A NEW CONCEPT FOR APPLICATION OF MICROCOMPUTER TECHNOLOGY TO REAL-TIME TRAINERS

CHARLES F. SUMMER
Naval Training Equipment Center
Orlando, Florida 32813

ABSTRACT

This paper presents a summary of an analysis of a new and unique concept of microcomputer technology application to real-time trainers that has been developed by NAVTRAEQUIPCEN. It summarizes the technical objectives, the conceptual analysis, technical feasibility, and life-cycle cost trade-offs. The required technologies for system implementation are identified and a status of the exploratory development to achieve the required technologies and demonstrate concept feasibility is provided.

INTRODUCTION

Over the past decade, the proliferation of digital computers embedded in trainer systems with the attendant proliferation of assembly-level languages has resulted in increased life-cycle costs of all types of real-time trainers. Large scale integration (LSI) and very large scale integration (VLSI) technology now available and embodied in state-of-the-art microprocessors and microcomputers suggests computer system architectural concepts that potentially can reverse this proliferation. Further, these concepts offer the possibility for extensive hardware - software modularity, hardware - software standardization and overall performance improvements with significant reduction in life-cycle costs of all types of real-time trainer systems.

Various concepts of microcomputer system architectures with applications for trainers have been under investigation at NAVTRAEQUIPCEN for over two years. A unique concept has evolved from this investigation. This concept will implement computer system requirements for trainers with a series of microcomputers organized in a multiple configuration. The system will control and provide the required computation for the total trainer with a stored program that has been functionally partitioned. The partitioned functions are dedicated to individual microcomputers.

This paper presents a summary of the conceptual analysis, the technical objectives, the feasibility and cost trade-offs. The required technologies for implementation are identified.

Finally, a status of the exploratory development to provide the critical technology and demonstrate concept feasibility is discussed briefly.

PROJECT OBJECTIVES

There were four primary objectives of this research project as follows:

a. To lower life-cycle costs of trainer computer systems and reduce proliferation of different computer types.

b. To improve computer systems processing performance without increasing costs or hardware complexity.

c. To achieve the ability to optimally tailor the computer system capability to the requirements of a specific trainer.

d. To preclude early equipment obsolescence as the commercial computer technology advances.

Conceptually, each of the above objectives will be achieved in the system architecture and application of microcomputers introduced and described in this paper. Ultimately, life-cycle costs of trainer software can be significantly reduced via modular standardization of generic (common) functions that will be programmed in the new DoD standard HOL, ADA.

TYPICAL TRAINING SIMULATOR

In order to place this microcomputer approach in the proper perspective, a typical training simulator will be discussed briefly. Figure 1 provides the generic or stereotyped system configuration of practically all modern real-time trainers. In such trainers a general purpose (GP) digital computer and stored program are the key elements. The instructor's station provides the instructor with the capability to interactively control, manage, and monitor the training task, training scenario and the performance of the trainee through the medium of the GP computer system. The GP computer system interfaces with the external trainer equipment via linkage equipment that provides signal conversion, translation, input-output, etc., in conjunction with the trainee

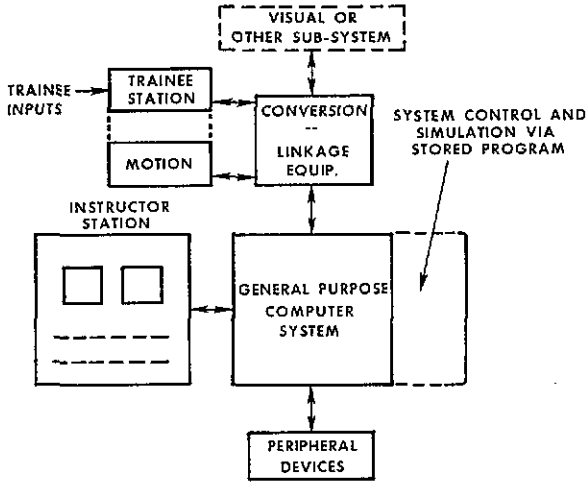

Figure 1. Typical Trainer System Concept

station and other subsystems. For a given trainer, there may be a motion system to provide the physical motion environment of the simulated weapon platform (e.g., an aircraft cockpit). The linkage and conversion subsystem also interfaces with any visual or other subsystem required for a particular trainer.

