

AD-A115 006

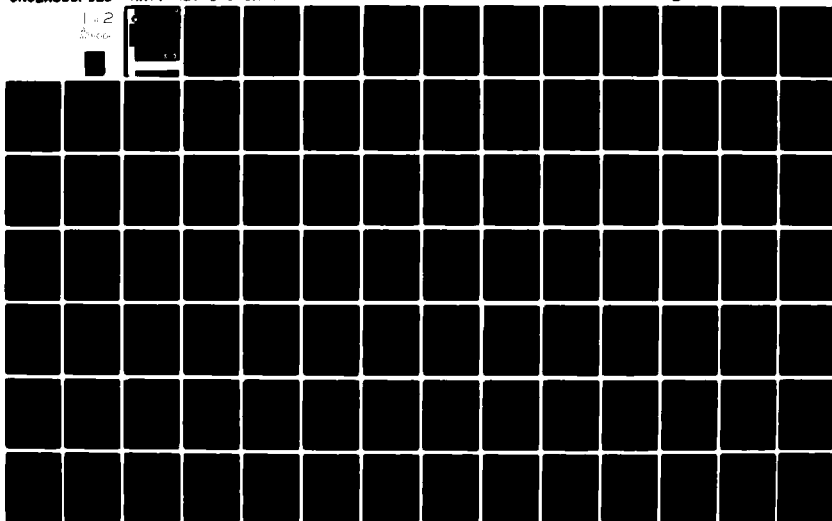
NAVAL TRAINING EQUIPMENT CENTER ORLANDO FL
COMPUTER ARCHITECTURE STUDY FOR VTXTS SIMULATORS.(U)
APR 82 L D HEALY, G A WYNDELE, B BAKER
NAVTRAEQUIPC-IN-336

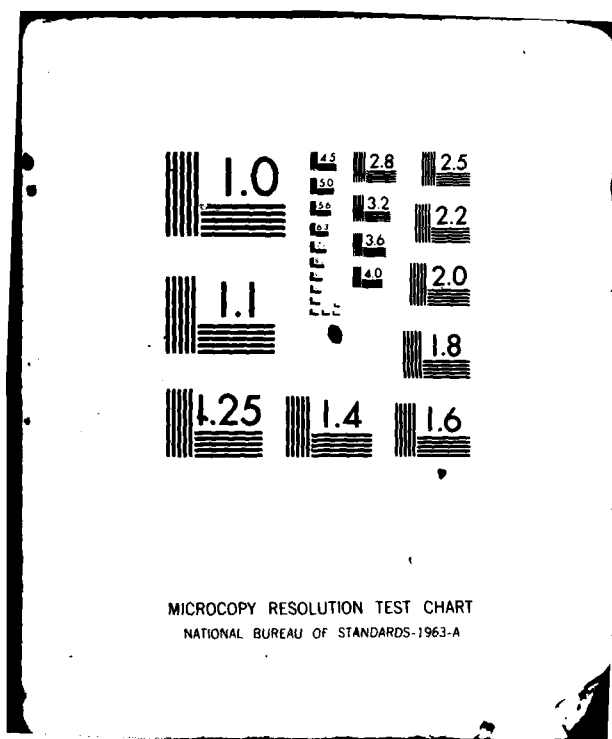
F/G 9/2

UNCLASSIFIED

NL

1-2
200000







Technical Report: NAVTRAEQUIPCEN IH-336

Computer Architecture Study for VTXTS Simulators

Leonard D. Healy
Gerald A. Wyndle
Computer Systems Laboratory
Research Department
Naval Training Equipment Center
Orlando, FL 32813

Bruce Baker
Servos & Simulation, Inc.

April 1982

Final Report March 1981 - November 1981

DOD Distribution Statement

Approved for public release;
distribution unlimited.

DTIC
ELECTE
JUN 01 1982
S D
E

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

82 06 01 098

NAV TRAEQUIPCEN IH-336

COMPUTER ARCHITECTURE STUDY FOR
VTXTS SIMULATORS

LEONARD D. HEALY
GERALD A. WYNDLE
Computer Systems Laboratory

BRUCE BAKER
Servos & Simulation, Inc.

APRIL 1982

APPROVED:

C. F. Summer

C. F. SUMMER
Acting Head, Computer Systems Laboratory

DAVID P. GLENN
Director of Research

GOVERNMENT RIGHTS IN DATA STATEMENT

Reproduction of this publication in
whole or in part is permitted for any
purpose of the United States
Government.

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FL 32813



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NAVTRAEQUIPCEN IH-336	2. GOVT ACCESSION NO. AD-A115006	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER ARCHITECTURE STUDY FOR VTXTS SIMULATORS		5. TYPE OF REPORT & PERIOD COVERED Final Report for Period March 1981 - November 1981
7. AUTHOR(s) Leonard D. Healy, Gerald A. Wyndle, NAVTRAEQUIPCEN Bruce Baker, Servos & Simulation, Inc.		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Systems Laboratory, Research Department Naval Training Equipment Center Orlando, FL 32813		8. CONTRACT OR GRANT NUMBER(s) NAVTRAEQUIPCEN WA 8129
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Training Equipment Center Orlando, FL 32813		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE APRIL 1982
		13. NUMBER OF PAGES 120
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) computers microcomputers/microprocessors minicomputers real-time simulation systems computer architecture		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report provides an evaluation of computer configurations that might be applied to VTXTS simulators. Computers ranging from the large, general purpose computer ordinarily used in data processing to the microcomputer which requires 40 separate units to implement a simulator complex are considered. Eleven computer configurations are evaluated with respect to availability, maintenance, expandability, risk and life cycle cost. The information is summarized in a table of advantages and disadvantages of the various configurations.		

DD FORM 1473

1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

NAVTRAEQUIPCEN IH-336

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
I	INTRODUCTION.	7
	VTXTS Cockpit Configuration	7
	A Preview	7
II	COMPUTER PERFORMANCE ANALYSIS	11
	Average Instruction Execution Time (AIET)	11
	Computer Capabilities	12
III	COMPUTATION REQUIREMENTS.	17
	Simulation for Training	17
	Simulator Components.	18
	Processing and Storage Requirement Analysis	19
	Description of Major Modules/Functions.	20
	Program Organization and Performance Requirement Analysis	22
	VTXTS Storage and Processing Requirements	23
IV	COMPUTER ARCHITECTURE	33
	Minicomputers	33
	Super Minicomputers	41
	Large Computers	41
	Microcomputers.	41
V	MINICOMPUTER CONFIGURATIONS	51
	Analysis of Dual CPU Architectures.	51
	Memory Sharing.	55
	Input/Output to Cockpits.	75
	Peripheral Configurations	83
	Redundant Systems	83
	Super Minicomputers	88
VI	EVALUATION OF ARCHITECTURES	91
	Availability.	91
	Maintenance	94
	Expandability	94
	Life Cycle Cost	95
	Risk.	100
	Cost Effectiveness of Adding a Redundant Computer	102
	Relative Advantages and Disadvantages	102
	Recommended Systems	107/108

NAVTRAEQUIPCEN IH-336

TABLE OF CONTENTS (CONTINUED)

<u>Section</u>		<u>Page</u>
VII	OTHER APPROACHES.	113
	Array Processors.	113
	Applied Dynamics AD-10.	114
	Heterogeneous Element Processor	114
VIII	SUMMARY AND CONCLUSIONS	115
	Recommended Systems	115
	Areas for Further Research.	116
	REFERENCES.	119/120

NAVTRAEQUIPCEN IH-336

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	VTXTS Cockpit Arrangement	8
2	Dedicated Minicomputer Systems per Cockpit.	35
3	Shared Minicomputers for OFT Only	36
4	Shared Minicomputers for All Simulators/Cockpits.	37
5	Separate Minicomputer Systems per Cockpit with a Redundant Computer	38
6	Shared Minicomputers for OFT Only with a Redundant Computer . .	39
7	Shared Minicomputers for All Simulators/Cockpits with a Redundant Computer	40
8	Super Minicomputer Configuration for all Simulators/Cockpits. .	42
9	Super Minicomputer Configuration for all Simulators/Cockpits with a Redundant Computer	43
10	Shared Single Large Computer for All Simulators/Cockpits. . . .	44
11	Shared Single Large Computer for all Simulators/Cockpits with a Redundant Computer	45
12	Dedicated Microcomputer Systems per Simulator/Cockpit	46
13	Microcomputer System Block Diagram.	49/50
14	Dual CPU with Shared Memory Only.	52
15	Dual CPU with Shared and Dedicated Memory	54
16	Dedicated CPU with Dedicated Memory Only.	56
17	CPU with Peripheral Processor and Dedicated Memory Only	57
18	Pairs of CPU's Sharing Switched Memory Bank	58
19	Pairs of CPU's Sharing Switched Memory Bank with Redundant Memory	59
20	Pairs of CPU's Sharing Wired Memory Bank.	61/62

NAVTRAEQUIPCEN IH-336

LIST OF ILLUSTRATIONS (CONTINUED)

<u>Figure</u>		<u>Page</u>
21	Pairs of CPU's Sharing Wired Memory Bank with Redundant Memory	63/64
22	Triple CPU's Sharing Switched Memory Bank	65/66
23	Triple CPU's Sharing Switched Memory Bank with Redundant Memory	67/68
24	Triple CPU's Sharing Multiple-Level Memory, 3N CPU's.	69/70
25	Triple CPU's Sharing Multiple-Level Memory, 3N+1 CPU's.	71/72
26	Triple CPU's Sharing Multiple-Level Memory, 3N+2 CPU's.	73/74
27	Single Bus Interface for All Cockpits	76
28	Multiple Bus Interfaces, One for Each Cockpit	77
29	Multiplexer Interface with One Bus for All Cockpits	78
30	Multiplexer Interface with One Bus for Each Cockpit	79/80
31	Multiple CPU Configuration for a Multiple Cockpit Trainer	81/82
32	Minimum Peripheral Configuration.	84
33	Configuration with No Shared Peripherals.	85
34	Shared Large Disk with Minimum Peripherals.	86
35	Shared Large Disk with Same Peripherals on Each CPU	87
36	Super Minicomputer Configuration.	89/90
37	Recommended Super Minicomputer Configuration.	109/110
38	Alternate Recommended System Using Minicomputers.	111/112

NAVTRAEQUIPCEN IH-336

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Instruction Mix for Simulators.	13
2	Average Instruction Execution Time.	14
3	Computer Capabilities Used in Analysis.	16
4	CPT Storage Requirements.	24
5	OFT Storage Requirements.	25
6	ACMT Storage Requirements	26
7	CPT Processing Requirements	27
8	OFT Processing Requirements	28
9	ACMT Processing Requirements.	29
10	Summary of Processing and Storage Requirements.	32
11	Computation Requirements Used in Analysis	34
12	Reliability and Maintainability Values Used in Analysis	92
13	Simulator Availability.	93
14	Hardware Cost Estimates Used in Analysis.	96
15	Life Cycle Cost of Various Computer Configurations.	97
16	Software Cost Estimates Used in Analysis.	99
17	Software Cost for Various Computer Configurations	101
18	Cost of Downtime.	103
19	Saving by Use of a Redundant Computer	104
20	Tradeoff Summary Table.	105/106

SECTION I

INTRODUCTION

The need for a broad analysis of computer architecture is the result of procurement under Office of Management and Budget Circular A-109. This procurement procedure gives vendors a wider latitude in proposing innovative systems. At the same time, it places upon the government a requirement to evaluate these proposals to determine which vendor's approach is best. The purpose of this study is to provide reference material and guidelines in one of the critical areas (computer systems) to be evaluated for VTXTS.

This study has been limited to the evaluation of computer configurations that might be applied to VTXTS rather than specific makes or models of computers to be used. The treatment of computers considers the general characteristics of each generic class of computers, with no attempt made to assess the advantages and disadvantages of the manufacturers' products within each class.

VTXTS COCKPIT CONFIGURATION

The baseline cockpit configuration selected for the analysis is the representative VTXTS simulator suite described by the VTXTS technical team[1]. The VTXTS training suite consists of six Cockpit Procedures Trainers (CPT'S), fifteen Operational Flight Trainers (OFT'S) without visual systems, nine OFT'S with visual systems, and six Air Combat Maneuvering Trainers (ACMT'S). The total training suite is divided into three identical cockpit arrangements. Figure 1 shows one of these divisions with its computer complex.

This study addresses the problem of what computer configuration should be used in the computer complex. Alternative computer architectures for performing the computation and control functions required to control each of the three sets of cockpits are evaluated to determine the advantages and disadvantages of each architecture. Eleven different computer configurations are analyzed. Six of these configurations use minicomputers, two use super minicomputers, two use large computers normally applied in the data processing environment, and one uses microcomputers.

A PREVIEW

This preview serves to put the discussion of computer architecture in perspective. The first step was to determine a measure of computer performance to be used in evaluating the various classes of computers considered for use in the VTXTS implementation. Section II provides such a measure and explains why that particular measure was chosen. The classes of computers considered range from the large, general-purpose computer of the type ordinarily used in data processing to the microcomputer which requires almost forty separate units to implement the system.

[1] Naval Air Development Center Report 9065-20, Naval Air System Command, Analysis of VTXTS Technology : Stu ss.

