AN/SLQ-32 STIMULATION USING A COMPUTER STIMULATION TRAINER ARCHITECTURE

L.M. Chaikin
AAI Corporation
P.O. Box 6767
Baltimore, MD 21204

ABSTRACT

The capacity and complexity of today's operational equipment require intensive training to realize full system's capabilities. To meet that training need by stimulation of the device at the RF or video level could require hardware that exceeds the operational equipment in size and complexity. It is therefore necessary to explore alternative simulation architectures. One such possibility is to interface the simulation at the operational device computer. Since much of the functionality of current operational equipment resides in the algorithms programmed into the device computer, high fidelity of simulation can be achieved by interfacing at this point, since these algorithms are fully stimulated. In addition the simulation will be stable if there are upgrades to the operational program. The simulators may even be used to test the response of the changed algorithms to complex environments. This is less expensive than using full RF or video stimulation, and the resulting simulator is packaged more tightly. Therefore, this kind of simulation closely approaches the organic training concept.

To illustrate these points, the SLQ-32 simulation developed as part of the Perry Class Pierside Combat Systems Team Trainer, Device 20B5, will be used. Not only the strengths, but some of the pitfalls of interfacing a simulator to the device computer will be discussed. Methods of solution and tradeoff requirements are covered which result from such causes as real-time constraints imposed by the operational program, simultaneous control of the simulation by the host and device computers, and simulation validation. The results demonstrate that with proper design consideration a flexible, high fidelity simulation can be achieved by the use of this interface point.

INTRODUCTION

As threats in the EW environment become more dense and complex, improved readiness training through simulation has become more vital to the country's defense posture. Proper operation of today's computer driven ESM/ECM/ECCM devices requires extensive training. Even though an operator may have several years of experience, dense tactical combat situations, and situations where his device has been saturated or shown serious misidentification or other anomalies may have been missed. To support advanced level training, a simulator must not only provide complex scenarios with the full range of threats, but also must model the responses of the device, in detail, including realistic response times and device anomalies. If this is not done, negative training may result. The difficulties of providing detailed device modeling are, however, greatly complicated by the rapidity at which devices of this class are upgraded and modified.

Since the anomalous behaviors to be modeled are specifically device dependent, changes to the device almost always effect them. In fact, these behaviors may have been the cause of the enhancement in the first place. The situation is worsened, from the simulation point of view, by the extensive use of embedded computers. Since many of the operational characteristics of the device are now due to the software, which is very easily changed, modifications have become much more frequent. Thus, to avoid the effects of negative training, the simulator must not only be able to accurately present complex scenarios to the device, but it must also be able to be quickly and inexpensively upgraded to keep it current, especially with respect to software changes.

Unfortunately, many of the simulator architectures in use today do not possess these attributes. For example, full stimulation at RF (widely used for device operability testing), is very accurate and does not need to be changed for device modifications, but it frequently cannot reasonably provide complex controlled scenarios. At the other extreme, if the device display is directly driven, it is simple to produce complex displays. However, it is difficult to provide accuracy, and whenever a change is made, the simulator must also be modified. For these reasons, alternative simulator architectures are now being explored. One of these - the one which will be examined in this paper - is interfacing the simulator directly to the device computer, which for the purposes of this paper will be called "computer stimulation."

In computer stimulation, the device receivers, and any special processing units associated with them, are disconnected from the computer, and the simulator is plugged in to replace them. The display/control console and the computer's operational software, clock, and memory are left intact. The interface to the computer is over its system bus. The computer controls the display, and sends commands to the simulator and receives responses from it, as if the sensors and processing units were still attached. It is the simulator's job to duplicate the responses of the disconnected units.

The remainder of this paper describes in some detail the attributes of computer stimulation architecture: the benefits achievable, comparison to other architectures and organic training, problem areas and implementation pitfalls. Finally, an actual implementation of a computer stimulation simulator is presented, to illustrate the concepts involved.

COMPUTER STIMULATION - SPECIFIC BENEFITS

Reduced Hardware/Small Packaging Size

The primary components of a computer stimulation simulator are a general purpose computer (generally a microcomputer) and a bus interface.

