SLAC-PUB-2973
September 1982
(I)

A MICROPROCESSOR-BASED CAMAC ISOLATED DIGITAL OUTPUT MODULE*

John Kieffer
Stanford Linear Accelerator Center
Stanford university, Stanford, California 94305

ABSTRACT

This paper describes a 32-channel digital output module, and the development of the hardware and software design. The microprocessor-based design allows versatility in output control in response to CAMAC commands. The unit may be tailored to different specific applications by modification of the internal applications software.

CAMAC commands, which may be stacked for serial micro-processing, allow outputs to be selectively DC or pulse controlled, and to control other outputs at the start or end of a pulsed output. Output pulse width may be pre-defined, or set within a range of values. Various outputs may be responding to any of the above commands concurrently. Outputs to be pulsed (on or off) may be preset with width and trigger options, with execution deferred. Action may be later initiated on a set of channels using a selective pulse command to initiate execution of the preset sequences.

INTRODUCTION

A 32-channel digital output module with some special features has been developed for use in a control environment. The features, and the reasons for them, are as follows:

1. The unit uses an 8751 microprocessor. The unit may be tailored for many different applications by modifying the applications portion of the control program stored in the EPROM of the 8751. Hardware changes are not required.

2. Any output may be DC or pulsed (on or off). The width of any pulse may be loaded via CAMAC, or assume the 250 millisecond (program variable) default value. A basic timer interval of 25 milliseconds (also a program variable) allows a pulse width range selection from 25 milliseconds to 6.375 seconds.

3. Using the microprocessor allows preloading a sequence of output functions. Execution may be later initiated by a single CAMAC command. An output channel may be loaded, via CAMAC, with a command for action to take place upon another channel at the time of the next occurrence of the start or stop of the pulse time (as specified) of the controlling channel.

4. CAMAC Write commands are buffered via a FIFO. This allows full CAMAC speed setup of the unit, without the delays which might be caused by the slower response time of the microprocessor.

5. CAMAC read and clear commands are executed immediately (within the current CAMAC cycle). Thus there is no delay when requesting status, or performing a general clear or initialize command.

---

* Work supported by the Department of Energy, contract number DE-AC03-76SF00515.

The unit is a single width CAMAC module. Intended for general use as an equipment controller, each output is rated for switching loads of up to 75 volts and 100 milliamperes.

DEVELOPMENT

The Isolated Digital Output Module (IDOM) is a microprocessor-based CAMAC module. The IDOM was designed for use by the Stanford Linear Collider project (SLC) at the Stanford Linear Accelerator Center (SLAC). The ability to preload and then initiate the execution of a sequence of variable width output pulses, makes this module useful as a process control sequencer. The outputs are arranged in two groups of sixteen, each group being brought out from the module through a 36-pin front panel connector (see Figure 1).

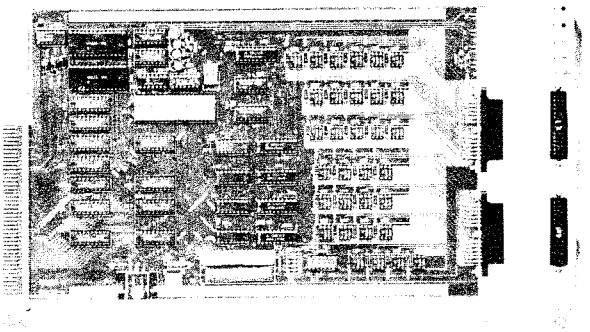

Figure 1: Component Side and Front views of the IDOM.

The preliminary specification required either a DC or 1/4 second pulse output. This could have been simply done if the selected outputs in a group could have been group pulsed. However, there was not sufficient circuit board space for the components needed to allow independent pulsing of the 32 outputs, and that requirement became part of the expanded specification.

Once the decision was made to use a microprocessor, it became apparent that the flexibility of the unit could be considerably expanded. This involved no increase in component count over that required to meet the specification. The increased flexibility was provided by the expansion of the microprocessor operating program.

