as on many newer N-channel processors. Included are 112 bytes of on-chip RAM for stack and data storage. A 15-stage counter is used for timer functions, such as periodic interrupt generation, pulse width measurement, and event counting. Sixteen bidirectional I/O pins are addressable as individual bits or as 8-bit ports. In smaller systems the only external element needed is a program ROM or electronically programmable ROM (EPROM).

The second set of elements for an all-CMOS system is memories. Bus-compatible ROM is available with the MCM65516, which contains 2K bytes of mask ROM in a compact 18-pin package. The multiplexed bus is compatible with the MC146805E2, as well as the NSC-800 and 80C35 microprocessors. Nonmultiplexed bus EPROMs such as 27C16s are now available for program storage in lower-volume applications. One source offers an address-latched version called the 67C16 for use with multiplexed bus microprocessors. For data storage 4K CMOS static RAMs have been available for some time in industry standard 4K x 1 and 1K x 4 configurations. The availability of 16K static CMOS RAMs may improve soon. Both bus-compatible and industry-generic CMOS memories are available for microprocessors. It is beyond the scope of this report to survey memories deeper.

The third major element of an all-CMOS microprocessor system is peripherals. Fortunately, CMOS users do not need as many peripheral integrated circuits (ICs) as NMOS MPU users. CMOS MPUs are seldom used with mechanical and
electrical devices that consume large amounts of power, such as floppy disks and cathode ray tubes (CRTs).

The most basic interface element is parallel-port input/output (I/O) connections. The MC146823 provides three 8-bit parallel interface ports along with handshaking port control signals in a bus interface peripheral. The latter portion of this report focuses on this parallel interface peripheral, since its bus interface allows direct connection to all new-generation CMOS microprocessors announced to date.

A frequent function of a CMOS MPU system is to keep the time of day, and often a calendar as well. The MC146818 Real-Time Clock Plus RAM maintains the time in seconds, minutes, and hours. It maintains a 100-year calendar, including day of the week, leap year, and daylight-saving changes. Some of the auxiliary system functions included are 50 bytes of uncommitted RAM, a periodic interrupt, an alarm interrupt, a square-wave output pin, and a microprocessor clock oscillator.

Other peripheral functions are available in the market. Generic asynchronous universal receiver and transmitters (UARTs) have been available for some time from two or three sources. Though not directly bus-compatible, they are easily interfaced. The RCA 1802 family includes some peripheral functions that could be useful with other CMOS processors. Some examples are a multifunction timer and an arithmetic chip. A little interface adapting is needed to use such parts with the MC146805E2 type of buses, but there are situations where it would be worthwhile. The two peripherals introduced with the NSC-800 are usable on other processors with very little adapting. The RAM plus I/O part is useful in low-volume cases where the needed memory-to-I/O ratio is close to that included in the part.

The fourth major element needed to assemble an all-CMOS system is the interconnect SSI and MSI logic. Few complex systems can be assembled totally with large-scale integrated (LSI) circuits. Until recently the CMOS standard logic functions have been slow; this trait frustrated attempts to take advantage of the available microprocessor performance. Full-speed "glue" parts are now becoming available in the 74HC00 family from Motorola and National. This line includes all the popular 74LS00 family functions, at the same speed as the low-power Schottky transistor transistor logic (LS-TTL) family, but with CMOS power usage.

The next section of this report looks at some of the bus interface uses for the CMOS SSI glue parts in the 74HC00 family.

**BUS INTERFACING**

The practicalities of putting together an all-CMOS MPU system are, of course, applications-dependent. This section outlines a few techniques that might be useful. The goal is to trigger user creativity with ideas, not to establish a standard way to interface to a microprocessor. For example, some applications need more memory than the program can directly access, so memory expansion techniques are appropriate. Many generic memories cannot accept the MPU bus control signal formats provided.

---

**Bus Control Signals**

Frequently the bus control signals that emanate from a microprocessor need to be modified for use by memories and peripherals. Figure 1 shows that the MC146805E2 microprocessor creates an Address Strobe (AS), a Data Strobe (DS), and a Read/Write level. Figure 2 shows how these three control signals are associated with read and write bus cycles. The memory and I/O cycles are identical, since a common address architecture is used. Figure 3 shows how three new control signals may be generated. Generic memories (industry standard Joint Electronic Device Engineering Council (JEDEC) pin functions, as opposed to directly bus-compatible) usually require a low-going read pulse, frequently called output enable. RAM writes are indicated by a low-going write pulse. Many peripherals accept similar signals. Figure 3 shows that two gates and an inverter create the needed read and write pulses.