Since most devices are built with off-the-shelf vendor supplied computers, the bus interface is typically well known and well documented. Thus, there is not much risk in the bus interface development. The simulator computer can interleave simulated pulses/responses, and can look to the device as if it were a number of different processing units. This plus the elimination of all analog signals (such as RF) typically leads to a smaller package size than for architectures using full stimulation.

Improved Security

There is no radiation sent from the simulator to the device, only digital information. Therefore, the chance of hostile intercept and derivation of radar parameters is lowered.

High Fidelity

Since the operational computer software is fully operational (and unmodified), the device *doesn't know that it is in a simulation mode.* Thus the real-time reactions and anomalous behaviors of most of the device are present.

Ease of Upgrade/Modification

Again, since the device software is fully operational, changes to it will not require changes to the simulator. Much of the code provides capabilities with which the simulator need not be concerned: operator communications, parameter derivation algorithms such as deinterleaving of signals, signal identification algorithms, and signal library maintenance among others. These functions are "above" the level of the simulator (receiver emulation), and so are used as they are.

The primary functions of concern are the device receiver control functions. The simulator software is designed to mimic the response of these units to all of the various commands. Therefore, changes in command sequences should not cause much change, if any, in the simulator. Of course, if changes are made to the device hardware, changes will have to be made in the simulator because these devices are being emulated. However, since the simulator is primarily a general purpose computer, these changes will be software modifications to the simulator's programming, a relatively rapid and inexpensive process.

Ability to Produce Complex Scenarios

The heart of the simulator is the general purpose computer. Therefore the limits on the number of signals produced and interleaved (and their complexity) depend upon the speed and memory of the computer used.

Simplified Device Interface

Since the simulator is being used to replace one or more whole devices connected to the computer chassis, installation requires the removal of certain cables and their replacement by cables from the simulator. Installation/removal is rapid and does not require device modifications or disassembly with their attendant risks. Maintenance on the disconnected receivers can be performed during training since they are not needed for the simulation. An alternative to utilizing operational equipment is to set up a classroom version of the simulator using only a device computer and display/control console, without the expensive, custom-made receivers and processing units.

COMPARISON TO OTHER SIMULATOR ARCHITECTURES

To place the capabilities and limitations of computer stimulation into perspective, a comparison with other possible interface points will be drawn. This comparison will include the following areas:

Fidelity - modeling of true real-time response rates and anomalies

Cost/complexity - simulator development and production cost

Interface difficulty - ease of installation and removal of simulator

Upgrade capability - effort required to keep the simulator current as the device and its software are modified.

Each of these areas will be considered for a simulator used for ESM tactical training purposes; that is, ability to produce a large number (over a hundred) of different, moving, target signals. A model device is pictured in Figure 1, and the interface points to be considered are listed below:

RF stimulation - injection of a radar signal into the device receivers

Video stimulation - injection of video signal into the device after the signal would have been sensed

Special processing unit stimulation - injection of digitized pulses into a tracking unit or other preprocessor of digital pulses before the computer

Computer stimulation - interface at the computer bus

Display/control console stimulation - direct handling of operator commands and display, bypassing all pulse production and device analysis

A comparison of each technique is given in Table 1. For advanced tactical training usage, the computer stimulation interface is very effective. It has capabilities for high fidelity and upgrading as do the full stimulation methods, while at the same time allowing the complex scenarios and ease of interface of full emulation. The costs and technical difficulties fall, as might be expected, between the two. Comparisons with interfacing at a special processing unit will vary, depending on the functions of that unit. However, in general, these units are connected with specific portions of the device, to perform services (such as digitization of video pulses), and so a simulator will need other interface points to complete the full functional