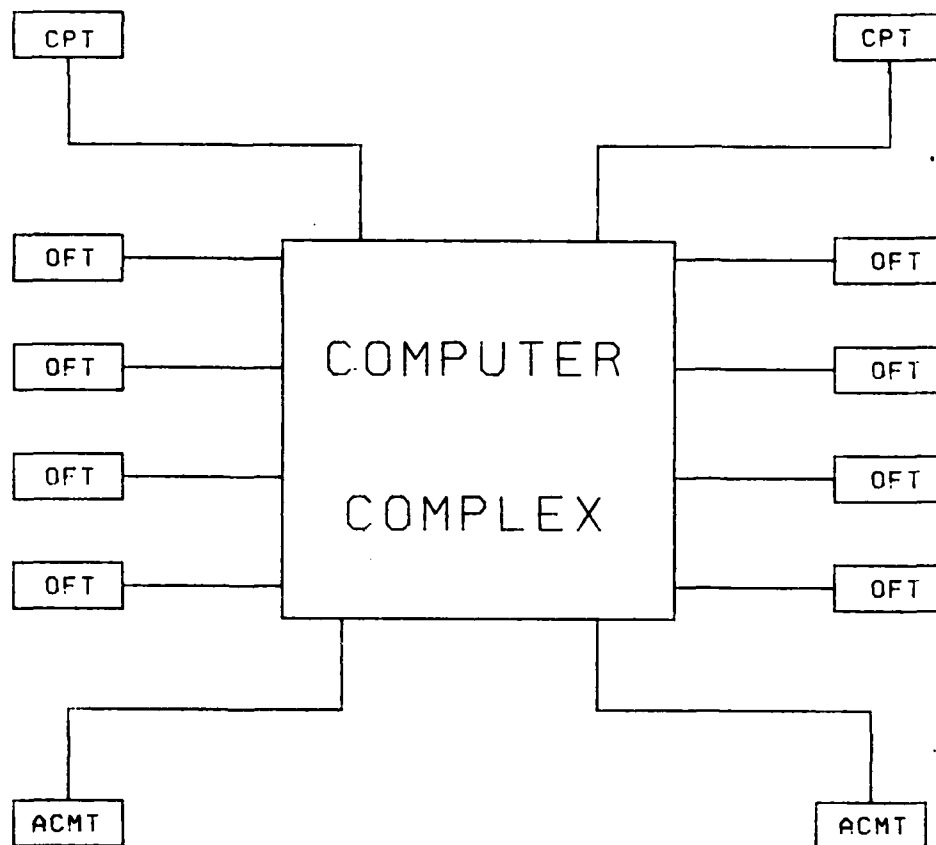


Figure 1. VTXIS Cockpit Arrangement.

Section III describes the trainers to be implemented and the projected simulation requirements for the VTXTS. It provides a discussion of the method for determining the processing and storage requirements for a training system and provides historical data on similar systems and an estimate of the VTXTS requirements.

Section IV describes the eleven computer configuration considered. Six different minicomputer systems, two super minicomputer systems, two large computer systems, and one microcomputer system are presented. Section V provides further details about minicomputer configurations. It describes various ways of interconnecting multiple computers to solve a problem that is too large for a single computer. Computer interconnection, memory sharing, connections to the cockpits, and peripheral connections are illustrated.

Section VI provides an evaluation of the architectures considered with respect to availability, maintenance, expandability, risk, and life cycle cost. It also considers the cost effectiveness of adding a redundant computer to increase system availability. The information is summarized in a table of advantages and disadvantages of the various configurations. This section concludes with the description of two configurations recommended for the VTXTS application.

Section VII provides a discussion of alternative approaches using special-purpose processing units such as array processors. Although these units are found unsuitable for the VTXTS application, a documentation of this result is included for completeness. Section VIII gives a summary of the results of this study and concludes with a description of research topics which should be pursued in support of this project.

SECTION II

COMPUTER PERFORMANCE ANALYSIS

Assessing a computer system's ability to solve a specific problem is a critical part of evaluating various computer architectures. Unfortunately, there is no standard measure of computer performance. Measures of computer speed such as MIPS (millions of instructions per second) and KOPS (thousands of operations per second) provide some indication of computer performance, but these measures are essentially unsupported by the vendors[2].

The vendors do not support measures of instruction execution speed, because the complexity of the computation problem makes these measures unreliable for most computer applications. Modern computers with cache memories, multiple input-output (I/O) channels, even multiple processing units are used to perform quasi-simultaneous processing of multiple programs through use of sophisticated operating systems. Evaluation of a computer's performance in this environment depends upon much more than the operating speed of the computer. The nature of the problem does not lend itself to a simple measure of computer performance which allows evaluation of a given computer system for a specific job.

Despite its disadvantages, instruction execution speed is a valuable tool in assessing relative computer capability for the present study. The simulator problem is more amenable to the use of such a measure than is a data processing problem, because the fixed, real-time problem being solved is likely to be computation bound rather than limited by external I/O functions. The critical I/O functions are typically handled by special-purpose hardware that operates by direct memory access and not through the computer. External hardware insures that data are available to the computer when needed.

Inaccuracies in estimating (1) the effectiveness of cache memory in reducing memory access time, (2) the instruction mix to be used in applying the measure, and (3) in the efficiency of translating from a high-level language to machine instructions remain. Fortunately, the problem addressed here is to estimate capability for a class of computers (e.g., minicomputers) to perform the computations required for the VTXTS simulation. The accuracy required for comparing one class of computer to another is not as great as that needed if the problem addressed were to select one vendor's computer instead another. In this less restrictive application, the use of instruction execution speed is appropriate.

AVERAGE INSTRUCTION EXECUTION TIME (AIET)

The problem to be solved by the computer complex will be specified in terms of computer performance and storage requirements. The computer performance requirement will be given in terms of the number of instructions to be executed during a given time cycle. The next step is to determine the capabilities of various classes of computers to solve the problem. The

[2] Edward J. Lias, "Tracking the Elusive KOPS," DATAMATION vol. 26, no. 12, Nov 1980, pp 99-105.

computer performance analysis is based upon determining an Average Instruction Execution Time (AIET).

The performance of any computer system involves several factors such as, hardware configuration, the problem to be solved, the program size and the skills, knowledge and experience of the programmer(s). In order to determine the relative performance of the classes of processors of interest a paper analysis was performed based upon the percent usage of each class of instruction (e.g., instruction mix) and the base case instruction execution time for the given computer. The execution times for the instruction set of interest were obtained for each computer system from the appropriate reference manual.

The problem in determining the relative performance of a computer system is that the instruction repertoires of general purpose digital computers selected for trainer applications have execution speeds which can vary significantly among instruction types. As an example, an ADD-type instruction of one computer may require only 1.5 microseconds for execution and a MULTIPLY-type may require 5.5 microseconds, whereas, another computer may be capable of executing ADD-type instructions in 1.2 microseconds and MULTIPLY-type in 6.75 microseconds.

In order to normalize such variations from machine to machine and to be able to compare two or more for execution capability, the actual computer hardware execution times are weighted by a percent-use factor representing the frequency each instruction is used in the program. The percent-use factors (program instruction mix expressed as a percentage) for the VTXTS OFT and ACMT analysis is shown in Table 1. The percent-use values for this table are a composite of actual instruction counts from several existing OFT's and WST's. Estimates have been used where the existing data were insufficient.

The percent-use of each instruction type is used to compute the AIET for a given computer. The AIET is calculated by multiplying the use factor for each instruction type by the execution time of the instruction for the computer being considered. The sum of the individual products gives the weighted AIET. The microsecond is the time unit used in this analysis.

The instruction usage indicates another aspect of the problem that will be treated in more detail in later sections. As indicated by the low usage of floating-point instructions, the computation performed in simulators is not primarily arithmetic. Logical decisions are a large part of the simulator problem. This factor has an impact upon the effectiveness of cache memory in the simulator application and upon the benefits that might be gained from the use of large, general-purpose computers for this problem.

COMPUTER CAPABILITIES

Table 2 shows AIET's for various classes of computer systems for the OFT and WST/ACMT instruction mixes. Results for the minicomputers (Digital PDP 11/70, Systems SEL 32/77, and Perkin-Elmer PE 3240) are taken from a report[3]. The performance of the other computers is estimated by comparing their speed to the speed of the minicomputers for a subset of the total instruction repertoire. The Amdahl computer performance is estimated from the

NAVTRAEQUIPCEN IH-336

TABLE 1. INSTRUCTION MIX FOR SIMULATORS

INSTRUCTION TYPE	USAGE	
	OFT	WST
Load	.157	.224
Load - Double Precision	.001	--
Store	.127	.133
Store - Double Precision	.001	--
Add/Subtract	.047	.096
Add/Subtract - Floating Point	.041	--
Multiply	.032	.068
Multiply - Floating Point	.015	--
Divide	.007	.007
Divide - Floating Point	.001	--
Logical	.076	.016
Shift (5 places)	.031	.091
Compare	.043	.046
Branch	.105	.147
Index	.003	.159
Register-to-Register Operations	.031	--
Input/Output Operations	.001	.013
Other	.279	--

NAVTRAEQUIPCEN IH-336

TABLE 2. AVERAGE INSTRUCTION EXECUTION TIME

COMPUTER	AVERAGE INSTRUCTION EXECUTION TIME (MICROSECONDS)	
	OFT MIX	WST MIX
Digital PDP 11/70	.779	.603
Systems SEL 32/77	1.111	1.166
Systems SEL 32/87	.191	.200
Perkin-Elmer PE 3240	.857	1.314
Amdahl 470V/7	.056	.054
Motorola MC68000	3.611	5.099
Zilog Z8000	7.476	10.458
Intel 8086	11.046	14.969

NAVTRAEQUIPCEN IH-336

ratio of ADD and MULTIPLY instruction execution times on the Amdahl 470V/7[4] compared to those of the minicomputers. The performance of the microcomputers (Motorola MC68000, Zilog Z8000, and Intel 8086) is estimated by comparing the instruction execution speeds given in a comparison of the microprocessors[5] to the minicomputer instruction execution speeds. Capability of the newest entry (System 32/87) is estimated from the ratio of performance on the British Whetstone benchmark program for the Model 32/87 and the Model 32/77 given in the manufacturer's data sheet.

For the purpose of this study, the computers considered have been divided into four classes (Large Computer, Super Minicomputer, Minicomputer, and Microcomputer), and an appropriate execution speed has been assigned to each class. The computers are divided into classes, because the purpose of this study is to determine computer configurations in terms of these classes and not to distinguish between the products of various manufacturers within a class. The AIET's shown in Table 2 were converted into a nominal instruction execution speed for each class. Table 3 shows this result.

The distinction between the Large Computer and the Super Minicomputer requires some explanation. The examples presented have a difference in computation speed, but this is not the criterion that distinguishes the two. The super minicomputer is an improved 32-bit minicomputer with an architecture which results in 3 MIP's or greater speed in executing instructions. The large computer is one designed for use in multi-user environments, both data processing and scientific computation. It has a high-speed arithmetic capability, but it also has features such as multiple I/O channels and sophisticated operating systems designed for the multiprogramming environment. Normally, the operating system for the large computer is not designed for time critical tasks in the frame considered in real-time flight simulators.

[3] Summer, C. F., Jr., G. A. Wyndle and David E. Daniel, Computer System Requirements Analysis, Device 2F112, F-14 Weapon System Trainer. Orlando, FL 1977. Report No. NAVTRAEQUIPCEN IH-262.

[4] GML Corporation, COMPUTER REVIEW. vol 21, no 1, p 5.

[5] Hoo-min D. Toong and Amar Gupta, "An Architectural Comparison of Contemporary 16-Bit Microprocessors." IEEE MICRO. vol 1, no 2, pp 26-37.

NAVTRAEQUIPCEN IH-336

TABLE 3. COMPUTER CAPABILITIES USED IN ANALYSIS

COMPUTER TYPE	CAPABILITY (1000 Instructions/Second)
Large Computer	16,000
Minicomputer	900
Super minicomputer	5,000
Microcomputer	200

SECTION III

COMPUTATION REQUIREMENTS

This section provides a description of a typical Cockpit Procedures Trainer (CPT), Operational Flight Trainer (OFT), and an Air Combat Maneuvering Trainer (ACMT), a discussion of the projected simulation requirements for the VTXTS aircraft, and the methodology employed in determining the processing and storage requirements for such a training system. The results of the analysis determine the computer performance and the storage requirement for each type of trainer in the VTXTS simulator complex.

SIMULATION FOR TRAINING

Simulation of a system for training provides for duplication and activation of the controls, performance, position and control instruments, communication and navigation system and other flight equipment of the cockpit controls and procedures in order to synthesize flight maneuvers and respond to induced emergencies. The trainee activation of controls and a presentation of instruments/displays duplicates the response of the aircraft throughout its entire operating range. The degree to which the various aircraft systems are simulated, if at all, varies for each type/class of simulator. The classes of trainers considered in this study are described below.

COCKPIT PROCEDURES TRAINER (CPT). A CPT simulates the system characteristics during cockpit preflight, engine start operation, communication, navigation, malfunctions and emergencies, and engine shutdown procedures. A typical CPT consists of a simplified replica of the specific aircraft cockpit mounted on a fixed base. The CPT also includes an instructor/operator console, a digital computer system with peripheral equipment and a power distribution station all housed in a permanent structure.

OPERATIONAL FLIGHT TRAINER (OFT). An OFT simulates the performance and characteristics during cockpit preflight, engine start operation through taxi and takeoff, navigation, flight, landing and engine shutdown procedures of a specific aircraft. A typical OFT consists of a full-size replica of the specific aircraft cockpit mounted on a fixed or motion base. The OFT also includes an instructor station/operator console, a digital computer system with peripheral equipment, sometimes a visual system, and a power distribution station, all housed in a permanent structure.

AIR COMBAT MANEUVERING TRAINER (ACMT). The ACMT simulates the performance of a fighter/attack aircraft engaged in one-on-one or in more recent ACMT'S two-on-one aerial combat by real-time simulation of maneuvers and relative positions of the engaging aircraft. In addition to simulation of an aircraft that is controlled by the trainee, the ACMT provides an unmanned (synthesized) aircraft or an aircraft operated by the instructor to participate in the combat situation. Performance simulation is provided for selected types of aircraft and air-to-air weapons. The selected aircraft and weapons are kinematically and computationally related to each other so that accurate visual simulation of the targets and weapon deployment is provided.

NAVTRAEQUIPCEN IH-336

The ACMT for the VTXTS does not include all of the features described above. Since the trainer aircraft has only a limited weapons system capability, the simulation is simpler than that for the usual ACMT. In addition, no takeoff, landing or navigational capability is designed into such a trainer.

