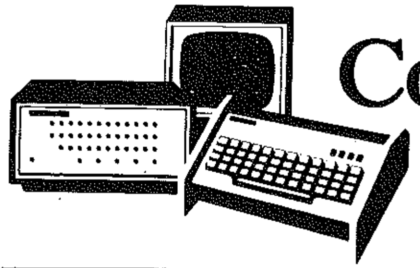


Computer Bits

By Jerry Ogdin

INTERRUPTS AND REAL-TIME

HOBBYIST applications of computers fall into two general categories: intellectual pursuits and practical uses. In intellectual pursuits, simplicity is generally the keyword and all the unessential details are stripped away to reveal the underlying simplicity. In practical (real) applications, however, we can't afford that luxury, and it is in these useful areas that the realities of "real time" operation strike home.

In any computer, the instructions are sequentially executed until the specific goal is accomplished, and all this happens without regard to the onward march of time. However, once you are forced to write a program that must finish within some limited amount of time, you are in the "real time" domain. A computer program to read thermostats once every fifteen minutes is not difficult at all. But a program that is required to shut off a valve when water reaches a certain level must be very short, unless you want to risk a flood.

In real-time programming, it is generally not too difficult to write a program that can perform the required work in the allotted time and produce the requisite output signals. What is difficult is sensing the external conditions so that the program can be activated. Ideally, the real-time program can be made to sit in a loop, sampling some input signal until the appropriate event occurs to activate the rest of the program and produce (hopefully) the appropriate response (see Fig. 1). However, this simple scheme limits the computer to handling one and only one chore at a time. The major virtue of a computer is its ability to do many things and do them all fast. It seems a shame, then, to limit the entire computer to continuously executing one sequence of input testing instructions.

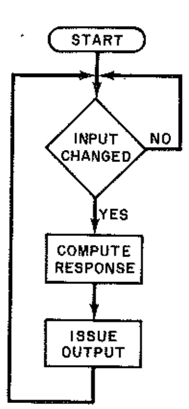


Fig. 1. Real-time looped program.

If you want to install a computer in the basement to monitor and control many of the sub-systems in the house (heating and air conditioning plant, burglar and fire alarm system, night lights, and a small diskette file that holds your family financial records, for example), you will need to "service" a wide range of input signals. Any of these signals can arrive at any random time, and unless you can afford to ignore some of them it is important that you create a hardware and software combination that is foolproof. This can be done with the *interrupt* circuitry of the computer or microprocessor.

An interrupt is a special kind of external signal that notifies the computer to pause in its current task, do something else for a while and then resume the interrupted task. For example, if you are reading a newspaper and the phone rings (you answer, talk, hang-up and resume reading the newspaper), you've been interrupted. Similarly, when the computer is executing a program to compute and generate some output, we should design the hardware and software to be interruptable. Then, when some new external stimulus comes along (via the interrupt hardware), the currently executing program is suspended and the interrupt-servicing software is activated. When the interrupt service subroutine is finished, it will cause the interrupted program to be reactivated.

An elegantly simple interrupt system is used in the popular Intel 8080A microprocessor. As shown in Fig. 2, the external signals are applied to an input called INT (interrupt). When the computer is about to fetch the next

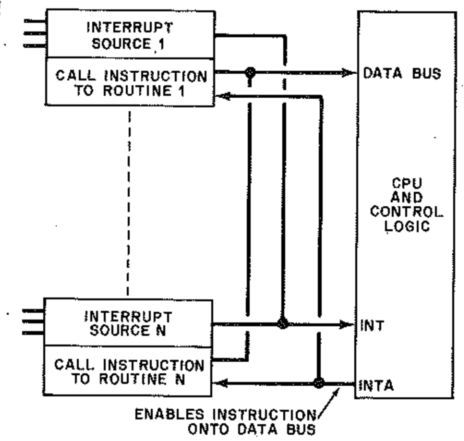


Fig. 2. Diagram shows how the interrupt works in the 8080A.

instruction in sequence, it senses the INT condition and does two things: it issues an interrupt acknowledgement signal (INTA) which allows the external hardware to supply an instruction to the CPU instead of the memory; and it inhibits the incrementing of the internal program location counter. Otherwise, instruction execution proceeds normally.

