What's in a microcomputer?

Look inside your home computer and you'll find a heck of a lot more than one 8080 integrated-circuit chip. Here's a fast rundown on what's in your black box.

by Pete Stark
Contributing Editor

They call them "the computer on a chip." So, when you see an advertisement in a leading catalog for a "100 percent Prime 8080A 8-Bit Microcomputer Chip" at $17.95 you might be tempted to think that with this integrated circuit and a power supply you will have a complete computer. Unfortunately, it's not that simple.

To make a complete computer with the 8080A or other popular microprocessors you will need quite a few other integrated circuits. Although several manufacturers have in the last few months developed microprocessor IC's which can work almost by themselves, they usually can be used only for special and limited applications. More IC's still have to be added for most uses.

Let's take a look at what we need besides "the chip." Figure 1 shows the typical pin connections for common microprocessor integrated circuits. As the name implies, the microprocessor IC is not a complete computer—it is just the processor portion of a computer system. The processor, or the *Central Processing Unit* (CPU), includes the arithmetic and logic circuits which perform calculations for the computer, as well as the control circuits which govern its operation. It acts as the central controller to which the computer's memory and all input and output devices are connected.

Running pulse

Like all IC's, the microprocessor requires a source of power. Modern microprocessors require only a well-regulated source of +5 volts. The CMOS processors such as Intersil's IM6100 or RCA's Cosmac may require just a few milliamperes, while others may require several hundred milliamperes. Older processors may require 12 volts or other voltages in addition to the +5 volts. The choice of +5 volts, of course, is to make the microprocessors compatible with standard TTL IC's.

Also required is a continuously running pulse signal, called the *clock*, which regulates the internal timing of the entire system. The clock oscillator which generates the clock pulses may be either inside the microprocessor IC or outside. If it is inside, then an external resistor-capacitor circuit or crystal is connected to two of the IC pins to regulate the exact clock frequency. External clock oscillators also use R-C or crystal circuits to generate a pulse wave of a constant frequency. Since crystals are more accurate than R-C circuits, they are more common except in very simple (and inexpensive) systems. Some micropro-

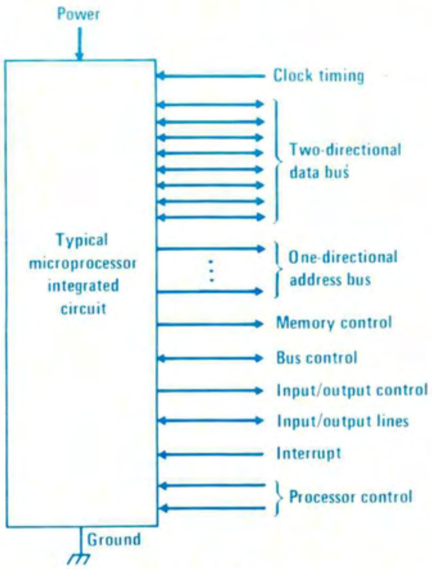


Figure 1 **Typical microprocessor connections**

cessors require two clock signals of the same frequency but carefully staggered so that they alternate pulses on two lines. This is called a two-phase clock and, in some cases, requires the use of several IC's just to generate the proper clock signals.

The data bus and address bus communicate with memory and input-output equipment. Figure 2 shows the connections to a typical computer memory. The term *bus* describes a group of wires which carry several related signals all at one time. For most systems, the data bus consists of eight wires and the address bus consists of sixteen. Rather than show the individual wires, however, most busses are shown as a thick bundle. The data bus, which carries data in both directions (not at the same time) has arrowheads on both ends, while the one-directional address bus has only one arrowhead.

Eight bits

The computer's memory is used to store numbers, usually written as binary numbers using only ones and zeroes. Each of these binary digits is called a *bit*, and each number stored in memory has a fixed number of bits. The most common number length today is eight bits. Since the data bus is used to carry these numbers to and from the memory, it too has eight wires. The eight-bit numbers stored in memory can represent one of three things—numbers used in calculations, letters or punctuation marks coded into binary form, and instructions.