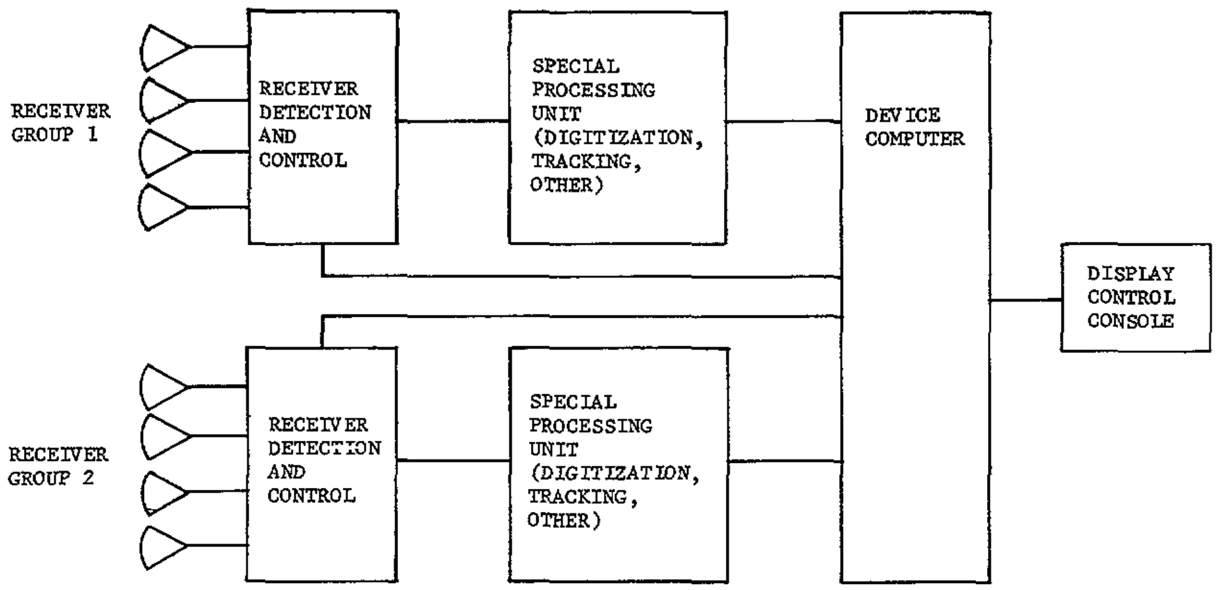
RECEIVER GROUP 1
RECEIVER DETECTION AND CONTROL
SPECIAL PROCESSING UNIT (DIGITIZATION, TRACKING, OTHER)
DEVICE COMPUTER
DISPLAY CONTROL CONSOLE
RECEIVER GROUP 2
RECEIVER DETECTION AND CONTROL
SPECIAL PROCESSING UNIT (DIGITIZATION, TRACKING, OTHER)

Figure 1. Model ESM Device

Table 1. Interface Point Comparisons

Interface Point	Fidelity	Cost/Complexity	Interface Difficulty	Upgrade Capability
RF Stimulation	Best: all subsystems fully stimulated	Prohibitive: realistic multi-threat dynamic scenarios require massive amounts of hardware	Very Difficult: either place physically around training vehicle or cluster at each antenna group	Automatic
Video Stimulation	Excellent	Very Costly: much hardware usually required for large scale scenarios	Difficult: may need partial device disassembly and many interface points	Automatic
SPU Stimulation	High: all software and some hardware stimulated	Costly: custom hardware interface components are required	Variable: usually will require multiple interface points for full simulation	Good: few if any software changes require simulator updates
Computer Stimulation	High: all software stimulated	Moderate: fewer custom components due to single interface point	Simple: can usually use cable connectors and small packaging	Good: few if any software changes require simulator updates
Display Stimulation	Poor: Each anomaly must be specifically modeled	Least expensive	Simple: use cables to display or mock up/generic display	Poor: all changes must be reflected in simulator revisions

simulation/stimulation. In many cases, the most convenient interface point for the remainder of the device subunits (for example computer controlled filters) will be at the computer anyway. For this reason, the interface can be simplified by using computer stimulation alone. Of course, this means that the anomalies and details of special processing unit functioning must be modeled in the simulator.

It should be emphasized that the remarks in this section are relevant to a simulator used for advanced level training. Obviously, a computer stimulation simulator will be of no use in testing the device for operability. The RF and special processing units would be missed, and in testing there is no requirement for long or complex scenarios. In addition the increased cost/complexity of computer stimulation is not warranted for lower level "generic" trainers where the anomaly modeling and fast upgrade to the latest version are not required.

COMPARISON TO ORGANIC TRAINING

Computer stimulation is similar in many ways to the concept of an organic trainer in which a trainer is built into the device itself as part of its design. Like other simulators used for training, as opposed to device testing; complex scenarios, high fidelity, and ease of upgrade will be essential. Since the organic trainer is part of the device it will probably stimulate the operational software - but will not use RF, IF or video, due to the large number of signals required. Thus, it will have many of the characteristics of computer stimulation. One major difference is that the organic trainer will partially reside in the operational software of the device, and so the operational software can be thought of as having been "modified" to accommodate it.

