N86-24514

1985

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

A TECHNIQUE FOR INCORPORATING THE NASA SPACELAB PAYLOAD DEDICATED EXPERIMENT PROCESSOR SOFTWARE INTO THE SIMULATION SYSTEM FOR THE PAYLO CREW TRAINING COMPLEX

Prepared By:	Dale A. Bremmer, Ph.D.
Academic Rank:	Adjunct Associate Professor
University and Department:	Oklahoma State University Education Research
NASA/MSFC:	
Division:	Operations Development
Branch:	Man/Systems Integration
MSFC Counterpart:	Charles M. Lewis
Date:	August 5, 1985
Contract No:	NGT 01-008-021 The University of Alabama in Huntsville

A TECHNIQUE FOR INCORPORATING THE NASA SPACELAB PAYLOAD DEDICATED EXPERIMENT PROCESSOR SOFTWARE INTO THE SIMULATION SYSTEM FOR THE PAYLOAD CREW TRAINING COMPLEX

BY

DR. DALE A. BREMMER
Adjunct Associate Professor
OKLAHOMA STATE UNIVERSITY
Stillwater, Oklahoma

ABSTRACT:

The NASA space shuttle missions offer an opportunity for a principle investigator (PI) to design and operate many experiments that were impossible to conduct prior to the advent of manned space flight. With a projected shuttle schedule of two launches per month, numerous PIs will be able to take advantage of the opportunities and challenges provided by the spacelab. Many of the onboard spacelab experiments are controlled by an experiment computer. In addition, the experiment computer serves as a focal point for much of the data acquisition and transmission activities of the experiments. The result of the high demand for payload management activities and the geriatics of the antiquated technology of the spacelab's experiment computer is a limited ability to control the operation of the PI's experiment. Hence, it is often necessary to incorporate a dedicated experiment processor (DEP) in the experiment design. Shuttle mission flight experiments that gather data or require control information on an uplink from the Payload Operations Control Center (POCC) may need a (DEP) as an integral part of the flight hardware. The DEP (1) offers some capability for immediate analysis, (2) provides additional data storage and transmission capabilities, (3) allows the PI to control and/or alter the experiment operations while in flight, (4) provides for feedback control, (5) permits some real-time data analysis required during the mission. (6) simplifies the design of bulky and weighty hardware, and (7) if autonomous, allows relatively complete checkout and verification prior to integration with the onboard experiment computer.

This study proposes to assess the feasibility of some off-the-shelf microprocessors and state-of-art software: (1) as a development system for the PI in the design of the experiment model, (2) as an example of available technology application for future PI's experiments, (3) as a system capable of being interactive in the PCTC's simulation of the DEP, preferably by bringing the PI's DEP software directly into the simulation model, (4) as a system having bus compatibility with host VAX simulation computers, (5) as a system readily interfaced with mock-up panels and information displays, (6) as a functional system for post mission data analysis.

ACKNOWLEDGEMENTS:

Acknowledgements are extended to the National Aeronautics and Space Administration (NASA), the American Society of Engineering Educators (ASEE), the George C. Marshall Space Flight Center (MSFC), and in particular the MSFC Man/Systems Integration Laboratory that funded and provided the facilities needed to make the NASA/ASEE Summer Faculty Fellowship Program feasible.

Thanks are extended to Charles M. Lewis for listening to the philosophy of this project and for offering guidance in its undertaking; to Tina Melton for assisting in my understanding of the operation and scope of the Payload Crew Training Complex (PCTC) and for the techinical editing of this paper; and to Rose Ann Kafer for reading and commenting on choice of words used to make this paper more readable and for adding a little sanity at the proper time.

INTRODUCTION:

The NASA space shuttle missions offer an opportunity for a principle investigator (PI) to design and operate many experiments that were impossible to conduct prior to the advent of manned space flight. With a projected shuttle schedule of two launches per month, numerous PIs will be able to take advantage of the opportunities and challenges provided by the spacelab. Many of the onboard spacelab experiments are controlled by an experiment computer. In addition, the experiment computer serves as a focal point for much of the data acquisition and transmission activities of the experiments. The result of the high demand for payload management activities and the geriatics of the antiquated technology of the spacelab's experiment computer is a limited ability to control the operation of the PI's experiment. Hence, it is often necessary to incorporate a dedicated experiment processor (DEP) in the experiment design. Shuttle mission flight experiments that gather data or require control information on an uplink from the Payload Operations Control Center (POCC) may need a (DEP) as an integral part of the flight hardware. The DEP (1) offers some capability for immediate analysis, (2) provides additional data storage and transmission capabilities, (3) allows the PI to control and/or alter the experiment operations while in flight, (4) provides for feedback control, (5) permits some real-time data analysis required during the mission. (6) simplifies the design of bulky and weighty hardware, and (7) if autonomous, allows relatively complete checkout and verification prior to integration with the onboard experiment computer.

