USING THE MICROPROCESSOR TO TAILOR COMPUTER SYSTEMS TO TRAINING SIMULATOR REQUIREMENTS

NICHOLAS A. SIECKO
Educational Computer Corporation

INTRODUCTION

The advent of the microprocessor has ushered in a new era of computer applications. The power and economy of the microprocessor opens entire new fields for computer applications and will change many existing techniques that now use mini and full scale CPU. It is this power and combined economy that allows the microprocessor to be tailored to each particular application rather than fitting system requirements to existing processors.

Educational Computer Corporation's use of multiple microprocessors in a "Federated Multiprocessing Architecture" is an example of tailoring the processing system to fit the application. It also represents changing an existing minicomputer application. The EC 3, as was its predecessor, the EC II, is specifically designed to accommodate the requirements of simulator operation. However, the microprocessor has provided the EC 3 a multifold increase in capability and power, plus a decrease in mainframe hardware costs.

Though the hardware cost has been reduced and subsequent software operational languages have, as a result, been made more efficient and powerful, the cost of preparing the simulation, if not controlled, can become much greater than in previous applications simply because of the added or increased capability of the simulation system. This requires that the simulation requirements specifications be precisely controlled.

The range of requirements that were considered in the design of the EC 3, the resultant microprocessor system structure, and some of the applications to which it has already been put are also discussed.

VERSATILITY OF THE MICROPROCESSOR

The microprocessor has often been heralded as a device that will create our next technological revolution. It is already replacing mechanical timers and integrators. It will change engineering application techniques and allow the development of new devices not economically feasible before. Some of this is already happening. We've already seen the cost of hand-held calculators come down while their function capacity increases; video games, children's toys, the home computer, and educational games are the more obvious commercial applications. In the industrial area there is a less obvious, but nonetheless broad application of the microprocessor: the auto industry, cameras, machine tool design, word-processors, test instrumentation, portable devices of all kinds. In the military there is a similar growth of application, especially in test devices, electronic systems, avionics, and training equipment.

This wide-spread application is made possible by the combination of computational power, size, and low cost of the microprocessor. As many as are needed can be put almost anywhere and without significant expense. This is the significant feature of the microprocessor -- its applications adaptability. Processing requirements can be more readily tailored to meet application requirements than is possible with processors available up till now -- namely the minicomputer.

APPLICATION AT EDUCATIONAL COMPUTER CORPORATION

The microprocessor has provided Educational Computer Corporation a means to capitalize upon our original concept of simulator design. The predecessor to the EC 3, the EC II, utilized a special purpose mini-computer of our own design. It was a logic processor, akin to a sensor based, or process-control system, capable of handling some 96 inputs and 96 outputs within a 58 microsecond cycle time frame. It was not a commercial, scientific or data processor. It was unique in that it was designed exclusively to support our simulator requirements. The software was also unique in that it related directly to the functions of the simulation which demands a diversity of input/output (I/O) and a high degree of logic complexity.

The EC 3 is both a continuation and expansion of that philosophy, but instead of being limited by the processor, the processor is now a system utilizing multiple processors that support all the requirements prescribed for the simulator. The microprocessor has also enabled the projection disk slide capacity to be increased from 100 to 150 with an access time of less than 500 ms.

SIMULATOR DESIGN CONSIDERATIONS

In designing a training simulator a choice has to be made by the designer. Either he choses off-the-shelf, commercial hardware and its accompanying software, or he makes his own. Choosing the former he has a computer, but it is a data processor, and not optimum for what he really needs It can do what it is designed to do, but can't do enough of what is required as a simulator controller. The greatest problems occur between internal and external operations and the "systems overhead" incurred by using available high level languages.

The simulator designer also has trouble communicating with the scientific or data processing computer. The languages available were designed for typical computing purposes, such as data handling and scientific processing, not extensive logic function handling. Hence, it takes longer to prepare source code programs and they, in turn, are generated into still less efficient machine code programs. These use up memory unnecessarily, take too long to run, and are cumbersome to control -- both from the programmer's viewpoint and logistically. They also take longer to test and debug because of the additional code.

To make the other choice, designing his own computer, the simulator designer must make a long term commitment. And he must understand the requirements that will be created for his simulator. Not only will he be devising a computing capability, but he must also devise a means to effectively communicate with his computer and then use it efficiently. This means creating his own software languages to match his computing system.