Because of these similarities, computer stimulation is considered an externally connectable organic trainer. It appears to the computer as part of the device and simulates/stimulates many of the same things as an organic trainer. However, being removeable provides some advantages. There are no weight, space, or performance penalties on the operational use of the device as there are with organic trainers. Also, unit costs of the device will be less, as it will not be necessary to produce a trainer for each device.

PROBLEM AREAS IN COMPUTER STIMULATION IMPLEMENTATION

The concept of computer stimulation is simple. A generalized microcomputer emulating disconnected receivers is hooked to the device computer through a standard computer bus interface. However, there are several problem areas which complicate the implementation and are particularly relevant to this kind of simulation. They are not insurmountable, but the method by which they are handled will effectively determine the quality of the resulting simulator and will determine whether or not the benefits achievable from computer stimulation will be realized. The balance of this paper will consider these difficulties and cite a particular example. SLQ-32 computer stimulation - describing how these problems were handled for that simulator. The problems have been divided into four areas (real-time constraints, computing power requirements, dual trainer/device control, and testing/confirmation), each of which will be discussed below.

Real-Time Constraints

Training on a computer stimulation device is a real-time process. The device computer expects the same real-time responses and timing from the simulation as from the disconnected sub-units. The responses from the simulation computer do not, in general, exhibit the same timing. Lacking parallel hardware, the simulation computer processes signals and messages sequentially, and, in many cases, more slowly than the specially designed hardware sub-units. Even worse, some of the real-time constraints are determined by the operational software. An example of this is device real-time background self-testing. If the correct response is not received within the proper time - the device may begin to report malfunctions in the disconnected units. Standard operational responses may also be timed - with the same effect.

For these reasons, a detailed understanding of real-time constraints and timing requirements is essential, not only for the hardware, but for message handling software as well. Some of the required responses may, in fact, be too rapid to be processed in the simulation software at all. For these, custom hardware may be required. Others may be handled by dedicating an extra microprocessor to handle specific messages or by responding on an interrupt basis. In some instances, the opposite of the above may happen - the simulator may actually be too fast for the actual device. This situation can arise when the simulator "knows" and can present an answer that the hardware must scan and process. One example of this is when there are no signals in a specific frequency band. The obvious solution of installing a wait in the software usually works well, therefore, this condition is not too problematic. However, this situation must also be considered when researching the real-time constraints.

Computing Power Requirements

Due to the desire for small packaging and low cost, microprocessors or other small computers are selected for computer stimulation. Often, when many interleaved signals must be produced, the computing power of the simulation processor can be exceeded. Maintaining a continuous flow of interleaved digital pulses at full radar rates is certainly a challenge for the typical unaided microprocessor. This problem is aggravated when multiple channels of parallel processing detectors must be modeled. Fortunately, there are some mitigating circumstances. The most important is that the device itself is only looking at some of the signals at a given time. The simulator, by having access to all computer control messages to the device sub-units, can determine precisely which signals are being analyzed. Thus, only those signals, not the whole RF environment need to be produced. Nevertheless, the data rates required may be quite large. Microprocessors lend themselves to multi-processor systems, and dedicating an extra processor board to the task of improving throughput is frequently a good solution.

It should be noted that when the data collection time is short, the time difference in human terms may not be noticeable. Also the operational software may not look at all of the sensed signal parameters or pulses. Therefore the required modeling might be reduced. However, this begins to place the design on shaky ground. If the simulator begins to take the operational software algorithms into account, software independence and consequently automatic upgrades of the simulator may be jeopardized.

Although there is always the temptation to direct the model to the software when dealing with the need to improve throughput or response times, such compromises should always be analyzed with respect to their potential effects on future upgrades. Ultimately, specialized hardware for signal production can be used if necessary. Of course, multiple processors or specialized hardware will increase the cost and complexity of the simulator, which is the reason that computer stimulation costs more than merely driving the display in a full emulation simulator. Even so, the hardware requirement for computer stimulation is likely to be substantially less than for the other interface points which all require even greater throughput.