Computer peripheral devices such as CRT control terminals, magnetic tape devices, disc units, printers and the like, provide the capability to input the simulation programs, to input initialization data, and data bases, to print-out performance parameters, and to log specific trainer events. The disc units provide mass storage for all types of programs as well as storage of real-time data and parameters for playback to the system.

The program stored in the GP computer provides the system capability to process and compute the math model that represents the vehicle or system being simulated for training and to control the total training exercise in real-time.

It is appropriate to comment on some of the characteristics and deficiencies of the generic trainer computer system. The complete real-time program is usually stored in contiguous memory space of the GP computer. For trainer systems requiring more than one computer, the program is divided according to the operations and functions assigned to the several computers. In this type of configuration, one computer is designated as the master computer and the other computers are designated as slave computers. They normally communicate or pass parameters via a common memory. In all cases, the program is executed sequentially and concurrently and thus usually requires very high speed GP computers.

The computer hardware for trainer systems is selected by the trainer system contractors based on NAVTRAEQUIPCEN procurement policies and the technical requirements of the NAVTRAEQUIPCEN specifications. Cost-effective and performance considerations for selection restrict the computer equipment to a very limited number of computer manufacturers. Current state-of-the-art in computer main-frame hardware designs are not sufficiently modular for flexible low-cost system synthesis and/or low-cost system expansion for a wide spectrum of trainer classes.

There is no standard assembly-level language among computer manufacturers. Even the available high order language (HOL) FORTRAN is not optimum for programming high fidelity real-time trainers. There are no standard software modules regardless of mathematical or other model similarities among trainer classes. Thus the NAVTRAEQUIPCEN must pay for "re-inventing the software wheel" with each new procurement.

Limitations on processing performance of available computer hardware that is cost effective for trainers, and significant acquisition and life-cycle support costs of software, were driving functions in a search for concepts to apply microcomputer technology to trainers. The concept as developed is primarily applicable to replacing the GP computer function as described above and as depicted in Figure 1.

The initial part of the analysis was directed toward deriving GP computer performance requirements for a modern aircraft operational flight trainer as an example. Average performance figures are shown in Table 1.

The GP computer processing requirements as indicated cannot be accommodated by a single cost effective computer. The computer system was organized in a master-slave configuration, requiring a minimum of three GP computers.

PROGRAMMING AND EXECUTION REQUIREMENTS - TYPICAL OFT

Because of the sampled data and discrete nature of the variables as processed by the GP computers in a modern trainer, the solution of the vehicle simulation equations of the typical OFT is iterative. Of significant importance in determining the rate at which the variables must be sampled and processed is the highest natural frequency of the system or vehicle being simulated. Studies have determined that adequate simulation fidelity requires the solution rate must be at least 20 times the highest natural frequency. For modern high

TABLE 1. TYPICAL MODERN OFT PROGRAM PROCESSING REQUIREMENTS

SIMULATION FUNCTION	SIMULATION PROGRAM REQUIREMENTS* (INSTRUCTIONS PER SECOND)
Flight Control, Equations of Motion, Aero, Cockpit Inst.	917,063
Navigation/Communication	70,623
Accessories	12,066
Instructor's Station and Displays	42,188
Executive	111,375
Real-Time Operating System	27,338
Total for Typical Modern OFT	1,180,653 IPS
Average Instruction Execution Time Required by a Computer System	0.847 μsec/inst

* Includes 15 percent for FORTRAN inefficiency, 25 percent spare and 10 percent estimating contingency.

performance aircraft the highest natural frequency can be of the order of 1.0 to 1.5 HZ or even greater. Therefore the solution rate (iteration rate) of the simulation equations representing such a system must be executed at a rate of 20 to 30 HZ. If a visual subsystem employing displays with the standard TV raster frame rate is a part of the trainer, then the solution rate should be at least 30 HZ or an integer multiple of 30 so as to reduce system time lags and frame synchronization problems.

The high solution rates of real-time trainers impose stringent requirements on the capabilities of the computer hardware. The average processing requirements in instructions per second (IPS) for the various software groups of a typical modern OFT are shown in Table 1.

A standard set of generic computer instructions with percent usage (Table 2) was selected and families of microcomputer components were analyzed for equivalent performance in executing the selected set. The results of that analysis are tabulated in Table 3. As will be noted, the processing performance capabilities of the selected modular families show a wide range of average instruction execution times (AIET). Execution time capability was only one of the analysis criteria applied to modular families of microcomputers.

