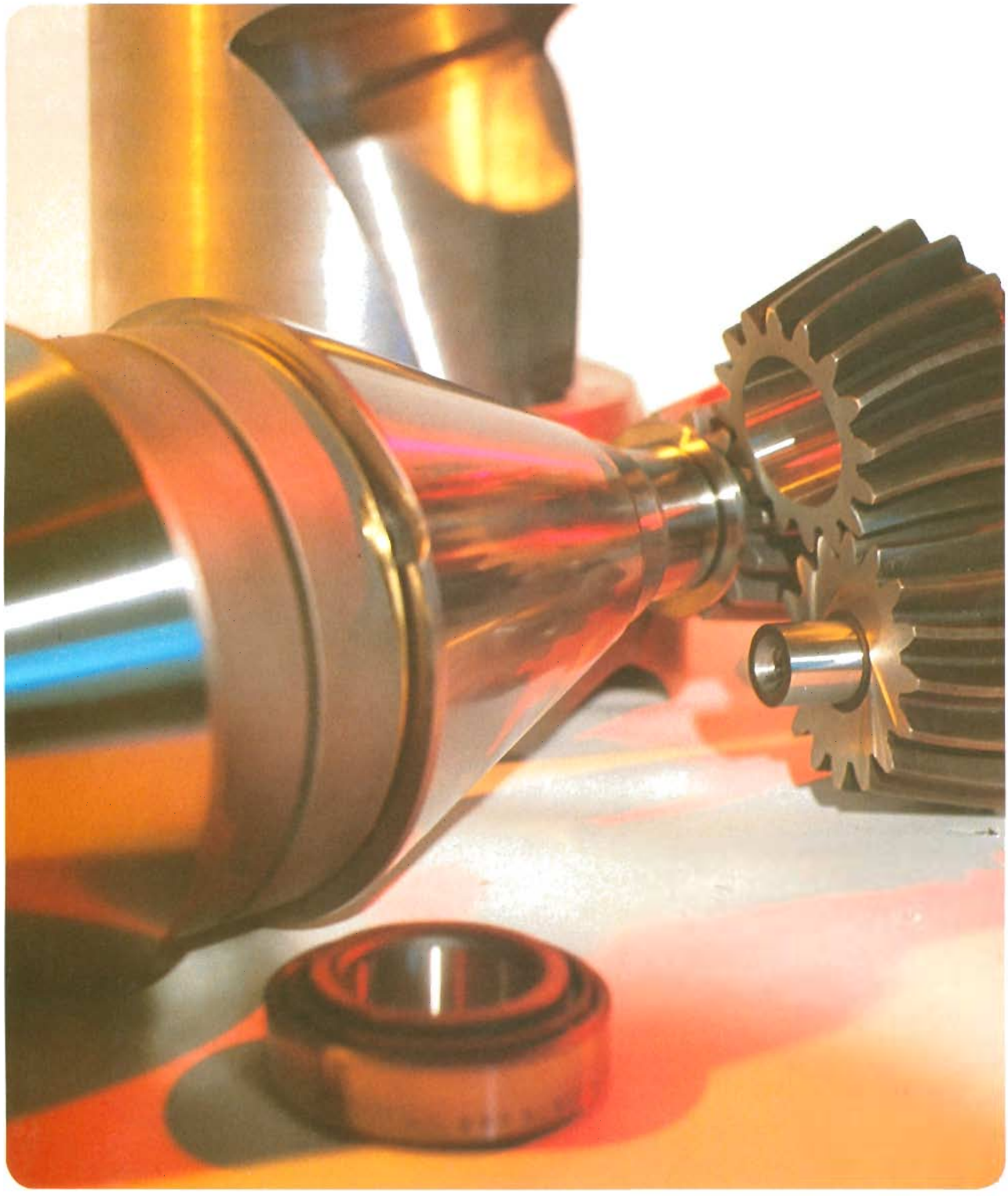
Keyboard

Mar-Apr/79 A Publication of Hewlett-Packard Desktop Computer Division



Keyboard
March-April 1979

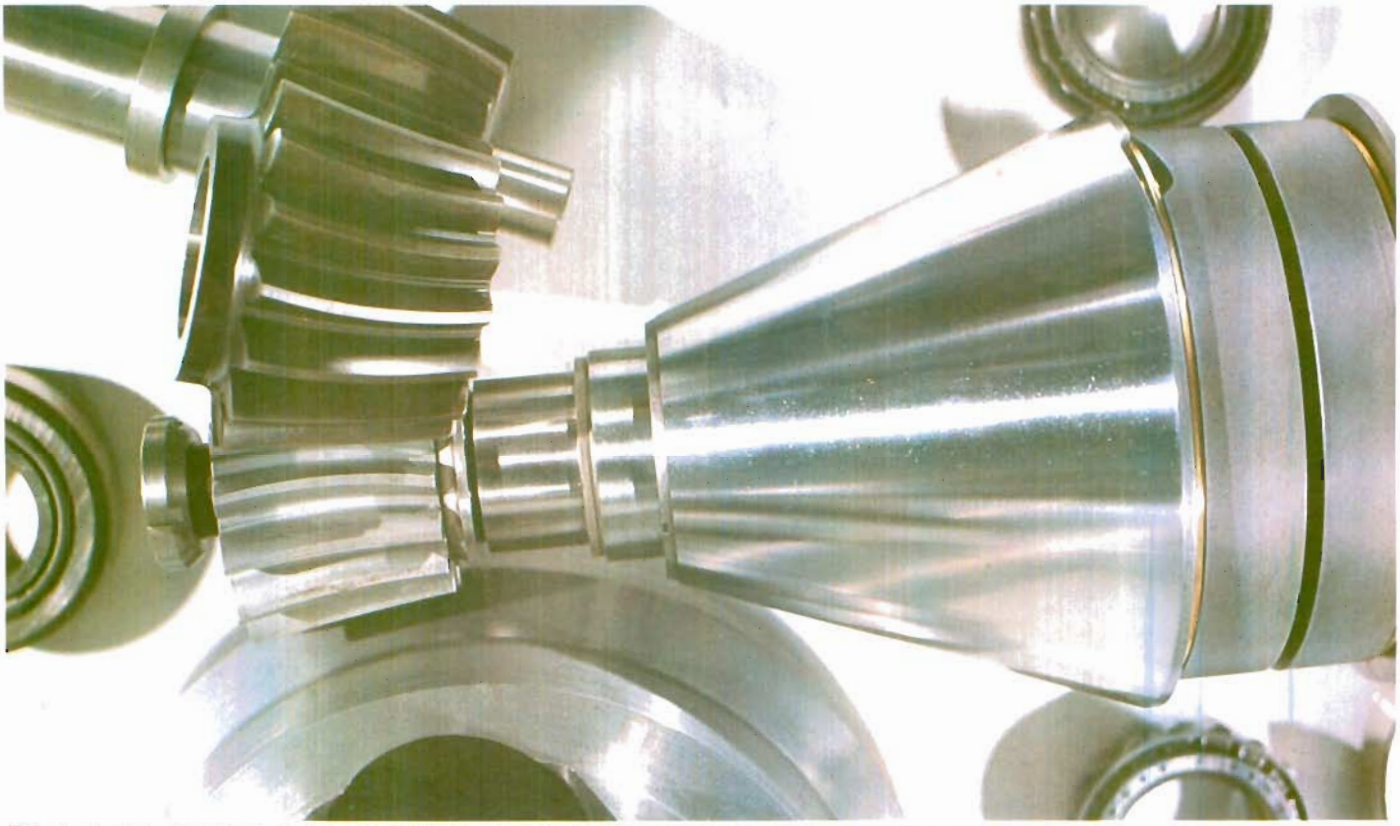
Notice

Have you sent us your *Keyboard* subscription renewal card? Your January-February 1979 issue included a renewal card to fill out and mail. If you have already mailed it in, you will continue to receive *Keyboard* without interruption. If you have not yet completed and mailed the card, this will be your last reminder. Send us the card by April 1, 1979, and you will continue to receive *Keyboard* without missing an issue. If you need another card, call your nearest HP sales office, or contact us: *Keyboard*, Hewlett-Packard, 3404 East Harmony Road, Fort Collins, CO 80525 U.S.A.; or call us at 303/226-3800.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Design Research Suggests Energy, Pollution Solution



This photograph, like the cover, shows some of the internal parts of a new type of transmission. Unlike the cover, this exposure was taken without colored lights. Torque is transmitted by the viscous shear of a thin oil film between two loaded rolling contact surfaces. High coefficients of traction are possible due to the rheological behavior of the oil or traction fluid at high pressure. Viscosity of the synthetic fluid increases greatly under pressure, becoming glassy under the stresses prevailing between the rolling contact surfaces, a concept upon which the transmission design is based.

Vadetec Corporation, Troy, Michigan, is a research and development and patent licensing company specializing in the transmission field. Its end products are engineering software and preproduction prototypes. As described in this article, Hewlett-Packard equipment plays an important role in both of these areas.

During the past seven years, Vadetec has developed new types of traction drives based on a unique kinematic principle involving the wobbling of a body during rotation about an axis, or nutation. As a result of this effort, Vadetec has acquired a vast amount of experience in the transmission field and a portfolio of more than 200 patents filed in 12 countries. Among Vadetec's licensees are some of the world's major vehicle manufacturers.

by Edward G. Trachman

Today, all industry is under pressure to solve, at great expense, problems created by the energy crisis and pollution. A driveline using a continuously-variable-ratio transmission (CVT) represents a quantum jump in overall automotive efficiency that could be one of the solutions to these problems.

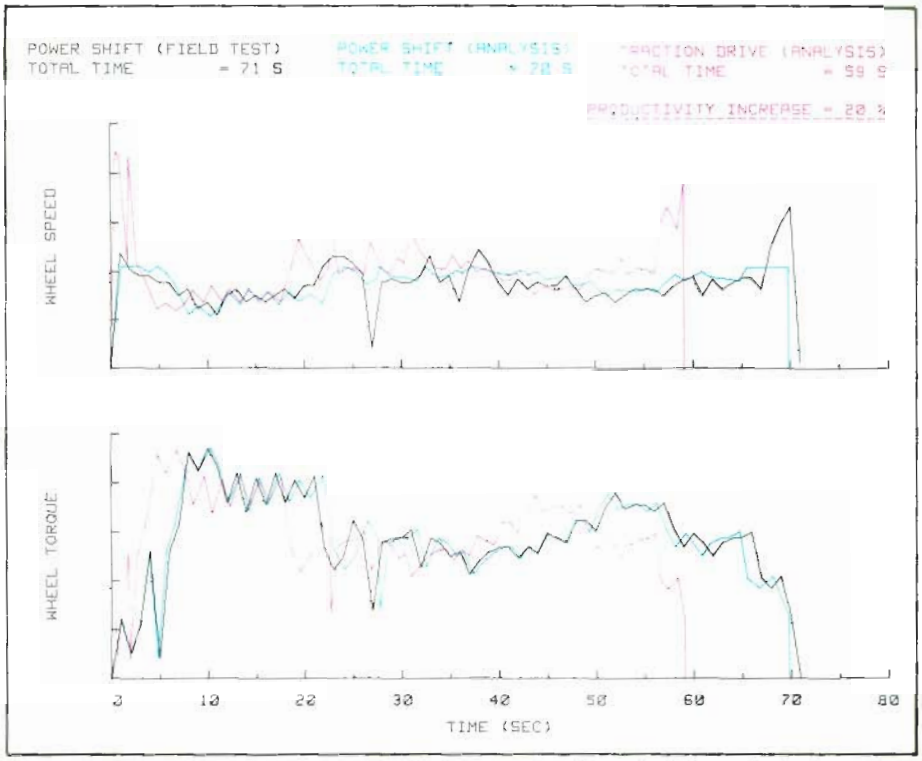
Vehicle performance simulation

In the study of the overall efficiency of a vehicle, we simulate the performance over typical duty cycles. We have performed these simulations for several automobiles over the U.S. EPA Urban and Highway Driving Schedule, and the European city cycle.