Each shuttle mission is an aggregation of many tasks and potentially many DEPs, each from a different PI. Once the mission's tasks have been integrated into a performance timeline, the Payload Crew Training Complex (PCTC) utilizes simulators to provide operations training for the payload and mission specialists and the ground operation teams. To produce a well trained flight crew, this simulation system must simulate faults and equipment failures, accommodate alternate mission flight profiles and model the PI's experiment operations. These software models must realistically simulate the experiment's hardware as well as the DEP and experiment computer software. This results in a complex computer programming task requiring significant data base modification with any variation in the timeline of the mission. As changes are made to the actual experiment hardware and software, the existing simulation code has to be significantly modified to accurately reflect the experiment operations.

Currently the following software simulators (models) of major flight subsystems are required to be developed for the complete simulation and training capability;

- Experiment Computer Operating Systems (ECOS) Model
  - relatively constant from mission-to-mission
- Experiment Computer Applications Software (ECAS) Model
  - special processing software which runs on the spacelab experiment computer under ECOS
  - unique to each experiment
- Experiment Hardware Model
  - simulates experiment hardware functions, feedbacks, telemetry, science data, etc.
- Experiment DEP Software (where implemented) Model
  - independent of ECOS/ECAS
  - unique to each experiment
  - interacts with the Experiment Hardware Model

As a result of the freedom offered the PIs in the configuration of their DEP software, (this is feasible because the software is the farthest removed from the interfacing spacelab systems) there are, typically, some last-minute changes to the flight DEP up to the final stages of the flight experiment integration and verification. These late changes, while often benign relative to flight software, frequently require massive changes in the corresponding DEP simulator software with consequent deleterious effects on training and simulation schedules and validity.

Neither the training function nor any other echelon should succinctly dictate the DEP utilized by the PI. However, the scientific and engineering objectives of the shuttle missions can hopefully be more near optimal in flight and be more easily simulated at the PCTC by establishing some relevant computer hardware architecture and software attributes in line with the aforementioned conditions.

It would be impractical to propose system homogeneity throughout the entire hardware and software systems utilized by the PIs, the onboard experiment computer, the Payload Crew Training Complex, the simulation computer, and any system for post mission data analysis. However, the hardware and software entities, where practical, should be relevant to the state-of-the-art, off-the-self items, compatible (on the back plane) with the PI's experiment sensors, and compatible with the host simulation computers via buses or communication protocols.

Attempts to improve the incorporation of the DEP model, hardware and software, into the PCTC simulation models should encompass the PI's modeling and development techinques and the hardware and software utilized in the experiment model. Thus, there is a need to study some currently available hardware and software relevant to the DEP's function, the shuttle mission objectives, and the PI's freedom to alter the DEP software as required by the experiment, and the inherent limitations in the PCTC simulation.

It would be advantageous to the PI and the mission objectives, if systems employed by the PI during experiment development were also interactive with the PCTC simulation computers during the training of payload and mission specialists. There would be a reduction the setup time and mock-up initializations if the PI's development system were interactive and bus compatible with processors used in conjunction with mock-up panels and displays. There would be additional benefits if the same system could be used in any post mission data analysis.

If the type of development hardware and software used by the PI is also functional in the programming and the manipulation of the simulation model by the PCTC, then there will be much higher integrality between the PI's experiment model and the PCTC's simulation activities. If the PI's software is in an appropriate language, such a C-language, then incorporating the DEP into the simulation model may be little more than doing a cross-assembly of the PI's software under the operating system of the PCTC's host simulation computers.

This study proposes to assess the feasibility of some off-the-shelf microprocessors and state-of-art software: (1) as a development system for the PI in the design of the experiment model, (2) as an example of available technology application for future PI's experiments, (3) as a system capable of being interactive in the PCTC's simulation of the DEP, preferably by bringing the PI's DEP software directly into the simulation model, (4) as a system having bus compatibility with host VAX simulation computers, (5) as a system readily interfaced with mock-up panels and information displays, (6) as a functional system for post mission data analysis.

