INCORPORATION OF DIGITAL AVIONICS SYSTEMS IN THE B-1 TRAINING SIMULATOR

O. R. MOYEN-VAN SLIMMING
Aeronautical Systems Division

INTRODUCTION

The B-1, which is intended to replace the B-52 in the Strategic Air Command (SAC) inventory, has several simulator requirements, ranging from study carrels to full mission simulators. The aircraft has a sophisticated avionics package, similar to that in the FB-111. It controls the navigation, weapon delivery and defensive functions for the aircraft. It utilizes a number of analog and digital computing devices whose functions have an impact on the simulator. A detailed analysis of the avionics system was accomplished in order to identify the problems that may be encountered in fulfilling the avionics function requirements for the crew training simulator and indicate which approach(es) to implementing the avionics functions would be most cost effective over the life cycle of the B-1 simulator. There are several simulator requirements which have no equivalents in the aircraft. One such requirement is the capability to freeze the mission, usually with the intent to reset the simulated aircraft to another position, or to instruct the crew concerning a portion of the mission without their having to concentrate on aircraft status. Another such requirement is parameter freeze, i.e., the ability to freeze certain parameters such as fuel status, pitch, altitude, etc. This allows the instructor to make a task easier by freezing one or more parameters so that the crew can concentrate on other tasks. These requirements represent a definite challenge to the implementation of avionics functions in the simulator.

AIRCRAFT OVERVIEW

The B-1 is a long range, supersonic, swing-wing bomber. Its mission includes tanker rendezvous, low level penetration at high subsonic speeds and the weapon delivery of high energy conventional stores, nuclear gravity stores, and SRAMs. The B-1 has a crew of four: pilot, copilot, Offensive Systems Operator (OSO), and Defensive Systems Operator (DSO). The OSO and DSO are physically located separately from the pilot/copilot in an aft station. There are presently provisions for two instructors, one in the cockpit and one in the aft station.

Pilot/Copilot Station

The most striking departure from previous aircraft instrumentations is the Vertical Situation Display (VSD), essentially an electronic Attitude Direction Indicator (ADI). It utilizes a computer driven CRT, which can also be used to display Forward Looking Infrared (FLIR) information. In production models this CRT may also be used to provide offensive station data, such as weapon status, to the pilot. Although the traditional dial-type instruments are still present (particularly in the backup instruments) much of the pilot/copilot information is provided on vertical tape type instruments and some in digital format (such as airspeed, mach number, and altitude).

The B-1 has a Terrain Following Radar (TFR) which has a display between the pilot and copilot. This system has two analog computers which are dual redundant and can operate in a manual or automatic mode.

In the first three prototype aircraft, the escape system consists of a crew module; however, in the production models this will be replaced by individual ejection seats. This change has necessitated some redesign of the instrument panels, particularly in the aft station. This may delay development of the aft station in the simulator.

The B-1 is also equipped with an automatic flight control system (AFCS) which provides a variety of pitch and roll, automatic guidance and automatic throttle control. This system uses two analog computers in dual redundant mode.

Aft Station

While Rockwell International has the airframe responsibility, Boeing (Seattle) and AIL share responsibility for the aft station, with Boeing being the overall avionics system integration contractor. The Offensive Station utilizes a Forward Looking Radar (FLR) for navigation and weapon delivery and a Forward Looking Infrared System (FLIR) for use in a nuclear environment. The FLIR presentation appears on a multi-function display (MFD) which is physically located above the FLR display. This display can also be used as an alphanumeric display (AND) in conjunction with the other three CRTs. In the normal mode twin ANDs are used to display navigation/weapon delivery information, while another AND is used in conjunction with the integrated keyboard (IKB) to select different options using the Offensive Flight Software (OFS) logic trees. In case of failures of a CRT, any of the other three can be used to display the required information. Navigation and Stores Management System (SMS) data are displayed on the NAV and SMS panels respectively. The OSO can also steer the FLIR sensor.