Buses, trucks and off-road vehicles also have been studied, each with an appropriate cycle. Power plants studied for use in these vehicles have included both diesel and spark ignition engines, and driveline systems both with and without flywheels.

The measure of overall efficiency varies for each application. For example, the overall efficiency of automobiles is measured by fuel consumption. In this area, U.S. federal legislation has set the rules. There are required minimum mileage performance levels, and maximum emissions levels over a defined driving schedule. But for off-road equipment, the measure of efficiency may be productivity, as in cubic yards of earth moved per hour.

Vadetec's approach has been to study the overall system efficiency of any given automobile or off-highway vehicle, and design a transmission and driveline which will optimize that particular vehicle's desired performance. This individual approach to each application requires constant development of new computer programs and revision of existing programs.



Efficiency increases of more than 20% have been demonstrated in typical duty cycles for farm equipment, trucks and off-road equipment. In this four-color figure from the HP 9872 Plotter, a typical duty cycle for off-road equipment is defined by the plots of wheel speed and wheel torque versus time. In the figure, the results of the performance simulation of the equipment under test (blue) can be compared with the actual test results (black). The simulation of the same vehicle with its transmission replaced by a traction drive (red) shows the same work cycle can be completed in 20% less time.

Vehicle performance simulation has proven to be a valuable tool in the design of new propulsion systems. Analysis of the results of the simulations has disproven some commonly accepted driveline ideas and guided the development of Vadetec's unique and patented systems using the continuously-variable-ratio transmission.

Vadetec has developed a general vehicle performance simulation program on the System 45 that can be applied to all of the vehicle duty cycles, engines and drivelines mentioned above.

We have found the HP System 45, with its convenient editing and debugging features, very well suited to this environment, where there is a relatively high proportion of program development time to program execution time.

Program interrupt during execution

We can interrupt the program during execution, make a change, and then continue. Even with many sophisticated minicomputers, this is not possible. Because you can pause exactly where the problem occurs and query the current value of any variable when using the System 45, you can constantly debug a program during its development. And there is no need for recompiling during program development with the BASIC language. We have complete control of the system.