Prior to designing the EC II,a minicomputer driven simulator which preceded the EC 3, ECC had experience with its hardwired simulator, the SMART. We therefore knew what kind of computer was needed and how we had to describe the simulation; i.e., how we would have to communicate to the processor. But, there was a problem, we tried using FORTRAN and BASIC. There was nothing available that could do what we wanted. We built our own computer and created our own language. That was 1970. We were sticking our necks out, but soon we were using all our available I/O (96 Input and 96 Output lines). Even multiplexing. Memory increased from 4K to 12K, and in a few cases to 16K.

In 1974, we started research on a redesign. We started with a microprocessor and, in fact, built some so-called "hardwired" devices which in reality utilized a microprocessor and a ROM chip for memory. Figure 1 shows one of these "stand-alone" simulation models.

Then in 1976, we produced the EC 3 which uses four or more microprocessors. Though we don't build the processors themselves as we do for the EC II, we have structured them to satisfy our requirements and we are achieving results we would not be capable of with standard off-the-shelf systems.

STRUCTURE OF THE EC 3 COMPUTER SYSTEM

In designing the EC 3 computer system, ECC design engineers had to consider all the potential requirements that would be placed on the system. As already mentioned, the requirements of a computer-controlled simulator are more like those of a process-control or sensor-based system rather than like those of a commercial or scientific data processing system. A simulator-controlling computer has a relatively limited data movement and numerical computation requirement. But it must have the capability to handle a high volume of diverse I/O requirements and be able to economically and quickly handle large complex logic models in a Boolean format.

Commercial, off-the-shelf, processing systems and their associated high-level languages are designed for scientific or business applications, and perform poorly in handling process-control or sensor-based system requirements. The EC 3 hardware, firmware, and software, were tailored from their inception to provide the optimum simulator capability.

The design effort had to also take into account the requirement for a multitude of unknown peripherals or special devices that may be configured on the system to meet some future requirement. In the EC 3, separate processors have been incorporated to simplify system management and peripheral growth. ECC software has been designed to manage the configuration, which results in more efficient programming and memory usage. This in turn reduces the time and cost of programming.

Since the development of high density, integrated circuits, the major cost of any computer system is in soft-