Each number stored in memory is in a separate memory location. The traditional way of describing the organization of a computer memory is to visualize it as a row of post office boxes, each big enough to contain exactly one number and each having an *address*. In most microprocessors, the address is a binary number having 16 bits. Since it is carried over the address bus, the bus also generally has 16 wires.

Memories come in two types—RAM and ROM. The letters RAM stand for Random Access Memory, although Read-Write Memory might have been a better choice. It's a memory which can be written into—meaning that numbers may be put into it by the microprocessor. It also can be read from—meaning that its contents can be sent to the microprocessor. Such a RAM memory can be used to hold instructions and data, as well as intermediate results and answers to problems run by the computer.

The letters ROM stand for Read Only Memory, meaning that numbers can be read out of the memory by the microprocessor but not written. Such a memory can store instructions or long-term data but not results of calculations. At some point, of course, something had to be written into the ROM to be read, but in this case the writing occurs during the manufacture or installation of the ROM.

The set of instructions stored in memory for doing some particular problem is called a *program*. Programs can be stored in either RAM or ROM memory, depending on use. A program which is changed often or which is used only now and then would most likely be in RAM. On the other hand, a program which never changes and is run continuously would best be stored in ROM so it could not accidentally be erased in case of power failure or computer malfunction.

A good example might be the new computerized cash registers used by some of the national hamburger chains. Whenever the clerk pushes the button labelled double hamburger, for instance, the microprocessor inside follows a program stored in ROM to calculate the price and add it to the total. Even in such a system, though, some RAM is needed, both for data involved in one particular sale, as well as to keep totals for daily sales. In the case of the double hamburger, the RAM stores the total number of hamburgers ordered by the customer, total price for those hamburgers, and eventually total price for the entire order. At the end of the day, the RAM also contains somewhere inside the total number of double hamburgers ordered all day and other data needed to total up the day's sales.

Read or write

Several types of memory control lines are used. To keep diagrams simple, the various control lines have short abbreviated names such as R/W or VMA to indicate their function. The R/W line tells the memory whether to read or write. Since the same voltage levels used by TTL integrated circuits are used in most microprocessors, the R/W line takes on one of two voltages: 0 volts or +5 volts. In a typical system, placing the R/W line at +5 volts indicates the microprocessor wants to read from memory, while placing the line at 0 volts indicates it wants to write.

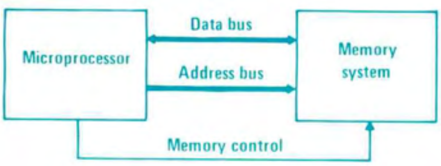


Figure 2 **Connections to memory**

VMA stands for Valid Memory Address, a signal used by the microprocessor to tell the memory that an address has been placed on the address bus and is ready to be used. In addition to these, the memory also may get some timing signals from the microprocessor to indicate the precise instant when to read or write. Sometimes these signals have names such as 02 (which stands for *phase two of the clock* and doesn't mean much to the newcomer). Other times they may have specific names such as MEM SEL for MEMory SELect.