SIMULATOR COMPONENTS

Each class of aircraft trainer consists of a trainee station (replica of the aircraft cockpit), instructor station and computer system. Most OFTs and all ACMTs utilize a visual system and both use a motion system. A brief description of the above major systems is provided below.

TRAINEE STATION (AIRCRAFT COCKPIT). The aircraft cockpit contains all of the cockpit equipment that is normally visible to the trainee and visually represents the particular aircraft being simulated with respect to floors, bulkheads, panels, shelves, consoles, and other structural items. The aircraft cockpit station replicates the interior of the design aircraft (aircraft to be simulated) and includes interior lighting which functions in response to appropriate cockpit lighting controls, an intercommunication system, and may include the following as required (1) modified production ejection seat with buffet and g loading simulation; (2) simulated g-suit system; and (3) a simulated oxygen system. Canopy, mirrors and supports are provided as appropriate.

INSTRUCTOR STATION. The instructor station for these types of simulators is usually located external to the trainee station. It provides trainers control including implementing the flight, inserting malfunctions, monitoring trainee action and evaluating trainee performance. The instructor's station is utilized to provide information to the trainee concerning the appropriateness of his action as well as the actions he should have taken at any point during the problem. The instructor station also permits the instructor to maneuver the simulator through a planned flight for demonstration purposes. The instructor's station provides a continuous display of real-time information such as airspeed, altitude, heading, ground position, and the like to the instructor. For simulated aerial navigation and control, data are also available in order that he can control the simulation problem within the flight capabilities of the design basis aircraft. There are also displays which provide information to the instructor pertaining to the operation of the simulator. The displays consists of the following types (1) cockpit instruments, (2) switch position indicators, (3) instruments repeaters, (4) CRT's, and the like.

COMPUTER SYSTEM. The embedded computer system for a represented CPT is usually a single computer/processor configuration with private memory, peripheral equipment, mass storage memory and cockpit input-output conversion equipment (linkage). However, the embedded computer sytem of a modern OFT or ACMT is usually a multiprocessor configuration with private memory, shared or common memory, peripheral equipment, mass storage memory, and cockpit

NAVTRAEQUIPCEN IH-336

input/output data conversion equipment. This system is typical of the conventional master-slave computer system configuration.[6,7]

In a multiprocessor configuration, one processor is designed as the master unit which controls the training system. The master processor usually contains the software for instructor station functions, data recording/playback, simulator modes of operation. The slave processor contains the software for flight, engines, accessories, communications/navigation, weapons and all the other simulation functions.

MOTION SYSTEM. The motion system is designed to provide the trainee pilot with kinesthetic cues of the simulated accelerations, angular rates and other disturbances. The motion system is usually comprised of a g-suit, a g-seat, a buffet system and/or motion base platform. The g-suit provides the trainee with an indication of the normal g-loading on the aircraft. The g-seat gives the trainee a sensory indication of the magnitude and direction of the aircraft's total acceleration vector. These two systems are sometimes used in conjunction with a motion base platform to provide the required motion cues. The buffet system provides the trainee pilot with the effects of aerodynamic buffet, engine rumble, control surface deflection, stores release, weapon strike and aircraft crash by pulsating the pilot's seat with a signal of variable frequency and amplitude.

VISUAL SYSTEM. The visual system provides the out-of-the-cockpit view for the required training situation such as taxiing, field takeoff and landing, carrier takeoff and landing, day, night and dust scenes, air combat and weapon delivery. The three types of visual systems which are used in aircraft trainers to provide the required visual presentation and proper visual cues are: (1) point light source; (2) model/model board; and (3) computer generated imagery.

PROCESSING AND STORAGE REQUIREMENT ANALYSIS

The procedure used in determining the storage (memory) and computational requirements for the real-time simulation problem for the VTXTS trainer aircraft was that described in a NAVTRAEQUIPCEN report[3]. The major computer analysis steps are:

- a. Determine all simulation and control functions and assign each to a program module.
- b. Determine or estimate the number of program statements, data and constant storage requirement for each program module.

[6] B. A. Zempolich, Future Avionics Systems Architecture. Naval Air System Command, Report No. AIR-360, Oct 1977

[7] W. J. Dejka, Microcomputers and System Design. Naval Electronics Laboratory Center, Report TD-507, Feb 1977

[3] Summer et al, ob. cit.

NAVTRAEQUIPCEN IH-336

- c. Determine or estimate the worst-case instruction count for each program module.
- d. Determine the computer solution iteration rate for each program module.
- e. Determine the processing requirements for the total trainer simulation program.
- f. Derive the percent usage factor for a specified class of typical instructions.
- g. Derive an average instruction execution time for the specific computer hardware being considered.
- h. Determine computer hardware configurations which will satisfy the performance requirements as determined by e. and g. above.
- i. Determine the computational load per processor (average number of instructions to be executed per second) which is the sum of the products of the worst-case instruction count/module and the corresponding iteration rate (resulting in a total instructions per second (IPS) figure for each processor).

DESCRIPTION OF THE MAJOR PROGRAM MODULES/FUNCTIONS

The major program modules/functions for the VTXTS trainer suite are identified and described below.

FLIGHT. The flight modules perform the real time simulation of the aircraft aerodynamics. Using such information for inputs as engine thrust, control surface deflections, stores loading, and fuel loading, this module solves the six degree of freedom equations of motion for the aircraft. The primary outputs include the aircraft attitude (Euler) angles, attitude rates, and aircraft velocities and moments in inertia space as well as in the local air mass. In addition the flight module performs simulation of both the primary and secondary flight control systems for the aerodynamic response of the control stick, rudder, trim switch, flaps and speed brakes.

ENGINE. The engine module simulates the engine dynamics along with thrust forces, fuel consumption, and engine startup and shutdown operations. Single engine and multiple engine failure can be provided by instructor inserted malfunctions.

MOTION SYSTEM. The motion system module provides the conversions of the appropriate body axis translational and rotational accelerations and angular rates from the flight module. The motion system servo command data are used to generate motion cues in consonance with pilot and flight control system inputs and resultant flight dynamic effects.

G-SEAT/G-SUIT. This module uses the present center of gravity location and the location of the pilot's seat in aircraft body coordinates to compute the

translational accelerations at the pilot's station using the appropriate body axis translational and rotational accelerations and angular rates. The G-seat drive equations are designed to establish a linear relationship between air pressure delivered to the seat air cells and the accelerations acting on the aircraft. The lap drive equations provide signals for simultaneous belt movement which enhances negative acceleration cues. The G-suit drive equations are structured to establish a linear relationship between normal acceleration and commanded suit pressure.

COMMUNICATION/NAVIGATION. The communication/navigation module performs the simulation of the VHF, TACAN, UHF and IFF systems which include the channel/frequency selection, in-tune, and in-range functions. Also, navigational functions of the navigation computer and inertia navigation system (INS) are provided to the flight control system, propulsion system and other aircraft systems.

VISUAL. The visual system module computes the geometric relationship between the out of the window scene and the aircraft's position; provides command data for the target, missile, and background projectors and the visual system; and provides instructor required data for the CRT displays.

ACCESSORY. This module consists of the following functions: (1) hydraulic system which simulates the hydraulic pressure indication for the flight controls and the associated caution displays and indicators. Dynamic simulation of the system pressure and demand capacity relationship can be included as well as instructor inserted malfunctions; (2) fuel system which simulates the fuel quantity indications, fuel available logic, normal fuel transfer, weight, and center of gravity for all fuel tanks for both normal and emergency operations; (3) electrical system which performs simulation of aircraft ac and dc electrical systems for both left and right engine. This simulation includes the cockpit indications and the control and logic for aircraft power distribution to other aircraft systems. Both normal and emergency indications should be simulated; (4) landing system which performs simulation of the cockpit indications for the landing gear, nosewheel steering, anti-skid, parking brake, launch bar and arresting hook system. Simulated malfunctions should be included; (5) instruments which perform the dynamic simulation of the various cockpit instruments such as the attitude indicator, airspeed, altimeter, rate-of-climb, magnetic compass, angle of attack, and radar altimeter; (6) fire detection/extinguish system which simulates the cockpit indications for warning lights and switches; (7) egress system which simulates cockpit indications for the canopy system and ejection seat, and freezes the training mode cycle on detection of an ejection; (8) environmental control system which simulates cockpit controls and indicators which control and relate the status of the cockpit air conditioning, temperature, and oxygen supply; (9) controls/indicators interface which provides the capability of transferring the analog/digital information to/from the trainee cockpit controls and indicators; and (10) aural tones/cues simulation which provides the audio system command data which are derived from the trainee control inputs and aerodynamics.

WEAPON SYSTEM MODULE. This module is comprised of the following functions: (1) weapon flight dynamics which simulate the aerodynamic characteristics, flight trajectory and scoring(hit/miss) parameters for several types of

NAVTRAEQUIPCEN IH-336

armament; (2) radar detection/tracking which performs the radar threshold calculations used to establish air target detections and maintain target tracking; (3) antenna simulation which simulates the antenna scan/track motion of the tactical sensor for the air-to-air radar mode. The resultant pointing angle is used by the radar detection/tracking module to determine target detection and maintain target tracking; and (4) ECM simulation which provides the ECM threat environment and detection capabilities.

AML PROGRAM TARGET MODULE. The AML target is a version of the NASA 1-9115 program which provides the simulation of a computer-controlled target while engaged in complex air-combat maneuvers with the trainee controlled aircraft. The program performs the decision logic plus the equations of motion for an instructor designated target.

ADVANCED TRAINING MODULE. This module is comprised of the following functions: (1) instructor controls which performs the logic associated with the instructor station switches which are functional during the various simulation modes. This logic includes trainer management controls; (2) repeater displays which provides the display data selection and preparation associated with the displays of the trainee cockpit instruments; (3) instructor displays provide the display data selection and preparation of the interactive display presentations for flight data, air combat situation, out-of-cockpit view, aircraft armament, weapon status, weapon scoring, approach area, ground approach, and procedure monitor displays; (4) record/playback consists of several routines which, during the training mode, gathers and stores simulation data, retrieves and replays the previously recorded mission; (5) procedure monitoring provides the logic associated with the selection, activation, and trainee monitoring of the designated procedure used in both normal and emergency operations of the aircraft equipment; (6) malfunction module performs the logic necessary for the instructor to select, insert, activate and de-activate simulated malfunctions which affect the simulated aircraft equipment operation and the associated cockpit displays and indicators; (7) parameter recording performs the data collection, formation and output of the data previously specified by the instructor; and (8) instructor training/demonstration provides the logic associated with replaying one of several predetermined scenarios which are resident on disk. The module presents all visual displays, aural and motion cues, instrument drives, and graphic displays as they appeared at the trainee station and instructor station during the training mission.

OPERATING SYSTEM. This module consists of the following modules: (1) system monitor provides the top level control capabilities common to the major simulation modes. This capability includes the handling of tapes and disc, I/O requests, interrupts, inter-processor communications and computer-operator interface and (2) executive performs the interrupt servicing, mode activation, intra-computer task scheduling, inter-computer task scheduling and system monitor interfacing for the various modes.

PROGRAM ORGANIZATION AND PERFORMANCE REQUIREMENT ANALYSIS

The module size, data and constant storage requirements for each functional group were determined by using the methodology described in the NAVTRAEQUIPCEN report[3]. The summation of the instruction count of each

identified module, and constant and data counts for each module determines the total storage (memory) requirements for that major functional group. For the purpose of this study effort, estimation of the storage and computational load within 20 per cent is considered acceptable. FORTRAN IV was considered to be the programming language for the VTXTS training system. The inefficiencies in the code generated by FORTRAN compilers as compared to routines coded in assembly level language are reflected in the estimates. In order to allow for future expansion, spare memory and spare processing capability considerations were included in this study effort. Spare memory and spare processing capacity is defined as a percentage of the installed memory or processing capacity per processor.

WORST-CASE INSTRUCTIONS EXECUTED. In determining the time required to execute the various program modules, it is necessary to determine the worst-case number of instructions of each major program module which is logically possible to be executed in a single path through the routine for each computation cycle.

The worst-case instruction execution counts for various simulation functional groups for each class of VTXTS simulator were derived from past experience and other trainer programs with similar processing requirements. The various program modules, execution rates, worse-case instruction count, and the worse-case number of instructions to be executed for each class of VTXTS trainer are based upon current simulators presently in use.

INSTRUCTION EXECUTION RATE. The highest execution rate for the VTXTS OFT and ACMT was selected to be 30 HZ. The basis for selecting this execution rate was influenced by several factors which are: (1) selected type of aircraft (trainer), (2) future performance considerations, and (3) minimization of overall simulation system time delay and subsystem time lags. Submultiple rates of 30 (i.e., 15, 10, 5, 1) were assigned to other program modules according to the required responses of those modules.

The minimum real-time execution rate required by the total simulation program (expressed as instructions per second (IPS)) is the summation of the product of the worst-case instruction count for each module and required solution rate for that module. The program modules with assigned solution rates for each of the various simulation functions for the CPT, OFT and ACMT are derived from current trainers in the Navy's inventory.

VTXTS STORAGE AND PROCESSING REQUIREMENTS

The random access memory requirements for typical CPT's, OFT's and ACMT's are provided in Tables 4, 5 and 6. The processing requirements are given in Tables 7, 8 and 9. The simulators for which data are provided are described below.

VISUAL TECHNOLOGY RESEARCH SIMULATOR. This simulation of a T-2C aircraft consists of two CPU's driving one cockpit with a motion and model board visual system. This system is used as a research tool. The computer system provides for the simulation of aircraft flight dynamics, aircraft systems, data

[3] Summer et al, ob. cit.

