A PROGRAMMABLE WAVEFORM CONTROLLER*

H. T. Yeh
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830

ABSTRACT

A programmable waveform controller (PWC) was developed for voltage waveform generation in the laboratory. It is based on the Intel 8080 family of chips. The hardware uses the modular board approach, sharing a common 44-pin bus. The software contains two separate programs: the first generates a single connected linear ramp waveform and is capable of bipolar operation, linear interpolation between input data points, extended time range, and cycling; the second generates four independent square waveforms with variable duration and amplitude.

INTRODUCTION

In the development of superconducting magnets for future fusion machines, various nonstandard waveforms are needed to simulate different types of heat-releasing events in the conductor. A dedicated programmable waveform controller (PWC) is needed to allow the easy entering of the desired waveform from a keyboard. Microcomputers seem to be the logical choice for dedicated real-time monitor/control applications.

Both single board microcomputers and microcomputer systems are commercially available. Single board microcomputers, although low in cost, are not easily expandable. However, microcomputer systems are more expensive, their turnaround time for repair is usually slow, and available options are limited. Thus, for reasons of economic constraints, flexibility for future development, and ease in hardware maintenance, we decided to build our own boards, using the Intel 8080 family of chips.

HARDWARE

The system hardware consists of a power supply, a bus, and several modular boards.

a. Power Supply

We used a power supply made by Power One, Inc. (Model HBAA-40W). It can provide 5 V at 3 A and ±12 V at 1 A. We added a voltage regulator for -5 V.

b. Bus

A common bus was chosen to simplify wire connection and signal tracing. The number of pins is minimized to 44. The pins are divided as follows: 10 for power lines (±12 V, ±5 V, ground), 16 for address lines (A0-A15), 8 for data lines (D0-D7), and 10 for control lines.

The bus system is based on a printed circuit socket with wire-wrap tails (we used Viking Industries, Inc. 2VH22/1AND3). Power lines are soldered to pins continuously with 18-gage solid wire. All other lines are wire-wrapped continuously with 28-gage wire with a Vector Electronics Slit-N-Wrap tool P180. A bus that can accommodate ten boards is shown in Figure 1.

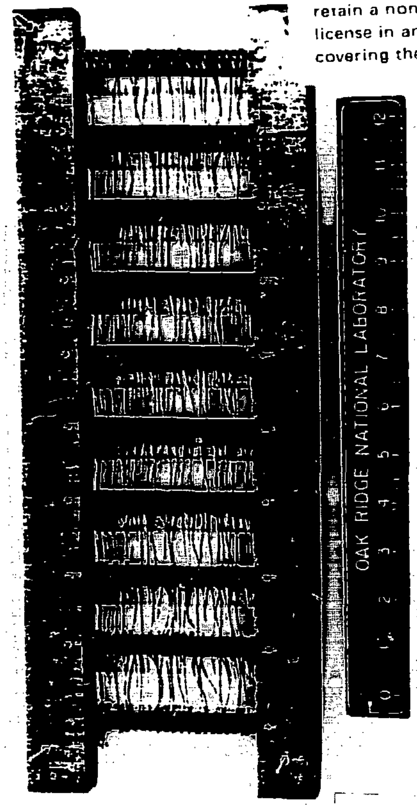


Fig. 1
The bus system

c. Modular Boards

There are two advantages in the modular board approach. One is ease of maintenance; the other is that one needs to put in only those boards needed for the particular application. Most boards are wire-wrapped, based on a circuit board made by Douglas Electronics, Inc. (22-DE-1). The circuit side of a 4K PROM (programmable read-only memory) board is shown in Figure 2. The printed circuit fingers are 0.156 in. center to center, and the board is 4.5 x 4.375 in.

The board was first lined with wire-wrap pins along the printed circuit fingers and perimeter. Then +5-V and ground lines were connected along the perimeter with 22-gage solid wires. Wire-wrap sockets were used for all IC (integrated circuit) chips to ease wire connection and replacement of ICs. The continuous Slit-N-Wrap technique was used whenever possible.

Ten boards are used for the PWC. They are: one CPU (central processing unit) board, one 1K RAM (random access memory) board, one 4K PROM board, one 2K PROM board, one TIMER board, one TTY (teletype) interface board, and four DAC (digital-to-analog conversion) boards. The designs of the CPU, RAM, PROM, and TTY boards are based on splitting a single-board microcomputer developed by Dodd and Connell[1] into multiple boards. All memory and I/O (input/output) boards have address decoding and board select on the board. Since we expect a small system, signal lines from CPU are not buffered. However, all TTL (transistor-transistor-logic) chips used are of the low power Schottky type.

The DAC board is based on Burr-Brown's 8-bit DAC90.

*Research sponsored by the Office of Fusion Energy, U.S. Department of Energy under contract No. W-7405-eng-26 with Union Carbide Corporation.