TABLE 2. SELECTED INSTRUCTION MIX (WITH PERCENT USAGE)

INSTRUCTION TYPE	PERCENT USAGE
Load	0.158
Store	0.128
Add/Subt	0.090
Multiply	0.047
Divide	0.008
Logical	0.076
Shift (5 places)	0.031
Compare	0.043
Branch	0.105
Index	0.003
Reg-to-Reg Opns	0.031
Miscellaneous (e.g., calls to O/S, etc.)	0.279
Input-Output	0.001
Total	1.000

TABLE 3. AVERAGE FIXED POINT INSTRUCTION EXECUTION TIMES (AIET)

MICROCOMPUTER FAMILY	AIET μSEC/INST
TI-SPP-9900	14.294
INTEL 3000 BIPOLAR SET	1.926
FAIRCHILD 9440	5.711
INTEL 8080A	52.782
INTEL 8085A-2	21.013
AM2900 FAMILY	1.401
MOTOROLA-MC-10800	1.209
TI-SN74S481, 482	1.531

MICROCOMPUTER SYSTEM ARCHITECTURAL CONCEPT

A new and unique multiple microcomputer system architectural concept for real-time trainers is graphically illustrated in Figure 2. The system concept consists of N microcomputers that are functionally dedicated to assigned portions of a total trainer program. They function as applications microcomputers. A control microcomputer exercises control over the complement of application and input-output microcomputers.