NAVTRAEQUIPCEN IH-336

TABLE 4. CPT STORAGE REQUIREMENTS

PROGRAM MODULE	STORAGE (32 BIT WORD)	
	MIN	MAX
Aircraft System	5,000	6,000
COM/NAV	1,400	1,600
Tactical Environment	5,000	6,000
Instructor Station	9,600	14,400
Executive/Operating Sys.	12,000	32,000
Total	31,250	60,300

NAVTRAEQUIPCEN IH-336

TABLE 5. OFT STORAGE REQUIREMENTS

PROGRAM MODULE	SIMULATOR				
	VTRS	2F90	2F132	2F101	2F129
	32-BIT WORD	32-BIT WORD	32-BIT WORD	32-BIT WORD	32-BIT WORD
Flight	3,900	5,600	52,000	5,200	15,200
Engines	1,000	--	1,800	1,600	5,600
NAV/COM	2,700	--	11,400	3,800	7,800
Accessories	8,100	700	10,300	4,000	4,600
G-Seat/G-Suit	1,100	--	700	--	--
Motion	1,200	--	--	500	500
Instructor Station	19,900	7,900	47,600	7,900	11,200
Executive and OP SYS	18,300	600	16,400	11,100	18,900
Total	56,200	14,800	140,200	34,100	63,800

NAVTRAEQUIPCEN IH-336

TABLE 6. ACMT STORAGE REQUIREMENTS

PROGRAM MODULE	SIMULATOR	
	DEVICE 2E6	DEVICE 2E7
	32-BIT WORD	32-BIT WORD
Flight	16,500	21,800
Engines	1,400	3,000
G-Seat/G-Suit	1,800	4,100
Accessories	3,800	3,100
Weapon System	8,900	8,400
Visual System	4,200	3,700
AML	9,900	12,200
Instructor Station	15,800	32,600
Executive	14,000	16,000
Total	76,300	104,900

NAVTRAEQUIPCEN IH-336

TABLE 7. CPT PROCESSING REQUIREMENTS

PROGRAM MODULE	PROCESSING REQUIREMENTS INSTRUCTIONS PER SECOND	
	MIN	MAX
Aircraft System	1,300	1,900
COM/NAV	2,800	4,200
Tactical Environment	32,800	49,200
Instructor Station	16,600	24,900
Executive	17,800	26,600
Total	71,300	106,800

NAVTRAEQUIPCEN IH-336

TABLE 8. OFT PROCESSING REQUIREMENTS

PROGRAM MODULE	SIMULATOR				
	VTRS (IPS)	2F90 (IPS)	2F132 (IPS)	2F101 (IPS)	2F129 (IPS)
Flight	136,250	97,900	254,000	128,110	437,800
Engines	14,500	50	19,000	7,000	51,000
NAV/COM	52,000	8,950	107,000	36,000	107,000
Accessories	69,750	28,900	34,000	24,500	130,000
G-Seat/G-Suit	108,000	---	30,000	---	---
Motion	52,500	---	---	---	1,500
Instructor Station	37,800	9,000	120,000	42,250	7,500
Executive	9,000	18,000	36,000	4,000	87,000
Total	479,800	162,800	600,000	241,860	804,800

IPS - Instructions Per Second

NAVTRAEQUIPCEN IH-336

TABLE 9. ACMT PROCESSING REQUIREMENTS

PROGRAM MODULE	SIMULATOR	
	DEVICE 2E6 (IPS)	DEVICE 2E7 (IPS)
Flight	231,200	174,000
Engines	4,800	26,000
G-Seat/G-Suit	96,000	10,2000
Accessories	48,000	34,500
Weapon System	23,120	158,500
Visual System	112,000	123,000
AML	174,200	195,000
Instructor Station	345,600	221,000
Executive	67,200	75,000
Total	1,102,120	1,109,000

IPS - Instruction Per Seconds

NAVTRAEQUIPCEN IH-336

recording, reduction and analysis and provides the means for entering an extensive number of malfunctions and augmentation of problem/experiment development and control through an interactive CRT/keyboard display system. The highest solution for this research system is 30 Hz.

DEVICE 2F90. This TA-4J OFT uses a multi-processor system architecture. In this training system two CPU's share the computational for four simulated TA-4J aircraft. The computational system provides for the simulation of the four independent simulated aircraft, the aircraft systems, data recording and performance measurements and the instruction control functions. The highest solution of this training device is 20 Hz.

DEVICE 2F132. This F-18 OFT consists of four CPU's to drive one cockpit with a visual system. The computational system provides for the simulation of a complex fighter/attack aircraft and the aircraft systems, the ability to introduce an extensive number of malfunctions, and the ability to record/playback training and performance data, the capability for scenario generation and problem control using an interactive CRT/keyboard system. The highest solution rate for this training device is 60 Hz.

DEVICE 2F101. This T-2C OFT consists of four CPU's driving four simulated cockpits. The computational system provides the complete simulation of all flight, engine, navigation and accessory systems of the T-2C aircraft. It also provides for data recording and problem control in conjunction with the instructor station and performs the operation of the executive system. The highest solution rate for this training device is 20 Hz.

DEVICE 2F129. This T-44A OFT consists of a dual CPU configuration to drive a single cockpit with a motion system. The computer system provides flight simulation for the T-44A aircraft, simulation of the aircraft systems including navigational aids, and the instructors station control functions. The highest solution rate for this training device is 30 Hz.

DEVICE 2E6. This ACMT simulates two manned fighter/attack aircraft and one unmanned, synthesized, interactive aircraft, or an instructor controlled aircraft, engaged in one-on-one or two-on-one aerial combat with each other. The ACMT consists of two trainee stations, an instructor station, two target model image generators, visual projection system and eight CPU's. Each trainee station consists of a fixed-base cockpit mounted inside a 20-foot radius spherical vision envelope. The cockpit of one station simulates the F-4J aircraft and the other simulates the F-14A aircraft. The instructor station provides for scenario generation, trainee monitoring, problem control, data recording/playback and demonstration/critique. The highest solution rate for this training system is 30 Hz.

DEVICE 2E7. This WTT simulates two manned fighter/attack aircraft and one unmanned, synthesized, interactive aircraft, or an instructor controlled aircraft, engaged in one-on-one or two-on-one aerial combat with each other. The WTT consists of two trainee stations, an instructor station, CIG visual system and twelve CPU's. Each trainee station consists of a fixed-base cockpit mounted inside a 20-foot radius spherical vision envelope. The cockpit station simulates the F/A-18 aircraft. The instructor station provides for scenario generation, trainee monitoring, problem control, data

NAVTRAEQUIPCEN IH-336

recording/playback and demonstration/critique. The highest solution rate for this training system is 60 Hz.

The storage and processing requirements for each class of VTXTS trainer are provided in Table 10. These estimates were derived from available data on each class of trainer which is currently in the Navy inventory. This table provides a summary of the minimum and maximum storage and processing requirements for each class of VTXTS simulator. A 50 per cent spare/growth factor was added to both the minimum and maximum values. This is the spare/growth factor being used in current trainer procurements.

The minimum storage and processing requirements for the OFT and ACMT are based upon a minimum system configuration in the areas of NAV/COM, accessories, G-seat/G-suit, motion, and instructor station functions. Furthermore, the minimum configuration for ACMT does not include a full weapons system or programmed threat target (AML) simulation.

The aircraft performance envelope, training and system management requirements will determine the complexity of the VTXTS trainer suite and the computational system requirements. However, the stated minimum processing and storage requirements give a realistic estimate for the VTXTS suite.

TABLE 10. SUMMARY OF PROCESSING AND STORAGE REQUIREMENTS

DESCRIPTION	SIMULATOR			
	CPT	OFT	OFTV	ACMT
Processing Requirements: (Instruction Per Second)				
Minimum	111,500	284,000	296,000	767,000
Minimum plus 50%	167,300	426,000	444,000	1,150,500
Maximum	--	767,000	784,000	1,290,000
Maximum plus 50%	--	1,150,500	1,176,000	1,935,000
Storage Requirements: (32-Bit Word)				
Minimum	19,000	15,000	17,000	76,000
Minimum plus 50%	28,500	22,500	25,500	114,000
Maximum	28,000	14,000	144,000	1,049,000
Maximum plus 50%	42,000	21,000	216,000	1,573,500

SECTION IV

COMPUTER ARCHITECTURE

The problem to be addressed is the configuration of a computer system to service a number of different simulators at a site. The computer architectures considered can be classified in four categories: (1) use of a single computer to service several different simulators, (2) use of a single computer to service a single simulator, (3) use of several computers to service a single simulator, and (4) use of several computers to service several simulators. A simulator complement of 2 CPT's, 8 OFT's (5 with visual systems), and 2 ACMT's is chosen to illustrate the computer configurations considered. However, the results are carried through in general terms applicable to different simulator arrangements. Table 11 shows the processing requirements assumed for the analysis. These requirements are based upon the information given in Tables 7 through 9.

The analysis will be restricted to consideration of four classes of computers. These cover the spectrum from the large computers capable of handling all of the simulators at a site to the microcomputers which must be massed in dozens to handle even one large simulator. However, the major part of the analysis is spent on implementation of the VTXTS simulators using minicomputers like those used on other present-day flight simulators such as the 2E6 and the 2E7 and using the super-fast minicomputers now becoming available.

MINICOMPUTERS

Three different minicomputer configurations are considered in this analysis. The simplest in terms of implementation is shown in Figure 2. This is an attempt to use a one-on-one match of computer to cockpit, modified to take care of gross mismatches in the simulator requirements and the computer capabilities. For the OFT's the match in capabilities and requirements allows the one-on-one arrangement. Since the CPT's use less than half of the capability of a single computer, one computer is used to drive two CPT's. The ACMT implementation introduces the opposite problem. Since a single computer does not have enough capability to perform the computations required, two computers must be used for each ACMT.

The alternative to a one-on-one arrangement is to use multiple computers to drive multiple cockpits. Figure 3 shows an arrangement in which this principle is applied to the OFT's only, with the CPT and ACMT configurations unchanged. This arrangement requires two fewer computers, since the total computation requirement for four OFT's can be met with three computers. Figure 4 shows this concept applied to the entire complex of cockpits. This results in considerable saving in the number of computers required, reducing the 13-computer configuration shown in Figure 2 to only 8 computers. The disadvantage of this approach is an increase in software cost and complexity.

Each of these configurations lends itself to use of redundancy in order to improve system availability. Figures 5, 6 and 7 show configurations in which a spare computer is added into each of the configurations presented in

NAVTRAEQUIPCEN IH-336

TABLE 11. COMPUTATION REQUIREMENTS USED IN ANALYSIS

VARIABLE	SIMULATOR		
	CPT	OFT	ACMT
Average Inst. Exec. Rate (1000 Instructions/Second)	125	500	1200
Storage Requirements (1000 32-bit words)	24	64	96

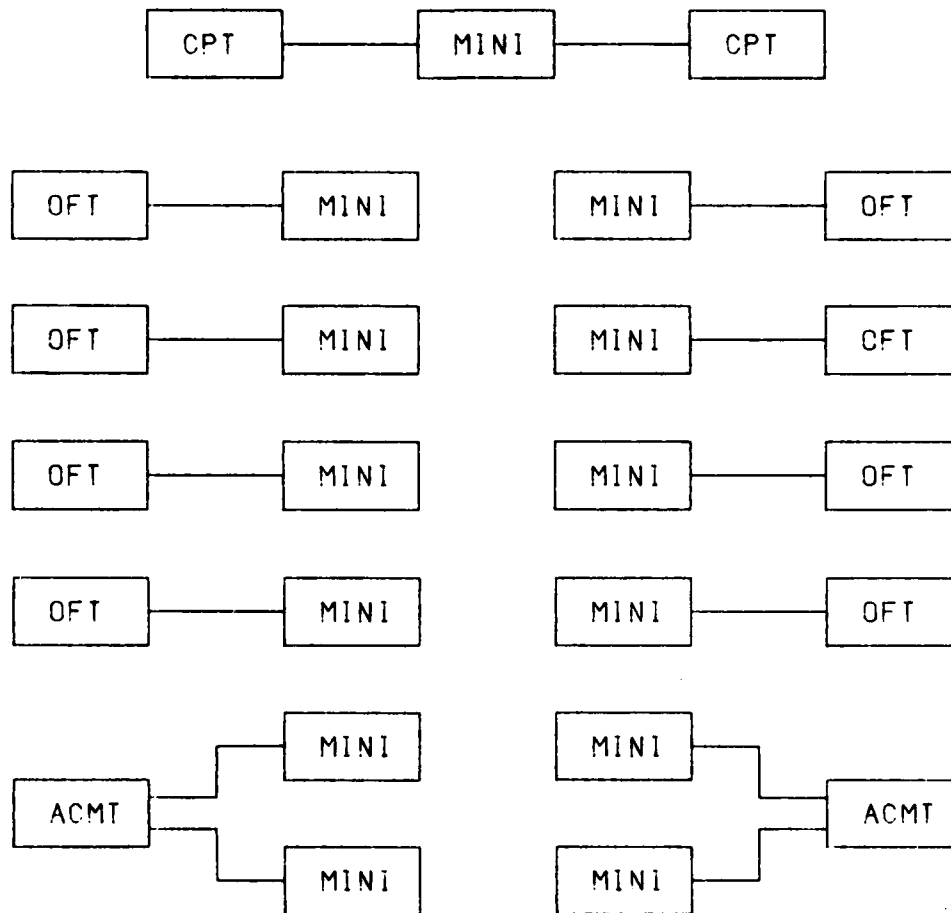


Figure 2. Dedicated Minicomputer Systems per Cockpit

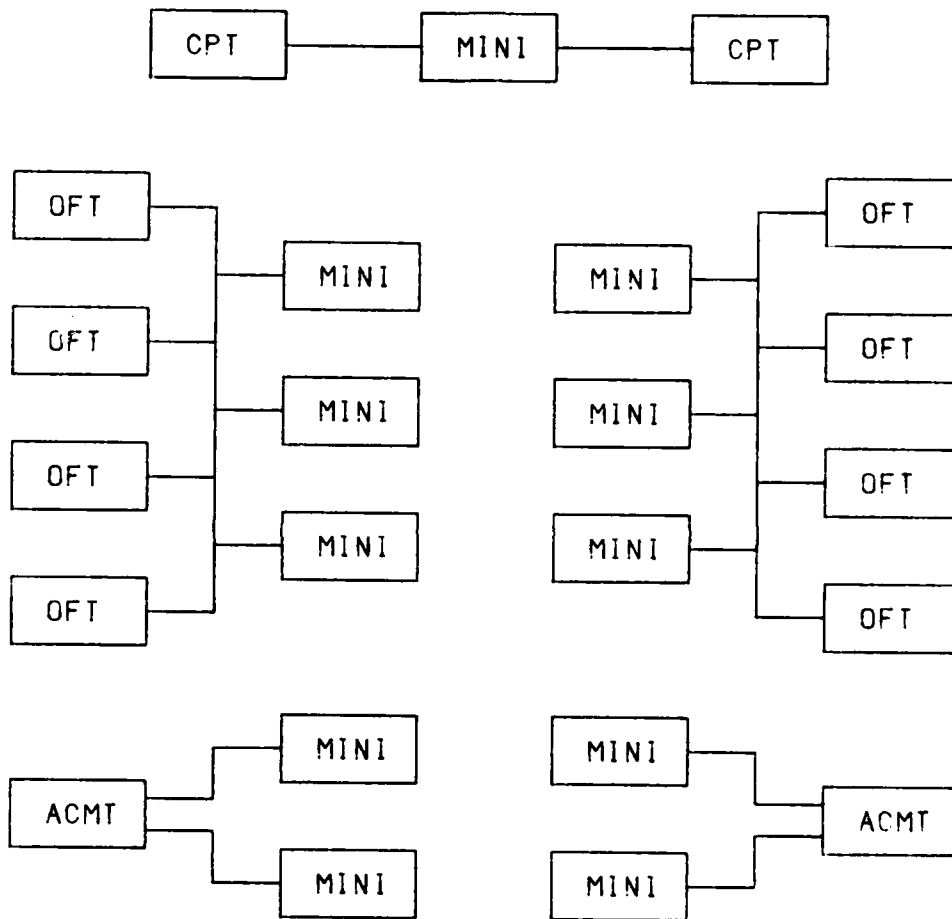


Figure 3. Shared Minicomputers for OFT Only.