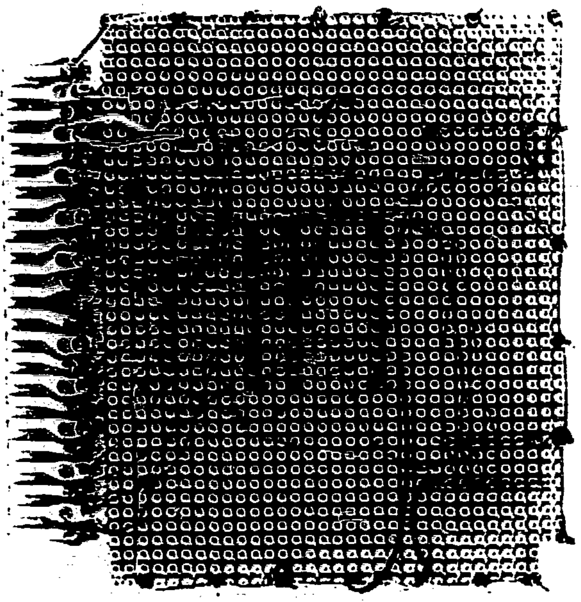


Fig. 2
Circuit side of a 4K PROM board

Because several units are required, a printed circuit board was made, as shown in Figure 3.

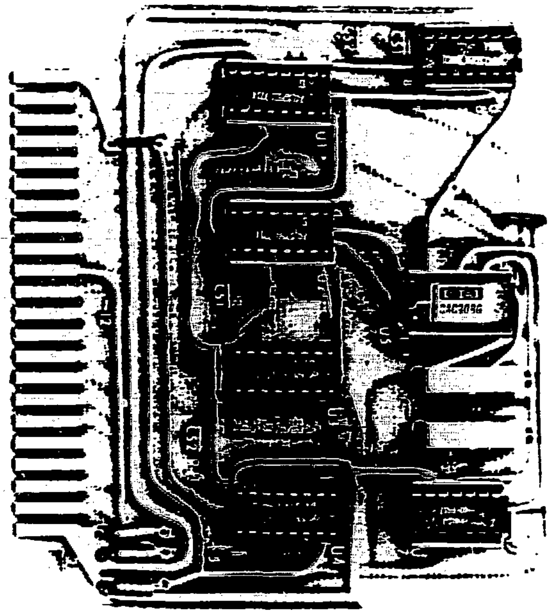


Fig. 3
8-bit DAC printed circuit board

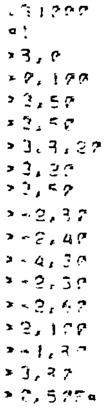
Programs are written in assembly language and assembled by a cross-assembler (Boston Systems Office, Inc. CA8080) available on the PDP-10 computer at Oak Ridge National Laboratory. In this way, a powerful editor on the PDP-10 is available, and subroutines and macro libraries can be built up and stored on magnetic tape. Information (data, object code, etc.) can be passed back and forth between the PDP-10 and the PWC by paper tape. A 2K resident monitor on the PWC controls paper tape read/punch and other system functions.

PROM programming is presently done by connecting a small circuit board to the TTL output port of an Intel SBC 80/10 single board computer Both the PROM programming program and the circuit are adapted from Ref. 1.

PROGRAM FOR A CONNECTED LINEAR RAMP

The first program generates a single connected linear ramp waveform. It consists of five modules: MAIN, INPUT, COMPUTE, CYCLE, and GO. After execution, the program start executes the MAIN module. It accepts a character from TTY and branches to the appropriate module if it is one of the command characters. Otherwise, it returns with an error flag.

The INPUT module reads in from TTY the desired waveform in the format of successive pairs of voltage values (in volts) and time instants (in milliseconds). An example of input is shown in Figure 4.



```
.G1000
*I
>3,0
>0,100
>3,50
>2,50
>3.3,20
>3,20
>3,50
>-2,30
>-2,40
>-4,30
>-2,30
>-2,60
>2,100
>-1,30
>3,30
>0,500F*
```

Fig. 4
Example of input for connected linear ramp program

The input string is terminated by the character F. Input numbers may be integers, floating point numbers, or scientific notation and signed or unsigned. A program by M. L. Bauer converts each input number into a 3-byte binary floating point number. Input characters are checked for legality. Errors are detected and can easily be corrected.

The COMPUTE module uses K. J. Caserta's 8080 floating point arithmetic package (Intel's INSITE library). The bit width of the output data is set by a constant in the program. COMPUTE first computes the number of equivalent digital steps from one input voltage to the next input voltage and then computes the duration of each digital step from the two consecutive time instants.

The program contains three software timers (40 μs, 1 ms, and 1 s). COMPUTE selects the appropriate timer according to whether the duration is greater or less than 1 s or greater than 60 s, and computes the number of repetitions required when using the selected timer.

This number and other relevant information, such as whether the step is a ramp up or a ramp down, are stored in RAM memory, arranged in order, so that they are easily accessible to the GO module.

The CYCLE module is optional. After COMPUTE, the desired number of cycles (up to 65,000) of the waveform can be entered from TTY by using the CYCLE module. The default is one cycle.

