Bulletin of the Psychonomic Society
1983, 21 (6), 473-475

APES: A digital-circuit simulation program for real-time control of behavioral and physiological data collection

RONALD N. EHRMAN, CHARLES P. O'BRIEN, and J. W. TERNES
Veterans Administration Medical Center, Philadelphia, Pennsylvania

This paper provides a brief users' description of an all-purpose experimental system, APES, which was developed for use in a microprocessor-controlled behavioral pharmacology laboratory. APES simulates the functioning of the modular electromechanical and solid-state digital circuitry commonly used to control experiments. Descriptions of the simulated modules are provided.

Recently, we described an assembly language program that can be run on any of the DEC PDP-11 processors under an RT-11 single-job operating system (Ternes, Ehrman, & O'Brien, 1982). This system is designed to simulate and replace most of the modular electromechanical digital circuits commonly used to control psychology laboratories. Since it can support almost any common psychopharmacological or psychophysiological paradigm, it has come to be known as the "all-purpose experimental system" (APES).

APES was specifically written to provide real-time control of a series of psychopharmacological experiments with monkeys. The system provides simultaneous control of up to eight behavior-conditioning schedules, controls eight infusion pumps on fixed schedules, and monitors physiological data for heart rate, respiration, and skin temperature for up to six animals. All of these functions are performed simultaneously. APES also has a flexible scheme for storing data in a machine-readable format for later statistical analysis or graphics construction.

The schedule format is flexible enough to allow implementation of any Pavlovian or operant-conditioning schedule. This includes accurate timing and provisions for handling signals both entering and leaving the computer. Perhaps the most attractive feature of APES is that the program is interactive and permits individuals who have no computer programming experience to program conditioning schedules and to implement experimental protocols.

APES operates as a software simulation of digital circuitry. A digital circuit requires both a set of devices to wire and a set of wires. The devices that give APES its distinctive character are simulations of electromechanical and solid-state components such as AND gates, OR gates, flip-flops, counters, and timers that are typically used in the psychological laboratory. These devices are called modules. In APES, these modules are conceptual rather than real. They have what can be thought of as syntax and semantics. The syntax consists of a list on connection points, and the semantics is a description of how the outputs of a module change as a function of changes to its inputs.

MODULES

There are four general types of modules incorporated into the APES system. These include timing, state, counting, and miscellaneous modules. A brief description of the specific modules available within each general type follows.

Timing Modules

UNIVERSAL TIMER. The UNIVERSAL TIMER is capable of timing intervals from 100 msec to 45 days. It provides an output while the timer is active as well as a 50-msec end pulse when the timing cycle is complete. The module has six inputs: operate, repeat operate, reset, hold, stop, and inhibit.

TIME BASE. This module, when enabled, provides a steady stream of pulses at any one of five rates: 1, 2, 5, 10, and 20 Hz.

State Modules

AND. The AND module is organized into three separate three-input gates. Any input of an AND gate that is left unconnected is assumed to be high. Thus, a two-input AND gate may be simply constructed by using only two of the inputs to a three-input AND gate.

OR. Similar to the AND module, the OR module is organized into three separate three-input gates. Unlike the AND module, any inputs left unconnected are

This project was supported by Veterans Administration Merit Review grants to J. W. Ternes and C. P. O'Brien. The authors' mailing address is: Drug Dependence Treatment and Research Center, Veterans Administration Medical Center, Philadelphia, Pennsylvania 19104.

assumed to be low. Therefore, a two-input OR gate may be constructed by only using two inputs.

FLIP-FLOP. The FLIP-FLOP module is a two-position internally controlled switch that has five inputs and two outputs. The outputs are the normally low and normally high value of the switch. The inputs provide means to set, clear, and toggle the value of the outputs.

STEPPER. The STEPPER module is a multiple-position switch that has 16 separate positions. Its only output is a single high level that can move in either direction from one output position to the next as a function of an input to the step-up or step-down operates of the module. When the output reaches either its maximum or its minimum travel, the output wraps around. The output may be prevented from appearing on any of the output positions by applying a high level to the inhibit input. In addition, a positive assertion on the reset input of the module causes the output to jump directly from its current position to Position 1, without affecting other output positions.

Counting Modules

PREDETERMINING COUNTER. This module counts down from its preset value to 0. The preset count can be arbitrarily set from 1 to 65,535. Upon reaching 0, the done output of the module is asserted, after which the counter automatically resets to the preset value. In addition to the count input, the module has two additional inputs, reset and hold.

INCREMENTAL COUNTER. This module consists of one five-digit counter that can be used to tally the number of occurrences of an event. It has no logic-level outputs but four inputs: count, reset, enable, and log. The log input, when asserted, deposits the current value of the counter into the permanent data file.