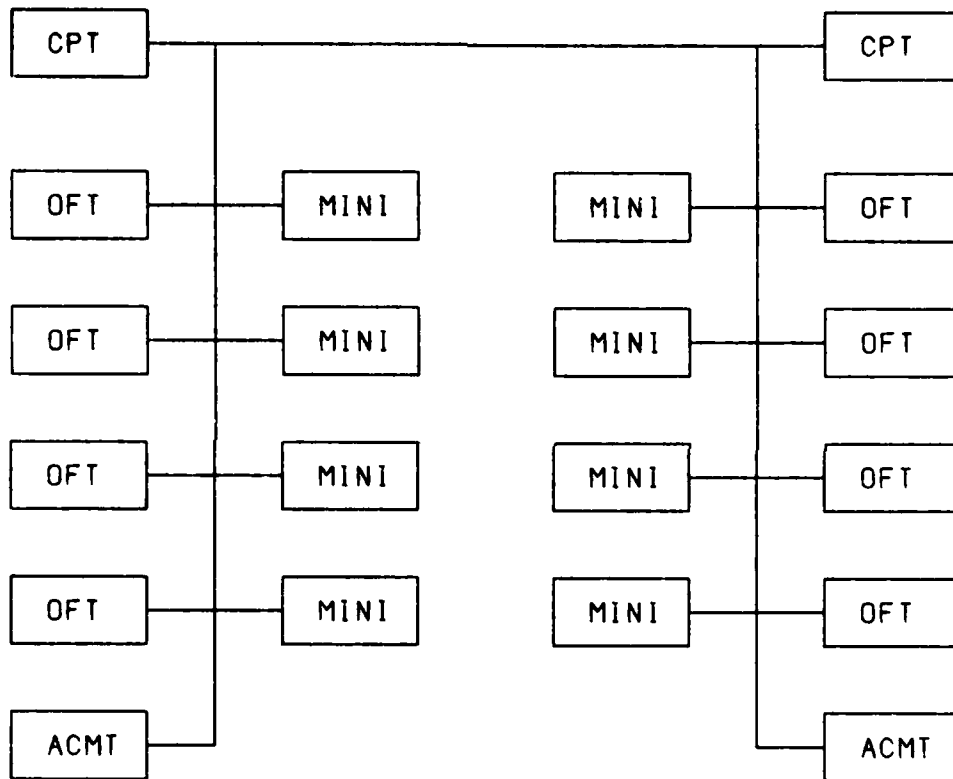


Figure 4. Shared Minicomputers for All Simulators/Cockpits.

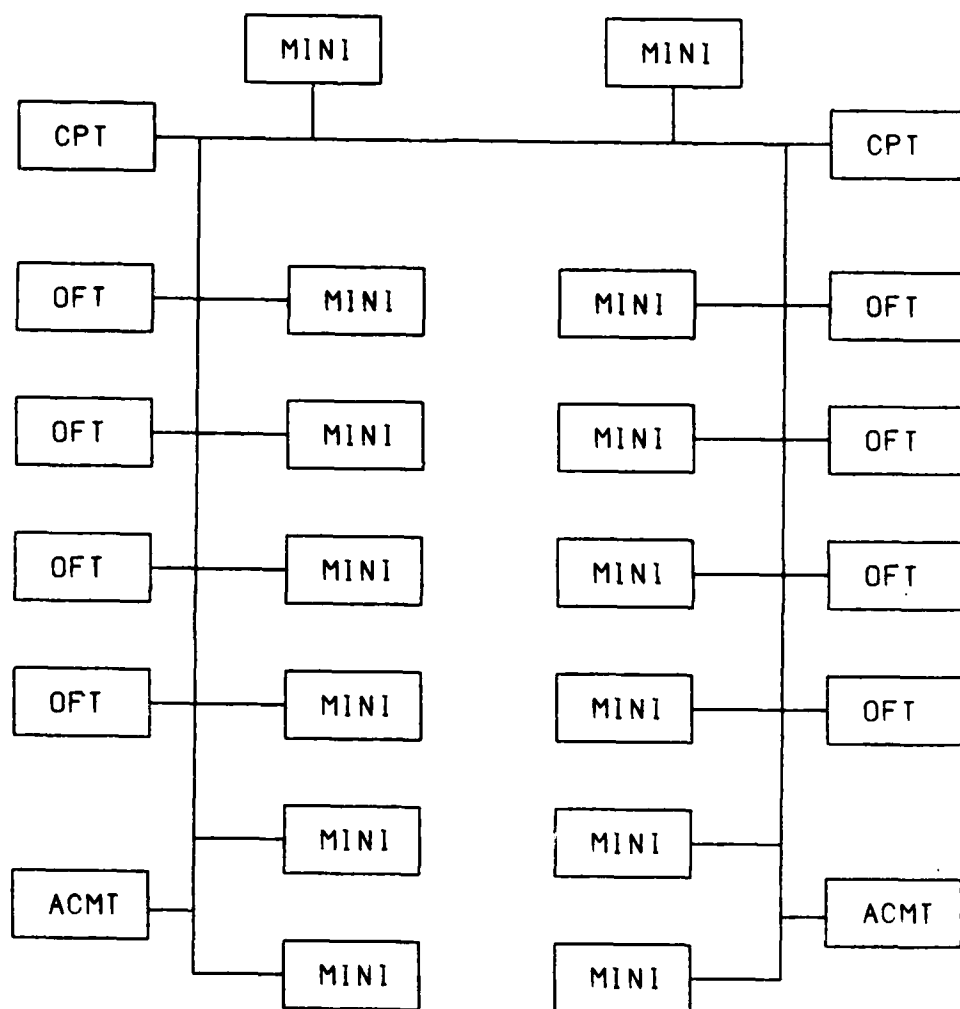


Figure 5. Separate Minicomputer Systems per Cockpit with a Redundant Computer.

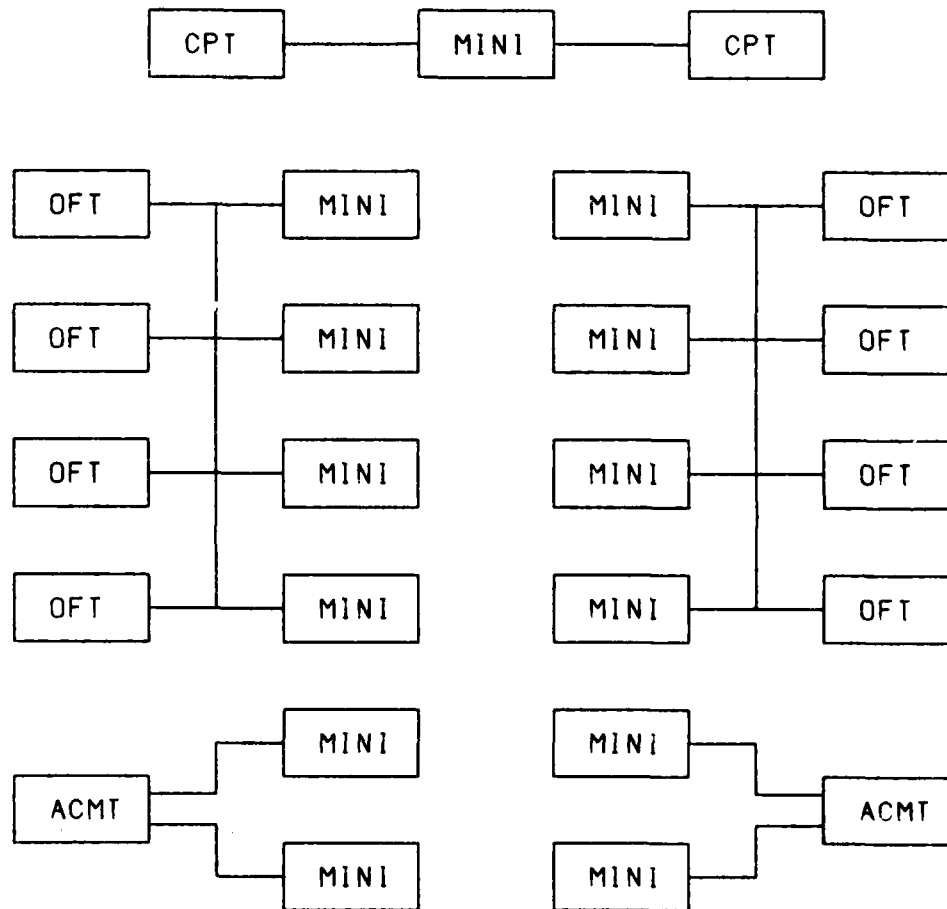


Figure 6. Shared Minicomputers for OFT Only with a Redundant Computer.

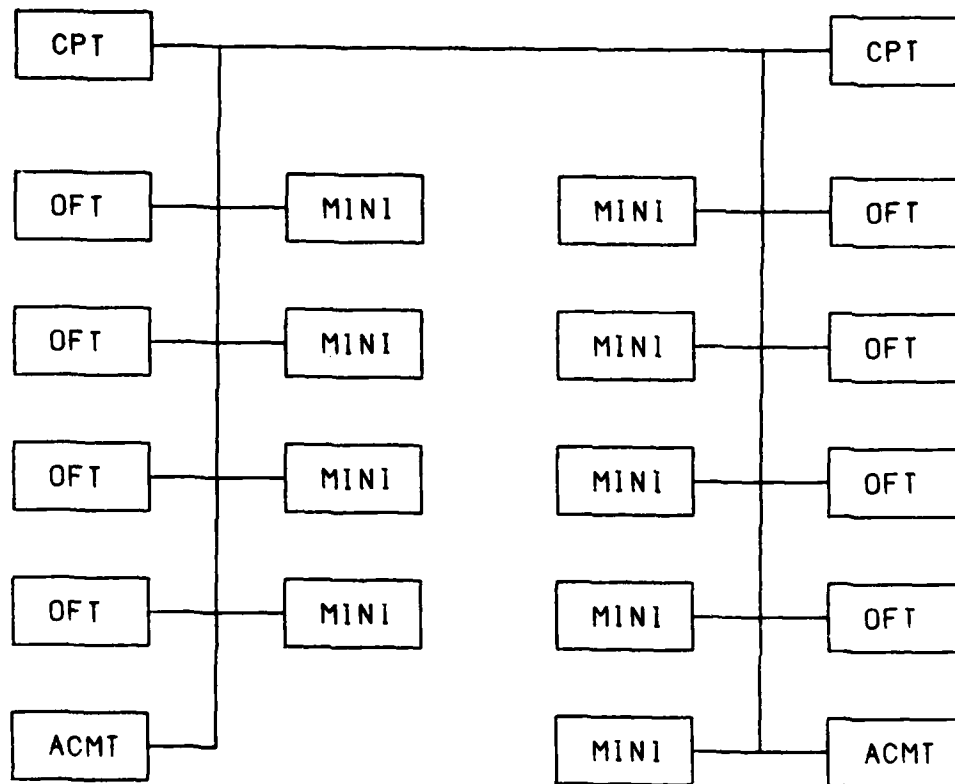


Figure 7. Shared Minicomputers for All Simulators/Cockpits with a Redundant Computer.

Figures 2, 3 and 4. It is assumed in this analysis that the additional computers do not interact and that the device for switching computers in and out of the circuit does not affect the reliability. Although these assumptions cannot be realized in practice, the effect is minor and does not change the result of the analysis.

SUPER MINICOMPUTERS

Figure 8 shows an arrangement of Super-Minicomputers suitable for the VTXTS problem. The same configuration with the addition of a redundant computer is shown in Figure 9. Use of Super Minicomputers is attractive because, unlike the large computers described next, the Super Minicomputer provides a greater increase in processing capability than the increase in cost. The reason for this is that the major additional cost is in the high-speed arithmetic element rather than I/O devices or sophisticated hardware for improving the computer's performance in a multiuser environment.

It might seem at first glance that there would be no reason to add a redundant computer to a system that requires only two computers to perform the job. However, when one super minicomputer is down, half of the cockpits are down. The trade-off to evaluate whether the use of redundant computers is cost effective is presented in Section VI.