An Intel 8751 microprocessor was selected primarily for three reasons: 1) the software development support was already available at SLAC for the Intel 8080 series processors, 2) this microprocessor includes both RAM and EPROM on the same chip, thus reducing component count, and 3) the 8751 operates faster than the 8748 (already in use in other SLAC designs). The extra RAM and EPROM of the 8751 (twice that of the 8748) allowed space for the expanded software needed to implement the output flexibility of the IDOM.

(Presented at the Nuclear Science Symposium, Washington, D.C., October 20-22, 1982.)

IDOM outputs are individually opto-isolated, with each output fully floating. The outputs are specified to switch 75 VDC, 100 milliampere circuits. By changing the opto-isolators to another unit of the same family, the open-circuit switching voltage may be increased to 100 VDC.

CAMAC CONTROL

Each 16-channel output group may be overwritten, selectively set or cleared, selectively pulsed on or off, reset or, read back using various CAMAC commands. In addition, the entire module may be reset by either a module specific or Crate Initialize command. (See Appendix for a full listing of the implemented CAMAC commands.)

Two additional CAMAC commands are implemented, both related to pulsed output and each relating to one channel at a time. The first allows setting variable ouput pulse widths, and the second is used for loading the sequence information needed for Pulse Transfer Mode operation (see Figure 2).

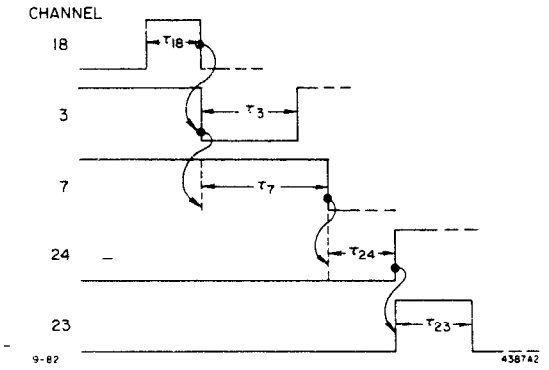

Figure 2: Typical Pulse Transfer Mode Operation.

The output pulse width is dependent upon several factors. These are the basic time unit (TU, equal to n counts of the crystal-controlled microprocessor clock), a default output width (mTU) which is defined by the program in EPROM, and a CAMAC command which may be used to load a count (in the range of 1 to 255) for a specific output channel. As presently implemented, when a count has been loaded for a given channel, it is used to determine the width (count times TU), for the next pulsed output only, for that channel. If no count has been loaded prior to a pulsed output trigger, the width becomes the default value mTU.

Additional data contained in this command allows setting the pulse polarity (i.e. 'on' or 'off') and also either generating the pulse immediately, or deferring the pulse until a later trigger.

The unit may also be used to perform a delayed switch transition, in the following manner: Assume that an 'on' output is commanded to pulse on. At the time of the command there is no change, as the output was already on. Then, at the end of the pulse period, the output would switch off, thereby effecting a delayed output. An 'off' channel, pulsed off, would effect a delayed turn-on.

Pulse Transfer Mode operation allows any pre-selected pulse to trigger any one other pulse, and the trigger may be selected to occur on either the leading or trailing edge of the pulse, regardless of pulse polarity. This command loads a channel with the number of another channel to be triggered, and a bit indicating the edge on which to generate the trigger. All execution is deferred, and must be initiated by either a subsequent command, or the trigger from another channel.

INTERNAL OPERATION

Command decoding is done by Field Programmable Logic Arrays (FPLAs). (See Figure 3.) The FPLAs either select immediate execution of a command (Reads, Resets, and Test Status), or load the command and related data into a First-In-First-Out memory (FIFO). X and Q responses are generated for each command as received, and each command is either executed or loaded into FIFO, within a normal CAMAC Dataway Cycle.

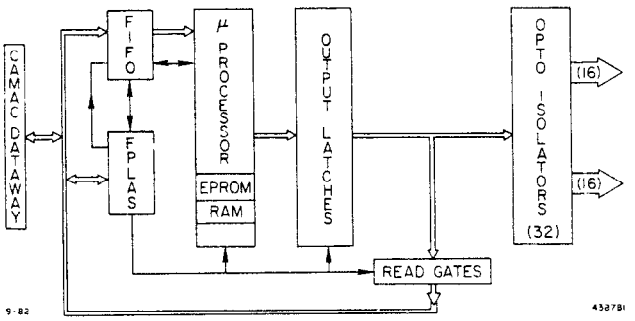

Figure 3: Block diagram of the module circuit.