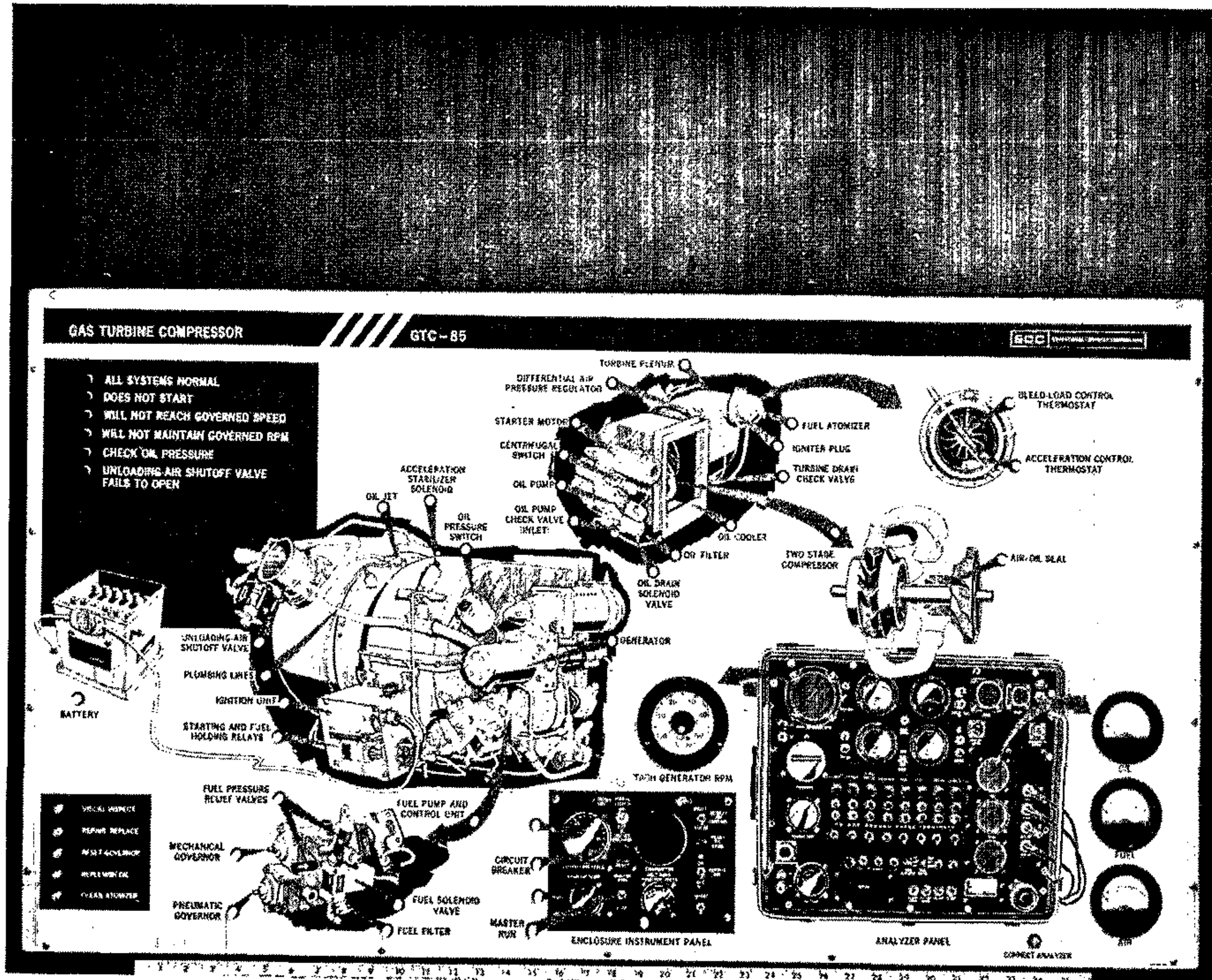


Figure 1. GAS TURBINE COMPRESSOR (STAND ALONE)

ware development. Hence, the hardware design in a total system development must enable programming to be as simplified and efficient as possible.

Unlike commercial data processing systems which utilize one central processing unit, the EC 3 employs multiple microprocessors in a "federated multiprocessing architecture." The executive processor is free for the processing of the mainline program while channel interrupts, control and scheduling are handled by the auxiliary processors. Peripherals requiring constant attention by the EC 3 are handled by dedicated or rapidly multiplexed processors which communicate through shared memory, making data transfer essentially transparent.

Where I/O operations are time consuming or complex, such as transfer between controller and memory or moving simulation panel instrumentation, these operations are controlled by separate auxiliary processors. With data transfer at full operational speed via shared memory, the executive processor is freed from I/O transfer wait time and can concentrate its resources on processing the main simulation program.

The EC 3 processing system can test or set a large number of Boolean I/O devices in one instruction cycle with no set-up or "handshaking" required. This is an order of magnitude faster than single-bit I/O implementation used by the commercial data processing systems. The EC 3 also places I/O ports in specific memory locations so the vast majority of I/O devices are treated as data types in the EC 3 programming language. This reduces the assymetry between internal and external operations and much of the "systems overhead" incurred by compilers such as FORTRAN or COBOL . This capability, which is unique to the EC 3, not only reduces memory required for a simulation model program, but also reduces processing time by elimination of I/O control operations which, by themselves, do not produce meaningful results.

Programming efficiency and operational time is enhanced by PROM resident within the EC 3 operating system. This eliminates utilizing memory for the standard I/O drivers (CRT, Teletype, Diskette drive, etc.). PROM is also used for basic system diagnostics and common sub-routines such as arithmetics, integrations, text handling and error reporting. PROM storage protects critical and commonly used program routines from loss due to power failure and especially inadvertent modification by the user. With only the main line application program resident in the RAM, savings are realized with reduced programmer coding and optimization of the RAM.

PROGRAMMING

ECC's prior experience with simulation controllers, including the use of commercial hardware and software, had already proven that most simulation programming time and cost was associated with operations that could easily be described through concise Boolean statements. (In fact, a major portion of the EC II and EC 3 programs is a series of Boolean equations.) These operations are the ones most poorly implemented using commercial languages. Even with languages such as FORTRAN, in which it is possible to write Boolean equations ("TYPE LOGICAL"), the implementation penalty is severe. The EC 3 is designed specifically to handle Boolean equations and single bit I/O rapidly and efficiently and the EC 3 compiler maximizes the use of this capability. (Logic design engineers use Boolean algebra statements in the description of logic problems and logic systems design.)

If the system being simulated is able to be described with Boolean statements it can be directly implemented on the EC 3.

Random Access Projector

The Random Access Projector (RAP) used in the EC 3 might be called a "SMART" projector in that it has its own microprocessor. The first of these projectors built by ECC, (RAP I), were controlled directly by the processor. Later models, RAP II and RAP III, do by themselves, what the computer system asks. Here again is an application of the microprocessor.

In order to provide a high degree of availability and rapid random slide selection, the slides on the disk are arranged in a spiral on a transparent disk which rotates over a fixed light source while it traverses laterally. Originally the slides were placed along fixed radii of the disk with considerable space being wasted in the outer coils of the spiral. Now that space is able to be condensed with the microprocessor keeping track of all the slide positions. It simply "knows" where it is all of the time.