The Data Entry Unit (DEU) is a cartridge tape device by which the OSO can load the avionics computers with their computer programs and mission data. He has the capability to erase classified data and to shutdown the offensive computer complex. The DSO has the same capability over the defensive computers. Both operators have access to the Central Integrated Test System (CITS) Control and Display (CCD) panel, and a few flight instruments.

The DSO has an IKB and CRT which are identical to the ones in the offensive station, except that navigation/weapon delivery functions are automatically locked out (likewise the OSO cannot enter any DSO functions from his IKB). The DSO has two electronic display units (EDU) which can be used to display the threat environment in panoramic or situational format. The information displayed on the EDUs is entirely computer generated. Aural warning is provided for immediate threats such as a detected missile launch. Additionally, the DSO has the option to request reconstituted audio for any detected emitter.

The defensive system is basically an automatic system, but the DSO has the option to override the system and change jammer power, threat priorities, etc. In the production aircraft, the DSO will also have control over expendable ECM (flares and chaff) and possibly a tail warning system.

ON-BOARD COMPUTER SYSTEMS SIMULATION ALTERNATIVES

The presence of digital processors in the aircraft presents a challenge in the development and support of training simulators, particularly if the processors are *software programmable. The question of* providing the best way of simulating systems which utilize such processors is a vital one, since it has a great impact on development risk, performance, facility for change, and on initial and life cycle costs of the simulator. It is usually not advantageous to use aircraft equipment in the simulator because of the high cost of such equipment and because the simulator operates in a simulated instead of a "real-world" environment. It is neither necessary nor cost-effective, for example, to use an aircraft operable UHF transceiver, since any ground communication is usually handled by the instructor; hence, an inexpensive intercom set is sufficient. Likewise, it probably is impractical to use the nuclear-hardened devices used in the aircraft, since the simulator will not encounter an environment which requires such protection.

With programmable digital devices an additional consideration must be made, i.e., what kind of a change activity is expected over the life cycle of the aircraft (and consequently of the simulator)? Devices in which the program is expected to change frequently are usually core memory driven, while devices in which the software change activity is low frequently use Programmable Read Only Memories (PROMs), which are more rugged than core memory but are also more expensive. This limits their use in the simulator.

There are several alternative to simulating programmable digital devices. The following paragraphs will cover the approaches considered as candidates in a training simulator application.

Aircraft Hardware

In this approach the actual On-Board Computers (OBC) are used in the simulator with the aircraft Operational Flight Program (OFP). These OBCs may be softer (non-hardened) versions of the aircraft computer using non-mil spec parts, but still being functionally identical to the aircraft computer. The intent of this approach is to *obtain a hands-off update capability for the* simulator. The OFP support facility for the aircraft would send a tape, which is identical to the aircraft tape, to the simulator support team whenever the OFP is modified. Theoretically then, all that would be necessary to update the simulator would be to read the new tape into the OBC.

Advantages of Aircraft Hardware: Rapid OFP update for all software-only changes can be achieved, as well as one-to-one correspondence functionally with the OBC, thereby maximizing realism of signal stimuli and response to inputs. Also, the aircraft OFP is exercised much more in the simulator than in the actual aircraft, thus giving the possibility of finding bugs in the OFP on the ground which may not be found in the aircraft. This, however, does not necessarily improve the training environment.

Disadvantages of Aircraft Hardware: High recurring hardware costs are incurred and hardware life-cycle costs may be high also, since any aircraft computer change will have to be accomplished in the simulator. The interface between the aircraft computer and its environment in the simulator must be designed one-to-one, i.e., signals must be represented exactly (timing, amplitudes, etc.) as in the aircraft. This interface is complex and expensive. Simulator peculiar functions, such as freeze, may be difficult to implement, since the simulator must use the unmodified OFP. Particularly, parameter freeze is difficult to implement. One suggested method is to include simulator peculiar functions in the aircraft OFP. This would eliminate one of the advantages of simulation, i.e., the flexibility to implement any training requirements independent of the aircraft design implementation. Additionally, any future trainer peculiar requirements may require further OFP modifications. This would increase the size of the aircraft OFP without attendant benefits to the flying mission and may preclude mission essential changes. Another problem would arise if a new OFP change comes in and it does not play in the simulator. The fault may be in the OFP itself, in the simulator, or in the interface, and may be difficult to resolve since the OFP support team is not knowledgeable in the detailed design of the simulator, and the simulator support team is not necessarily knowledgeable in the details of OFP development. Close coordination of these two software support teams would lessen the problem but would not eliminate it entirely.