LARGE COMPUTERS

A single large computer can be used to replace all of the minicomputers at a site. Figure 10 shows this arrangement. In general, use of a single computer will add to the programming requirements for a system, but this slight addition is ignored in the analysis. The major effect of using a single large computer is a decrease in total availability of the system. Figure 11 shows the addition of a redundant computer to improve availability.

A major consideration in using a large computer for the VTXTS is the cost of the large computer. In general, the cost of a larger, faster computer grows more than linearly (i.e., a computer ten times as fast usually costs more than ten times as much). For example, the Amdahl 470V/7 costs about 25 times the price of a Perkin-Elmer 3240 [4] but has only a 20 to 1 advantage in computation speed. Use of a large computer cannot be discarded in all cases, but this type of architecture does not seem to provide any advantages in the VTXTS application.

MICROCOMPUTERS

Multiple microcomputers can be used to replace the minicomputers in a simulator as shown in Figure 12. This arrangement adds overhead for a control processor or embedded control and also depends upon how well the problem to be solved can be partitioned in order to divide the computation among the microcomputers. These are second order effects not considered in determining the number of microprocessors required.

[4] GML Corporation, ob. cit.

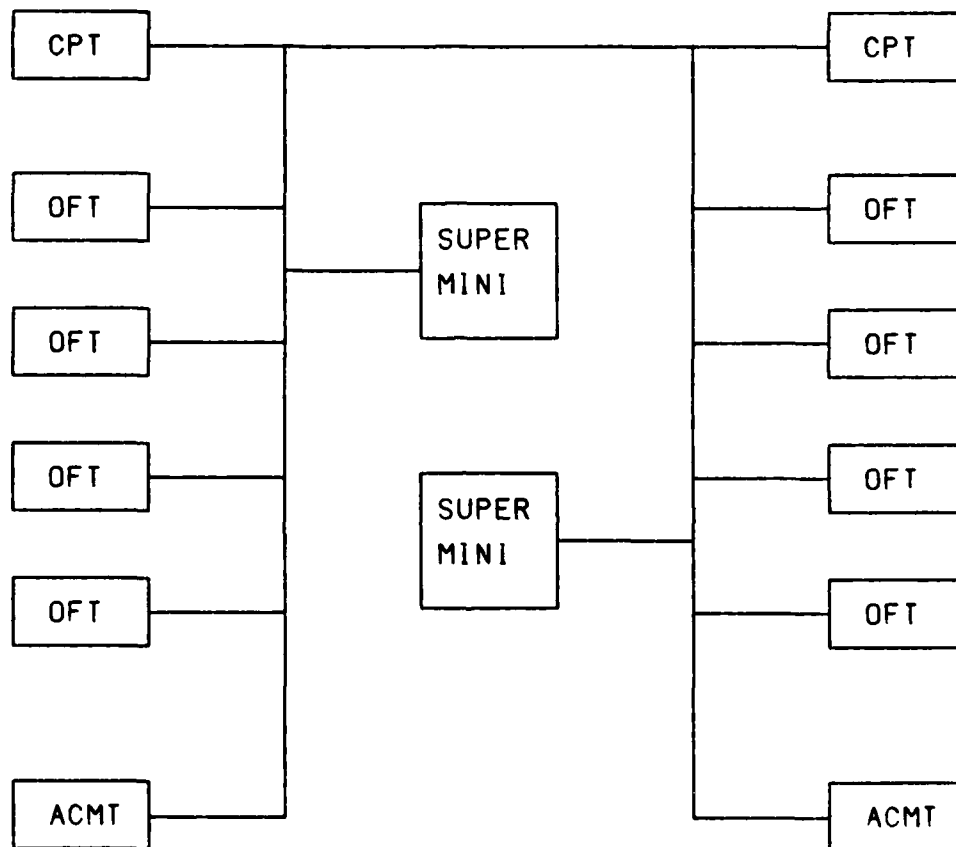


Figure 8. Super Minicomputer Configuration for all Simulators/Cockpits.

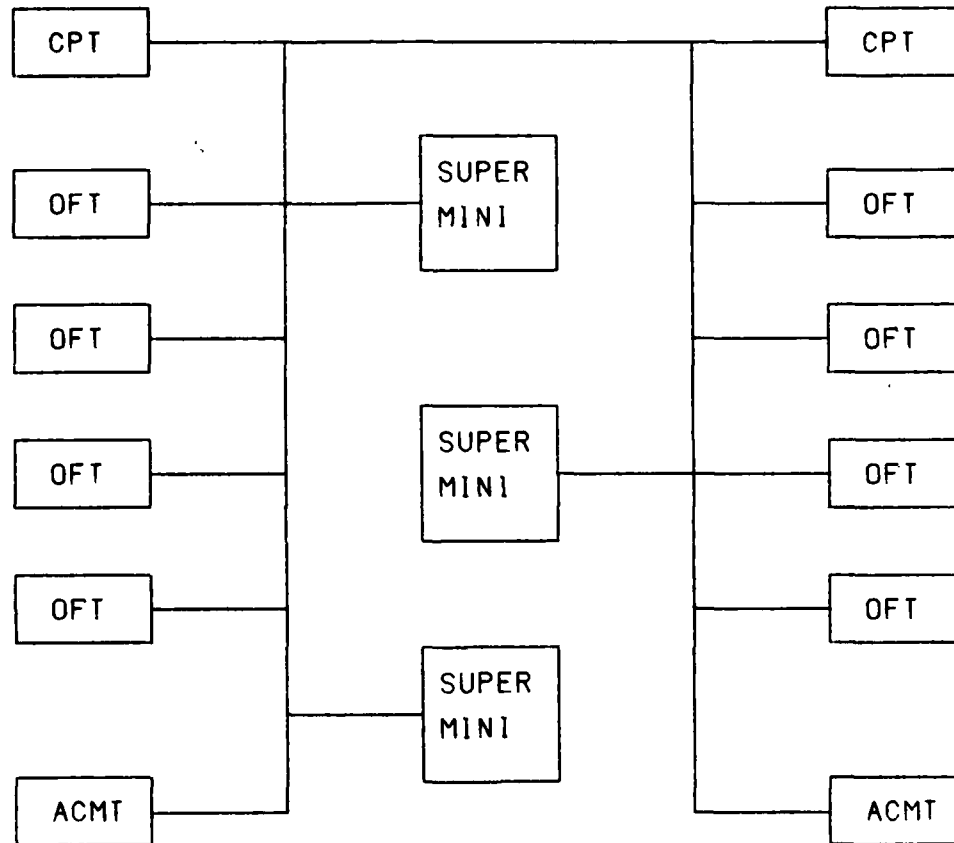


Figure 9. Super Minicomputer Configuration for all Simulators/Cockpits with a Redundant Computer.

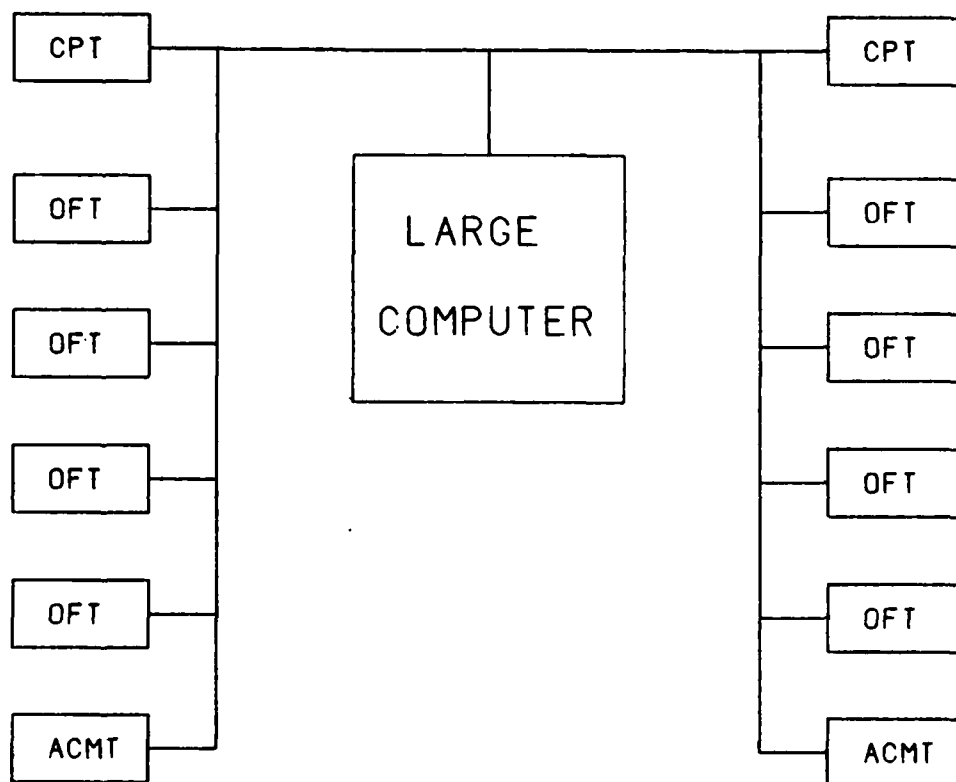


Figure 10. A Shared Single Large Computer for All Simulators/Cockpits.

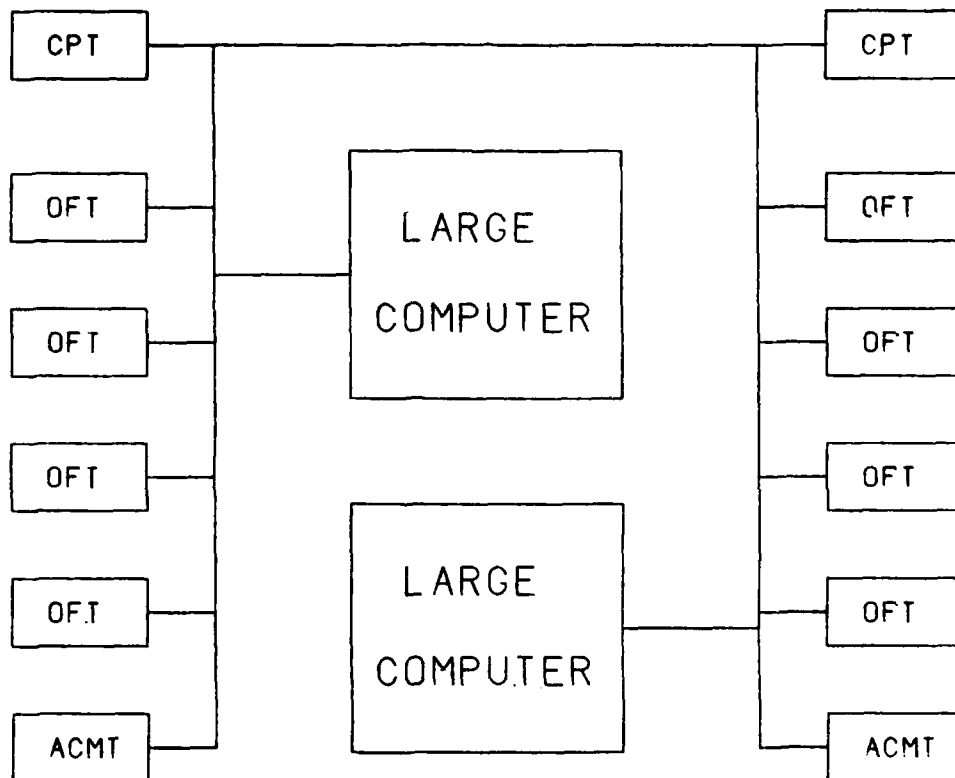


Figure 11. Shared Single Large Computer for all Simulators/Cockpits with a Redundant Computer.

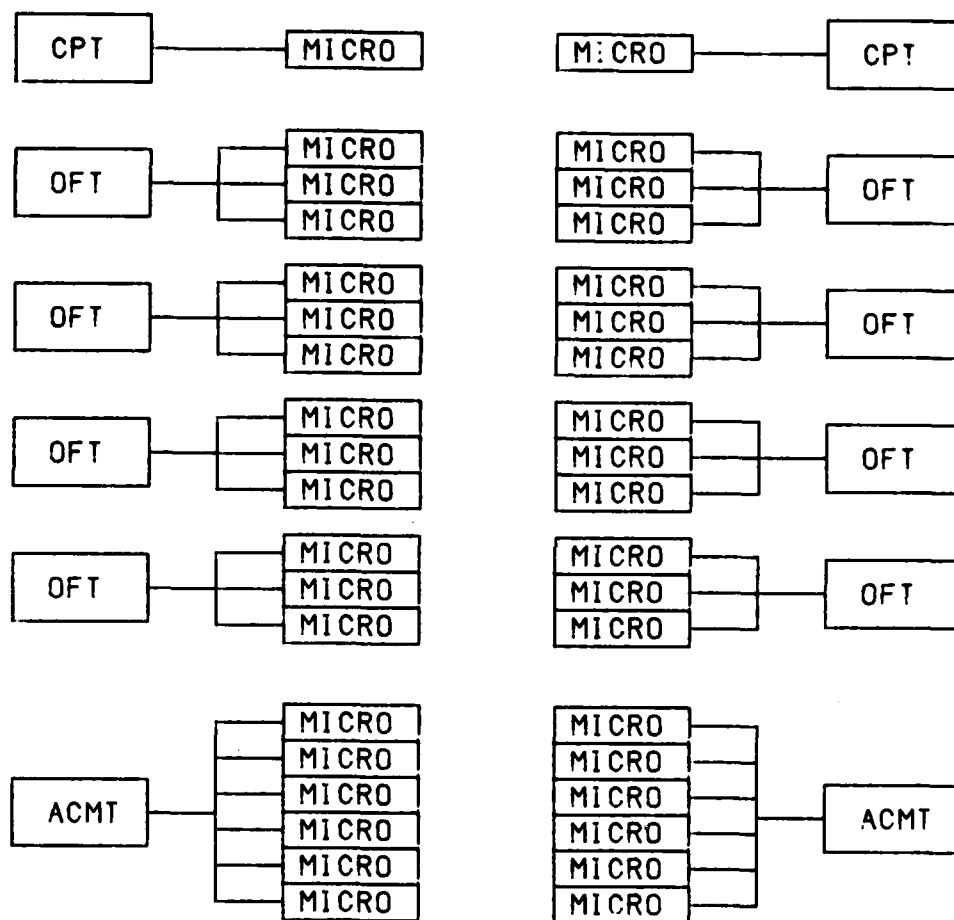


Figure 12. Dedicated Microcomputer Systems per Simulator/Cockpit.