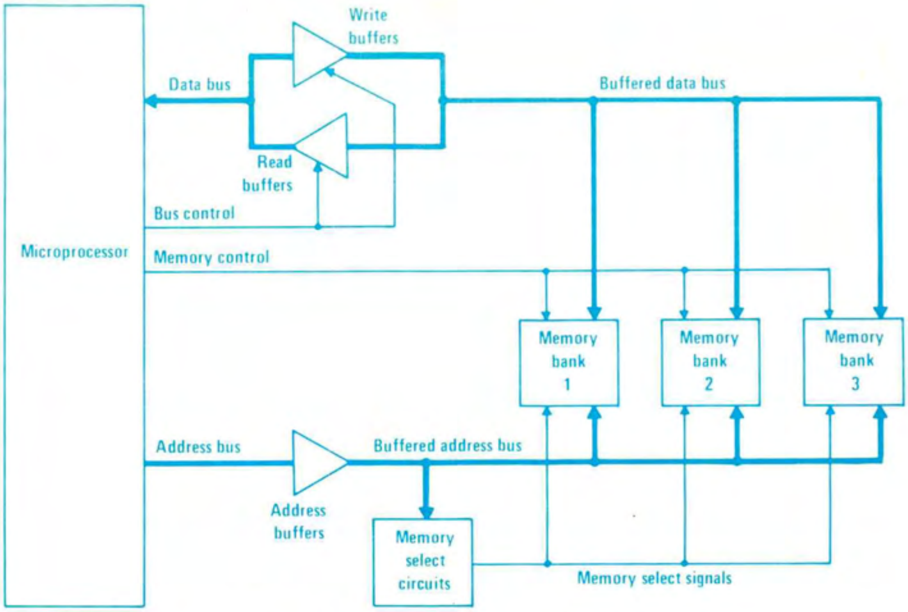


Figure 3 **Memory connections in larger systems**

At this point, let's pause and see just how all these signals work together. Suppose the microprocessor IC wants to read some number out of memory. The particular sequence of signals would work like this:

- The microprocessor sends out the binary address of the desired location, waits a short time to allow the memory circuits to receive it, and then sends the Valid Memory Address signal.
- At the same time, the microprocessor puts the R/W line at the correct voltage level to signify a read.
- A short time later, a clock pulse on the 02 or MEM SEL line appears, and the memory sends the appropriate memory contents to the processor on the data bus. At the end of the 02 or MEM SEL pulse, the processor accepts the number from the data bus.

Data bus

If the memory were just a single IC, then it might be possible to connect that IC directly to the microprocessor. In very simple systems that is exactly the case, since some memory integrated circuits have been designed to be directly compatible with certain microprocessors by connecting directly to the address and data busses and to the memory control lines. In larger systems this is not possible, however, for two reasons. First, to provide larger memory it often is necessary to provide many different memory integrated circuits, with each IC storing just part of the needed memory locations. Additional circuitry then is necessary to select the particular IC which contains the location addressed at any instant. The second reason is that to keep the processor IC small and to reduce its power requirements and heat output, it is necessary to take some short cuts. One of these is to design the output amplifier circuits which feed the busses and other outputs in such a way that they can drive only a limited load. If more than just one or two external IC's need to be connected to any given output, then external amplifiers have to be added to provide enough power. The catch is that some of these, such as those on the data bus, have to amplify in both directions if the line they handle is also two-directional.

As a result, the memory often looks like Figure 3. To keep the diagram simple the memory is shown divided into only three sections, called banks. In an actual system, many memory banks might be used, possibly with some as RAM and some as ROM memory.

Both the data bus and the address bus have amplifiers inserted called *buffers*. Since the address bus is a one-directional bus, only a single set of buffers is needed. That portion of the bus which lies between the microprocessor and the buffers still is called the address bus, whereas the output of the buffers is called the *buffered address bus*. The data bus is also buffered; but since it is a two-directional bus, two sets of buffers are needed—write buffers for data going from the microprocessor to the memory to be written, and read buffers for data being read. To avoid feedback and oscillations, however, only one set of buffers can be turned on at any one time, and so a bus control output from the microprocessor is needed to switch off the set of buffers not being needed at any one time. In most systems, this bus control is handled by the R/W signal, but in some systems which permit several processors to connect to the same data bus (such as National Semiconductor's SC/MP microprocessor) other bus control inputs or outputs may be present.

The buffered address bus is split into two portions, one portion going to all memory banks, the other portion going to memory select circuits which select one of the banks, depending on the actual address. Some microprocessor systems are even more complicated, since part or even all of the address may actually travel over the data bus. Instead of address buffers, a group of flip-flops and control circuits would be needed to catch the address as it is placed on the data bus for an instant, and retain the address for the entire time that it is needed by the memory, even after the data bus is used for other data transfers.