Prior to using the microprocessor in the projector, the computer system had to track the position of the visual disk and send positioning commands to the projector through an interface. Now the computer system simply asks for a slide position and the projector takes care of itself. This also allows the RAP III projector to be used as a stand-alone device or easily interfaced with some other system.

EC 3 SYSTEM

ECC developed the EC II primarily as a device to provide procedural and diagnostic (troubleshooting) training for the maintenance technician trainee. The EC 3 has been designed for these same applications and beyond. It is able to interface to actual equipment, operate CPT's (Cockpit Procedures Trainers), perform detailed intermediate level maintenance simulation and provide CAI and/or CMI support.

The EC 3 system consists of a mainframe or console which houses the computing system and control console and then, as required, additional peripherals: CRT, visual display, printer, or other display device.

Figure 2 shows the typical, individual system console which contains the power supply, diskette, computer system, projector, projection screen, and system control console. Figure 3 shows the computer system console connected to a large classroom display panel. Figure 4 is a detailed view of the Control Center.

The computing system is designed as a general purpose simulator control and becomes a system specific trainer with the addition of a simulation model. A standard simulation model consists of a unique display panel on which is presented a modified two-dimensional pictorial or schematic of the real equipment, a magnetic diskette on which is stored the simulation computer program, a visual display disk or 35mm slide tray, an instructor's guide and, where necessary, auxiliary equipment such as meter probes.

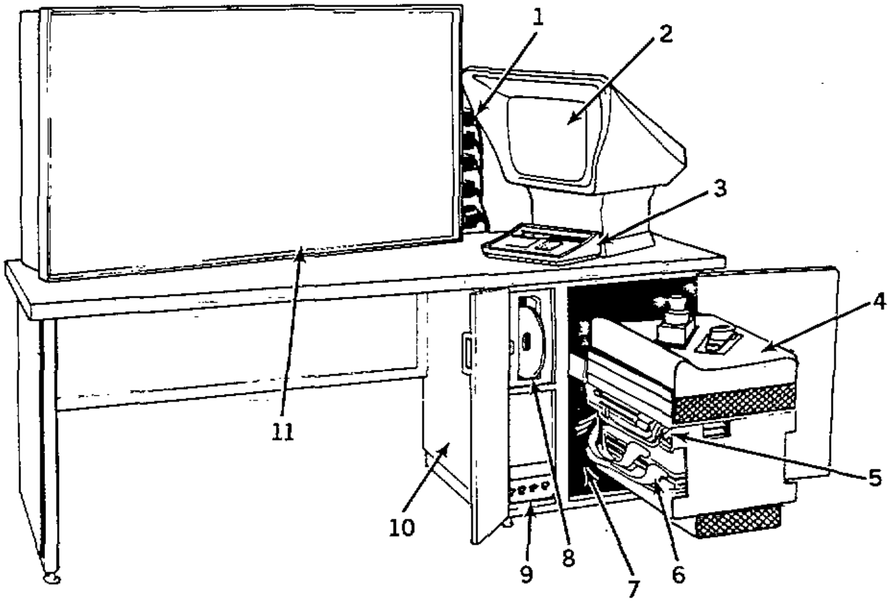
To operate the system, the student or instructor quick-connects the display panel to the system, places the visual disk or slide tray on the projector, and inserts the magnetic diskette in the diskette reader. This takes about two minutes. Inserting the diskette automatically loads the simulation program from the diskette into the computer memory. The system then functions as the actual equipment in response to student actions. Visual inspections can be made of various components, volt/ohm meter tests can be made on electrical circuits and special test equipment can be simulated. A malfunction can be entered in the system simply by keying in a predefined code on the control console.

The student can troubleshoot the system to determine the problem and indicate via the control console or entry buttons on the display panel what remedial action is required to repair the system. While the student is operating the system, records are maintained by the system of elapsed time and the number of tests or replacement actions taken by the student.

If program changes are required, they can be made either by ECC or the user.