As implemented, the FIFO is 16 words deep. (Some special applications could possibly utilize a deeper FIFO structure.) This allows a sequence of at least 16 commands to be streamed to the module at maximum dataway rates, for execution by the slower microprocessor. Attempting to overwrite the FIFO will cause a Q response of 0.

The microprocessor is interrupt-driven by a signal generated by the FIFO when a command is ready at the FIFO output. The microprocessor accepts and completes operation on any given command prior to clearing the FIFO output and re-enabling the microprocessor interrupt.

The TU pulse counter is also interrupt-driven to insure reasonably stable pulse widths, regardless of other microprocessor activity. This interrupt is always enabled, and has been assigned the highest processing priority. Each time this interrupt occurs, the RAM counter for each active output pulse is decremented, and tested for zero. Transition of a counter to zero causes termination of the related output pulse.

As approprate, the latch for each output is set on or off by the microprocessor. The microprocessor does not read back the output latches, but instead keeps a copy of the output pattern in RAM. A CAMAC read of the outputs does read the latches, showing the data present at the time of the read. Note that with no additional CAMAC commands, the output data may change subsequent to (or during) a read, should the microprocessor have operations pending. No provision has been made for the readback of pending commands. It is the responsibility of the host CAMAC controller (usually a computer) to keep track of the sequence of pending commands. If the IDOM is to be used in a multi-host CAMAC environment, host proto-

cols must be established which will insure that proper command sequencing of outputs is maintained where necessary.

APPLICATIONS

The IDOM was designed primarily as a controller for electro-mechanical and pneumatically operated devices (solenoids, servos, motor drives, etc.). It may be used simply as a digital switch. The floating outputs allow utilization directly as the trigger controller for 120 VAC SCR circuits*.

The flexibility of the microprocessor and EPROM makes the IDOM useful in applications requiring specialized sequencing. The microprocessor may be programmed with an expanded, or even an entirely different, set of output control instructions. For example, a complex sequence of output control may be pre-programmed with the execution being later initiated by a single CAMAC command. As this requires no hardware design changes, the maintenance of many different hardware sequencers becomes easy. All modules are identical in hardware, and it requires only the insertion of the microprocessor chip containing the desired program to configure a spare unit as the replacement for a special controller/sequencer.

Some special applications may even require that the IDOM actually respond to a modified set of CAMAC commands. The unit will easily accomodate these requirements by user reprogramming of the FPLAs as well as the 8751. Other than the chip exchange, no hardware modifications would be needed.

Some thought has been given to the possibility of on-line program loading of the IDOM, although this feature has not been included in the present units. The addition of a single RAM integrated circuit would allow the implementation of this feature. With care in the selection of the devices, an additional socket could accept either RAM or EPROM. This could mean that all units could contain the same basic program in the 8751 EPROM. Custom tailoring could then be done by either the insertion of a pre-programmed EPROM, or by down-loading a custom program from the host controller to RAM. The transfer to the custom RAM or EPROM program could then be initiated by the execution of a single CAMAC command.

ACKNOWLEDGEMENTS

Special thanks to Eric C. Yuen, a 1982 SLAC Summer Science Program participant, who wrote the development microprocessor program. Thanks also to R.S.Larsen and C.A.Logg for their editorial comments, and to L.Paffrath for both editorial comments and design review.

REFERENCES

1. SLAC Drawing Package 135-568 (complete documentation set).

2. SLAC IM-135-568-00 (Specifications and Instruction Manual only (a part of the complete documentation)).

---

* If the IDOM is used for 120 VAC control, it is suggested that a 16-channel output group be dedicated for this application, and not mixed with low-voltage DC circuits.

APPENDIX

CAMAC codes, X and Q responses:

1. Read functions:

   a. F(0)A(0): Read channel 0-15 output latch, X=1, Q=1 if J1 & J2 pins C9 are both above +12 volts.