Input/output control signals are used on some microprocessors—those which have specific input and output instructions. This includes the Intel 8008 and 8080, the Intersil 6100, and others. In these units, specific instructions exist which output pulses on these input/output control lines. When a program is supposed to input data from a device such as a keyboard, or output to a device such as a printer, an input/output instruction is placed into the program. This generates a pulse which goes out to all input/output equipment. It is used in conjunction with the address lines, which select a particular input/output device, and the data bus, which carries the data being handled, to perform the input or output operation. Some of the more recent microprocessors such as Motorola's 6800 do not use this idea. Rather than treat input/output equipment as a special case, the 6800 treats input/output equipment as if it was part of memory. In that case, the same memory control lines which control reading and writing of memory also control reading and writing to output or input devices.

IC Pins

Some microprocessors, especially those designed for use in small systems, have one or more input/output lines which may be one-directional or two-directional. These are pins on the microprocessor IC which can be set to either a high or low voltage by the program, or which can be connected to an external device and the voltage state on the line sensed under the control of the program. For very simple uses, these control lines may be all the input/output circuitry that is needed. For example, National Semiconductor's SC/MP microprocessor has seven such lines. In one simple application, this processor can be used as the central control in an electronic lock. The input/output lines are connected to several switches, lights, and a solenoid to unlock the door. Along with a ROM to hold the control program—no RAM is needed for such a simple use—the microprocessor monitors the switches. When a particular sequence of switch closures is detected, it opens the door.

Virtually every microprocessor has one or more interrupt inputs. The interrupt system is designed to allow external devices, such as input/output devices, to interrupt a program being performed by the processor and switch to a different program. A simple example might be a computer which is normally used to handle bookkeeping for a small motel but which is also connected, via the interrupt inputs, to the motel's burglar alarm. Part of the burglar alarm might connect to the television set in each room. If, while the computer is processing the payroll, a guest decides to disconnect the tv set, then an interrupt immediately stops execution of the payroll program and forces the computer to switch to a burglar alarm program which alerts the operator to the room number, looks up the name and address of the guest occupying that room in the computer file, and alerts the police.

There are generally at least, two processor control connections, labelled halt and reset. The halt lead causes the processor to stop executing a program. When the halt signal is removed, the processor simply continues from wherever it had stopped. The reset signal is a bit different—it halts the program but also causes the microprocessor to start from the very beginning by erasing all internally stored results and starting a new program from a different starting point.

Let's put all this information together and see exactly what is required to make a complete system. First are data and address busses and their buffering, con-