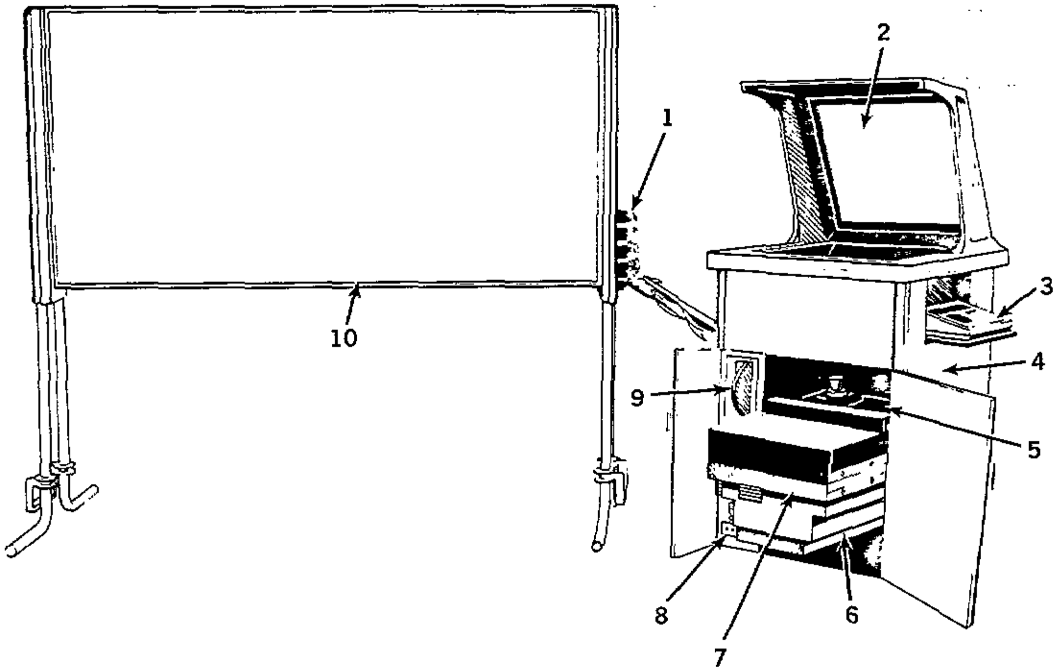
The typical EC 3 application is to provide a panel with a pictorial of the system or part of the system for which training is being given. Within the pictorial rendering are all the controls and test equipment, meters, displays and so forth that a technician would use if he were working on the actual equipment in the real world. The student can then operate the system just as he would in actuality and the simulated system will respond.

He is provided simulated test equipment and can use, for example, probes from his simulated VOM or oscilloscope in jacks or at test points that are actually represented on the simulation. For those actions which can't actually be performed, the student is provided push-buttons which replicate the results of that action. For instance, if he wanted to visually inspect a component on the projector screen he presses a visual inspect button and then a button at the component he is inspecting. If the system was simulating a malfunction mode and that component would in actuality appear abnormal, he would see a picture of the abnormal component. If he is using the



1. DISPLAY PANEL CONNECTORS
2. VISUAL PROJECTION SCREEN
3. EC3 CONTROL CENTER
4. RANDOM ACCESS PROJECTOR
5. EC3 COMPUTER SUBSYSTEM
6. SYSTEM ELECTRONICS
7. SYSTEM POWER SUPPLY
8. PROGRAM DISKETTE DRIVE
9. CIRCUIT BREAKERS
10. EC3 CONSOLE CABINET
11. SIMULATION DISPLAY PANEL

Figure 2. INDIVIDUAL EC 3 SYSTEM



1. DISPLAY PANEL CONNECTORS
2. VISUAL PROJECTION SCREEN
3. EC3 CONTROL CENTER
4. *EC3-LP CONSOLE CABINET*
5. RANDOM ACCESS PROJECTOR
6. SYSTEM POWER SUPPLIES
7. EC3 COMPUTER SUBSYSTEM
8. CIRCUIT BREAKERS
9. *PROGRAM DISKETTE DRIVE*
10. SIMULATION DISPLAY PANEL