The configuration we use at Vadetec is the System 45S, which includes 62K bytes of memory, the graphics package, a second tape cartridge drive and an internal thermal line printer. We have a group of four System 45s to meet our program development and computation needs. We also use the HP 9872A Plotter and the HP 9871 Impact Printer to assist in report preparation.

From foreground, Shan Shih and Patrice Elu of the Research Engineering Analysis Department of Vadetec make heavy use of both the computational and graphics capabilities of their System 45s.

Easy program manipulation

The program has been assembled with the logic of the system and the energy balance in the main program and the engine map, CVT map, vehicle characteristic parameters, duty cycle data, etc. in subprograms and data files. This allows easy manipulation of the program.

For example, the same vehicle can be run on different duty cycles, or different vehicles can be compared on the same duty cycle by simply attaching a different data file for the appropriate change.

Vadetec has simulated the performance of several automotive propulsion systems operating over the U.S. government's defined cycle and, as an illustration, this article will outline the procedure of simulation for a general driveline consisting of an engine, a flywheel and a continuously-variable-ratio transmission.

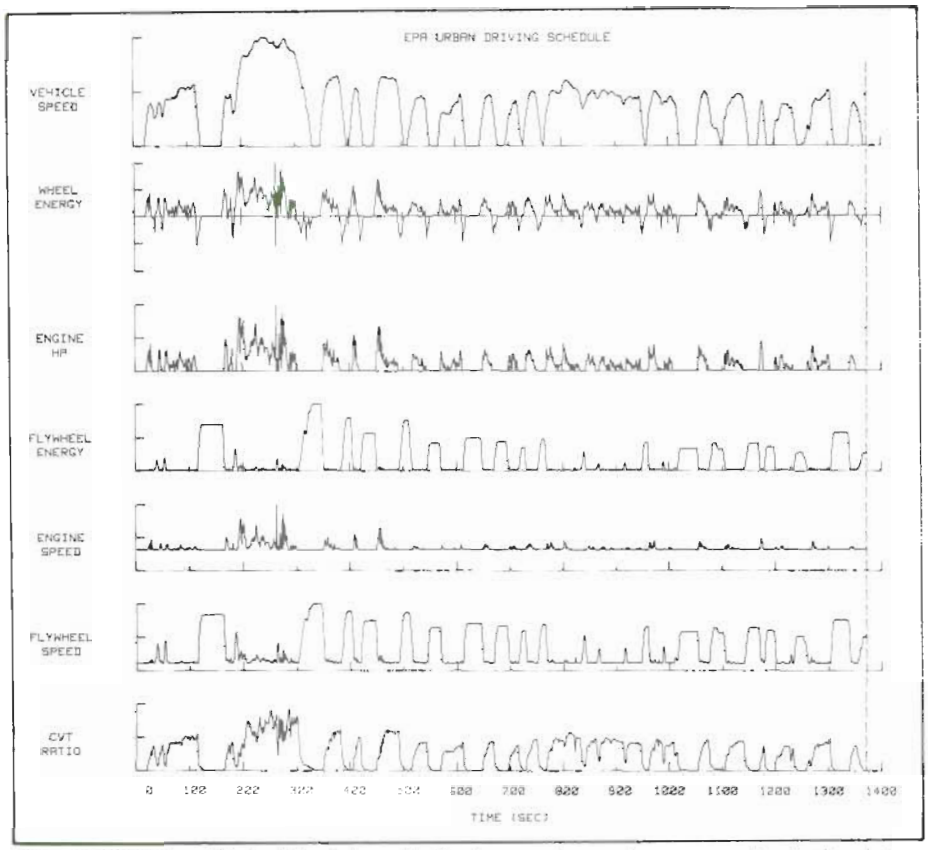


Figure 1. EPA Urban Driving Schedule and testbed parameters are shown versus time in this plot.

The EPA Urban Driving Schedule, as defined in the U.S. Code of Federal Regulations, is a speed-versus-time sequence consisting of 17 stop-drive-stop cycles. The total test time is 22 minutes, 52 seconds; the total mileage is 7.45 miles and the maximum vehicle speed is 56.7 mph. The cycle is the top line in the multiple-graph plot of the EPA Urban Driving Cycle, figure 1.

Energy loss sources

When an automobile operates on a city driving schedule such as the one described above, the following are among the main sources of energy loss.

- Fuel is wasted from not operating at maximum thermal efficiency, hence at minimum BSFC (brake specific fuel consumption), due to the inability of discrete speed transmissions to select the optimum ratio.
- Kinetic energy is lost during deceleration and braking.
- Fuel is consumed during idling when the vehicle is stopped.

The load on the system is the energy which must be supplied to the wheels to propel the vehicle according to the speed versus time sequence specified by the EPA cycle. The energy required for this task consists of three components — the energy to accelerate the mass of the vehicle, the aerodynamic drag of the vehicle, and the rolling resistance of the tires.

The acceleration term may be positive (acceleration) or negative (deceleration). The two additional forces, aerodynamic drag and rolling resistance, are always resisting the motion of the vehicle and therefore are always positive. The algebraic sum of these three quantities is the "wheel energy".

The required wheel energy and the upstream losses in the transmission and driveline must be supplied by the engine and/or flywheel.

The flywheel is also an energy storage device. It allows the negative wheel energy (due to deceleration of the vehicle) to be stored and used when required for subsequent acceleration of the vehicle. This is called regenerative braking.

Continuously-variable-transmission

In order to make this system work, we need a CVT. The flywheel speed is set by its energy content at any given time and the wheel speed is set by the speed-time requirements of the EPA cycle. These two independent speeds can be connected at all times only by a continuously-variable transmission.

The performance simulation program marches forward in time. At each step of time, the program looks up the required vehicle speed, calculates the required "wheel energy," and supplies sufficient energy from the engine and/or flywheel according to the logic of the system being simulated.

The fuel consumption and mileage traveled is accumulated and the miles per gallon rating is calculated at the end of the cycle. Hydrocarbon, carbon monoxide and nitrogen oxide (NO_x) emissions are also taken from engine test data at each time step and accumulated for the cycle.

The results of the program, the vehicle's performance over the cycle, are summarized in graphic format in figure 1. The EPA Urban Driving Schedule is plotted first, followed by the wheel energy, engine power, flywheel energy, engine speed, flywheel speed and CVT ratio, all plotted against a common time axis.

The energy balance calculations require the following data for the components of the system: the aerodynamic drag coefficient; the tire rolling resistance; the mass of the vehicle and the driveline efficiency. These are the results of tests and have been obtained from the manufacturer for each vehicle studied.

Bench testing of Vadetec CVT prototypes depends upon a 9830 desktop computer for control of data acquisition and processing. Vadetec plans to modify the system further in the near future to control the entire test process.

The performance simulation program also requires a fuel consumption and emissions map of the engine over the entire range of speeds and power levels. This data has been obtained from tests by the engine manufacturers and government testing laboratories. The CVT efficiency map which is required is supplied by Vadetec.

The complex performance simulation program and graphics usually associated with "big" computers is handled competently by our desktop computer system.