Dual Trainer/Device Control

For tactical team training, many device simulators can be integrated into a single training device with centralized control of the scenario. Since the computer stimulation simulator is compact and self-contained, (due to the necessity of rapid response) it lends itself well to this concept. However, from the device point of view, the simulator is a slave, actually a part of the device. The central training device controller also treats the simulator as a slave, of itself. A conflict of control can arise. Both the trainer and the device assume they can control the simulator. Therefore, a means of serving both must be found without undue interference. In general, since the unmodified operational software cannot make allowances for external interference, it is the trainer messages, commands, and responses which will have to be queued or serviced at lower priority.

Testing/Confirmation

Since computer stimulation interfaces into the middle of the device's information flow rather than at either end (RF or display), testing of the simulator tends to be more difficult. Being a high fidelity simulation technique, anomalous device behaviors are observed, especially near saturation or in complex scenarios. To validate the simulation, one must verify that these anomalies represent actual device behavior and do not arise from causes such as errors in the simulation hardware or software, device design misunderstanding, test case errors, or test cases that are too idealized. Complicating this process is the fact that, short of combat, many of these complex scenarios are unlikely to have been seen in the real world. For generic (full emulation) simulator modeling, anomalies may not be important. For an RF stimulator, verifying that the signals sent have the right parameters ensures that the device response must be genuine. Computer stimulation is not as simple to verify. Researching device schematics and software listings, in addition to obtaining opinions from expert operators, may be required to separate the genuine from the eroneous responses.

AN EXAMPLE: THE 20B5 AN/SLQ-32 SIMULATION

As an illustration of the concepts presented in this paper, an actual simulator using computer stimulation is described. The specific device to be simulated was the AN/SLQ-32(V)2.

The Device

The AN/SLQ-32(V)2 ESM suite contains two wide-open receiver systems in frequency bands associated with most threats and a superheterodyning receiver in a low threat band. All are controlled by the same central device computer (a ROLM 1606 UYK-19). In the high threat bands (bands 2 and 3), the SLQ-32 receives pulses continuously using one set of antennas, an IFM (Instantaneous Frequency Measurement) sub-unit to determine the frequency and DFRs (Direction Finding Receivers) with encoders to determine bearing and amplitude. These are combined into digital pulses along with a time-of-arrival in the DFC (Direction Frequency Correlator) and then sent to a tracking unit.

The Digital Tracking Unit (DTU) sorts the pulses by bearing and frequency into "emitters," and reports new, moving, and deleted emitters to the central computer. The computer will periodically request pulses from specific emitters (up to 14 at once), and the DTU will send them. They are analyzed to determine pulse repetition interval (PRI) scan period and type. An identification/threat library comparison is then performed to determine the symbols to present to the operator on the Display and Control Console (DCC). Computer controlled limit switches and frequency filters are available to eliminate unwanted signals via the IFM and Serial I/O Controller (SIOC).

In the low threat band (band 1), a computer controlled receiver steps through operator requested frequency ranges, "dwelling" at each frequency for a short time, looking for signals. When one is found, the pulses are digitized and sent to the computer by the Digital Control Unit (DCU) for analyses, identification, and presentation as is done with the high threat band data. The computer performs all tracking and receiver control functions for band 1.

Other functions performed by the computer include DCC control, operator command interfacing, and the handling of real-time Built-In-Test (BIT). Built-In-Test includes not only monitoring the fault detection circuitry and isolating any reported faults, but also testing the active background of certain sub-units constantly performed in real time. Figure 2 is a simplified SLQ-32 block diagram.

The Simulation

The SLQ-32 simulator was developed by AAI as part of the Perry Class Pierside Combat System Team Trainer Device 20B5. This trainer has an instructor station/scenario control center housed in a van on the pier. Carry-on units are connected to the weapons systems and sensors of the FFG-7 class ship on which tactical team training is to be performed. The carry-on units are linked to the van by a fiber optics, high-speed data link, to enable coordinated stimulation of the sensors and weapon system control panels. The SLQ-32 ESM suite is, an important part of the ship's equipment being stimulated. Virtually the whole simulation for the SLQ-32 is done in the carry-on unit. After downloading the simulation software and scenario signal library to the unit, the van will update the bearing, aspect, received