Figure 3. LARGE PANEL EC 3 SYSTEM

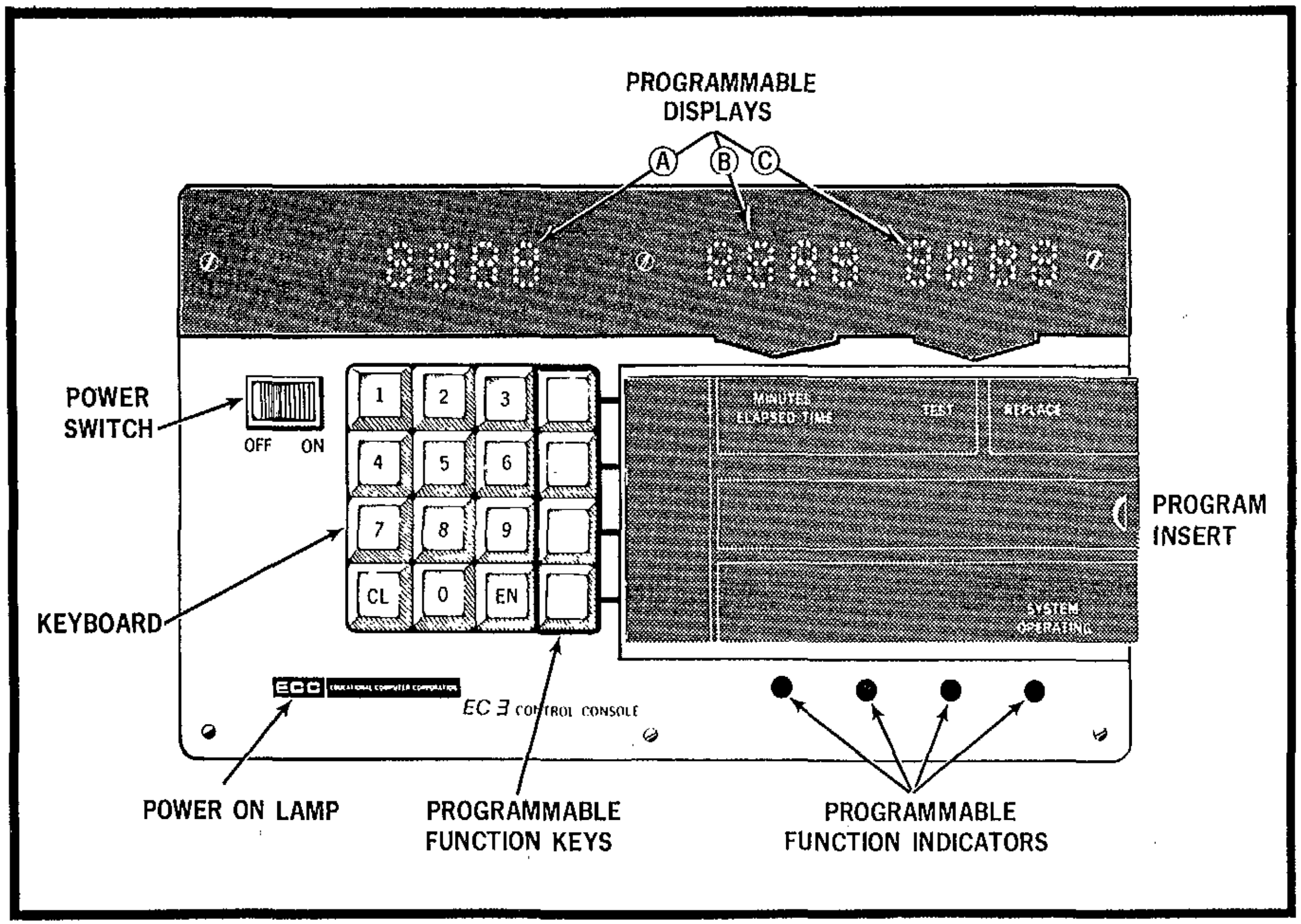


Figure 4. EC 3 CONTROL CENTER

simulation in a normal configuration and does something which would create an abnormality, the simulation would respond accordingly.

The panel simulations present prime equipment systems in a reduced physical fidelity format, but with full functional fidelity. If the training requirements dictate that the simulation have full physical fidelity, then instead of a panel, a three-dimensional representation is used. Figure 5 is a 3-D simulation of a tank turret system trainer.

How the simulation operates is determined by the training analyst. For instance, whether or not the student is given cues or admonitions is dependent upon the training design requirements. And if required, these can be changed under program control as the student advances and becomes more proficient. Figure 6 shows a specific simulation application. It is a simulation incorporating a TSGMS (a test set) for troubleshooting the TOW Missile System on the COBRA helicopter.

EC 3 FEATURES

Standard Features

SYSTEM

- o Multiple Microprocessors (four)
- o 16K Dynamic Random Access Memory (RAM)
- o 9K Programmable Read Only Memory (PROM)
- o 512 words static RAM
- o Multiple RS232 communication input/ouput ports
- o 256 input lines/128 output drivers
- o 64 discrete lines for simulated probe points
- o 8 Digital/Analog - 8 Analog/Digital Converters

RANDOM ACCESS VISUAL SYSTEM

The RAP III projector, a high reliability display device, stores 150 images on a 9-inch plexiglass disk. Access to visuals for display is under computer control. Average access time is less than half a second, with maximum access time from position 01 to position 150 of approximately 2 seconds. The RAP III, with its own microprocessor, can be used as a stand-alone device or interfaced with other systems.

DISKETTE

An IBM compatible diskette with 256K words of storage is used as the program storage media. The use of a direct access storage device permits easy configuration of an EC 3 system for CAI/CMI applications.