Miscellaneous Modules

PROBABILITY GATE. The functioning of this module is straightforward. Some preset proportion of pulsed inputs to the module appears as high-level pulsed outputs. The module has one parameter that is under program control: the probability that an input will appear as an output. The probability ranges from 0% to 100%.

SYSTEM. The SYSTEM module is the entity through which a behavioral schedule communicates with the outside world. It is analogous to an external experimental chamber, in that it has inputs such as lights and tones and outputs such as barpresses. At the present, the module has 15 input and 15 output channels. When a schedule is used, the software input and output channels are associated with the hardware input and output channels on the parallel interface boards of the computer by means of an interactive dialog. Theoretically, the user can connect any or all of the SYSTEM module inputs and outputs to any interface pins he or she desires.

INITIAL. The INITIAL module may be thought of as a power-up pulse. It has one output and no inputs. The output produces a single pulse (low-high-low) as soon as the schedule is initiated. Its function is to set the desired initial state of the various modules in a configuration.

PHYSIOLOGICAL. This module is used to automatically turn physiological data collection on or off for an individual experimental chamber. The physiological module allows the experimenter to use the timing facilities available in a behavioral schedule to control physiological data collection.

FUNCTIONAL AREAS

In addition to the configuration modules, there are four functional areas that the user is concerned with.

Library File

Behavioral schedules are stored in the Library File. In general, every schedule is made up of two parts: a wire list and a configuration. The configuration is simply a set of modules needed to develop the schedule circuitry. A wire list describes the set of interconnections between the configuration modules that define the circuit. The Library File has the capacity to store a maximum of 256 different independent configurations, with up to 256 separate wire lists associated with each configuration.

Box Area

The Box Area is where a schedule resides when it is being used. It is divided into a number of segments called box sections. The number of box sections available at any one time is a parameter that typically is set to the number of peripheral devices being controlled. We generally assign a number to each of our experimental chambers, and let each one correspond to a particular box section. However, within a box section, it is possible to run more than one schedule simultaneously. Furthermore, more than one experimental chamber can be controlled by schedules within a single box section. The upper limit to the number of chambers controlled in a single box section is determined by the maximum number of inputs and outputs available on the system module.

Work Area

The Work Area has three basic functions. First, it acts as an intermediate point for all schedule transfers from the Library File to the Box Area. Second, it is the place where all schedule creation and editing is performed. Finally, it is general-purpose scratch area for APES operations.

Log File

The Log File stores compactly various types of information, including all physiological and behavioral data. Important information about commands that affect the box sections are also stored here when the commands are issued. The result is a record of which schedules were loaded into which boxes, when each box was started and stopped, and when physiological data collection began and ended. In addition, the onset and/or offset of any event, response, or stimulus, and the time each occurs, can also be stored in the Log File.

SCHEDULE PROCEDURE

The normal procedure for entering and starting a schedule is as follows:

1. The user creates a configuration in the work area. At this point, space is allotted for it and the configuration is given a name.
2. The user adds modules. He or she adds as many of whatever types are needed, taking care that the first module in the configuration is a SYSTEM module and that there is no more than one INITIAL module per configuration.
3. The user saves the configuration. This makes it a library configuration, and it is no longer modifiable.
4. The user creates a wire list under the current configuration (i.e., the configuration in the work space). At this point, the empty skeleton of the wire list is created, and the wire list is given a name.
5. The user connects wires. He or she makes connections between the outputs and inputs of the configuration modules. He or she is not allowed to connect outputs together, since that may cause an indeterminate state at the point of junction.
6. The user saves the wire list. This makes it a library wire list, and he or she will no longer be able to modify the connections.
7. The user sets a current box section with the box section command. The following commands apply to that section.
8. The user loads the section with the schedule he or she has just created. At this time, connections between the SYSTEM module and the interface pins are made. The section may be loaded even if the user has omitted Step 6, because it is possible to have a nonlibrary wire list in a box section. However, it is not possible to use a nonlibrary configuration in the box section because such a configuration cannot have had a wire list associated with it.
9. The user starts the schedule running with the Run Schedule command. During the running of a schedule the experimenter can: (a) issue the Watch command, which displays the status of all incremental counter modules in all box sections that are currently loaded; (b) change parameters in any module by using the Set Parameter command; and (c) change the type of information being stored in the log.

To facilitate training operators in the use of APES, we have developed a users' manual. This manual details the command set, the syntax, and the semantics of each module. Diagrams and explanations of each module type, as well as examples of the circuitry that can be used to run simple behavioral studies, are provided. Details of the interactive process and the rules involved in creating, debugging, and using APES for real-time experiments and for data storage are also provided.

REFERENCE

Ternes, J. W., Ehrman, R., & O'Brien, C. P. A software system for real-time control of psychological experiments. *Behavior Research Methods & Instrumentation*, 1982, **14**, 160-164.

(Manuscript received for publication August 4, 1983.)