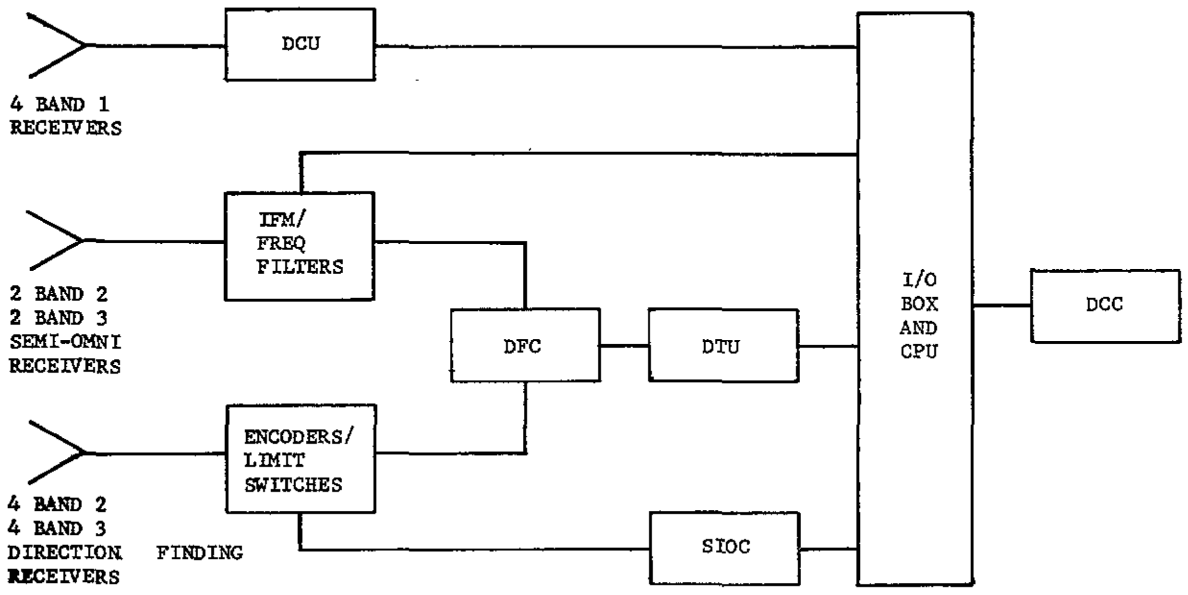

Figure 2. SLQ-32 Simplified Block Diagram

power, and signal identifiers of the currently active signals once per second.

All other processing is done within the carry-on unit including: filter and blanker modeling, pulse production, frequency agility and continuous wave effect simulation, message, data, and interrupt sending/ receiving, audio generation, self-test modeling, and tracking unit modeling. Currently, the maximum number of signals per scenario is 1024. The maximum number simultaneously active is 150. Both of these figures are software parameters and may be substantially increased by changing them and recompiling.

Figure 3 shows the interface of the carry-on unit to the SLQ-32 common I/O bus. The installation is very simple: the special processing units are disconnected from the bus expanders by unhooking their cables, and cables from the carry-on unit are plugged into one of the expanders. A cap is placed on the other expander to provide continuity. The figure does not show that the audio cable to the Display and Control Console (DCC) is disconnected and replaced by a carry-on unit cable. No disassembly of any of the SLQ-32 components is required, and disconnected units need not be present to perform training.

Figure 4 shows the contents of the carry-on unit. There are four microprocessors (CPUs 1, 2, 3, and 4) all sharing access to the entire system memory and hardware I/O interfaces. The functions of each CPU are listed below:

CPU 1 - Main simulation processor: handles all trainer communication processing; audio frequency agility, and continuous wave effect simulation; limit switch, blanker, frequency filter, and tracking unit modeling.

CPU 2 - Pulse generation processor: generates all band 2 and 3 pulses for the 11 interleaved and 3 non-interleaved data collection channels.

CPU 3 - Device communication processor: queues and sends interrupts/messages to the SLQ-32 and handles (on an interrupt basis) incoming messages/ commands. This ensures proper response times which for most commands is 500 microseconds or less before device failure alerts are generated. Work is given to CPU 1 and CPU 2 via control flags.