SNAP ON SIMULATION DISPLAYS

Quick disconnect-connect receptacles are used for the interchangeable simulation models. The individual trainer panels are standardized in size, but classroom system panels are varied dimensions. In training situations where three-dimensional test equipment or additional panels are required, multiple panels can be cable-connected to one main panel attached to the system.

CONTROL CONSOLE

- o Digital Keyboard for entry of student information and preprogrammed conditions
- o Special Alphanumeric Functions for multiple choice testing results
- o 12 LED Condition Display for instructional feedback, in monitoring elapsed time, student performance, or special tracking
- o System Operating Condition Display, including audible tone warning signal

Optional Features

ADDITIONAL:

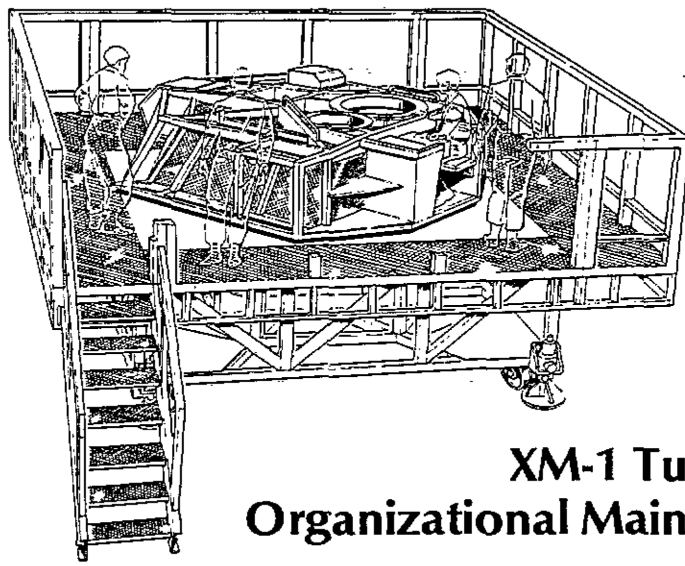
- o Diskette Units
- o D to A/A to D Converters (increments of 8)
- o Input/Output increments of 256 inputs; 128 outputs, 64 discrete lines

RAM MEMORY

- o Expansion from 16 to 56K in 4K increments

CRT DISPLAY

- o 1024 characters on a 7x9 inch screen for Computer Aided Instruction or Systems Programming



XM-1 Turret
Organizational Maintenance Trainer

Figure 5.