CVT performance testing

Hewlett-Packard computers and instruments also play a role in our testing program. Our test bench has a computer-controlled data acquisition and processing system using the HP 9830 desktop computer as the controller.

An HP 3495A scanner directs analog signals from torque, speed, flow and temperature measuring instruments in proper sequence to an HP 3455A digital voltmeter (DVM). The DVM performs the analog-to-digital conversion and communicates the result to the 9830 for processing and report printout.

The computer commands and communicates with the scanner, the DVM and an HP 59309A digital clock over the HP-IB interface (the Hewlett-Packard Interface Bus is HP's implementation of IEEE Standard 488-1975). The HP 9862A plotter is used to generate report graphics of the test right at the test bench. The system works well for immediate analysis and reduction of the test data. Future plans for the system include automatic control of the test.

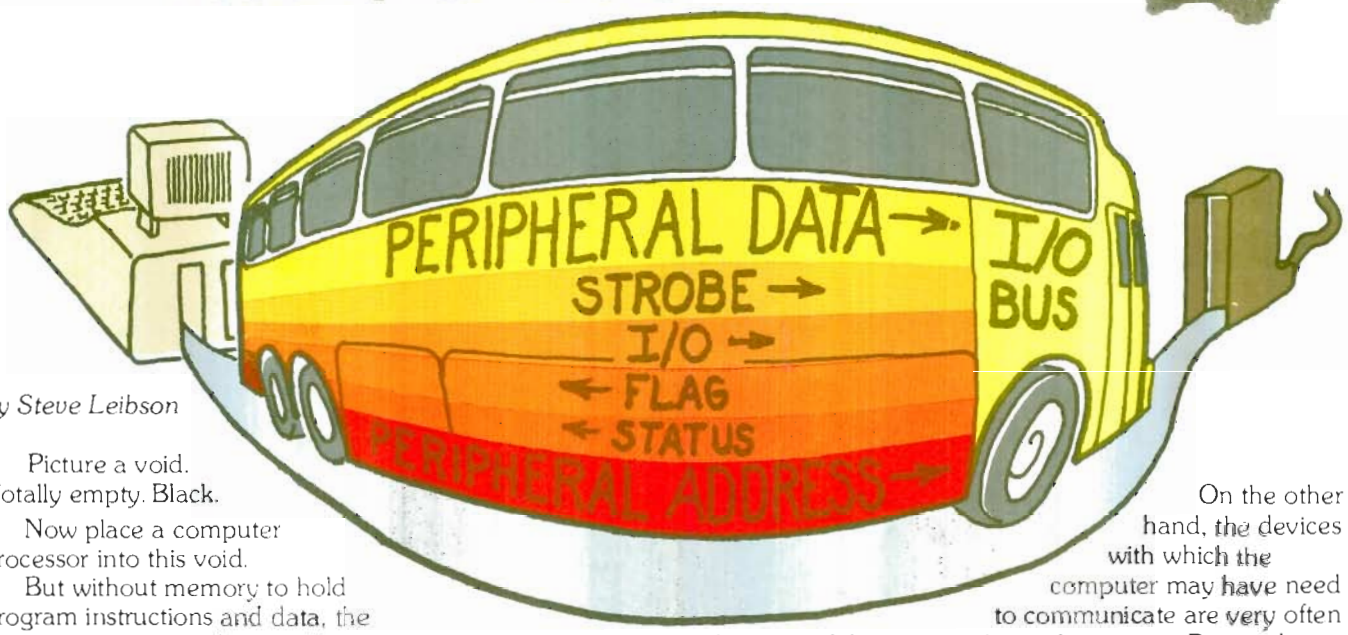
NDT's future

The results of our vehicle performance simulations indicate that with the new Vadetec driveline, NO_x emissions can be reduced to the new EPA target of 0.4 grams per mile. At the same time, the fuel economy of any automobile, regardless of weight, or whether powered by a spark ignition or diesel engine, can be increased by 30 to 50% over the U.S. EPA Urban Driving Schedule. K

Dr. Edward G. Trachman is head of the Engineering Analysis Department at Vadetec. Prior to joining Vadetec, Dr. Trachman conducted research in elastohydrodynamics and tribology at General Motors Research Laboratories, Warren, Michigan, and at RCA Laboratories, Princeton, New Jersey. He received a bachelor's degree at The Cooper Union, New York City; a master's degree at California Institute of Technology, Pasadena, California; and a PhD from Northwestern University, Evanston, Illinois. All three degrees were in mechnical engineering.

Leibson on I/O part II

The I/O bus



by Steve Leibson

Picture a void. Totally empty. Black.

Now place a computer processor into this void.

But without memory to hold program instructions and data, the processor is useless. So we will provide a memory and some wires to connect the processor to its memory. Our creation floats in the void quietly running its program, performing its assigned task in the scheme of things. Suddenly, it arrives at *THE* answer — but alas, we have given it no voice, no I/O with which it can announce the conclusion.

In the first installment of this series, we discussed several basic concepts relating to computer systems and I/O (Input/Output). We are now ready to provide the computer with a voice, some means of supplying the answers to the questions asked of it by some programmer.

Bus is a set of conductors

The first order of business is to create an I/O bus leading from the processor to the outside. As we discussed last time, the I/O bus is a set of conductors carrying signals that represent the information which the computer is trying to transmit from the processor to the interface.

In addition, several conductors carry control signals that make it possible for the computer to signal the recipient at the other end of the bus when the data on the bus is valid and should be accepted. The recipient must also have some signals to communicate to the processor its readiness to accept data and its operational status. Finally, a signal is needed to dictate the direction of data flow on the I/O bus since we want the computer to receive as well as transmit data.

Our figure of the data bus shows that it has a number of connections. The topmost connection represents a group of 16 data lines, and is shown with arrowheads at both ends. This is the peripheral data bus. It is capable of carrying data in either direction, depending on the immediate need. Under the data lines is a single wire called "strobe". This wire is a synchronizer and is used by the computer to signify that data is available.

I/O wire is the traffic cop

The next line is called I/O and controls the direction of data flow. It is the traffic cop of the I/O bus, allowing bidirectional data flow, but only in one direction at a time. The recipient signals the computer on two wires called "status" and "flag". Status is a very simple signal, used to represent the presence or absence of the recipient. It is impossible to communicate with a device that isn't there.

"Flag" is a more complex signal and to explain it requires a brief study of speed. In the scheme of things electronic, computer processors are very fast. The only moving parts inside a processor are the speedy electrons carrying the digital signals.

On the other hand, the devices with which the computer may have need to communicate are very often mechanical in nature. Disc and tape mechanisms, printers and plotters all have moving parts which take relatively long periods of time to perform their assigned tasks.

Taking a printer as an example, let us examine the interchange between the computer and this peripheral device.

The computer first addresses the printer using the last set of wires on the I/O bus, the "peripheral address" bus. If there is a device at the selected address, it will respond on the status line and inform the computer of its presence.

The computer will place the information it wishes to transmit on the data lines, set the I/O line to "output" (data direction is always from the computer's perspective) and finally cause the strobe line to clock the information into the printer. If the printer is operational, it will accept the information and print it.