The interrupting equipment is then responsible for supplying the CPU with an instruction in the 8080's architecture. The interrupting hardware places an eight-bit instruction on the data bus whenever the interrupt acknowledge signal (INTA) appears. This instruction is a subroutine call which causes the current program location counter contents to be pushed onto the stack in memory and is the first step in the process called "preserving the state of the program." After the subroutine call instruction preserves the interrupted program's next-instruction address, it places a specified address into the program counter; that is the address of the interrupt servicing program. At this point, the interrupt servicing program may not modify anything in the computer or memory without tampering with the execution of the interrupted program.

When any program is in execution, part of the status of that program is in the CPU's registers. The contents of the program location counter and the accumulator and the flag bits are all

important. If any of these is changed from outside the program, then the program's correct execution is nearly always prevented. The interrupt servicing program must preserve this part of the interrupted program's status, and in most computers there are instructions that allow the state to be preserved by copying the CPU registers into memory (Fig. 3). It is this particular operation that the 8008 is unable to perform without some extra outside hardware. The 8080, however, has a set of instructions for preserving the state of the computer; for example, the PUSH instructions save the relevant information in the stack. The 6800 microprocessor automatically preserves the entire state of the CPU on the stack whenever an interrupt is sensed.

memory. The execution of a subroutine return instruction will cause the interrupted program's program location counter to be restored. The interrupted program then continues as if nothing had happened.

There is a potential problem that arises at the interface between the real-time interrupt servicing program and the non-real-time "background" software. There is a part of each routine, called the "critical section" where confusion may reign unless some care is taken. Imagine the worst possible case: the interruptable program is interrupted just at the instant that it is about to examine the interrupt counter. In most computers, the counter must be fetched up from memory into a CPU register to be examined and decremented and then re-

```
;   INTERRUPT SERVICE ROUTINE
;   (INTERRUPT #7)
            ORG     56
;   COMES HERE TO SERVICE INTERRUPT AUTOMATICALLY
            PUSH    PSW         ;   PRESERVE STATE
            PUSH    B           ;   OF INTERRUPTED
            PUSH    D           ;   PROGRAM
            PUSH    H
;   INTERRUPT SERVICE ACTIONS HERE
            LXI     H,COUNT
            INR     M
            POP     H           ;   RESTORE STATE
            POP     D
            POP     B
            POP     PSW
            EI                  ;   RE-ENABLE INTERRUPTS
            RET                 ;   GO BACK
```

Fig. 3. Routine to allow state to be preserved by copying the CPU registers into the memory.

After the interrupted program's state has been preserved, the interrupt servicing program performs its work. Generally, the minimum amount of work is accomplished. The longer the interrupt servicing program takes, the more likelihood there is that some other interrupt will be lost. The absolute minimum is to record the fact of the interrupt somewhere in memory. Each time the interrupt occurs, a specific location in memory is incremented by one; and, whenever the interruptable softwear gets time, the counter is examined and the appropriate actions are taken. The interruptable software then decrements the counter by one to account for the action taken.

After the interrupt servicing routine is finished, the entire state of the computer is restored to the conditions that existed *prior* to the interrupt. The registers of the CPU are restored from

stored to memory. This sequence must not be interrupted. If it can be interrupted, you may not get the expected results. The best way to avoid this problem is to prevent the interrupts from being serviced while the interrupt counter is being examined or modified (Fig. 4). There are instructions in all computers that can be used to disable and then re-enable interrupts.