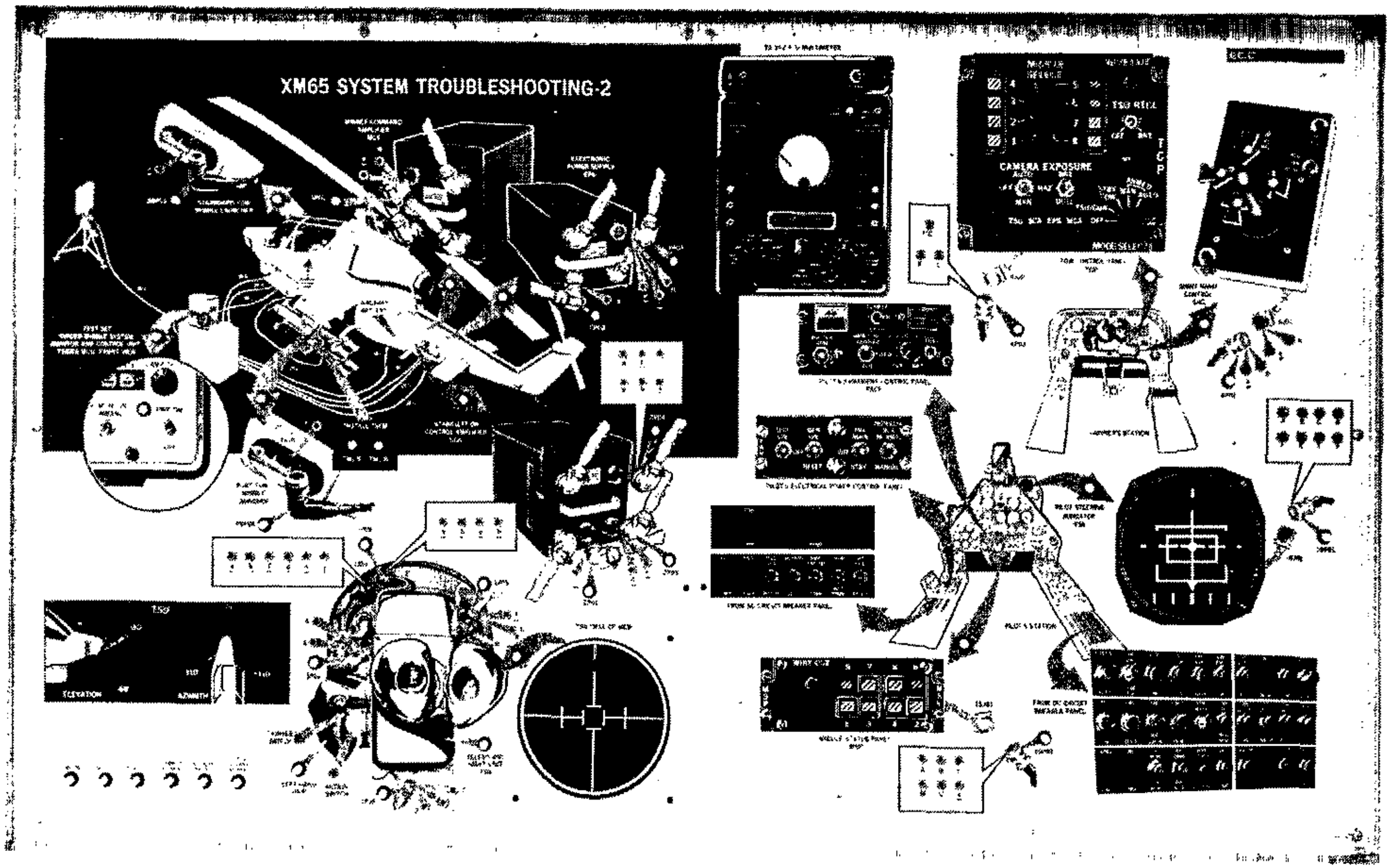


Figure 6. XM65 SYSTEM TROUBLESHOOTING 2

TELETYPE

o ASR 33-35

STORAGE SCOPE DISPLAY

o For use in Graphics Simulation

AUXILIARY PRINTER

o High speed character or line printer for student performance records and program compilation listings

VISUAL PROJECTION

o Single or multiple 35mm Random Access Projectors

RANDOM AUDIO SIMULATION

o As required

INCREASED CAPABILITY AND TRAINING REQUIREMENTS

From the preceding it is obvious the EC 3 has a capability many times that of a non-dedicated system and, at the same time, the basic system hardware is less costly. Our EC 3 is less expensive than the EC II.

The microprocessor has greatly extended the range and detail available in simulation. The EC 3 is not limited to the two-dimensional panel. It can drive just about anything: cockpit procedures trainers, three-dimensional trainers, interface with actual equipment, modular add-on test devices. Even detailed intermediate level maintenance activities can be simulated. The EC 3 can readily simulate electronic component testing using meter or oscilloscope probes.

But all this comes with its price. As everyone should know, the greatest cost in putting a computing system on-line is generated by the software development. The added capability created by the microprocessor is something of a mixed blessing. The hardware procurement cost is going down, but with increased simulation detail and fidelity readily available, the software development, if not controlled, not only makes up the difference, but can become prohibitive. However, this cost can be controlled by accurately prescribing the fidelity requirements.

This is not simply a matter of whether or not to use a 3-D or 2-D simulator, but how much detail in physical and functional fidelity will be necessary to meet the learning objectives. Microprocessor technology isn't making the training technologist's job easier. It is placing greater demands on him. No longer is he conveniently restricted and bound to the confines of the available equipment. He must now sharpen his techniques to be able to accurately specify the training equipment support requirement. It becomes his burden to accurately specify the training and media requirements. Cost effective training simulator design becomes his liability.

It is very easy to retrench and stay within the confines and capability offered by existing off-the-shelf commercial hardware, but that is passing up the opportunity to increase the effectiveness of applicable training.

ABOUT THE AUTHOR

MR. NICHOLAS A. SIECKO is Vice President of the Educational Research and Development of the Educational Computer Corporation, Orlando, Florida, and directs all of the educational projects for the company. He directed the federally sponsored experimental and demonstration manpower training center (Northeastern Pennsylvania Technical Center). Mr. Siecko is a co-holder of the patent for the design of the company's EC II general-purpose simulator and was responsible for establishing the training concepts which are now being implemented. He established the approach used in the design and development of simulation models used with the EC II, and the instructional programs for the SMART simulator. While with the Systems Operations Support, Inc., he developed, prepared, instructed, and evaluated vocational high school technical courses in one of the earlier attempts to apply a systems approach in a public school district vocational technical high school. He was employed by Burroughs Corporation as a programmer and later as a systems analyst on the Atlas Missile Project. Mr. Siecko is a member of the American Society of Training and Development, National Security Industrial Association, and a charter member of the Society for Applied Learning Technology. He has a B.A. degree in mathematics and has done graduate study in psychology at the Villanova University.