STATEMENT OF THE PROBLEM:

The PI needs a DEP (hardware and software) that is easily obtained, somewhat universal in its application, readily adaptable and reliable to the experiment. It should also be functional and/or compatible with computers used for post mission data analysis. It is important that the PI be able to make changes to the DEP software late into the pre-flight period. The hardware and softwarte support systems, used in the development of the PI's experiment, should have a high level of technology integrity and integrality with the DEP, with the simulation computers, and with computers used in post mission analysis. Additionally, the configuration should support commonality between mission objectives.

The PCTC needs both a DEP hardware and software model that can be readily incorporated into the simulator as a module that is compatible with the host VAX computer. This DEP software should be easily transportable or readily assembled under the PCTC simulator environment. Thus, ideally the hardware and software support system used by the PI should be functionally compatible, in an on-line mode, with the simulation host computer.

Efforts, to assist in the above problems, are intended to lead toward a more efficient experiment design for the PI and better use of hardware and software expenditures. The efforts should produce a simpler, more efficient, and more relevant simulation of the mission in the PCTC. The efforts will detail a current technological example for future PIs and study systems that will also assist the PI in post mission data analysis.

OBJECTIVES OF THE PROPOSAL:

The objectives of this proposal are:

(1) To study the INTEL Model 310 series, utilizing a 80286 miroprocessor chip and C-language operating under a RMX-86 operating system (Such a system would serve as a functional DEP development system capable of assembling the necessary software for the PI's DEP and converting or transferring it into VAX simulation compatible code for use in the PCTC.)

(2) To study the feasibility of interfacing the Intel Model 310 series MULTIBUS to a VAX 785 UNIBUS via a commerically available plug-in, self-contained boards

(3) To assess the functionality of the Intel Model 310 as a system to be an interactive system in the PCTC with host VAX simulation computers and the transportability of the DEP software model into the VAX computer via cross-assembly techniques

(4) To assess the value of using the Model 310, and the DEP, or DEP compatible processors as a tool for post mission data analysis

(5) To assess the feasibility of using a 80286 single board computer (compatible with the Model 310, the DEP, and the simulation computer) as an intelligent package in a behind-the-panel arrangement within the training mock-up, functioning as an interface to the experiment control and display panels

(6) To study the feasibility of using the Model 310 in a local area network or in a bit bus communication arrangement to initialize and manage 80286 single boards in the mock-up panels during simulation operation

GOALS OF THE OBJECTIVES:

Goals to be reached in this study are:

(1) To employ the 80286 microprocessor (as a Intel Model 310 series) to serve as a development tool for the PI in the development of his software for the flight experiment

(2) To demonstrate the compatibility and ease of using a 80286 microprocessor as a DEP with the Intel Model 310 and VAX computers and associated software as support systems

(3) To use the C-language in the design and programming of a DEP

(4) To use the Intel Model 310 remotely, at the PI's parent facility, in the experiment design and the DEP software development

(5) To use the Intel Model 310 as an interactive system in the PCTC simulation operations

(6) To interface the MULTIBUS of the Intel Model 310 with the UNIBUS of the VAX simulation computers

(7) To transport the DEP software, developed under C-language, into the VAX simulation computers via a cross-assembly program operating under the VAX operating system

(8) To use the Intel Model 310 series, located at a PI's parent facility, as a post mission data analysis support computer

(9) To set precedents for future PIs designing an experiment model and developing the associated DEP hardware and software systems.

(10) To simplify and standardize development of simulator control and display panels used in the PCTC by employing DEP compatible processors behind the mock-up panels and using networking techniques to link the various panels to the host system (This would additionally reduce the current panel cabling scheme to a single co-axial or optical fiber cable.)

HARDWARE & SOFTWARE:		
Two Intel 286/310-17 Real-time Starter Kits (a $42,000 market value) a 80286 microprocessor based system with an 80287 80 bit numeric processor, 1/2 MByte RAM, 320 KByte Floppy, 19 MByte Winchester, RMX operating system, Assembler 86, Link 86, Locate 86, PL/M 86, AEDIT full screen editor, PSCOPE (debugger), FORTRAN	$9,995.00	
C-86 Compiler	300.00	
C-86 Cross Compiler for VAX	300.00	
Interface boards (MTC Model 86/11W, Mesa Technology Corp.) for Intel 310 to VAX 785	6,000.00	
HARDWARE & SOFTWARE TOTAL		$16,595.00
PERSONNEL SUPPORT:		
On site (Huntsville) housing.(Nov 2 to Feb 3)	$1,000.00	
Travel expenses	500.00	
SUPPORT TOTAL		$1,500.00
TOTAL		$18,095.00

NOTE::

THE ABOVE IS ONLY AN ESTIMATE AND DOES NOT INCLUDE ANY OVERHEAD COSTS.