A serial printer, similar in operation to a typewriter, would have to move to the next character position, select the proper character and mechnically fire some mechnism to strike the paper and leave a mark.

All these proceedings may take .01 of a second or so. That may not seem like too much time but processors generally execute an instruction in .000001 second. From the processor's perspective, the printer is taking forever. Fortunately, computers are

patient and will wait if told to do so. Our example computer will courteously wait for a signal on the flag line, which informs it that the printer has finished the task assigned.

Processor interrupts

If it seems wasteful that such an expensive tool as a computer should have to wait most of the time on slower equipment, you are definitely thinking ahead. In a future article, we will explore a different communictaion mechanism known as interrupt. This feature allows the processor to go on about its business after transmitting a piece of information, later to be called back when the printer is ready for more.

We have already covered the peripheral address lines, though only briefly. Computers generally communicate to several peripheral devices. There are two ways to accomplish this. The first is to have a complete I/O bus for every peripheral device connected to the computer. Such a scheme would rapidly create a rat's nest of wires resulting in a totally unproduceable system.

Our I/O bus has a set of lines called peripheral address lines that are used to specify the device in which the computer is currently interested. This greatly simplifies wiring the system together and results in major cost savings. It does limit the computer to communicating with only one peripheral device at a time, but for most computer processors that is the limit anyway.

Peripheral lines for multiplexing

The peripheral address lines allow the I/O bus lines to be shared or "multiplexed" by many devices. Each device must have its own unique address or conflicts will arise when two devices try to use the information at the same time. For instance plotters are very useful for graphing data but

Our example computer will courteously wait for a signal on the flag line.

are terrible program storage devices. Therefore, when accessing the disc storage device, the computer would prefer that the plotter ignored the transactions on the I/O bus.

Clearly each peripheral device must have a unique address. But it is more advantageous for each to have several unique addresses. Think of the peripheral device as an apartment building having a unique street address. Apartment one gets the daily newspaper, mostly general information, while apartment two gets the *Wall Street Journal* which reports information relevant to the economic running of the country. Both apartments receive information, but of differing types.

A peripheral device must also receive varying types of information. A printer not only has characters to print, but also information relating to the running of the printer such as: line spacing, number of characters to print per line and print font to name a few. We therefore create subaddresses within the peripheral device so that information of different types can be directed to the relevant section of the peripheral.

The peripheral address lines are split into a "select code" to specify the peripheral's address and a "register code" to specify the subaddress. A "register" is a location which will temporarily hold information until it can be used by the peripheral.

Setting up subaddresses

For our purposes, it may be sufficient to create four subaddresses within each select code. And being obstinate as all computer designers are, we will call these addresses 4,5,6 and 7 because nobody likes to start at zero. These are four registers which serve as portals to the peripheral device.

Four registers require two lines to specify the register address, since two lines can take on four states. These states are:

State of line #1	State of line #2	Register Addressed
0	0	4
1	0	5
0	1	6
1	1	7

(Remember that digital signal lines can take on only two states: on and off or "1" and "0".)

Although we have four registers, we still have a complication. The data lines which carry information to and from these registers are bidirectional and therefore the registers must be also. Actually, what we must have is a total of eight registers.

Four input registers contain information for the computer to input and four output registers receive information from the computer. We will select between these two sets of registers based on the state of the I/O signal line. Remember that the I/O line was used to specify the direction of data flow over the data lines and is ideal for selecting between the input and output register sets.

We have now completed construction of a simple I/O bus which can be used to convey information

between the computer and external devices. It isn't the most advanced I/O bus but it will satisfy our present needs. We will be upgrading this "bunch of wires" in the future but first there is a more pressing problem to solve.

Introduction to interfaces

Our I/O bus happens to be a subset of the I/O bus used in the Hewlett-Packard Desktop Computers 9825A, System 35 and System 45. It would be most convenient if all peripherals were available with circuitry installed in them that directly interfaced to our I/O bus.

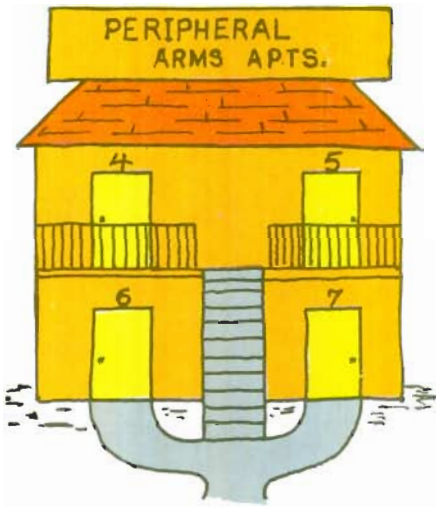
Unfortunately, the reality of the situation is far from the ideal. Our bus is parallel-oriented, meaning every binary digit (bit) of a piece of information such as a character is available simultaneously on the 16 data lines.

Not all peripherals use 16 bits of parallel data, some do not have parallel data lines at all but send and receive one bit of data at a time in a serial fashion. No peripherals use the 8 register scheme exactly as discussed above and some do not even use the same voltage levels to represent 0 and 1. This can present quite a problem to the person attempting to interface a computer to a peripheral.

Interfaces act as translators

It is necessary to interpose some specialized circuitry between the I/O bus and the peripheral device to adapt the signals from one to those of the other. This specialized circuitry is called an interface. The interface is the actual recipient of the I/O bus. The interface acts as an intermediary, translating between the two interfaced devices.

If every peripheral manufactured in the last decade required a different interface, it would be impossible for a computer system to communicate with even a small fraction of the range of



Think of the peripheral device as an apartment building having four unique subaddresses.

devices available. Fortunately, a large number of devices can be interfaced using only four basic types of interfaces: parallel I/O, serial I/O, HP-IB and BCD.

The parallel interface connects to the peripheral with a set of wires very similar to those in our I/O bus, less the address lines. This interface is the most common among current peripherals. Major variations involve the physical connector used, and the sense of the control and data lines (Does zero volts mean a 0 or a 1?). A flexible parallel interface is available with several connectors as well as with an unterminated cable so that a custom connector may be installed. It is adjustable as to logic senses used and even logic levels used.

The serial interface takes the data from the I/O bus and serializes it into a stream of bits. Incoming serial data is converted to parallel data and sent to the computer. One type of connector is usually encountered, though not always. Many specialized control lines exist in this type of interface because serial I/O is found in the specialized data communications environment where special channels are used for long distance communications.

HP-IB is a relatively new interface standard. It is formally known as IEEE 488-1975 and the signals, connector, logic levels and logic senses are well defined. This interface allows connection to a large number of devices in a very simple manner; the cable connector is simply bolted to the connector on the peripheral device.

The hardware connection is not in question and the software for communication can be directly addressed. As a bonus, one interface can be used to service up to 14 peripheral devices. The HP-IB (Hewlett-Packard Interface Bus) is also known as the General Purpose Interface Bus (GPIB).

Older instruments use a different type of interface known as BCD. Data is dealt with four bits at a time to form numerals. This interface is used predominantly by instrumentation where the data to be transferred is only numeric.