Use of multiple microprocessors offers saving in hardware cost over the use of larger computers. Summer and Wyndle [8] estimate the cost of a microprocessor OFT to be little more than one-half the cost of an OFT implemented with minicomputers. Their analysis of life cycle costs show a similar saving in software and maintenance costs. Indeed, the multiple microcomputer network provides a number of advantages over other approaches. However, these advantages are outweighed by its one disadvantage--such a system has not yet been developed. Use of multiple microcomputers for the VTXTS would result in high development costs and an associated high risk in added cost and in schedule slippage.

The development cost and high risk are a result of problems that must be solved in order to use multiple microcomputers effectively. The basic architecture of the multiple-microcomputer system concept is to take advantage of the inherent parallelism that exists in many applications. To do this successfully depends upon three major criteria:

- a. Successful partitioning of the problem into disjoint tasks.
- b. Provisions for some form of centralized control by an operation.
- c. Developing a run-time structure which provides for the passing of system parameters between tasks and/or processors while preserving precedence.

NAVTRAEQUIPCEN has conducted an active research program in this area since 1974 [9,10]. Implementation problems which have been solved include:

- a. The development and demonstration of a control algorithm for "N" microcomputers and embodied in hardware, firmware and software. This control algorithm is identical in each microcomputer and functions as an applications task manager (ATM) for distributed control.
- b. The development and demonstration of a distributed cache concept whereby the address space of a shared (common) memory is distributed among each applications processor. Data and parameters to be passed to various processors from other processors are broadcast globally on the system data bus but are read locally in each processor in a parallel manner. This significantly reduces the system data bus bandwidth requirements and the likelihood of bus contention.

[8] Summer, C. F., and G. A. Wyndle, An Analysis of Microcomputer Technology for Application to Real-Time Trainers. Orlando, FL, July, 1979, Report No. NAVTRAEQUIPCEN IH-313.

[9] Pettus, R. O., R. D. Bonnell, M. N. Huhns, L. M. Stephens, and G. M. Wierzba, Multiple Microcomputer Control Algorithm. Orlando, FL, Sep, 1979, Report No. NAVTRAEQUIPCEN-78-C-0157-1.

[10] Pettus, R. O., M. N. Huhns, L. M. Stephens, and M. O. Trask, Multiple Microcomputer Control Algorithm Feasibility Breadboard. Orlando, FL, Aug, 1981, Report No. NAVTRAEQUIPCEN-79-C-0096-1.

NAVTRAEQUIPCEN IH-336

- c. The processing frame scheduling (unique to real-time sampled data processing) is carried out by the control processor. Groups of microcomputers are scheduled according to the processing required in a given frame time. All interprocessor communications are controlled by the ATM which is stored in a PROM.

Figure 13 shows a block diagram of a system using multiple microprocessors to solve the simulation problem. Evaluation effort is being conducted by NAVTRAEQUIPCEN on an in-house research tool using Motorola 6809 microcomputers. Task scheduling, bus accesses and distributed control are vested in the applications task manager firmware. Initial investigations will address program partitioning and limitations imposed by task scheduling and bus traffic.

Figure 12 shows a microcomputer configuration to solve the VTXTS problem. This implementation is interesting for future applications, but the research program has not yet provided the answers needed to use this approach. This configuration is included in the evaluation with the high risk of such an approach made part of the analysis.

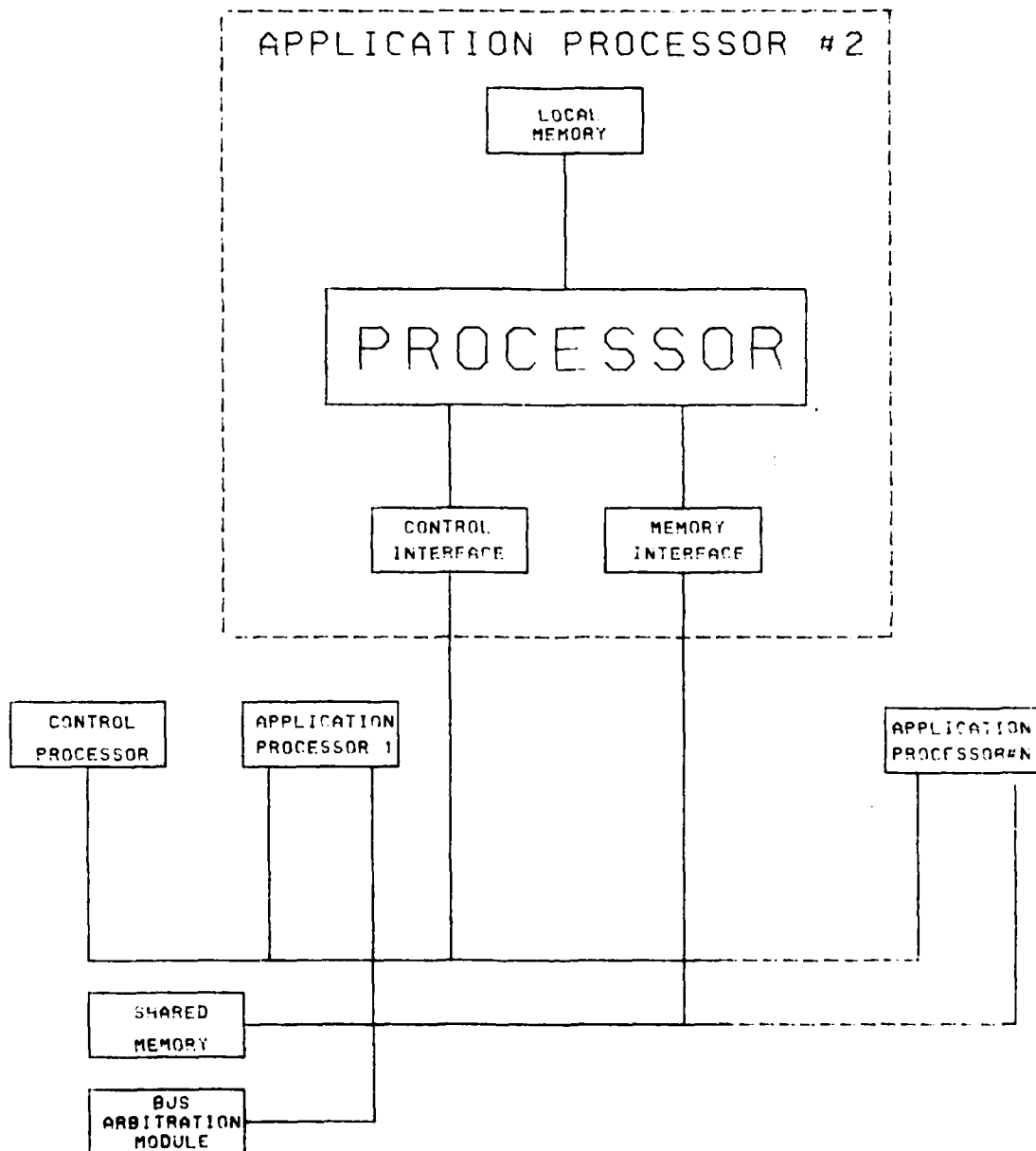


Figure 13. Microcomputer System Block Diagram.

SECTION V

MINICOMPUTER CONFIGURATIONS

This section provides more detail about the minicomputer configurations that might be used in implementing the VTXTS trainers. It provides a discussion of dual CPU architectures and extends these to multiple CPU architectures suitable for the VTXTS. Two suitable configurations result from this analysis: (1) one using conventional minicomputers and (2) one using new super-fast minicomputers.

ANALYSIS OF DUAL CPU ARCHITECTURES

Dual CPU architectures are used to illustrate the various methods that might be used to provide communication between computers.

DUAL CPU WITH SHARED MEMORY ONLY. This configuration is shown in Figure 14. Each processor requires a port on each memory bank. Since all the memory is shared, a problem arises with the interrupt and traps for the slave processor. This problem can be described as follows:

Using the floating point divide by zero trap as an example, assume the slave processor pulls the trap. The trap location contains an instruction which saves the present location counter and transfers control to a trap handler which does something with the attempt to divide by zero. Should the master CPU also pull a divide by zero trap while the slave is executing the trap handler, the master would execute the instruction at the trap location which would store its location counter in the same location used by the slave CPU and start executing the trap handler. The return address now is the value of the location counter of the master CPU when it pulled the trap.

When the slave CPU finishes the trap handler and attempts to return to its program, it instead returns to the master's program since that is where the return address sends it. When the master CPU finishes the trap handler, it also returns to its own program, and we have both CPU's executing the same code.

The problem is solved by providing an address translator which moves the trap and interrupt addresses for the slave CPU to a part of memory different from that used for interrupt and traps by the master CPU. This provides each CPU with its own trap locations and trap handlers.

This approach provides a simple way to augment the capability of a computer system by a factor of nearly two. However, it has the following disadvantages:

- a. The operating system and the loader must be modified to accommodate use of a slave computer.

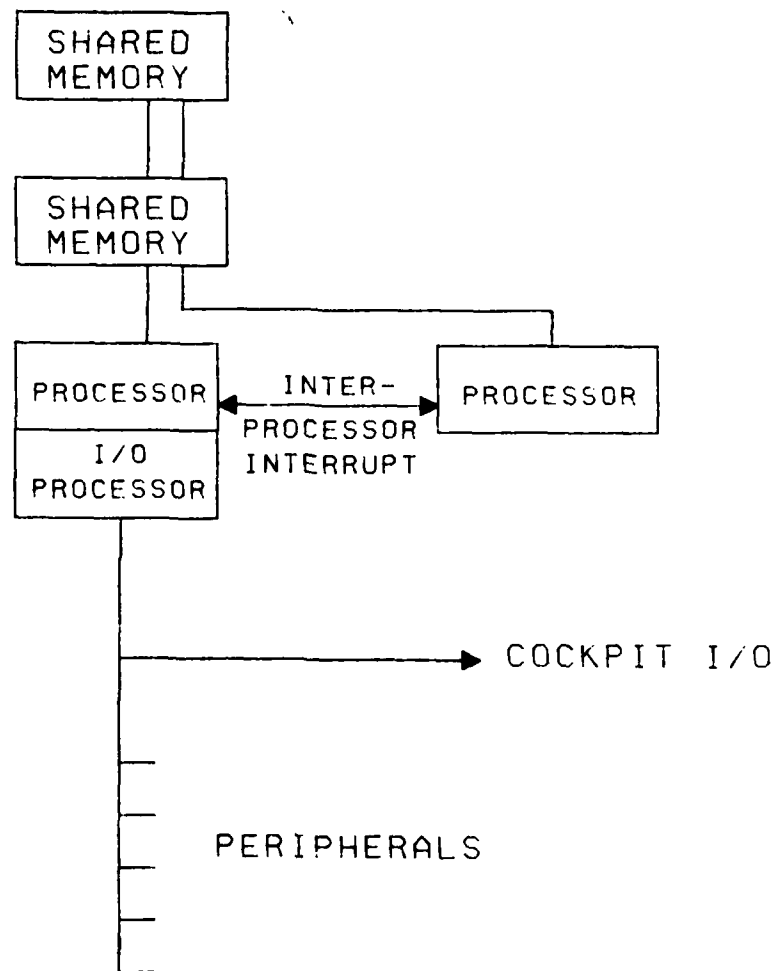


Figure 14. Dual CPU with Shared Memory Only.

- b. If a FORTRAN math library routine like SIN(X) is to be used by both CPU's, two copies of the routine are required.
- c. The slave CPU cannot perform background tasks or perform I/O to the peripherals (although it can communicate with the cockpit interface). The slave CPU's unused time is not available for background processing.
- d. The system cannot be easily reconfigured if the master CPU goes down. The master CPU cannot be repaired without having the entire system available to the computer technician.
- e. The system still has very little redundancy. The I/O processor and all the peripherals are not redundant, and loss of any one of them will bring the system down.

DUAL CPU WITH SHARED AND DEDICATED MEMORY. This configuration is shown in Figure 15. The addition of a dedicated memory solves some of the disadvantages listed above and provides a configuration that can be extended to multiple CPU's.

Each CPU has its own copy of the operating system which resides in dedicated memory. Each CPU also has its own 80MB disk and set of peripherals. A 300MB disk is shared between the CPU's. The 80MB disk contains the operating system and system temporary files for each CPU. The 300MB disk contains user programs which can be accessed from any CPU. Organizing the system this way allows the operating system to be identical for each CPU. Use of two disks in this manner has a significant advantage during program development as the throughput is significantly better than it would be with a single disk. Use of a small dedicated disk for each computer is also required for storing time history data during a training exercise and for storing data for a preprogrammed demonstration. Use of the 300MB disk for this purpose would probably result in cueing problems with the 300MB disk, since it is shared with other processors. It would be possible with this system to design the operating system to run off of the 200 MB disk in case of a RAD failure on one of the CPU's.

The interprocessor interrupt allows one CPU to poll the other CPU to determine its status. This feature is particularly applicable when the dual CPU configuration is expanded to multiple CPU's. Each CPU needs to know whether the other CPU's are dead or alive. The interprocessor interrupt can also be used for synchronization of real time programs. In particular, if one CPU does all the I/O to the cockpit, the interprocessor interrupt can be used to "wake up" the other CPU's and get them started on the processing for this time frame.