Static memories are not always fully static today. Though memory retention may be static, the internal decoders and buses may need to be cleanly waked up to initiate a successful bus cycle. Careful reading of data sheets reveals that chip enable inputs can no longer accept address decoding transitions. Once chip enable is asserted, it must remain, without bounce, until the access cycle is complete. The Data Strobe microprocessor signal is a clock with clean edges that can be ANDED with address decoding to create a clean chip enable. The problem is that fast memories are then needed, since DS begins quite a while after the addresses are stable. Figure 3 shows how an earlier chip enable strobe can be generated using less than one-and-a-half packages of SSI logic. Figure 2 shows that the Enable Strobe (ES) begins with the falling edge of Address Strobe and lasts until the end of Data Strobe. The address is stable at the leading edge of ES, and the un-multiplexed address remains stable for the duration of ES.

---

![Figure 1—CMOS microprocessor bus interface adaptations](image-url)
Address Expansion

The second step in generating a larger system is to create an extended unmultiplexed address bus. Figure 4 shows a typical example. Most new-generation microprocessors time-multiplex portions of the address onto the data bus. Address bits appear on the bus during the first part of the bus cycle and are identified with an Address Strobe signal (Address Latch Enable [ALE] in other processors). Then, after the memory begins its access time, the bus is switched over to carry data during the latter portion of the bus cycle. Data Strobe identifies the data portion of the bus cycle (Read and Write with other processors). Figure 2 shows the time-multiplexing relationships.

An increasing percentage of available memories and peripherals are including address latches. Some accept an AS

beginning the address range with pages.

Figure 2—Auxiliary bus control signal timing

Figure 3—Creating auxiliary bus control signals

Figure 4—Extended unmultiplexed address bus
(ALE) signal directly, whereas others need an ES signal as a part of a chip enable input. Since many memories still need unmultiplexed addresses, Figure 4 shows an octal latch to save the low-order eight address bits.

The address expansion technique illustrated in Figure 4 uses one MSI part, a dual one-of-four selector, and six port bits to add two bits to the 13 address bits created by the microprocessor. Figure 5 shows the resulting page-addressing arrangement. A program sees 8K of logical address space split into four 2K byte pages. A common convention would be for the interrupt and other I/O routines to be in the fourth page, along with a centralized page-changing routine. The remaining three 2K byte pages could freely include programs and data in whatever mix was appropriate. Each of the three logical user pages can be mapped to one of four physical pages of 2K bytes. The physical address space would include 12 pages of 2K bytes each (24K bytes total), of which only 6K bytes are visible at any time. When a program needs data that are not visible, it calls the executive in the fourth page (probably with a software interrupt instruction) to have the pages switched. The executive can switch pages via the I/O port without losing control, since the fourth page is reserved as always visible.

The address expansion scheme outlined above is an example of one of the possible techniques. Figure 6 shows that other variations of the method can allow addressing to over a quarter of a million bytes by using only three glue parts and only the 16 I/O pins included on the MC 146805E2 processor. Other expansion techniques could be as easily used, but they are beyond the scope (or space) of this report. The processor program may only have a 13-bit address, but it can access as much memory as is needed.

Typical Expanded System

It has been shown that generalized bus control signals can be easily created and that an unmultiplexed address bus of 15

<table>
<thead>
<tr>
<th>TO OBTAIN THIS TOTAL ADDRESS ABILITY</th>
<th>ONLY THESE GLUE PARTS ARE NEEDED</th>
<th>USING THESE MC146805E2 PORT PINS</th>
<th>THE ADDRESS BUS SIZE IS</th>
<th>THE SIZE OF THE LOGICAL PAGE IS</th>
<th>THE NUMBER OF LOGICAL PAGES IS</th>
</tr>
</thead>
<tbody>
<tr>
<td>8K</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>8K 1</td>
</tr>
<tr>
<td>16K</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>14</td>
<td>1K 8</td>
</tr>
<tr>
<td>29K</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>15</td>
<td>2K 4</td>
</tr>
<tr>
<td>44K</td>
<td>3</td>
<td>0</td>
<td>14</td>
<td>15</td>
<td>1K 8</td>
</tr>
<tr>
<td>290K</td>
<td>0</td>
<td>3</td>
<td>16</td>
<td>16</td>
<td>1K 8</td>
</tr>
<tr>
<td>26K</td>
<td>0</td>
<td>3</td>
<td>16</td>
<td>19</td>
<td>2K 4</td>
</tr>
</tbody>
</table>

Figure 6—Various extended memory sizes

Figure 8 shows how the above address map could be easily decoded with available glue parts. Three-to-eight decoders are shown to decode chip enable signals for the RAM, ROM/EPROM, and the multiple parallel interfaces. NAND gate address range detection is shown for the peripherals. The fourth page ROM is disabled when the peripherals are accessed. In systems where demultiplexing is not needed, the decoders could include an address latch—the MC74HC137, for example.