ANY COMPUTER OPERATING COST RELATIVE TO TEST OR CHECK OUT ARE NOT INCLUDED.

TIMELINE OF THE PROPOSED PROJECT:

July 20, 1985 to August 10, 1985, - on going.
November 2, 1985 to February 3, 1986.
May 10, 1985 to August 30, 1985 - (Under NASA/ASEE support)

ACTIVITIES LIST:

List of activities for the research project under this proposal to the MAN/SYSTEMS LABORATORY.

A. Phase I - Early Activities:

(1) Identify the PI's for the laboratory missions:

MISSION	LAUNCH DATE
ASTRO-1 61-E	3/6/85
EOM-1/2 61-K	9/3/86
ASTRO-2 71-B	10/30/86
IML-1 71-K	5/7/87
ASTRO-3 71-N	7/15/87
SUNLAB-1 71-Q	9/21/87
EOM-3 81-C	11/23/87

(2) Identify the type of processor the PI will use for any DEPs designed for the above missions

(3) Identify how many present, past, and future PI's are using an Intel system in the development of the DEP flight software or in the development of their original experiment model.

(4) Identify the software employed and, if used, the host mainframe systems.

(5) Determine the experiments that were, and those that could have been, coded in C-Language.

(6) Determine the limitations, if any, of using C-Language in the DEPs employed in the missions.

(7) If a PI's DEP is other than an Intel and uses (or could use) C-Language, determine what is the possibility of doing a cross-assembly of the DEP flight software under the PCTC VAX simulation computers.

(8) Identify those experiments suitable for study in this proposal.

(9) Investigate the feasibility of the PI doing his DEP modeling using an Intel Series 310-17 system employing iRMX^{TM} 86 Real Time Operating System.

(10) Investigate the possibility of rolling the PI's DEP into the simulation model via a cross-assembler under VAX control.

(11) Investigate the possiblility of doing the above (10) for DEPs not compatible with C-Language.

(12) Investigate the feasibility of, and identify the constraints against the provisions of fault insertion techniques (for simulations and crew training) despite the use of flight configuration DEP software.

B. Phase 2 - Intermediate activities:

(1) Obtain two Intel Model 310-17 systems with 80286 micoprocessors.

(2) Construct one model relevant to the selected mission DEP.

(3) Transport the constructed DEP model into the PCTC simulation.

(4) Obtain a hardware board, bus coupler, to make the Intel Model 310's MULTIBUS function in a compatible bus extension mode with the UNIBUS of the VAX 785 of the PCTC.

C. Phase 3 - Follow-up activities:

(1) Demonstrate that the payload and mission specialists can undertake crew traing at a remote site as well as at the PCTC.

(2) Demonstrate that the 80286 system with the C-Language forms a viable structure for the flight DEP, the original modeling of the DEP, and for post mission data analysis.

(3) Verify the existing techniques of accessing NASA data banks, after the mission, for PI's post mission data analysis.

(4) Demonstrate the feasibility of using the 80286 microprocessor as an interactive interface with the simulator mock-up panels and information displays

(5) Demonstrate the feasibility of using a video disk player under a DEP control to generate the fault display scenes for simulation training activities.

(6) Demonstrate incorporating the DEP behind the panels of the mock-up and communicating to the PCTC via an optical fiber, coaxial cable, or an appropriate high speed bit bus using software controlled ports.

(7) Demonstrate using the Intel Model 310 system to load and control the processors behind the panels of the mock-up.

TIMELINE - SCHEDULING CHART:

AUG 85	NOV 85-FEB 86	MAY 86-SEPT 86
PHASE 1		
	PHASE 2	
		PHASE 3

REFERENCES:

1. Intel Corporation, C-86 Compiler User's Guide, Intel Corporation, Santa Clara, California, 1983.

2. Intel Corporation, iRMX^{TM} 86 Programming Techniques, Intel Corporation, Santa Clara, California, 1983.

3. Plum, Thomas, Learning to Program in C, Prentice-Hall Englewood Cliffs, New Jersey, 1983.

These references are not cited in this paper. These references were supportive in the development of the philosophy implied and the hardware and software denoted.