Hardware Emulator

The emulator approach requires a microprogrammable general purpose (GP) computer for each aircraft OBC that is being emulated. The internal structure of the GP computer is changed through microcode to allow it to accept the instruction set of the OBC. This requires a detailed knowledge of the internal workings of the aircraft computer that is being emulated, to insure that the emulator internally looks exactly like the OBC. With this approach, the actual aircraft OFP is directly loaded into the emulator and the OFP "thinks" it is executing in the actual aircraft computer.

Advantages of a Hardware Emulator: Because the emulator uses a GP computer, it is much easier to integrate into the simulator than the actual OBC. Interface problems are minimized because of commercially available peripherals. This approach has the same advantages as the OBC approach of rapid OFP update and simulator exercise of OFP software, while the hardware cost is less because a less expensive GP computer is used.

Disadvantages of a Hardware Emulator: The emulator has all the disadvantages of the aircraft hardware approach, except as stated above. Additionally, the GP computer must be of higher performance than the OBC it emulates, and a detailed description of its internal architecture must be available.

Software Translator

In this approach a computer program is designed to accept as input a set of OFP instructions and provide as output a set of computer program instructions which, when executed in the simulator computer, accomplish the "same" performance as the actual aircraft OFP. The translation process is usually accomplished off-line and may require several passes before an executable code is generated. The translator can use the aircraft programs and media, but must operate on the OFP before it can be used in the simulator.

Advantages of a Software Translator: Development costs are nonrecurring and hardware costs are minimized. Aircraft OFP and media can be used and updates can be accomplished relatively quickly.

Disadvantages of a Software Translator: High level organic support capability may be required to resolve aircraft OFP coding not recognized by the translator, including programmer tricks which take advantage of the hardware for which the OFP was written. There is no guarantee that every tape change will work in the simulator since verification of the translator is very difficult. Development of the translator must be considered a high risk item, both in the technical and cost area.

Cross Compiler

In the past, weight and time constraints had made it necessary for the OFP to be written in assembly language. This made the OFP very efficient as far as core and CPU time was concerned, but it was hard to read and to modify. With the advance of hardware technology, it became feasible to write the OFP in High Order Languages (HOL), such as Fortran or Jovial, which are easier to read and allow easier code modification. The usual procedure is to compile and assemble the HOL program on a large ground-based computer (such as an IBM 370) and execute the resultant machine code in the aircraft computer. It would be feasible to design another cross compiler which would take the same HOL program and process it to obtain a program which can be executed in the simulator computer. Thus the aircraft OFP would be executed in the simulator without the use of aircraft hardware.

Advantages of a Cross Compiler: Development costs and technical risks are minimized because of extensive industry experience in compiler development. The aircraft OFP in HOL format can be used (allowing exercise of aircraft software in the simulator) and relatively quick updates of OFP changes can be accomplished.

Disadvantages of a Cross Compiler: If the OFP is not 100% in HOL and/or changes are made after the compilation process and before the OFP is finalized, changes may still have to be made to the simulator OFP to obtain the same effect as in the aircraft. This would require a software maintenance team; however, as with the functional simulation approach, such a team is required for simulator software support anyway, and the use of a cross compiler would give the team a better handle on the problem than would, for instance, a translator because of the greater visibility of HOL over assembly code.

A problem may still arise with this approach in case of programming tricks (although they are less likely to be found in HOL than in assembly language programming). Another possible problem may arise in implementing training requirements such as freeze and reset; however, these could probably be implemented by additional software modules. If the OFP uses internal clocks, the timing of which are system dependent, the program may not execute exactly the same as in the aircraft; however, this is a poor programming practice and unlikely to occur.