The GO module calls the appropriate timer and outputs the generated waveform data to the DAC board. The voltage range of output is ±4 V. The waveform generated by the input of Figure 4 is shown in Figure 5. The figure shows two cycles. The picture was taken using a PDP-12 data acquisition system. Duration between consecutive points is about 1.3 ms.

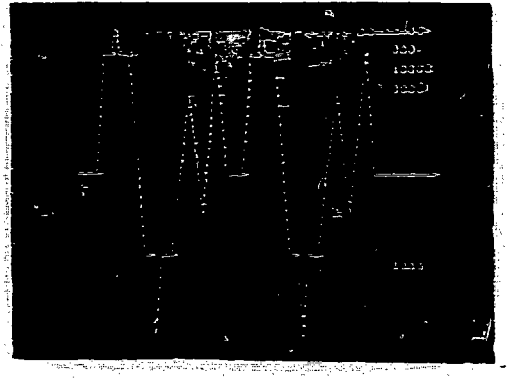


Fig. 5
Example of a connected linear ramp program

The arithmetic package and the decimal-to-binary conversion package occupy about 2K of PROM. The connected linear ramp program occupies about 1K of PROM.

PROGRAM FOR FOUR SQUARE WAVEFORMS

The second program generates four channels of independent square waveforms, each with variable duration (0.5 ms to 8 s) and amplitude (-4 to +4 V). This program occupies 285 bytes of PROM. The crystal clock signal from the CPU is used with three 74161 4-bit binary counters to generate the 0.5-ms timing pulse. This in turn is used to activate the system interrupt.

At program execution, the program first initializes various address pointers and stores the initial duration and magnitude in registers and designated memory locations, ready to be used for output. Then it enables the interrupt and enters a wait loop for the interrupt. After each interrupt, in order to maintain accurate timing, output is sent to each of the form channels immediately. Then the program proceeds to update the output.

To update the output, the program first checks whether all channels are done. If not, then it checks each channel in turn to keep track of the remaining number of timer cycles for the present output. If all cycles are done, it updates counters for the new output and timer cycles. After this, it returns to the wait loop for the next interrupt. It is not necessary to save registers after the interrupt as all computations are done within 0.5 ms, even under the worst combination of input data. Input data for each channel are input from TTY into preassigned RAM locations. Consecutive pairs of bytes are input, the first byte for duration (number of 0.5-ms cycles) and the second for magnitude. Each channel can accept up to 64 pairs of input data. A FF indicates the last pair of input for the channel.

An example of output is shown in Figure 6. Only two of the four channels are shown in this photo. The time scale is 0.5 ms per division. The voltage scale is 5 V per division. The corresponding input is shown in Figure 7. Values stored at 3000 (3E80) correspond to the lower (higher) waveform in Figure 6. For voltage values, 00 represents -4 V and FF represents +4 V.

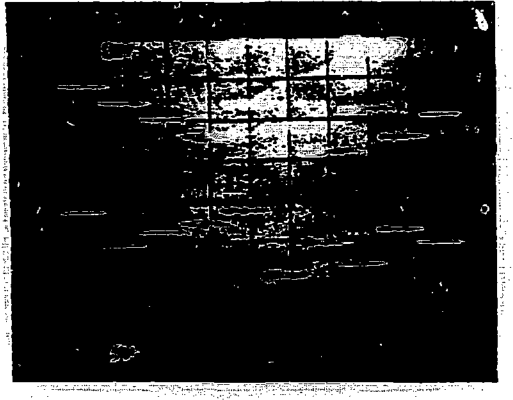


Fig. 6
Example of multiple variable square waveforms

```
.D3D00,3D09

3D00 41 FF 41 33 41 34 41 00 41 FF
.D3D0A,3D13

3D0A 41 34 41 44 41 33
3D11 41 44 FF 44
.D3E80,3E89

3E80 41 FF 41 C0 41 44 41 43 41 C1
.D3E8A,3E93

3E8A 41 54 41 11 41 41
3E91 41 34 FF 4A
.
```

Fig. 7
Example of input for multiple variable square waveforms

ACKNOWLEDGMENTS

I want to thank my colleagues M. L. Bauer, C. V. Dodd, J. W. Pearce, and S. S. Shen for much helpful advice and numerous consultations. I also want to thank Paul Rudd for technical assistance in fabricating the hardware.

REFERENCES

1. C. V. Dodd and G. D. Connell, *The NDT-COMP8 Microcomputer*, ORNL/TM-5773, Oak Ridge, Tennessee (March 1977).

Hsiang T. Yeh, research staff of Fusion Energy Division, Oak Ridge National Laboratory, received a B.S. in physics from National Taiwan University in 1960 and a Ph.D. in physics from the University of Illinois in 1967. Currently, he works on real-time monitoring, control, and protection of fusion superconducting magnets.

"Reference to a company or product name does not imply approval or recommendation of the product by Union Carbide or the U.S. Department of Energy to the exclusion of others that may be suitable."