The interrupt then, is basically a hardware trick to allow us the illusion of having two or more different cooperating computers sharing a common memory. It is important not to let the interrupt service routines do too much as they may be difficult to debug if they are complex, and the effects of the bugs may be obscure and seemingly unrelated. If the interrupt service routines are simple, they will be easy to "prove" correct without

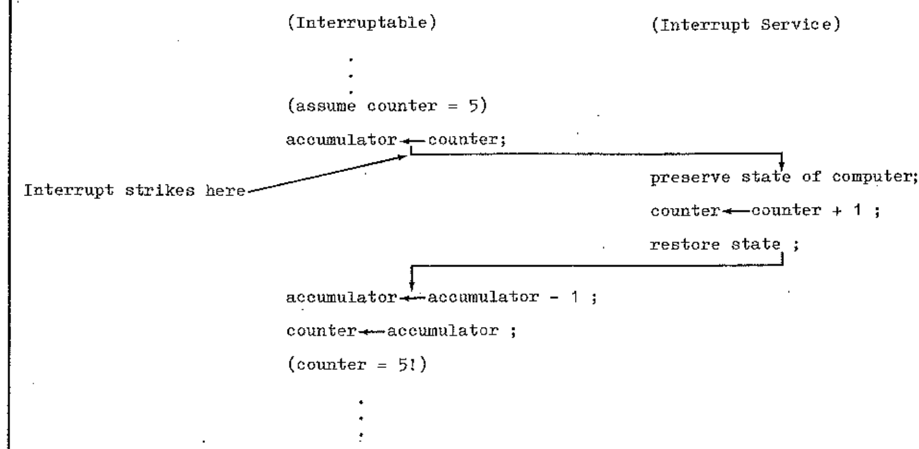


Fig. 4. Instructions can be used to disable and then re-enable interrupts.

much ado. Interrupts, like any other riches, can be overindulged; so resist the impulse.

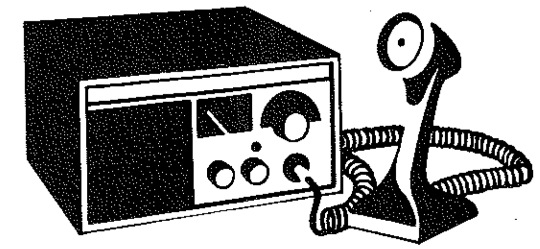
We Get Letters. Since we introduced the HIT (Hobbyist Interchange Tape) system in our September 1975 column, we have received a considerable amount of mail concerning it. There are a number of such cassette systems, each manufacturer using his own scheme. At a November meeting in Kansas City, Mo., an effort was made to establish a standard format. Looks like a 1200/2400-Hz system will be adopted. We'll keep you posted on the future developments.

The HIT format was not suggested as a computer industry standard, but as a simple, low-cost, easy-to-implement system for hobbyists so that they could exchange data and programs. In fact, one supplier of OEM kits (Cramer Electronics) has its software recorded in the HIT format, and a HIT interface is included in their kits.

News Items. AMD announced an OEM price reduction on its 8080A (to below $30), and Intel followed suit. Then, Texas Instruments announced its 16-bit 9900. Not due until March, the 9900 operates at about the same speed as the 8080, but will probably cost about $50. When price reductions start on the OEM level, they eventually sift down to the hobbyists.

Hobbyist Clubs. Information on computer hobbyist clubs continues to come in. In addition to the ones listed below (to which we will add from time to time), check out your local high schools and universities. Many of them have clubs which were formed by students, with outsiders generally welcome to join.