Functional Simulation

A functional simulation of the OBC functions consists of computer programs developed from the OFP documentation. These programs could be designed as separate modules, communicating through a common data base, such that no special interface software is required. Control and execution of these OFP functional modules would be handled in the same manner as the other simulation modules and would be designed and modularly organized into the total Computer Program System (CPS). The usual procedure of implementing a functional simulation is for the simulator designer to analyze the documentation of the aircraft OFP and design his own software to duplicate those functions which can be observed by the crew in the simulator and are required for the training mission.

Advantages of Functional Simulation: This approach can be successfully implemented with low design risk and also offers the maximum flexibility to integrate into the total system design. Technical risks are low, due to extensive industry experience, provided the OFP functions are well documented. Instructional features, such as freeze, reset, malfunctions, etc., are readily designed into the CPS and present minimum difficulties in implementation. Any future instructional features are easily incorporated, because the simulator software is under complete control of simulator software support personnel and can be easily modified to accommodate new requirements. This approach minimizes both development time and computer equipment cost, since the software needs only to handle those functions which can be perceived and are necessary in the simulator.

Disadvantages of Functional Simulation: The accuracy (i.e. one-to-one correspondence with aircraft functions) by which the software operates will not be exactly the same as the actual OFP. However, this may not be of any consequence, since the simulator does not require the high fidelity required in the aircraft (since the training task is only concerned with what the crew can perceive, iteration rates may not need to be as high as in the aircraft, for example). Software costs may be high, since the simulator OFP will have to be developed from scratch, but this may be minimized due to the fact that there are many functions in the aircraft OFP which are not required in the simulator. Operational costs may also be high, since a team of software maintenance personnel must track all aircraft OFP changes and incorporate those changes which impact the training environment. However, the OFP is only a small part of the overall simulator CPS and there must be a software maintenance team anyway to provide organic support. Another possible disadvantage of this approach is that the simulator may lag behind the aircraft in OFP updates. Certainly, instances of this occurring can be pointed out, but the problem is more a management than an engineering one. It can be resolved if the simulator team is kept informed of future OFP changes, so that they can work on those changes which impact the simulator concurrently with the aircraft OFP team. Control of the OFP configuration is maintained by means of a Configuration Control Board (CCB) which must approve all proposed OFP changes. Notification of all CCB approved changes would allow the simulator support team adequate time to incorporate these changes. There should be no reason then for the simulator to lag behind the aircraft.

SIMULATION OF B-1 ON-BOARD SYSTEMS

Devices using ROMs or PROMs can be functionally simulated, emulated, or the actual hardware may be used. Because the program resides in hardware, neither translator nor cross compiler are viable options. With computers using core memory, all of the above options may be viable. Most of the processors used in the B-1 use PROMs. It was found that all of these devices could be best simulated functionally, either because they monitored real-world environment (such as the air data computers or the engine instruments signal converters) or because their function could be implemented in software in a more cost effective manner (such as the electrical multiplex processors, which operates on sets of Boolean logic equations). Table 1 lists several B-1 devices, their function, and the reasons for functional simulation. In addition, four major systems utilizing programmable digital processors are discussed in greater detail in the following paragraphs.

Electrical Multiplex System (EMUX)

This system provides the control of electrical power in the aircraft and provides the means of transferring data between subsystems. The B-1 has two EMUX systems. The forward avionics bay and the forward and aft weapons bays are serviced only by the left EMUX, while the central weapons bay and aft avionics radar bay are serviced only by the right EMUX. The crew compartment, the central avionics bay, and the wheel well equipment bay are serviced by both EMUX systems. Each EMUX system has two control boxes (one being primary and the other backup) which control message traffic and access data, and provide logic processing, format and distribute outputs. In addition, each side has a Central Integrated Test System (CITS) interface box. EMUX samples the status of switches, circuit breakers, and power controllers in the aircraft. It operates on a set of Boolean logic equations and issues power controller commands if the equation's logical value is true. In addition to data transfers, EMUX also does serial to discrete and discrete to serial data conversion; conducts self-tests; outputs data to CITS and accepts test commands from CITS in the ground mode. The structure of EMUX makes it convenient to