CPU 4 - Band 1 processor: all band 1 functions including pulse generation and SLQ-32 band 1 receiver/Digital Control Unit message handling. The SLQ-32 band 1 superheterodyne system is being redesigned. To minimize changes to the simulator when a redesign is implemented, all band 1 functions have been isolated in a separate processor.

The effects of the implementation problems discussed in the previous section can be clearly seen in the resulting simulator structure. Aside from the main simulation processor, extra processors have been added to resolve the real-time constraint problem (CPU 3) and the computer throughput problem for pulse generation (CPU 2). In fact, more was required to deal with the throughput problem. For 11 of the 14 possible pulse collection channels only a few pulses are taken for each signal (PRI analysis). Therefore if the processor is producing pulses more slowly than real time (even by a factor of two or three), the effect will not be noticeable in human terms.

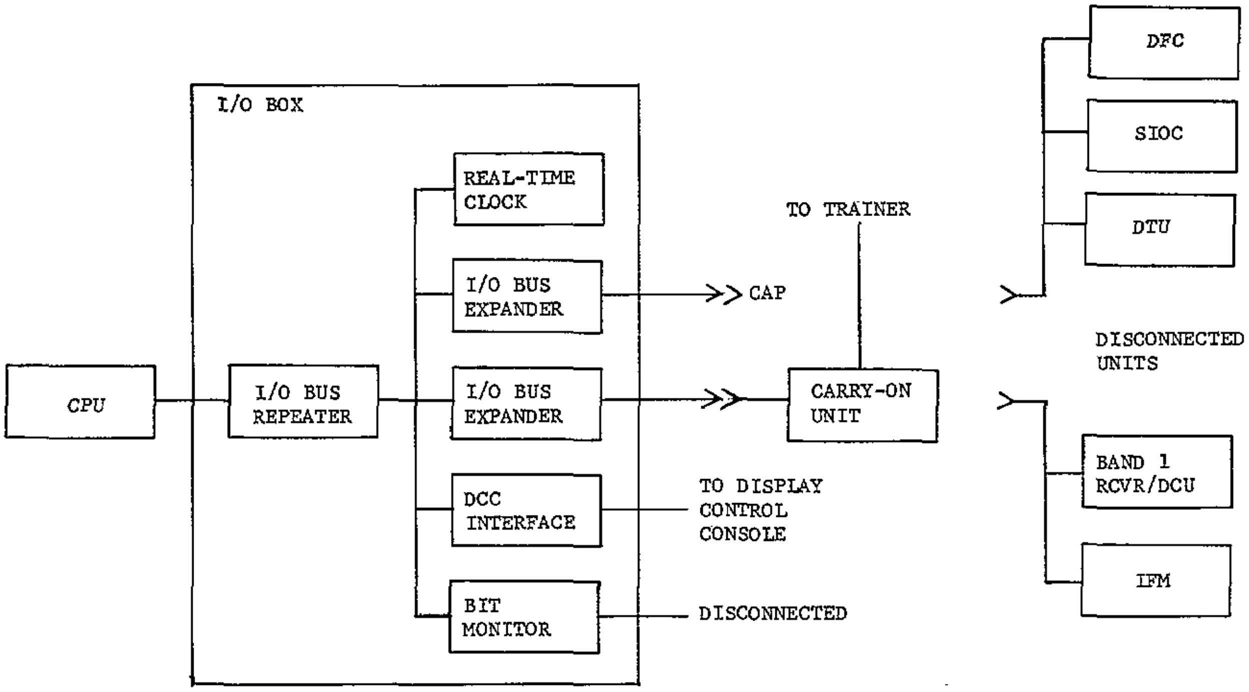

Figure 3. SLQ-32 Simulator Interface Point

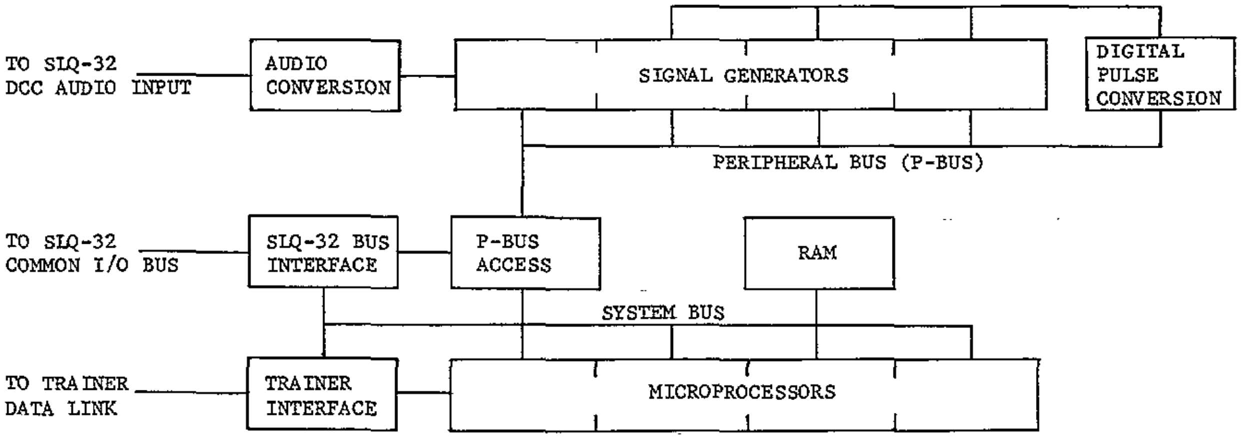

Figure 4. 20B5 SLQ-32 Carry-On Unit

However, for audio and the three remaining channels (scan analysis), many thousands of pulses over long periods of time may be required. For this reason, signal generators were added to produce pulses with precisely the same timing as the true signal. The carry-on unit has expansion slots available for more signal generators to be added if further channels are allocated to scan analysis. To preserve ease of simulator upgrade in the event of expected, but unknown, changes to the device, band 1 processing hardware (as well as software), simulation of the low threat band was isolated as much as possible in a separate microprocessor (CPU 4). As only a narrow frequency range of generally lower PRF (Pulse Repetition Frequency) signals can be seen by the band 1 receivers at one time, the processor is generally faster than the true device response time. Therefore, properly timed waits are performed where necessary within the software.

Trainer/device dual control of the simulator has been achieved by placing the interactions of the carry-on unit with the trainer and SLQ-32 into separate microprocessors, CPU 1 and CPU 3 respectively. Since all four processor run indpendently, no interference occurs.