In future articles, we will be examining these various interfaces in depth. We will find that in order to interface many devices to our I/O bus, we will be fighting problems of lack of information, faulty information and worst of all — faulty assumptions. In the process of this examination, we will develop the means for overcoming these problems and finally provide that "computer in a void" with a voice.

Digitizing system cuts multi-hour jobs down to minutes

Keyboard staff report

Installation of a digitizer, desktop computer, and printer system at a major California utility has dramatically reduced the labor input necessary to complete many jobs. Compared with manual drafting methods, the utility found the digitizer system at least four times faster.

When the Southern California Gas Company installed its digitizing system more than five years ago, it was expected that the system's capabilities could be expanded to meet future engineering requirements. Those expectations proved true, and today, the utility is undergoing an evolutionary step in computerized drafting.

Requires fewer man-hours

Since installation, the digitizing system has significantly reduced the number of man-hours required to scale millions of feet of gas mains on atlas sheets or tract maps. Formerly done by hand, the tedious procedure required a draftsperson to place a ruler over the gas main designated on the atlas sheet, note and record the footage, and total it at the end for each different size of pipe.

A Hewlett-Packard system consisting of a desktop computer, a digitizer and a thermal printer provided a convenient, time-saving way of electronically calculating the total pipe footage, and thus the utility's total dollar investment within specific tax and franchise districts. Last year, the gas company added a system-compatible 22-inch by 34-inch plotter to accommodate large tract maps.

Mapping automation program

Based on the success of this original program, engineering personnel began experimenting with additional programs designed to utilize the equipment to its full potential. The result of their experiments is MAP — Mapping Automation Program — combining many individual programs which were formerly separate functions.

Gas company workers labor to install a delivery line in the complex network needed to transport the fuel to customers. Workers depend upon accurate tract maps to make their efforts safe and efficient.

"We began by taking a look at our basic needs, and asking ourselves how we could adapt our digitizing, plotting and storage facilities to meet them," said Bob Coker, drafting supervisor. "Little by little, we built quite an extensive master program," he said.

Mike Bielec, a research engineer who designs the individual programs for MAP, broadly described some of the expanded system's present capabilities:

"First of all, we can digitize the information on a tract map, store it, and then re-plot it at different scales. We can permanently store the information on a cassette. We can receive a print-out of all the points that have been digitized, and we can automatically replot any portion of that information," Bielec said.

"It's really a multipurpose system; however, the digitizer itself is used primarily for mapping," said Coker.

The department has another HP desktop computer with plotter for additional engineering purposes, such as pipe network balancing, calculating

flow rate, pressure and temperature and interfacing with large computer systems to maintain a 23,000-page atlas-sheet inventory.

The digitizer assembly includes a surface area, hand-held cursor and electronics package. The gas company ordered a 42-inch by 53-inch working area board, large enough for tract maps, and an oversized cursor with 2-inch sight glass, making it easier to trace map points with the cross hairs.

Accepts data two ways

With the digitizer, there are two ways of feeding data into the desktop computer for analysis. In the menu approach, the draftsperson gives commands to the computer by placing the cursor over a special block on the digitizing board. Each block represents a preassigned function, such as pipe size, circle, dashline, delete point, etc. Then, as the desktop computer requests data from the digitizer via the ENTER statement, it receives an X-Y coordinate from the location of the free-moving cursor.

Point-of-origin may be set anywhere on the digitizing board. The digitizer can send data as a single sample or a continuous series of points. Coordinate units are transmitted in inches, and for most drafting programs a scale of 1 inch = 100 feet is used.

The other way of entering information is by digitizing the "CONSOLE" menu block, which transfers control to the keyboard on the computer.

Compensates for different scales

"By entering the appropriate command into the desktop computer, there are several things we can do," said Bielec. One unique function of the computer/digitizer combination is its ability to compensate for maps of different scales and proportions, when

Digitizing has made it possible to phase out 'scissors drafting' methods of maintaining extensive gas company tract maps.

it is necessary to combine information from several of them.

The utility receives tract maps from dozens of different sources, an engineer explained. Often, the maps are not exactly the same, even when they represent the same thing. For the draftsmen and designers, it's not simply a matter of enlarging or reducing a drawing to match-up the points. This is where MAP comes to the rescue, by automatically compensating for various X-Y scales as the information is digitized and electronically re-plotted.

To combine information from several source maps, each map is digitized in sequence. Menu commands, called out during digitizing, link the maps together. Throughout the process, accumulated data can be output to the plotter. Sometimes it is necessary to digitize as many as ten pages of information to form a single composite map.

Phasing out 'scissors drafting'

Said Coker, "Prior to this, we utilized 'scissors drafting' almost exclusively for this type of procedure. We had to photograph each page of information, scaling it up or down, cut out the various portions, mosaic them together and trace in their proper places. Now, using the digitizer, we can simply capture the information and feed it direcly onto our own company map in some cases.

"Although this program is still in the experimental stage, I would estimate that matching three maps with the digitizer is at least four times faster than doing the same job by the manual, photographic method," he said.

Leonard Verner, drafting assistant, uses the digitizer full-time for mapping purposes. He added that some jobs that would normally take 3-4 hours by hand can be completed in as little as 35 minutes with the electronic equipment.

Bielec said, "In addition to consolidating several maps into one, the MAP program allows us to do many other things with the information. We can delete points — a method of correcting errors. In addition, we can plot an arc by digitizing only three points. For instance, we could use the continuous mode and digitize perhaps 100 points. But by digitizing just three points, the computer will determine the circle that goes through those points and plot the arc, thus saving 97 points of storage.

"We can direct the plotter to draw dashed lines through any two points, instead of solid lines. The dashed lines represent private streeets," Bielec said.

Maintains a running inventory

Specific pipe sizes can also be assigned to the lines being digitized, in order to maintain a running inventory of pipe sizes within a franchise district. At the end of the program, the operator can ask the 9830 to inventory, and he'll receive a printout and total of each and every pipe size that has been digitized. This is

important for tax purposes, as different sizes of pipe are taxed at different values.

One new program leads to another, and gas company engineers are still experimenting with the digitizer's potential as they explore computerized drafting technology.

According to Harlon Striff, mapping and drafting supervisor, the system offers nearly unlimited opportunity for more efficient map processing and use of manpower.

Said Striff, "We have a very small staff and our workload is steadily increasing. The digitizer is presently used up to eight hours a day in order to handle the workload better and faster. We look forward to becoming more and more on-line with this important engineering tool as we develop new programs."

Mike Bielec is presently working on more programs and mapping shortcuts, with the goal of further eliminating manual drafting tasks. The digitizer plays an integral part in these programs.

For instance, a new program currently in the design stages will permit the draftsperson to dissect digitized data; a process currently done via the time-consuming scissors-drafting method.

The plotter has a capacity for only one map at a time, explained Bielec. "If the data being digitized must be allocated to several maps, the prototype program will dissect the data, store it on file and enable us to pull out the desired segment for each new map," he said.

For the Southern California Gas Company, the digitizer is already changing drafting technology as we know it. And as these engineers are proving, a little imagination can certainly go a long way.