TABLE 1

SUMMARY OF B-1 DIGITAL DEVICES

Device	Function	Reasons for Using Functional Simulation
Central Air Data Computer (CADC)	Uses air pressure & temperature to calculate true airspeed, mach number, and related data.	It is mainly an A-to-D converter and is stable in design and not likely to change.
Rotation-Go-Around/ Angle-of-Attack/ Air Vehicle-Limit (RGA/AOA/AVL)	Provides standardized AOA, air vehicle limits, and pitch steering information to cockpit instruments.	It is mainly an interface for cockpit instruments and is stable in design and not likely to change.
Fuel and Center of Gravity Management System (FCGMS)	Automatically maintains the CG by transferring fuel between tanks.	This device can be implemented with simpler algorithms in the simulator computer to provide crew training in its operation.
Gyro Stabilization System (GSS)	Provides heading, pitch, roll attitude, and rate-of-turn reference data.	The simulator "moves" in a simulated environment, hence there is nothing for the GSS to sense. Its output should be functionally simulated.
Engine Instruments Signal Conditioning & Distribution Unit (SCDU)	Processes engine sensor data for the cockpit engine instruments.	It is an A-to-D converter for the flight instruments and is stable in design and not likely to change.
Flight Instruments Signal Converter (FISC)	A-to-D and D-to-A conversion to interface the ACUC with various aircraft systems.	Functional simulation of this interface is simplier and less costly.
Vertical Situation Display (VSD)	Generates the real-time video display of flight parameters such as roll and pitch attitude, AOA, airspeed and altitude.	In a large simulator buy, it is more cost-effective to functionally simulate its functions and use non-aircraft parts for the display.

implement in the simulator. In the past, simulator engineers in designing the simulator had to search documentation to ascertain what effect each and every switch had on the characteristics of the aircraft, i.e., what events would the crew perceive if a switch were thrown, a button were pushed, etc.? The EMUX documentation provides the logic equations in a very simple format. Implementation of these functions should be simplified over previous simulator developments. It is not, however, necessary to physically include the EMUX system, since the simulator has no Station Logic Units (SLU), such as bombbay doors or launcher racks, to control, and data transfer will consist of core-to-core communication within the simulator host computer. Thus, only the functions of EMUX should be simulated.

Central Integrated Test System (CITS)

The CITS function is to detect faults and isolate faults to a unit, and display the fault detection/isolation information to the crew. It also records this information on a printer and maintenance recorder for post-mission debriefing and maintenance purposes (neither is accessible in flight). The CITS collects information via Data Acquisition Units (DAU), and interfaces with EMUX and the avionics complex. Crew information is displayed on the CITS control and display panel (CCD) which is situated in the aft station between the two operators. Upon a detected failure, CITS will transmit a warning message to be displayed on the CCD's Alphanumeric Display (AND) and turn on the light indicating which system failed. The crew can retrieve any previous messages and access core memory of the CITS Dedicated Computer (CDC) to obtain further information on the fault. CITS communicates with its peripherals on a CITS multiplex system (CMUX). In the simulator the aircraft malfunctions that are considered necessary for training are preprogrammed and under control of the instructor. Thus there is no need to detect faults, only to display the indication of a fault to the crew. Since the printer and recorder are not needed by the crew in the simulator, these CITS functions are not necessary. Likewise, the ground functions of CITS in which it controls EMUX are not needed since they are maintenance functions only. The C&D function, including the capability of the crew to access certain portions of core in the CDC, can be handled by functional simulation in the simulator host computer.