      # R1-R16 = Channel 0-15, 1=on, 0=off,
      # R17 =1= J1-C9<+12 volts,
      # R18 =1= J2-C9<+12 volts,
      # R19 =1= FIFO input not ready (full),
      # R20 =1= FIFO data present (waiting for processing).

   b. F(0)A(1): Read channel 16-31 output latch, X=1, Q=1 if J1 & J2 pins C9 are both above +12 volts.

      # R1-R16 = Channel 16-31, 1=on, 0=off,
      # R17 =1= J1-C9<+12 volts,
      # R18 =1= J2-C9<+12 volts,
      # R19 =1= FIFO input not ready (full),
      # R20 =1= FIFO data present (waiting for processing).

   c. F(1)A(0): Read status data only, X=1, Q=1.

      # R1 =1= J1-C9<+12 volts,
      # R2 =1= J2-C9<+12 volts,
      # R3 =1= FIFO input not ready (full),
      # R4 =1= FIFO data present (waiting for processing).

2. Write functions:

   NOTE: For all write functions described below: X=1; Q=1 if a) the voltage on pin C9 of both output connectors is above +12 volts, and b) the unit command input FIFO has space to accept the present command, and can store that command for action; else Q=0.

   a. F(16).A(0): Write the W1-16 bits to channels 0-15, bit =1= output on, bit =0= output off.

   b. F(16).A(1): As above, to channels 16-31.

   c. F(17).A(0): Write an output time interval to a specified channel.

      # W1-W5 = channel#, 0 thru 31, encoded.
      # W7 = polarity, 0 = pulse off, then on; 1 = pulse on, then off,
      # W8 = execute, 1 = defer execution, 0 = execute immediate,
      # W9-W16 = time units, 1 thru 255 decimal, of the program encoded basic time interval (nominally 25 milliseconds).

   d. F(17).A(1): Define a transfer channel (channel to be triggered).

      # W1-W5 = channel#, 0 thru 31, to initiate transfer, encoded.
      # W8 = transfer polarity, 1 = transfer at the pulse start time, 0 = transfer at the end of the pulse time, of this channel,
      # W9-W13 = channel#, 0 thru 31, to be triggered, encoded.

Action to be taken as defined by a previous F(17).A(0) command to that channel. If undefined, the default is to pulse on for the default time interval.

e. F(18).A(0): Selective set channels 0-15 as defined by W1-16, bit =1= output on, bit =0= no change.

f. F(18).A(1): Selective set to channels 16-31, as above.

g. F(19).A(0): Pulse on selected channel 0-15 output(s), as defined by W1-16, bit =1= pulse output, bit =0= no change.

If the output is off, it will pulse on for the defined time interval. If the output was on prior to the command, it will go off after the defined time interval.

The pulse period may be set by a prior F(17).A(0) command. If it has not been preset, the default pulse period is used.

h. F(19).A(1): Pulse on selected channels 16-31, as above.

i. F(21).A(0): Selective clear channels 0-15 as defined by W1-16, bit =1= output off, bit =0= no change.

j. F(21).A(1): Selective clear to channels 16-31, as above.

k. F(23).A(0): Pulse off selected channel 0-15 output(s), as defined by W1-16, bit =1= pulse output, bit =0= no change.

If the output is on, it will pulse off for the defined time interval. If the output was off prior to the command, it will go on after the defined time interval.

The pulse period may be set by a prior F(17).A(0) command. If it has not been preset, the default pulse period is used.

l. F(23).A(1): Pulse off selected channels 16-31, as above.

3. Command functions:

a. F(9).A(0): Clear channel 0-31 outputs (all off), clear the FIFO, clear and reset the microprocessor, X=Q=1.

b. F(10).A(0): Clear channel 0-15 outputs, X=1, Q=1 if the FIFO has room to store the command.

c. F(10).A(1): Clear channel 16-31 outputs, as above.

d. F(27).A(0); test status, X=1, Q=1 if both a) the FIFO is not full, and b) both J1 & J2 C9 voltages are above +12 VDC. If Q=0, the status of the individual conditions may be determined using any of the Read commands above.

4. Other X and Q responses: X=Q=0 for all other codes and conditions.

5. C, Z, and I: Z.S2 shall clear all registers and outputs. C and I have no effect on this module.