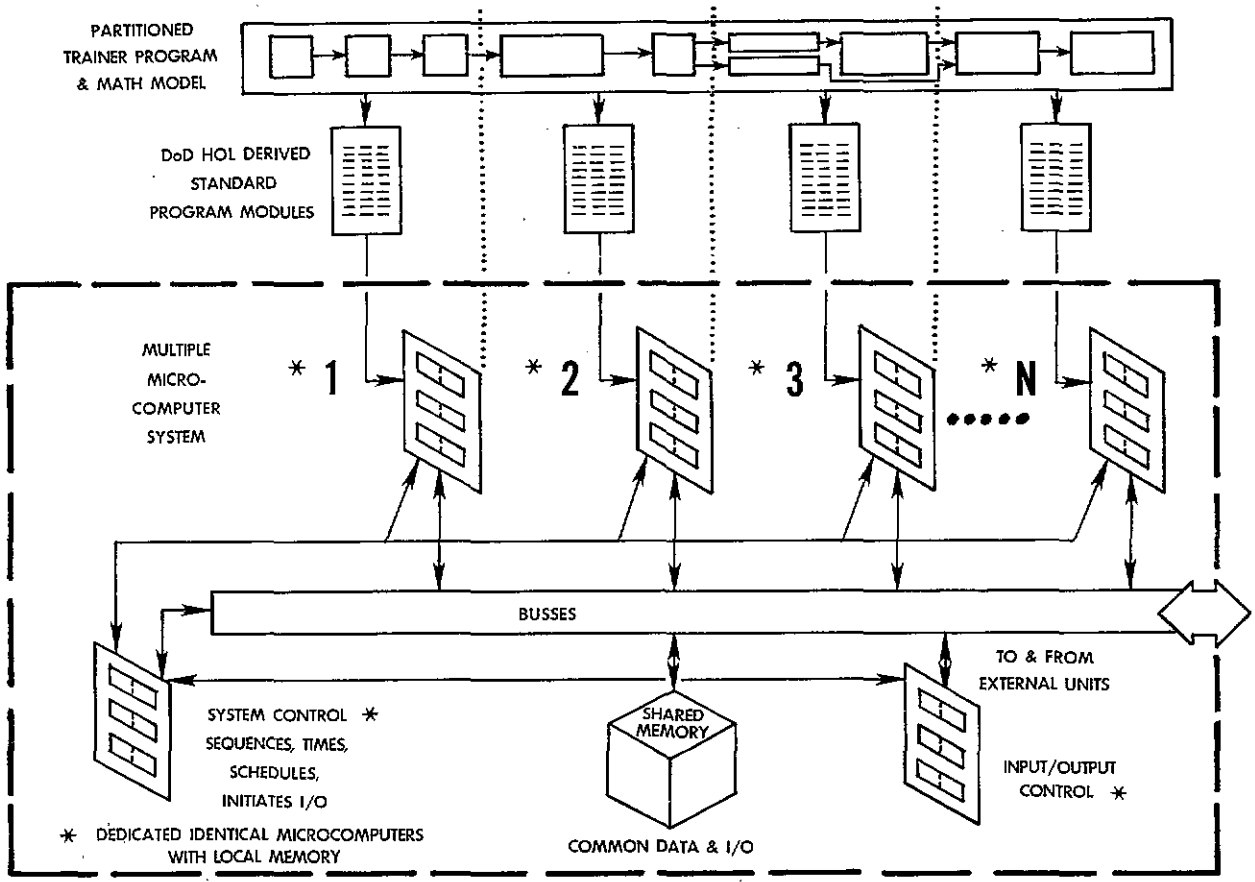

Figure 2. Functionally-Modular Multiple Microcomputer Concept For Trainer Systems

TABLE 4. TYPICAL OFT PROCESSING REQUIREMENTS PER APPLICATIONS MICROCOMPUTER

FUNCTION	INSTRUCTIONS PER SECOND	ASSIGNED MICRO-COMPUTER
Instructor Station, Misc.	169,000	1
Coefficients and Forces	308,000	2
Flight and Propulsion	265,000.	3
Axis Transformations and Integrations	181,000	4
Angular Position, Motion, TACAN	288,000	5
Instruments, Accessories, NAV/ COMM	241,000	6
Total Applications IPS	1,452,000	

TABLE 5. SUMMARY - MEMORY AND AIET REQUIREMENTS

APPLICATION MICRO-COMPUTER	ESTIMATED PROGRAM SIZE (WORDS)	ESTIMATED TOTAL MEMORY (WORDS)	REQUIRED AIET (μSEC)
1	6784	7115	4.7358
2	4725	5325	2.5991
3	4556	5206	3.0157
4	2717	3129	4.4215
5	3459	3747	2.7821
6	7298	8423	3.2949
7	620*	1500	*
8	100*	750	*
9	100*	750	*
10	1000*	2000	*
COMMON MEMORY		16,000	

* Worst-case instruction execution should result in AIET's that are well within the actual AIET of the microcomputer technology analyzed.

The control microcomputer schedules and activates the application and I/O microcomputers that are to be active during each processing frame. This is illustrated in Table 6. The applications microcomputers that are active during each frame determine a system processing control state. Intermediate results, global variables, and other common data are exchanged between microcomputers via a common memory with a distributed cache address space. The distributed cache concept allows each microcomputer to write to all other microcomputers and to common memory,but each microcomputer can only read from the cache address space assigned to it. The control computer recognizes system interrupts that are generated by the elapsed time of a real-time clock or precision interval timer as determined by the highest solution rate (frame time). It also communicates with all applications and I/O microcomputers to initially load assigned program modules to each microcomputer. Ultimately, standardized functional code will be stored in a ROM in each microcomputer.

The typical OFT program mentioned previously was partitioned in a heuristic manner and assigned to individual applications microcomputers. As functionally partitioned, the simulation problem requires at least six applications microcomputers. The processing requirements (in IPS) of each functional group were analyzed and the results are tabulated in Table 4 with assignment to each microcomputer. Note that the total solution rate has been increased from 1,180,653 IPS (Table 1) to 1,452,000 (Table 4) for this specific example of microcomputer system implementation. This is the result of assigning all routines (of a functional group) dedicated to a specific microcomputer to be executed at the highest rate required within that functional group, rather than incorporating lower rates considered in the original problem analysis. This procedure requires a higher total computation rate but reduces the software (and ATM firmware) complexity and scheduling time overhead.

TABLE 6. MICROCOMPUTER SYSTEM FRAME/ CONTROL STATE SCHEDULING (EXAMPLE)

SYSTEM CONTROL STATE FRAME NO.	SCHEDULED MICROCOMPUTERS
1	1, 2, 3, 4, 5, 8, 9
2	2, 3, 4, 5, 6, 8, 9, 10
3	1, 2, 3, 4, 5, 8, 9
4	2, 3, 4, 5, 6, 8, 9, 10
5	1, 2, 3, 4, 5, 8, 9
6	2, 3, 4, 5, 6, 8, 9, 10
7	1, 2, 3, 4, 5, 8, 9
8	2, 3, 4, 5, 6, 8, 9, 10
'	'
ETC	ETC
'	'
60	2, 3, 4, 5, 6, 8, 9, 10

Table 5 is a summary of the estimated program size, total estimated memory requirements,and the AIET required by each microcomputer for the assigned functional groups of Figure 3. The microcomputers (#8, 9, 10) assigned the functions of input-output and instructor station control are shown along with estimated program sizes and total memory requirements. Specific AIET's were not derived,but by the nature of the assigned functions the AIET of 2.5991 μsec/instruction indicated for microcomputer #2 will be more than sufficient to handle the processing requirements of these functions.