Other features, of a more general nature, have been designed into the system to enable modifications to be easily performed, if necessary. All of the microprocessors share the full memory (with ample space) as well as equal access to all of the hardware interfaces via memory mapped I/O. Thus tasks or processors can be moved, added, or deleted as needed. Potential resource conflicts are handled by both hardware and software semaphores (flags) which temporarily prevent processors from accessing the resources protected. Thus, the processors need not be synchronized and can run independently.

Ultimately, the only way that a simulator designed for ease in keeping current can have that property tested is to present the simulator with a changed device. This happened suprisingly early in the case of the 20B5 SLQ-32 simulator. The original development operational software tape and source listing used for research were revision seven and they had a dummy threat library to ease some of the security problems. Late in integration testing, a request was made for tape with a genuine threat library. Revision eight arrived instead of revision seven. This is a major revision level (not a patch level) with over a year and a half of changes. No source listing or statement of what changes had been made accompanied the tape. Nevertheless when the new revision was brought up, the simulation was able to service it correctly. No changes were required, and the new tape was used shortly thereafter to conduct a successful demonstration to personnel, from the Navy code responsible for the maintenance of the SLQ-32 design.

SUMMARY

For the purpose of advanced level ESM/ECM/ECCM training, a simulator interfaced at the device computer offers many advantages: high fidelity, complex scenario capability, small package size, simple installation/removal, and ease of upgrade. To realize these benefits, however, work in the following areas is particularly important for this kind of simulator: real-time constraint determination, processing throughput, dual control of the simulator by both the scenario controller and the device, and testing/configuration. Comparison to other simulator architectures and the organic training concept shows that computer stimulation is a practical, effective solution to the readiness training requirement.

ABOUT THE AUTHOR

L. CHAIKIN is a Principal Development Engineer in the Training and Simulation Department of AAI Corporation. He is responsible for systems analysis, design, and development in the areas of ESM/radar simulation, interactive graphics, and image processing. He holds a B.A. degree in Natural Sciences from the Johns Hopkins University.