Bus Interface Flexibility

The typical extended system above includes one microprocessor, nine LSI peripherals, and 41 memory parts. Only 10 SSI/MSI glue parts are needed to assemble a practical
system with an impressive set of features. The processor is a powerful 8-bit control-oriented MPU. Interfacing to 192 interface pins is provided, of which up to 32 could be interrupt sources. The memory includes 16K bytes of RAM and 10K bytes of EPROM or ROM. In addition, the time of day is automatically maintained in a peripheral.

With all that system power, it is not valid to consider an advanced larger all-CMOS microprocessor system to be impractical or too expensive.

PARALLEL INTERFACE PERIPHERAL

CMOS microprocessors such as the MC146805E2 include some parallel I/O pins on the MPU part. When the on-chip I/O is insufficient, a parallel I/O peripheral part is needed. The MC146823 provides 24 I/O pins to an MC146805E2, NSC-800, 80C35, or 80C48.

Figure 9 shows the 8-bit bus interface on the left and the 24 parallel I/O pins on the right. The I/O ports are looked at first, followed by the handshake functions and the generalized processor bus interface.

**Three Parallel Ports**

The processor program establishes the purpose of the 24 parallel I/O pins. Figure 10 illustrates the program selection features.

Each 8-bit port includes a data direction register. With power-on reset the data direction is initialized so that all pins are inputs. All output drivers are in the high impedance state to avoid a situation in which two circuits try to drive the same pin to opposite states. After power-on, the program establishes each pin as an input or an output. Each data direction register bit establishes the corresponding port pin as an output or an input. There are no restrictions on the number of input or output pins, nor on which pins of an 8-bit port are inputs or outputs.

The data register associated with each port is used primarily for output storage. When the data direction register bit indicates output, the state of the corresponding data register is driven onto the I/O pin by the output buffers. When a pin is designated as an output, a program write transfers the state of the I/O pin to the data register. For output bits the data register is a read/write register. A program read of an output pin gets the state of the data register, not the I/O pin. This permits read/modify/write cycles, such as the MC146805E2 bit manipulation instructions, to read the port, change one or more bits, and write the result back to the port data register.

The data register in Port A has one additional use. A handshaking pin causes input data to be latched for subsequent program reading when this feature is enabled by the program.

Four of the pins on Port C may be either handshake control signals or Port C parallel pins. Port C thus has a pin function select register that allows the processor program to establish which pins serve handshaking rather than parallel I/O purposes. Many system applications need only parallel interface pins and can thus disable the handshake features.

**Port Handshake Control**

The four handshaking control signals on Port C may serve the following functions:

A. Digital inputs
B. Digital outputs
C. Interrupt inputs
D. A latch enable for Port A input data
E. An output pulse when Port A to read
F. An output pulse when Port B is written
Figure 11—Port handshaking modes
G. Request and response signals for Port A input data
H. Request and response signals for Port B output data

A series of compound combinations of the above functions are useful. For example, Items D, C, and E allow an external source to write a byte into the MC146823, which initiates an interrupt; then, when the program reads the data, a pulse is returned to the external source. Figure 11 shows the eight handshaking functions graphically. The digital input and output modes are the Port C usage as already discussed.

Part C of Figure 11 shows that all four handshake pins may initiate interrupts. Each interrupt is separately enabled; control bits are provided by the program in control registers and separately identified in a status register available to the program. Interrupt overrun is also separately indicated in a status register for consecutive interrupts that are not serviced fast enough. The four interrupt functions are ORed together onto the one output interrupt pin to the processor.

Function D in Figure 11 allows an externally provided signal edge to latch input data into Port A. This is a convenient way to accept asynchronous data bytes from a serial I/O, another processor, a mechanical peripheral, or other parts.

In Parts E and F of Figure 11, a program read or write causes the Parallel Interface to send a pulse one bus cycle wide. In the case of a Port A read, the output pulse is a response signal. The output pulse with a Port B write allows a byte of data to be latched into external hardware.

Closed-loop handshaking is provided with Functions G and H. One input and one output handshake control pin is associated with input data on Port A and two separate pins handshake with Port B output data. One signal requests data flow with an edge; the other signal responds with an edge on the other pin. One use for handshaking interlinking like this is to interconnect two processors.

There is not enough space in this report to look at all the useful combinations of the eight modes outlined above.

**Bus Interface**

The bus interface consists of eight bidirectional bus pins and four input control pins.

During the first portion of the bus cycle the 8-bit bus includes four address bits to select one of the 15 addressable MC146823 register locations. During the latter portion of the bus cycle, the processor provides a write data byte or the peripheral provides a read data byte.