The system control microcomputer (#7) was likewise not analyzed to the same degree of detail as the applications microcomputers. However, the system state control functions will not require a processing capability that should exceed that required by microcomputer #2 (the worst-case application AIET).

The total control of the system is implemented as a hardware-firmware-software algorithm. Within the context of this concept, algorithm is used to describe a set of hardware-firmware-software implemented procedures to obtain a given result. The criteria being used for the partitioning of the control algorithm into hardware, firmware, and software are processing performance, ease and economy of implementation and the ability to enforce necessary rules for concurrent programs. A general rule will be to implement the fixed portion of the system in hardware (microcomputer modular family) and to use software for the variable or application-dependent portions. Firmware will be used in place of hardware for those functions that will be common throughout the total system. Specifically, an applications task manager (ATM) concept has been developed and will be implemented in firmware as part of the total control algorithm for the ultimate system architecture.

All communications within the system (via a multiple bus structure) will be

MICROCOMPUTER NO.	SOLUTION RATE	ASSIGNED FUNCTIONS
1	30/sec	Instructor Station, DISPLAYS, Ground Reaction effects, taxiing
2	60/sec	Aero, Forces, Moments-Stab, Axis Comp. Atmosphere, Weight & Balance
3	60/sec	Flight Control, Propulsion system, Flight Control-CPU interface
4	60/sec	Stability Axis transformations, Earth-Axis acceleration Moments - Body Axis computation $\dot{P},\dot{Q},\dot{R}$ Integration, Air Data Comp.
5	60/sec	Angular Positions (Quaternions), G-seat/G-suit, TACAN Math Model
6	30/sec	Navigation, Radar Nav, Compass simulation
7	60/sec	System Control State
8	60/sec	Input-output - Cockpit
9	60/sec	Peripheral units control
10	30/sec	Instructor Station I/O and Controls

Figure 3. Microcomputer Designations (Example)

handled by a virtual machine implemented in the microcomputer firmware (ATM). The characteristics of the virtual machine concept will be independent of the hardware implementation. This will allow for substantial modifications or acceptance of new microcomputer technology without requiring changes in the fundamental control algorithm.

The ATM will function as a simple, highly efficient resource manager and real-time operating system for scheduling and controlling execution of the assigned tasks in each microcomputer. This also includes the control microcomputer whose assigned task is primarily that of master interrupt accounting and scheduling all other microcomputers for each frame or system control state (ref. Table 6). The firmware-implemented ATM is identical in each and all microcomputers.

MICROCOMPUTER MODULAR FAMILIES INVESTIGATED	ANALYSIS FINDINGS
INTEL 3000 BIPOLAR SERIES**	FAST, MICROPROGRAMMABLE, TOO INFLEXIBLE IN MICROCODE ADDRESSING, LIMITED MODULAR FAMILY
TI SPP-9900 SERIES	TOO SLOW, TOO INFLEXIBLE, NOT MICROPROGRAMMABLE, AVAILABLE ONLY FROM A SINGLE SOURCE, NOT MODULAR.
FAIRCHILD 9400 BIPOLAR SERIES	TOO SLOW, TOO INFLEXIBLE, NOT MICROPROGRAMMABLE, AVAILABLE ONLY FROM A SINGLE SOURCE, NOT MODULAR
AM 2900 BIPOLAR SERIES**	MICROPROGRAMMABLE, FAST, AVAILABLE FROM AT LEAST 5 COMMERCIAL SOURCES, HIGHLY MODULAR
TI SN74S481 BIPOLAR SERIES**	FAST, MICROPROGRAMMABLE, NOT REGISTER ORIENTED, AVAILABLE ONLY FROM A SINGLE SOURCE, LIMITED MODULAR FAMILY
INTEL 8080A MOS MICROPROCESSOR	TOO SLOW, NOT MICROPROGRAMMABLE, TOO INFLEXIBLE
INTEL 8085 A-2 MOS MICROPROCESSOR	TOO SLOW, NOT MICROPROGRAMMABLE, TOO INFLEXIBLE
MOTOROLA MC 10800 MECL SERIES**	FAST, MICROPROGRAMMABLE, AVAILABLE ONLY FROM A SINGLE SOURCE, LIMITED MODULAR FAMILY

** BIT SLICE MODULAR FAMILIES

Figure 4. Technology Evaluation Summary

AVAILABLE HARDWARE TECHNOLOGIES

All major hardware technologies to implement this microcomputer architectural concept are currently available. The computational complexity of real-time simulation dictates use of bit-slice microcomputer technology (ref. Tables 4 and 5).