Avionics Control Unit Complex (ACUC)

The ACUC contains three SKC 2070 computers, two for the offensive avionics and one for the defensive avionics. The two offensive avionics computers operate in parallel, but each can be configured to assume the functions of the other in case of failure of one. Communication among the computers is via an avionics multiplex system (AMUX). In addition to private core memory, the three processors share a mass storage unit (MSU). The MSU is a drum with 400K of 32 bit words storage capacity and contains all the computer programs and mission data for the avionics computers. The ACUs are loaded via the DEU or the MSU. The Weapon Delivery Avionics Control Unit (WDACU) is loaded first. From a cold start (MSU not loaded), this is accomplished by the OSO inserting the program tape into the DEU. After the WDACU is loaded, the general navigation avionics control unit (GNACU) can be loaded either from the DEU or from the MSU. The defensive avionics control unit (DACU) can only be loaded from the MSU. The Offensive Flight Software (OFS) is executed in the GNACU and WDACU. The basic iteration rate is 16, with each 62.5 msec major frame divided into four minor frames. The functions are executed alternately during each minor frame. In the backup mode (one computer down) all functions are executed in the same computer. During each minor frame, the foreground programs are executed first. Any remaining time is devoted to background tasks (Impact Point Calculations, Kalman filters, and self-tests).

The Defensive Flight Software (DFS) is executed in the DACU in major frames (62.5 msec) only. There are presently no provisions to backup the DACU, although technically it would be feasible to have one of the offensive ACUs operate in the backup mode and have the other reconfigure itself to the DACU. Implementation of this capability would require some engineering redesign. The OFS and DFS are programmed for the greater part (by contract at least 75%) in Jovial J3B.

Radio Frequency Surveillance/Electronic Counter Measures System (RFS/ECMS)

This system uses a pre-processor avionics control unit (PACU) to process the emitter data from the antennas and control the jammers. The PACU is loaded from the MSU through the DACU. It contains the Defensive Order of Battle (DOB) and any defensive mission data that has been entered by the DSO using the IKB. The PACU processes the incoming signals and correlates their characteristics with the stored data to produce a prioritized table of threats which it transmits to the DACU. The DACU prepares this data for display on the EDUs and the aural warning system when appropriate.

The PACU uses a Litton LC4516D computer. The pre-processor flight software (PFS) is presently coded in assembler language, but is in the process of being rewritten in Jovial J3B. A cross compiler will be used to generate the aircraft OFP. Since the simulator will not use actual antennas and jammers, the signal processing and control functions of the PACU are not needed in the simulator. What has to be incorporated in the simulator are the threat analysis and prioritization functions. In the simulator, the threat environment is known (i.e., it is a simulated environment), hence the functions of the PACU that are needed in the trainer should be functionally simulated. These functions should include the capability to change threat and jammer characteristics (PRF, power, etc.) and add new threats. An additional option exists if the cross-compiler approach is used to simulate the ACUC. In that case it may be cost effective to design a cross compiler for the PACU to incorporate its functions into the simulator. This would minimize the interface with the ACUC.

ACUC Simulation

There are several facts which must be taken into account while considering the simulation alternatives for the ACUC computers:

1. The aircraft SKC 2070 computer costs about $600K per copy in the nuclear hardened version and about $350K in the softer (non-hardened) version. It may be possible to acquire a still softer version, using non-mil spec parts, in the $100-150K range. The problem with the last approach is that we are essentially dealing with a new computer, whose performance, reliability, etc. is unknown. Additionally, one advantage of using aircraft hardware (i.e., reduced logistics problems) would be lost if non-hardened versions are used. Even the lowest cost version would still cost $300-450K per simulator for all three computers. It may be possible to eliminate one computer by operating one ACU in backup mode permanently, and this would eliminate some of the test functions, but they may not be necessary to meet the training requirements anyway. Even so, this would eliminate the capability to train crew members to operate in both modes.

Implementing trainer essential functions are difficult. Freeze, for example, could be implemented simply by initiating an interrupt stopping the real-time clock in the ACU. However, this would cause all displays to go dead, unless the information necessary for display is somehow retained outside the ACU. To implement parameter freeze requires modification of the OFP for simulator use. This means that either the simulator OFP will not be identical to the aircraft OFP, or the aircraft will have to fly with simulator peculiar functions in the OFP. Other problems include record and playback capability, auto demonstrations, crash override, and malfunction insertion and control.