"With our complete digitizing, plotting and storage facility, there's almost no end to what we can accomplish," Bielec added. "Our progress so far is but a hint of things to come." ◧

"We can digitize the information on a tract map, store it, and then re-plot it at different scales. We can permanently store the information on a cassette, receive a print-out of all the points that have been digitized and automatically replot any portion of that information."

Editor's Note: The system used by Southern California Gas is based on technology which has been updated by Hewlett-Packard in recent months. A new, state-of-the-art desktop computer/digitizing system would include the 9874A digitizer, described in Keyboard 1978/3, and either a System 45 or System 35 Desktop Computer.

While just now being installed world-wide, the new equipment represents an HP commitment to the science of digitizing and the applications which could benefit from it, such as described in this article.

Programming Tips

A structured approach to program overlays (9830)

by Tim D. Barringer, Data Processing Group Leader, Ames Research Center, Moffett Field, CA 94035 U.S.A.

Many of you with 9830 s have likely yielded to the temptation to write large programs requiring many overlays. If you have a mass memory, this temptation is increased and, as in this shop, you are routinely developing systems with five to 15 overlay segments. The straightforward or GOTO approach to structuring overlays results in code that looks like:

```
10 REM...MAIN PROGRAM
   .
   .
100 CHAIN "O'LAY1", 1000,
   1000
   .
   .
200 CHAIN "O'LAY2", 1000,
   1000
   .
   .
   etc.
```

The overlay module (after being loaded into memory) looks like:

```
1000 REM...OVERLAY1
   .
   .
1200 GOTO 110
```

There are two difficulties encountered with this structure.

1. The last GOTO in the overlay must be especially dealt with during program development. The whole overlay must be RENumbered beginning at the intended load point called out in the CHAIN or GET command to prevent renumbering during loading, and the final GOTO must be updated to the expected return point before storing this overlay module.

2. Each time the load point line number changes, the return point line number changes, or the overlay module is required at more than one place in the main program, the overlay module must be edited to account for these changes.

For only one or two overlays this structure is fine, but it quickly becomes burdensome after more than two overlays. An alternative approach to the structure of handling overlays is:

```
10 REM...MAIN PROGRAM
20 A=0
30 GOTO A+1 OF 40, 140 ...
40 REM...FUNCTION 1
   STARTS HERE
   .
   .
130 CHAIN "O'LAY1", 1000,
   500
140 REM...FUNCTION 2
   STARTS HERE
   .
   .
240 CHAIN "O'LAY2", 1000,
   500
   .
   .
500 A=A+1
510 IF A> number of
   modules to process
   THEN 540
520 GOSUB 1000
530 GOTO 30
540 END (or possibly GOTO
   20 if the program is to
   be automatically
   restarted)
```

This technique allows the programmer to insert, delete and renumber program lines in both the main and overlay modules without further editing or special handling of the code. The overlay module (after being loaded into memory) looks like:

```
1000 REM...OVERLAY 1
   .
   .
1200 RETURN
```

Using this method, we let the 9830's operating system take care of the return to the main program. However, the 9830 initializes its list of GOSUB returns on each CHAIN or GET command. Thus all hierarchy in program structure must be contained in the main program. This may be viewed as a benefit by structured programming buffs.

A two-level hierarchy might look like:

```
10 REM...MAIN PROGRAM
   .
   .
80 DISP "SUBSYS1,
   SUBSYS2, SUBSYS3";
90 INPUT A
100 IF A<1 OR A>3 THEN 80
110 GOTO A OF 120, 240, 350
120 REM...SUBSYSTEM 1
   STARTS HERE
130 B=0
140 GOTO B+1 OF 150, 160,
   170
150 CHAIN "O'LAY1", 1000,
   180
160 CHAIN "O'LAY2", 1000,
   180
170 CHAIN "O'LAY3", 1000,
   180
180 B=B+1
190 if B>3 then 220
200 GOSUB 1000
210 GOTO 140
220 A=A+1
230 GOTO 100
240 REM...SUBSYSTEM 2
   STARTS HERE
250 B=0
260 GOTO B OF 270, 280
270 CHAIN "O'LAY4", 1000,
   290
```

(continued)

```
200 CHAIN "D'LAY5", 1000,
    290
290 B=B+1
300 IF B>2 THEN 330
310 GOSUB 1000
320 GOTO 260
330 A=A+1
340 GOTO 100
350 REM...SUBSYSTEM 3
    STARTS HERE
    .
    .
    etc.
```

Even when using this method, the structure becomes quickly complicated if you attempt too many levels of hierarchy. However, this method maintains the advantage of a consistent pattern in the structure. No special modification of the code is required after simple editing, and the complexity is a function of the level of hierarchy, not the number of modules.

When using GET (or LOAD for cassettes) instead of CHAIN (or LINK), be sure to declare the module indicies A, B, etc., and include other pertinent information in a COMmon statement. K

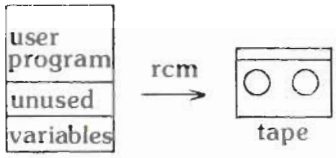
Temporary buffer (9825A)

Submitted by Don Albrecht, Ford Aerospace, Building 3C, Ford Road, Newport Beach, CA 92663 U.S.A.

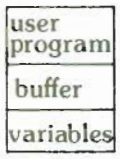
During the course of running a large program in a 9825A, it is sometimes desirable to allocate a temporary buffer, and, when the buffer is no longer needed, to reuse that area of memory. The diagram outlines the basics of the problem and its solution.

A sample program listing is also shown. This solution interrogates a data file containing either a "1", meaning the buffer is not currently allocated, or a "2", meaning that the buffer is allocated.

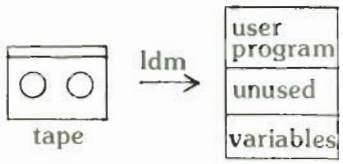
1. need temporary buffer:
Memory before buffer is neeeded.



2. establish buffer:
Buffer attached for I/O speed



3. buffer no longer needed:
Buffer is gone




```
0: "START":
1: cll 'DISPLAY
   AVAILABLE MEMORY'
2: rcm 3
3: ldf 2,A
4: if A=2; sto "MEMORY
   RELOADED"
5: buf "BUFFER", 15000,3
6: cll 'DISPLAY
   AVAILABLE MEMORY'
7: 2→A
8: rcf 2,A
9: ldm 3
10: "MEMORY RELOADED":
11: 1→A
12: rcf 2,A
13: cll 'DISPLAY
   AVAILABLE MEMORY'
14: stp
15: end
16: "DISPLAY AVAILABLE
   MEMORY":
17: dsp "MEMORY SIZE &
   AVAILABLE MEMORY ARE"
18: wait 3000
19: list -1
20: wait 3000;dsp
21: ret
*20749
```

K

Better label centering (System 45)