During the analysis part of this project eight microcomputer/microprocessor modular families were investigated. Performance,

modularity, microprogrammability and availability from multiple commercial sources were the primary analysis criteria. Figure 4 summarizes the results and findings of that investigation. Of the families of modules considered, only the AM-2900 series bit slice family is capable of meeting the criteria mentioned above. High performance 32 bit microcomputer architectures can be designed and implemented with the AM-2900 series modules. Other required hardware technologies are summarized in Figure 5.

RAMS, ROMS, PROMS, EPROMS, MAGNETIC CORE	AVAILABLE IN MANY CAPACITIES WITH 3-STATE INTERFACES - 50 NS TO 1000 NS ACCESS AND CYCLE TIMES
BUSSING CONCEPTS	STATIC BUSSES APPEAR TO BE MOST APPROPRIATE, REQUIRE NO ACTIVE COMPONENTS, BUS MANAGEMENT CAN BE ACHIEVED WITH EITHER SPECIAL HARDWARE OR AS A TASK ASSIGNED TO THE CONTROL MICROCOMPUTER
PACKAGING CONCEPTS	COMMERCIALLY AVAILABLE IN MANY FORMS & CONFIGURATIONS SUITABLE FOR TRAINER DESIGNS

Figure 5. Other Required Hardware Technologies - Summary

CRITICAL TECHNOLOGY

The most critical technology required to achieve the microcomputer system architecture concept of Figure 2 is the development of the system control algorithm previously mentioned. Optimal partitioning of a given simulation task is another technology area that must be addressed to some formal degree if standard software modules are to be developed in the future, but it is not considered critical.

The multiple microcomputer system control algorithm concept (employing ATM and distributed cache) has been developed in detail under contract to NAVTRAEQUIPCEN. All system control requirements have been achieved in the development.

CONCEPT FEASIBILITY

Concept feasibility addressed both technical and cost issues for full acceptance. The conventional general purpose (GP) computer approach was described earlier in this paper. A life-cycle cost model (Figure 6) was developed and reasonable GP cost parameters were identified and derived from vendor price information. Likewise, a multiple microcomputer system concept using the same functions was synthesized and microcomputer cost parameters were applied to the same model. The cost model results are summarized in Table 7.

SUMMARY AND CONCLUSIONS

A conventional three GP computer system for a typical OFT was configured and a life cycle cost figure derived from the cost model of Figure 6. The same OFT program was partitioned in a heuristic manner by grouping related processing functions for assignment to individual microcomputers. The processing requirements of each of these functional groups were derived and compared with the capability of the several microcomputer modular families previously analyzed. Microcomputers comprised of the AM-2900 series bit-slice family provided the necessary processing and control capability. The cost component of the required number of microcomputers was determined from vendor supplied application data and price information.

An examination of the life-cycle summary of Table 7 indicates the cost effectiveness of the multiple microcomputer system approach. The functionally modular microcomputer system architecture is conceptually feasible with current technology and should be cost effective in the projected implementation and for its life cycle.

TABLE 7. OFT COMPUTER SYSTEM 10 YR. LIFE CYCLE COST SUMMARY

Conventional GP Computer System Approach	$1,728,326
Multiple Microcomputer System Approach	$ 896,857

CURRENT STATUS

A Phase I exploratory development contract was competitively awarded in September 1978 for research into and development of the control algorithm for this architectural concept. Included in that contract was the initial design analysis of a fully microprogrammable 32 bit microcomputer that implemented the instruction subset used in the original performance analysis and with hardware characteristics required to implement the control algorithm. The AM-2900 series modules were the basis for the design.

The control algorithm has evolved in an implementation combination of hardware, firmware, and software. This approach embodies the ATM and distributed cache concepts

to implement the virtual machine idea. The contractor has issued a final report as of September 1979 and the contract has been completed.