2. Data concerning the internal operations of the SKC 2070 computer are company proprietary. This will make it difficult to design an emulator using a microprogrammable GP computer which reflects the structure of the OBC exactly. There is therefore no guarantee that a change in the aircraft OFP will have the same effect in the simulator and some simulator peculiar OFP support activity may be required with each tape release. As with the actual hardware approach, the same situation exists concerning simulator peculiar functions with the same problems of having either two different OFPs (one for the aircraft and one for the simulator) or having simulator peculiar functions in the aircraft OFP.

If this approach is used, it would be very cost effective to use the same microprogrammable GP computer for the simulator host computer(s) to allow commonality of equipment and ease the maintenance function. This would limit the choice of simulator computers to 32 bit microprogrammable GP processors. An indepth study of the SKC 2070 and available GP computers will be required to determine which computers are likely candidates for an emulator approach.

3. The present method of preparing the aircraft program tape requires the use of a cross compiler (this is also true of the pre-processor flight software, which was covered in the previous section). The design and cost risks of a cross compiler to accommodate the simulator computer would be low. The software is organized such that all programs are either in Jovial J3B or in assembler language (no in-line coding is allowed). This would make it easier to accomplish changes to the program. Simulator functions peculiar to the training environment could be inserted as additional code and new requirements could be implemented without modification of the aircraft OFP. The update capability is not as good as with the previous two approaches, since with each new tape release at least some analysis and possibly recoding will have to be done to obtain a simulator OFP. However, this approach gives much greater visibility of the total system and any bugs can be fixed more easily than with the hardware or emulator approach.

4. The translative approach has all the disadvantages of the cross-compiler approach, i.e., the translator requires off-line processing to produce a simulator OFP, including engineering analysis and computer run time. In addition, any code change would

have to be done at the assembler or machine language level. This removes the advantage of increased visibility.

5. From a development standpoint, functional simulation of the avionics has the lowest technical risk. This approach minimizes the total software and hardware, since the designer can eliminate all functions which are necessary in the aircraft, but have no equivalent function in the simulator. This is particularly true in the B-1 where much of the software is concerned with testing, statistical computations (Kalman filters), and control and display (C&D) - all of which are either unnecessary in the simulator or, as in the case of C&D, can be done more efficiently in a GP computer. This approach will require that OFP change specifications be made available as early as possible to allow simulator software support teams to keep the simulator OFP up-to-date with the aircraft.

6. There may be other approaches, including combinations of the above; however, they have not been proven in the training simulator environment.

CONCLUSIONS

All analog computers and all digital processors, with the exception of the three computers in the ACUC (and possibly the PACU, if the cross compiler approach is used) should be functionally simulated. Two approaches to implementing the ACUC computers are considered more cost effective over the life-cycle of the B-1 simulator than the other approaches. The approach which presently appears to be the better of the two is functional simulation. The following reasons are given in support of this conclusion:

1. Low risk. This approach has been implemented on many training simulators and it is a proven and reliable approach.

2. Short development time. Because of extensive industry experience with this approach, development time will be minimized. There are no unknown areas or surprises which can delay the program. This is particularly important in the case of the B-1 simulator which has a severe time constraint.

3. Low cost. The cost to rewrite the OFP functions for the simulator is nonrecurring and amounts to less than 1% of the projected total B-1 simulator cost. Even assuming that soft versions of the OBC at a reduced cost can be obtained, the hardware cost will still be several times higher than the cost of a functional simulation. This does not take into account the additional cost of the AMUX system, the cost of the interface with the simulator, contractor overhead charges for the hardware procurement, and the software cost of having to implement a more rigorous environment model if the OBC is used.

4. High flexibility. Simulator peculiar requirements such as freeze, reset, malfunctions, and parameter freeze are easily implemented. Any future requirements can be easily implemented. At the same time, the simulator OFP is not hardware dependent, i.e., if the aircraft computer is replaced or modified, the simulator OFP need only be modified if there is a change in performance that the crew can perceive. In such a case there would be no high hardware costs associated with the simulator.