A Chip Enable (CE) input pin tells the peripheral to accept or ignore the current bus cycle. As such, CE must be true after the address is stable and remain until the data is transferred. The Address Strobe pin allows the address on the bus to be latched within the peripheral.

Figure 12 shows the above bus functions as well as the two interpretations of the other two bus control pins. The two logical interpretations are called the MOTEL concept (for Motorola and IntEL compatible). This allows direct connection to processors, creating control signals in either de facto bus standard.

In the Motorola MOTEL mode the DS input is a positive pulse during the data portion of each bus cycle. The R/W pin indicates during the DS pulse whether a read or a write cycle is in progress. The other MOTEL interpretation is a low-going read pulse on the DS pin and a low-going write pulse on the R/W pin. The peripheral automatically decides which interpretation to use by sampling the DS pin at the time of the AS pulse.

**Used With Any CMOS Microprocessor**

The MOTEL bus interface concept allows a peripheral or memory IC to interface directly with any new-generation multiplexed bus microprocessor. The MC146823 peripheral was designed for use with the MC146805E2 processor. But the MOTEL concept allows it to also be used with other CMOS microprocessor and expandable single-chip microcomputers. Figure 13 shows the interface on a Motorola type of bus, while

![Figure 12-Bus interface signals](image1)

![Figure 13—Typical processor interface, Motorola bus](image2)

Figure 14 shows the same peripheral directly connected to other processors. No intervening glue is needed to adapt the peripheral to the other processor buses. Universal peripheral
and memory applicability has finally been achieved. (Incidentally, the MC146818 Real-Time Clock Plus RAM and the MCM65516 2K × 8 ROM also use the MOTEL concept.)

![Figure 14—Typical processor interface, other bus](image)

Memory-Mapped Registers

The port hardware functions and the bus interface capabilities of the MC146823 have been reviewed. The remaining element is the program use of the features.

From the program vantage point, the Parallel Interface is a block of 16 addressable locations, of which 15 are used. Figure 15 lists the seven major functions, shows which of the three ports the function is used with, and lists the hexadecimal address of the register within the 16 addressable locations.

![Figure 15—Register functions](image)

The port data registers are read/write locations for each of the three ports. Each port also has a read/write data direction register to establish pins as inputs or outputs. Port C has a 4-bit function select register to individually enable the handshaking function.

Two 5-bit registers establish the handshaking modes for Ports A and B. These bits establish whether the active input edge of the handshake function is a rising or a falling edge. Each of the four interrupts include an enable bit. The input latching, output pulsing, and closed-loop response functions are also established with bits in these registers.

A read-only status register contains flag bits for each of the four interrupt sources. Each flag bit represents a status condition denoting whether the corresponding interrupt has been enabled by the program. When a status flag is set and the corresponding interrupt has been enabled, the interrupt pin is asserted and an interrupt OR bit appears in the status register.

The program clears an interrupt by reading or writing a specific port data byte. When a handshaking byte transfer is to be acknowledged, the data read and write do so automatically. But in many cases, the data port is not directly associated with the interrupt function. In such cases a port read and write could cause a pending interrupt to be lost. Therefore the ports can also be read and written without effecting the interrupts. Three port data addresses are associated with each of the two handshake ports. One address does not affect the interrupt flags. The second address clears the interrupt status associated with one interrupt source as well as performing the data transfer. The third address does the same thing with the second port interrupt. An interrupt can thus be cleared with a pseudo-read of a port. The test (TST) instruction in the MC146805E2 does so without disturbing the accumulator.

The last addressable location is the overrun warning register, which indicates that a previous interrupt had not yet been serviced when a new one appeared. The 4-bit read-only register allows the program to find out that one or more events or data elements have been lost.

Feature-Packed Parallel Interface

Comparisons can be made to the popular N-channel parallel interfaces to see that the MC146823 includes more features than most. Three full 8-bit ports are included in a 40-in package where some others have fewer bits. The program establishes every bit separately as an input or an output, rather than establishing direction by groups of bits. Four separately selected interrupts are included, and the interrupts may be separately cleared. All registers are directly accessed by the program; none are hidden. Many handshake modes are included to allow easy interfacing to existing equipment.

CMOS IS NOW A PRACTICAL MICROPROCESSOR TECHNOLOGY

Many have waited a long while for full-performance microprocessor systems to be implementable entirely in CMOS. The CMOS MPU era has finally arrived. The microprocessors and memories are available. This report has shown that the needed parallel interface peripheral function is now covered, and the gate and flip-flop glue functions needed in larger MPU systems are also now practical in CMOS.