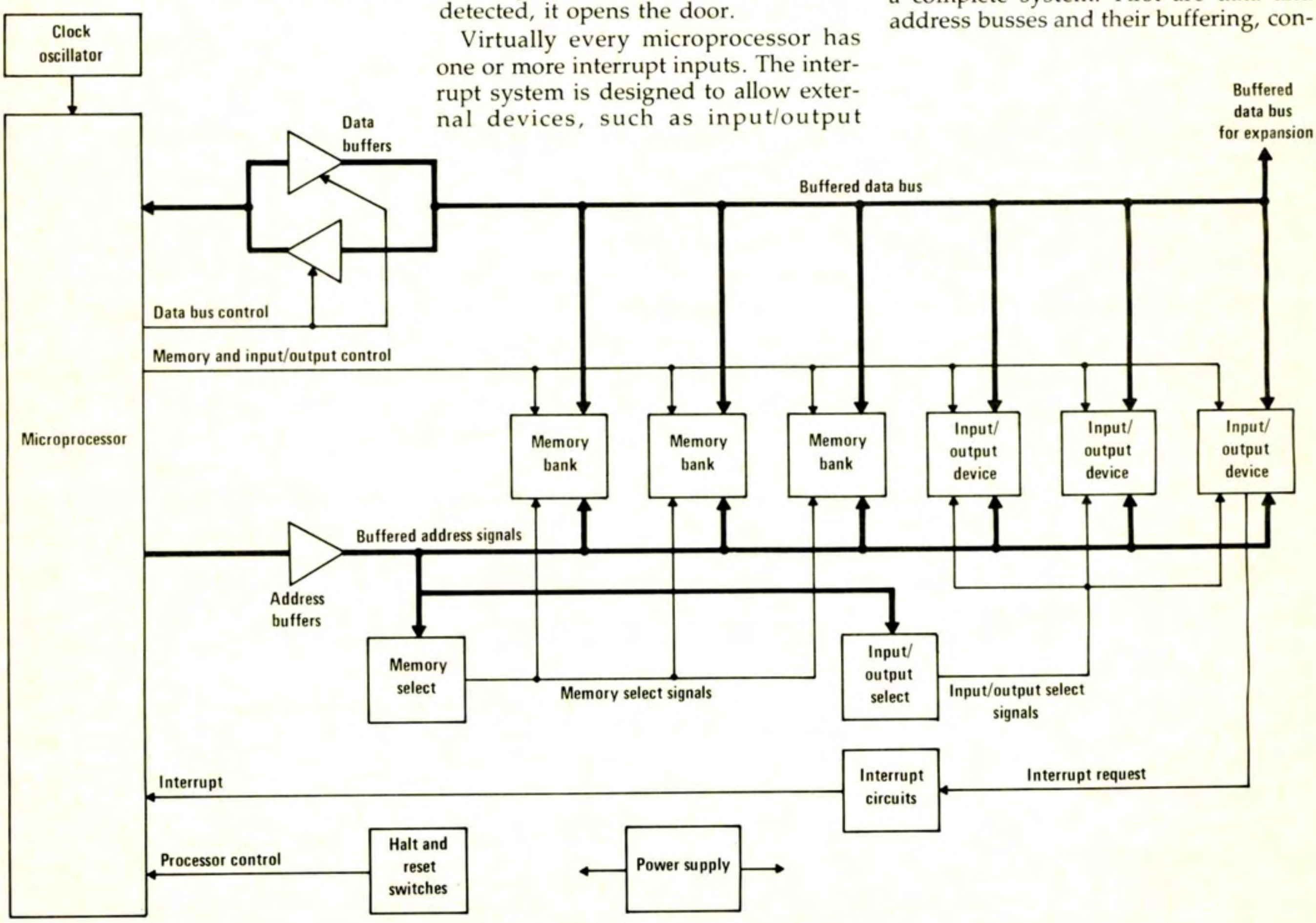


Figure 4 **Typical complete computer system**

necting the microprocessor integrated circuit to the memory and input/output devices. Memory select circuitry selects one of several memory banks, while input/output select circuitry selects one of several input and output devices. Some of the input/output devices may connect to the interrupt input through some additional interrupt circuitry. Finally, panel switches for reset and perhaps halt connect to the processor control inputs.

Wrapup

What is a means of entering the program into memory in the first place? The very first small computers solved this problem by adding a control panel. By means of switches and readout lights on the panel, it was possible to enter data into specific memory locations, check to see that it had been properly entered, and start and stop the computer. Such a control panel requires connection to both the address and data busses, some way of disconnecting the microprocessor from these busses so that the control panel can feed numbers directly into the memory without the microprocessor being involved, as well as connections to the processor, memory, and input/output control lines. This requires a substantial amount of circuitry.

A more recent approach has been to make at least one of the memory banks a ROM which is preprogrammed with a system program called a monitor. This monitor program allows an external device, such as a teleprinter or tv-type terminal, to enter data or output data from the system under monitor control, acting as a control panel but without all the circuitry a real panel would require. Some systems, such as the Heathkit H8, combine a monitor with a small control panel. The control panel contains just a series of pushbutton switches on a keyboard, and some light emitting diode readouts. The switches do not really do anything, as they simply connect to various combinations of wires on the data and address busses. Instead, the monitor does all the work by monitoring these switch closures and interpreting their meaning by means of a program.