California
Homebrew Computer Club, 193 Thompson Sq., Mountain View, CA 94043.
Sacramento Minicomputer Users, Box 741, Citrus Heights, CA 95610.
San Diego Computing Society, c/o Gary Mitchell, Box 35, Chula Vista, CA 92012.
Southern California Computer Society, Box 987, S. Pasadena, CA 91030.
UCLA Computer Club, 3514 Boelter Hall, UCLA, Los Angeles, CA 90024.
29 Palms Area, Sgt. Wesley B. Isrigg, Box 3558, C&E Schools M.C.B., 29 Palms, CA 92278; or Sgt. Stanley E. Herr, 13-C Copper Dr. M.C.B., 29 Palms, CA 92278.
Colorado
The Digital Group, Box 6528, Denver, CO 80209
Illinois
Chicago Users Group, c/o Robert Swartz, 195 Ivy Lane, Highland Park, IL 60035.
Massachusetts
Boston Area, c/o John Vullo, 21 Sunset Ave., N. Reading, MA 01864.
Minnesota
Bit Users Assoc., Resource Access Center, c/o Richard B. Koplow 3010 4th Ave. So., Minneapolis, MN 55408.
Southeast Minn. Amateur Computer Club, 2122 NW 17 Ave., Rochester, MN 55901.
New Jersey
Amateur Computer Society of N.J., 72 S. Railroad Ave., Staten Island, NY 10305.
Ohio
"Universe Unlimited" Users Group, c/o John E. Kabat, 11918 Forrest Ave., Cleveland, OH 44120.
Oklahoma
Oklahoma City Club, 2412 SW 45th, Oklahoma City, OK 73119.
Pennsylvania
Pittsburgh Area Computer Club, c/o Fred Kitman, 400 Smithfield St., Pittsburgh, PA 15222
Texas
Texas Computer Club, c/o L. Walker, Rt. 1, Box 272, Aledo, TX 76008.
Canada
Canadian Computer Club, c/o G. Pearen, 861 11th St., Brandon, Manitoba, Canada R7A 4L1. ◈



CB Scene

BY SHERMAN P. WANTZ

HOW TO USE CB RADIO "BUZZ" WORDS..

INTERSTATE truckers, abetted by a growing number of mobile and base station operators, meet each day, on Citizens Band (CB) channel ten. They use a colorful—often cryptic—language to communicate with one another.

The glossary below has been developed to help CB newcomers interpret the special terms used most frequently on the interstate highway channel and elsewhere in the CB Class D frequency spectrum. Save this list and add new terms and meanings to it as they develop among the truckers.

Using CB jargon correctly is interpreted by some as a mark of sophistication. Anytime you cannot remember the "proper" term to use, however, plain old everyday English will do. Even on channel ten.

TERM	MEANING	TYPICAL USE
Back	Over	Back to you.
Back door	Rear of vehicle	I'm watching your back door (protecting you from being overtaken unawares by police).
Back down	Drive slower	Smokey's ahead. Better back it down.
Back out	Stop transmitting	I'm going to back out of here.
Barefoot	CB set output signal not additionally amplified.	I'm running this rig barefoot.
Base station	CB set operated from a fixed location	I'm operating here from my base station.
Beam	Type of directional antenna	I've got my beam on you now.
Big switch	Turn off CB set	I'm going to pull the big switch.
Big 10-4	Acknowledged with enthusiasm	That's a big 10-4.
Bleeding	Interference from a nearby CB channel	A local (station) is bleeding all over you.
Break	I'd like to interrupt	Break, break.
Breaker	One who interrupts	Go ahead, breaker.
Catch	Talk to	We'll catch you later.
Chicken coop	Weigh station for trucks	The chicken coop is closed.
Clear	Out, final transmission	We're clear
Come again	Repeat your last transmission	Come again?
Comeback	Return call	Thanks for the comeback.
Come on	Over, invitation to transmit	What's your twenty? Come on.
County mounty	Sheriff's deputy	Two county mounties are parked under the bridge.
Covered up	Interferred with	You were covered up that time
Cut the coax	Turn off CB set	I'm going to cut the old coax now.
Bodacious	Shamelessly strong	You've got a real bodacious signal up here.
Ears	CB set	Got your ears on?
Eighteen wheeler	Truck and trailer	How about that northbound eighteen wheeler?
Eights (eighty-eights)	Goodbye (from amateur radio 88's meaning "love and kisses.")	Threes and eights to you.
Eyeball	Face-to-face meeting	Can we get together for an eyeball?
Feed bears	Pay a fine for speeding to highway patrol	I can't afford to feed any bears.
Final	Last transmission	I'll turn it over to you for your final.
Flip flop	Trucker's return trip	We'll catch you on the flip flop.
Four wheeler	Automobile	You've got a southbound four wheeler.