5. Control of configuration is maintained by the simulator community. This may be the most important advantage of functional simulation, i.e., the simulator is not dependent upon the aircraft OFP for its training capabilities. This would not be the case if the actual hardware or an emulator were to be used, since the aircraft OFP would then have to contain the freeze, reset, etc., capabilities which are necessary for the simulator. In other words, the simulator requirements would be in direct competition with the operational requirements for time and core in the OBC, and dependent on the aircraft community for implementation.

6. Good update capability. Traditionally this approach has been accused of causing long delays in simulator updates and expensive ECPs to keep the simulator current with the aircraft. These facts may have been true in the past, but are not necessarily true today. The Tactical Air Command (TAC) has had great success with their Development Technician Team (DTT) to update all simulator software, including the OFP portions. Last year TAC rejected a suggestion to incorporate the TC-2 computer into the A-7D simulator on the grounds that they had no update problems. They felt that as long as their DTT was kept informed of future OFP changes, they could beat the aircraft in implementing a change. They also felt that the approach was too expensive, degraded their training capability, and would take too long to implement. HQ USAF/XO and AFSC concurred with TAC's position.

The recommended method of implementing OFP changes in the simulator is to require that the SAC equivalent of the DTT be kept informed of any future OFP changes and that the funding for any aircraft ECPs which affect the simulator contain funding for simulator ECPs as well. This will insure a timely update capability for the simulator OFP without requiring a high cost/high risk approach to simulating the avionics system.

The other viable approach is the cross compiler. Its cost is comparable to that of the functional approach and is much less than that of aircraft OBCs or emulators. It has a slightly higher development risk than functional simulation, since it has not been proven in a training simulator, but it would give maintenance personnel a better chance of tracking any bugs in the system which may be impossible to find with the aircraft OBC or emulator approaches. It allows some of the functional simulation options of deleting code which has no impact in the simulator or adding subroutines to take care of freeze, reset, etc. It would also allow the option of using Jovial J3B as the simulator computer language to minimize interface problems.

To take care of the non-Jovial code (the executive, I/O, etc.), the simulator contractor could write new code, or else microprogram the simulator computer to accept the instruction set of the SKC 2070. This approach is not as technically proven as the functional approach, but it is the best of the approaches which use the aircraft OFP, and thus would be the best approach if exercising the aircraft software in the simulator is required. This does not mean that the other approaches are without merit. On the contrary, with different applications and requirements, the emulator, for example, may be the best approach. But for the B-1 training simulator program, those approaches are not recommended for the reasons stated above.

A further trade-off study should be made to determine which of these two approaches is best suited for the B-1 simulator in terms of life-cycle cost and training effectiveness. It should be noted that the aircraft aft station is behind in development compared to the front station, and hence a delay of one year in production of the simulator aft station would prevent costly ECPs to keep the simulator configuration current with the aircraft. This time period can be used to accomplish the recommended study. The study should, as a minimum, address the initial development cost; the time for development; a development risk assessment; an analysis of anticipated software changes and their impact on the simulator; the update capability; the number and qualifications of software maintenance personnel required; and predicted life-cycle costs for each of the two approaches.

ABOUT THE AUTHOR

CAPTAIN OTTO R. MOYEN-VAN SLIMMING is an Electronic Engineer in the Directorate of Equipment Engineering, Deputy for Engineering, Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio. He is presently the lead computation systems engineer in the B-1 System's Program Office for the B-1 Training Simulator program. He has worked on the B-52/KC-135 and A-7D Mission Simulators, the C-5 Cockpit Procedures Trainer, the Undergraduate Navigator Training System (UNTS), the Simulator for Electronic Warfare Training (SEWT), and the Digital Radar Landmass System for the F-111 (Project 1183). Previously, he has been a Quality Assurance Inspector for the Ground Electronic Engineering Installation Agency (GEEIA) and has instructed Royal Thai Air Force personnel in the installation and maintenance of ground-to-air communication systems. He received the B.S.E.E. degree from California State University at Sacramento and is completing thesis requirements for the M.S.E.E. degree from the Air Force Institute of Technology. He is a member of Phi Kappa Phi.