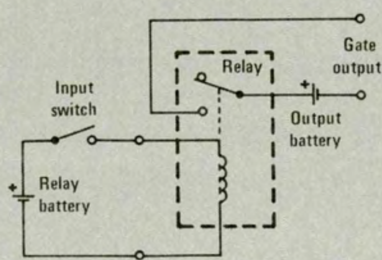
Despite the need to make hundreds and hundreds of connections to interconnect the microprocessor with all the other integrated circuits needed to make a complete system, the recent progress in hobby and personal computers has made a thorough knowledge of just exactly how this is done and why quite unnecessary. Whereas 10 years ago the amateur computer builder had to build his system circuit by circuit, connection by connection, today's computer builder can buy preassembled components and even complete systems. Still, sometimes it is comforting to know just what is inside. There's a lot more than just a $17.95 IC chip!

Basic building blocks for electronics projects: gates

Newcomers to electronics hear a lot about gates. But what the heck are they?

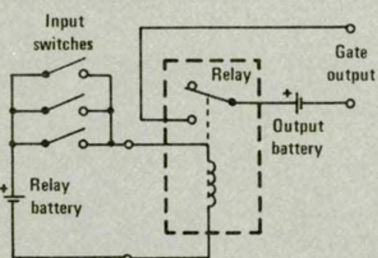
Thumbing through the pages of Modern Electronics, you'll find many small, inexpensive, and easy-to-build projects. The key ingredient in these "mini" projects, and the basic building block of even the world's largest computer, is the integrated circuit (IC) digital gate.

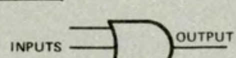
The digital gate in reality is nothing more than an educated "on-off" switch. You can get a better idea of what goes on inside an IC by looking at the diagram below. Here you'll see a simple relay circuit that



connects a battery to the output terminal when another battery is applied to the input.

Suppose you connected the relay battery through two or more switches as shown below. If you did, you could connect the battery

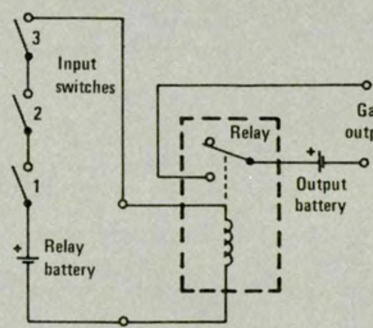


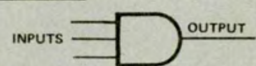


OR-GATE SYMBOL

to the relay, energizing it, by closing any one of the switches. Since you can connect the output battery to the terminal by closing input switch one *or* input switch two *or* any other input switch, this kind of circuit is called an *OR* gate.

If you connected your relay control switches in series, as shown below, you would have to close all

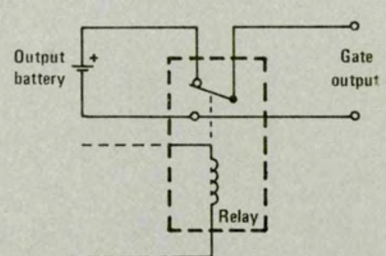


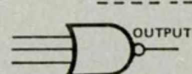


AND-GATE SYMBOL

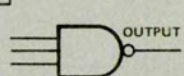
of them before the output battery would be connected to the terminal. Since you would have closed switch one *and* switch two *and* every other switch in the circuit, this kind of circuit is called an AND gate.

The OR and the AND gates described so far have no output *unless* the relay is energized by closing the appropriate switches. Sometimes, however, you need a gate that provides an output *except* when the relay is energized. This is done by adding a connection to the relay contacts, as shown below. Gates that provide *no* output when energized are called NOT gates. So an OR gate becomes a *not-or* gate, or simply NOR gate; AND gates become NAND gates. NAND and NOR gates are distinguished from





NOR GATE



NAND GATE

AND and OR gates by a small circle at the output end of the symbol in schematic diagrams. —*Bob Margolin*