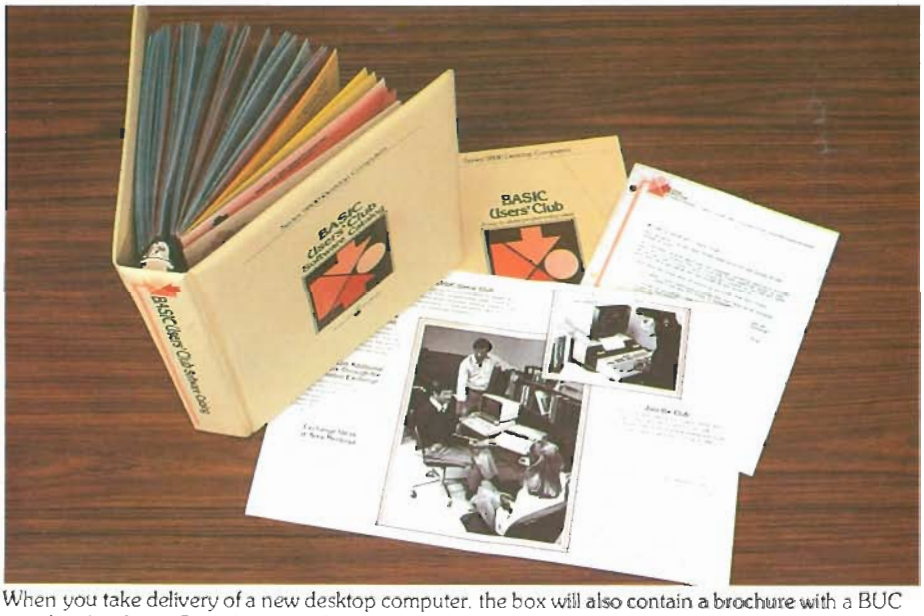
Submitted by Carl Johan Lamm, Astra-Draco AB, Fack, S-22101 Lund, Sweden

I am sorry to say that Label Centering on the HP System 45 should not be done the way proposed in Keyboard 1978/3. Ending a LABEL statement with a semicolon will cause the field to be buffered (as well as being output). Filling the buffer will ultimately lead to a linefeed being output, as is seen by running the following program.

```
10 PLOTTER IS "GRAPHICS"
20 GRAPHICS
30 SCALE 0, 20, 0, 20
40 FOR I=1 TO 10
50 MOVE I, I
60 LABEL "1234567890";
70 NEXT I
80 END
```

It is good practice always to use formatted output with LABEL. Change line 60 to LABEL USING "K"; "1234567890" and the program will run nicely. K

User's Club members get more than the basics



When you take delivery of a new desktop computer, the box will also contain a brochure with a BUC membership form. Completing the form and sending it to us will bring you a binder with descriptions of programs which may be useful to you. Many of these programs are written by other club members; others are translations of previously-available HP programs.

by Ron Mora, Ine Van Sabben, Hewlett-Packard Company, Desktop Computer Division

So you own a new desktop computer, and you have a job that you need to do with it. You find yourself thinking, "I know that this computer can do the work, but I need some special software for it. And I'd also like to find some odds and ends of programs that can save me time on other projects. I do some program writing, but I'd rather not have to start from scratch if somebody else has already been there before me."

Well cheer up. And join the club. A lot of other System 45 and System 35 owners have been in the same spot. Most of them are now members of the Hewlett-Packard BASIC User's Club. Belonging to the club costs you only the time necessary to fill out an application form. And once you're a member, you can submit a program that you've written to the club, and in return receive from the club three programs of your selection written by other members. Other programs are available for purchase from HP or from software suppliers through the club.

The Hewlett-Packard BASIC User's Club has evolved as an extension of HP's "after the sale" support to its customers. The club's objectives are to provide a single reference for all BASIC language software sources for the System 35 and System 45, to keep members informed of BASIC language programs and related HP equipment and inform and forward software information from software suppliers to the users of the System 45 or System 35.

Presently, there is no charge for membership in the club, and the benefits are numerous.

One reference source for software

This club is based on the successful experiences of the CUC (Calculator User's Club) for the 9825 and was started in Europe early in '78 at HP's Desktop Computer Operations in Böblingen, Germany. Each System 35 and System 45 shipped from HP includes a club brochure with a registration reply card which explains the club and invites the new owner to participate.

Members receive the benefit of having a single reference source for software available for the System 35 and System 45. Through the club and its operation, members share ideas with other System 35 and System 45 users. By attending area meetings, they can meet other members and see new equipment and software demonstrated.

Every member receives the software catalog which has a one-page description of each program in the club library. In each description is the relevant information concerning the particular program. This includes: program title, mainframe, program description (in English), language in which the program documentation is available, associated ROMs and peripherals, number of program lines, mainframe configuration and comments. Also included in the binder is a list of club rules, instructions for assigning application codes to submitted programs and submittal forms — everything needed for active club participation.

The programs in the library are classified in three categories:

User-submitted programs
HP-developed software
Programs from software suppliers.

A software exchange

The user-submitted programs are donated to the library by users, and are available on an exchange basis (submit one and receive three in exchange). The club requests that contributed programs be tested and proved in practice and be programs from which others are likely to benefit. The programs requested for exchange will be supplied as listings, or, if the

contributor supplies a blank initialized HP cartridge, the programs will be recorded and returned.

Information exchange

The club will publish information about programs from software suppliers in the software catalog from contributors who wish to make it available to other users under their own stipulated conditions. The description does not include the contributor's name — that is forwarded to other members upon request. The role of the club in the information exchange is that of coordinating information. Negotiations concerning availability, support, documentation, price, warranty, etc. will be the responsibility of the contributor and the user.

Hewlett-Packard software

HP-developed software is available by purchase through any HP sales office. The catalog includes descriptions of HP-developed software.

Newsletter

Each BASIC User's Club newsletter includes updates to the software catalog. The newsletter is also used as a channel of communication in keeping the members informed of new products, software updates and area meetings.

Updates of the existing catalog now are rapidly expanding the number of programs available to members of the BASIC User's Club.

The BASIC User's Club has two bases of operation. Europeans may write to Ine Van Sabben, Hewlett-Packard GmbH, West Germany. Other System 35 and System 45 users may contact the club by writing to Ron Mora, Hewlett-Packard Desktop Computer Division, U.S.A.

Ine Van Sabben
Hewlett-Packard GmbH
Werk Böblingen
P.O. Box 1430
Herrenbergerstrasse 110, D-7030
Böblingen, West Germany

Ron Mora
Hewlett-Packard
Desktop Computer Division
3404 E. Harmony Road
Fort Collins, CO 80525 U.S.A.

Keyboard

Hewlett-Packard, 3404 E. Harmony Road
Fort Collins, Colorado U.S.A. 80525
Phone: 303/226-3800

Publications Manager - Ed Bride
Editor - Bill Sharp
Editorial Assistant - Chris Stumbough
Art Director - Hal Andersen
Graphics - Wendy Hart
Typography - Paula Dennee

For further information on HP products or applications, please contact your local Hewlett-Packard Sales and Service Office or write to Keyboard

CHANGE OF ADDRESS: To change your address or delete your name from our mailing list please send us your old address. Send changes to Hewlett-Packard Keyboard, 3404 E. Harmony Road, Fort Collins, Colorado 80525, U.S.A.

March 1, 1979