A sole source Phase II exploratory development contract was issued to the Phase I contractor in June 1979 to design, develop, fabricate, test, program, and deliver a breadboard to demonstrate concept feasibility of the total multiple microcomputer system concept and system control algorithm. This breadboard with a suitable demonstration problem is scheduled for delivery to NAVTRAEQUIPCEN in September 1980 for extensive evaluation.

1. Initial acquisition cost of processor hardware - C_{01}
2. Initial acquisition cost of memory hardware - C_{02}
3. Initial acquisition cost of interface hardware - C_{03}
4. Initial acquisition cost of peripheral electronics - C_{04}
5. Initial acquisition cost of peripheral hardware- C_{05}
6. Initial acquisition cost of processor hardware documentation - C_{06}
7. Initial acquisition cost of memory hardware documentation - C_{07}
8. Initial acquisition cost of interface hardware documentation - C_{08}
9. Initial acquisition cost of peripheral electronics documentation - C_{09}
10. Initial acquisition cost of peripheral hardware documentation - C_{10}
11. Initial acquisition cost of OFT simulation software - C_{11}
12. Initial acquisition cost of utility software - C_{12}
13. Initial acquisition cost of maintenance spares for processor, memory, and interface - C_{13} hardware.
14. Initial acquisition cost of maintenance spares for peripheral electronics and - C_{14} peripheral hardware.
15. Initial cost of maintenance training - C_{15}
16. Initial cost of SSA programmer orientation - C_{16}
17. Initial cost of test equipment - C_{17}
18. Life cycle maintenance cost (0-5 yrs.) hardware (processor, memory, interface) - C_{18}
19. Life cycle maintenance costs (5-10 yrs.) - hardware (processor, memory, interface) - C_{19}
20. Life cycle maintenance costs (0-5 yrs.) - hardware personnel - C_{20}
21. Life cycle maintenance costs (5-10 yrs.) - hardware personnel - C_{21}
22. Life cycle maintenance costs (0-5 yrs.) - software personnel - C_{22}
23. Life cycle maintenance costs (5-10 yrs.) - software personnel - C_{23}
24. Life cycle maintenance costs (0-5 yrs.) - peripheral electronics hardware - C_{24}
25. Life cycle maintenance costs (5-10 yrs.) - peripheral electronics hardware - C_{25}
26. Life cycle maintenance costs (0-5 yrs.) - peripheral hardware - C_{26}
27. Life cycle maintenance costs (5-10 yrs.) - peripheral hardware - C_{27}
28. Life cycle maintenance costs (0-5 yrs.) - peripheral electronics hardware personnel - C_{28}
29. Life cycle maintenance costs (5-10 yrs.) - peripheral electronics - C_{29}
30. Life cycle maintenance costs - SSA programmer re-orientation - C_{30}

Figure 6. Elements of Ownership Cost Model - Computer System

REFERENCES

1. Moore School Report 55-20, University of Pennsylvania, "Simulation of a Supersonic Fighter Using a Digital Computer."
2. NAVTRADEVCEN 594-1, "Analog-Digital Computers for Real-Time Simulation."
3. C. F. Summer and G. A. Wyndle, "An Analysis of Microcomputer Technology for Applications to Real-Time Trainers," Naval Training Equipment Center, Orlando, FL, NAVTRAEQUIPCEN IH-313, August 1979.
4. R. O. Pettus, et aux, "Multiple Microcomputer Control Algorithm Investigation," NAVTRAEQUIPCEN 78-C-0157-1, September 1979.

ABOUT THE AUTHOR

MR. CHARLES F. SUMMER has been a Project Engineer and Principal Investigator in the Computer Laboratory of the Research and Technology Department of the Naval Training Equipment Center since 1970. At the Center, Mr. Summer has been involved in applied research in computer technology and computer systems architecture for trainers and simulators. He is also a consultant in computer technology and computer architecture to various Center departments and other Navy and Army agencies. He has more than 24 years experience in the design of digital computers and their application in complex real-time systems. Prior to joining the Naval Training Equipment Center, Mr. Summer was associated with various firms such as RCA, Harris Corporation and Martin Marietta Corporation. He is a senior member of the Institute of Electrical and Electronic Engineers and the IEEE Computer Society, a member of Tau Beta Pi and a registered professional engineer in the State of Florida. He holds a B.S. degree in electrical engineering from the University of South Carolina.