Use of some dedicated and some shared memory on each machine eliminates many of the problems encountered with the dual CPU configuration with shared memory only. Since each CPU has its own copy of the monitor, servicing the traps is exactly the same as for a single CPU. Copies of library routines such as SIN, COS, etc., can now be shared as long as their temporary stacks reside in dedicated memory. Each CPU can perform I/O. The system is highly

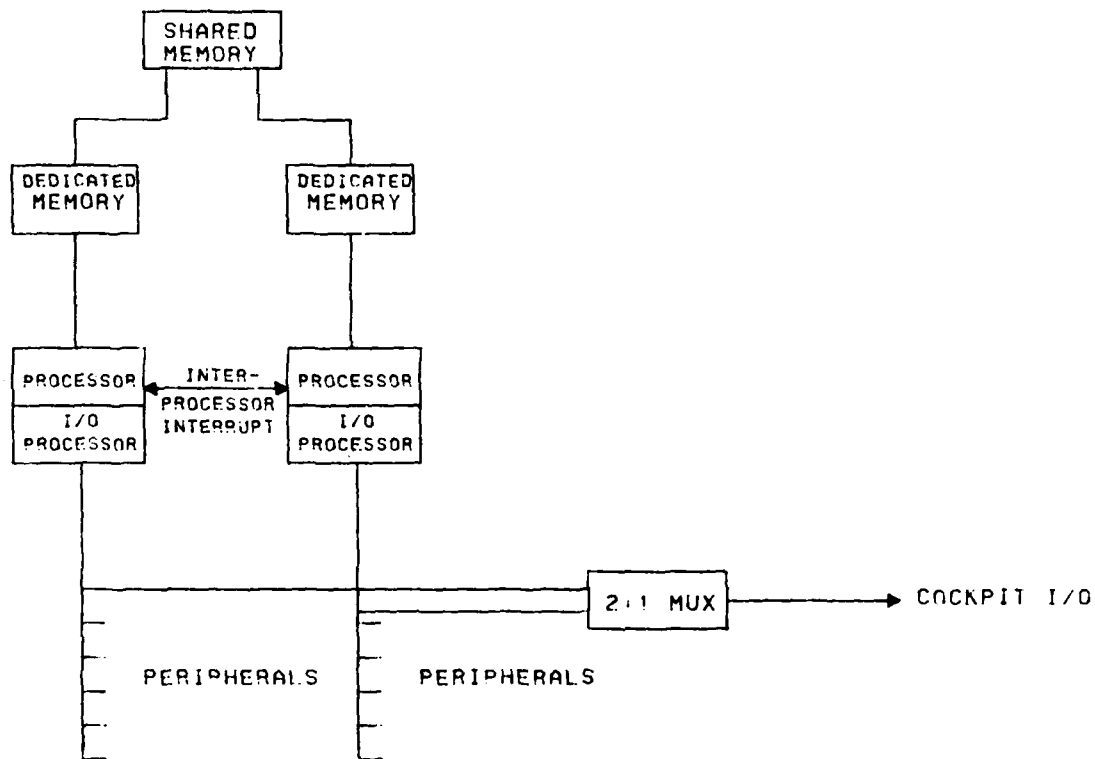


Figure 15. Dual CPU with Shared and Dedicated Memory.

redundant permitting operation with a significant portion of the system down. The system can be repaired off-line. All of the unused CPU time of each CPU is available to the background.

DEDICATED CPU WITH DEDICATED MEMORY ONLY. This configuration is shown in Figure 16. It uses a communication channel between the CPU's for transmittal of problem parameters during a computation cycle. This approach is extremely awkward unless the problem can be partitioned into two essentially independent parts. This is not usually the case in real-time simulator applications. However, this approach can be used in a multiple CPU configuration with communication over a shared bus.

CPU WITH PERIPHERAL PROCESSOR, DEDICATED MEMORY ONLY. This configuration is shown in Figure 17. It is similar to system described above except that the slave CPU has no peripherals. It therefore presents an even greater programming challenge. It also has the disadvantage that the two CPU's do not appear identical to the software. This prevents reassignment of CPU's in the event of failure, unless a significant amount of hardware reconfiguration was done. It also prevents the off-line check-out of a failed CPU since a failed CPU would necessarily be in a slave configuration and not in a master configuration.

MEMORY SHARING

From the above analysis, the dual CPU configuration with dedicated and shared memory is the most desirable configuration. It can be expanded to a multi-CPU configuration in one of several ways. Figure 18 shows one concept for a multiple CPU configuration with pairs of CPU's sharing a memory bank (in addition to the dedicated memory bank assigned to each CPU). The switches were introduced to allow switching out of a failed CPU. For example, if CPU 3 failed, all the switches to the right of CPU 3 would be switched to the right, thus leaving CPU 3 without a shared memory. CPU 3 could then be repaired as required. This approach requires a ported memory as do all the other approaches with shared memory. This type of memory may not be available from some vendors.

The disadvantages of this configuration are the following:

- a. The shared memory banks are not redundant, so a failure will disable two of the computers. Memory failures are more common than CPU failures.
- b. The switches would require some development, and could lead to speed problems due to the speed of today's memories.

Figure 19 shows an alternate memory configuration using switches which is suitable for either an even or odd number of CPU's. It has the disadvantage that twice as many shared memory modules as used in Figure 18. It has the advantage that loss of a memory bank will not cause the loss of a pair of CPU's since redundant memory is available.

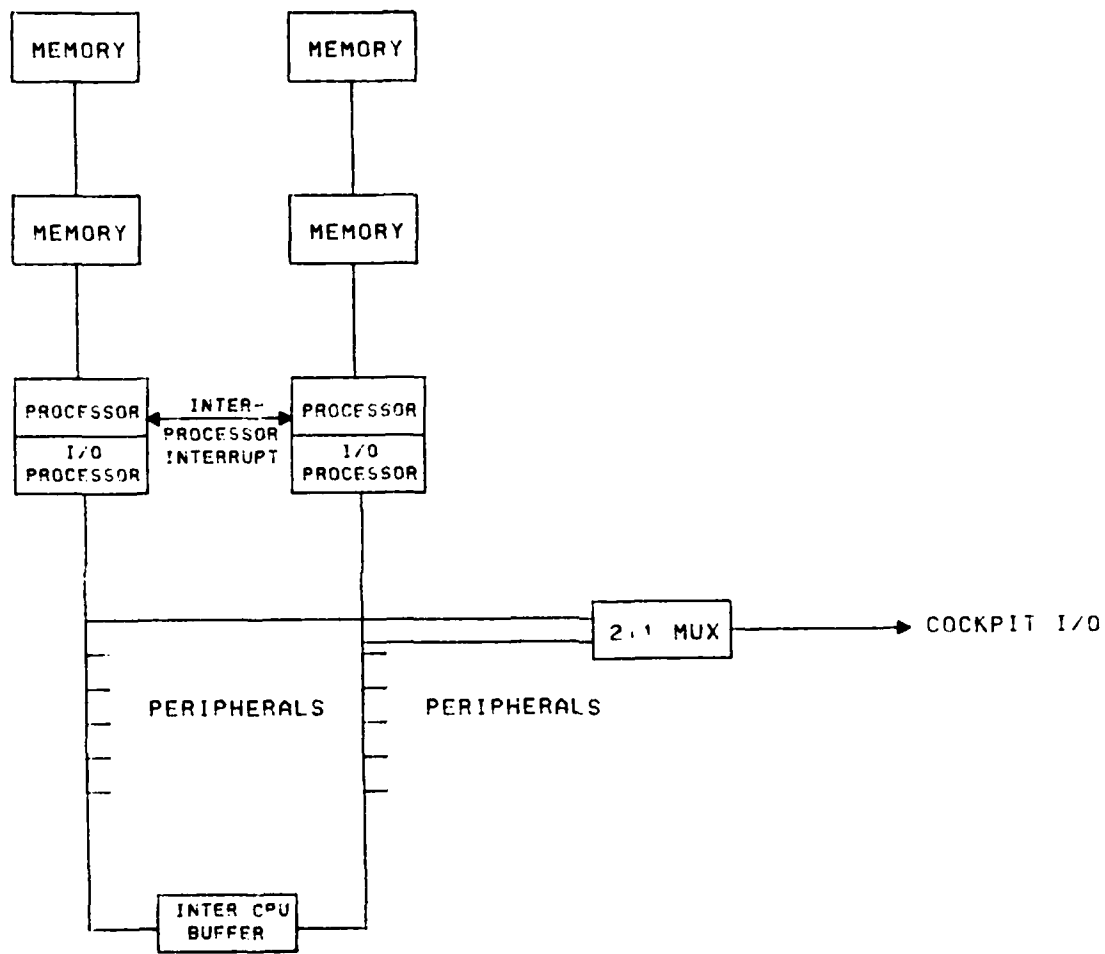


Figure 16. Dedicated CPU with Dedicated Memory Only.

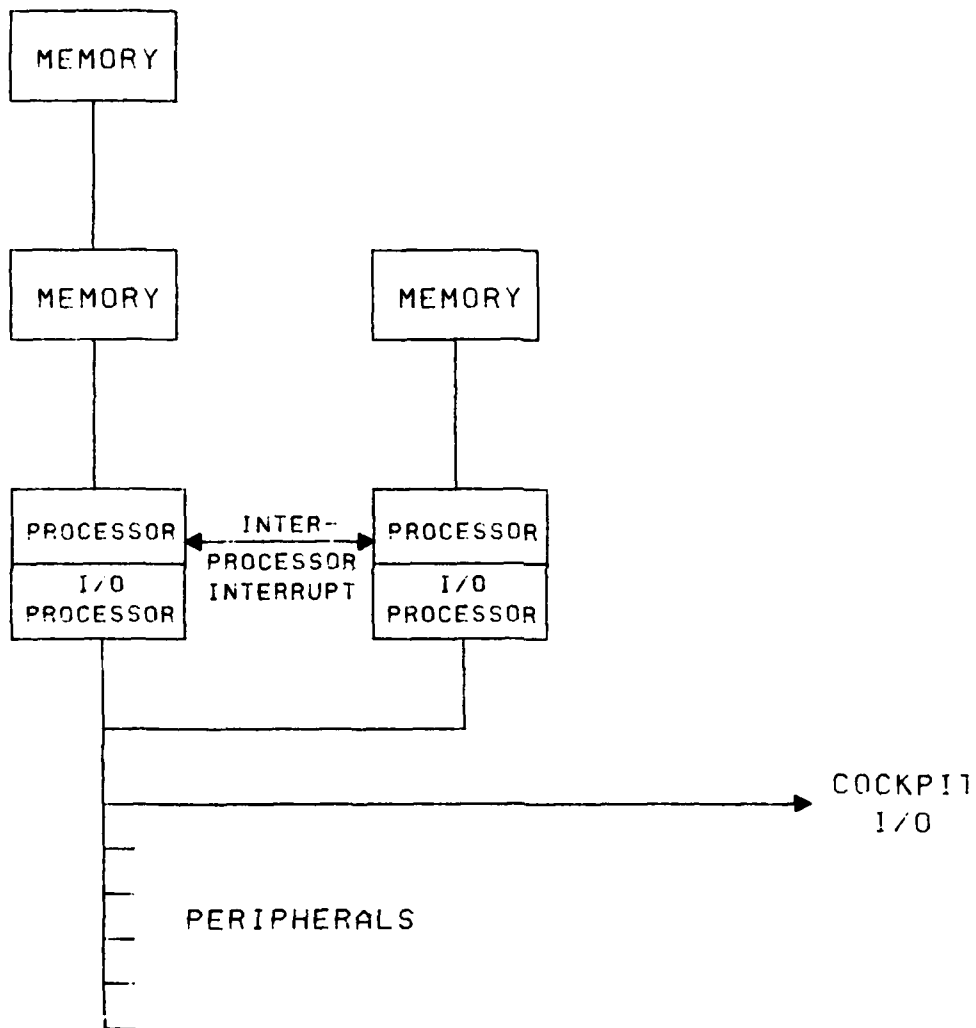


Figure 17. CPU with Peripheral Processor and Dedicated Memory Only.

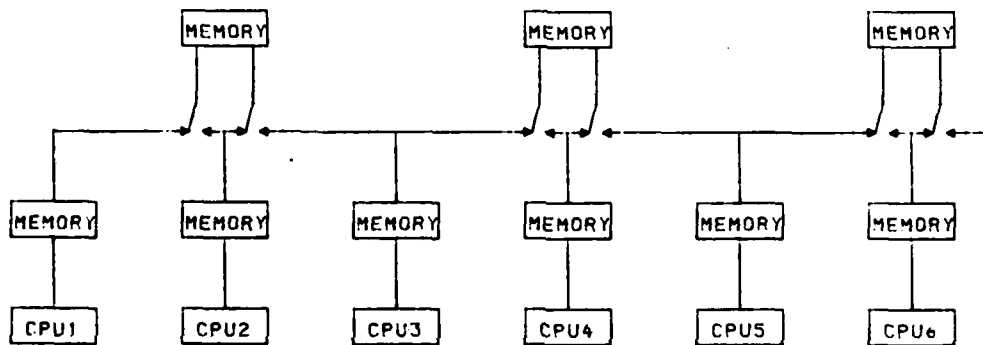


Figure 18. Pairs of CPU's Sharing Switched Memory Bank.

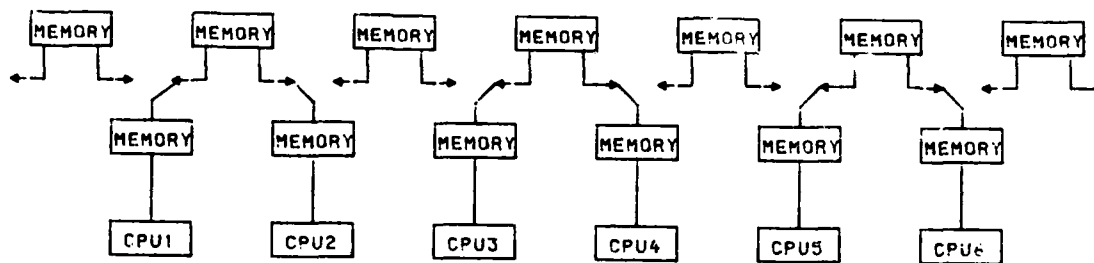


Figure 19. Pairs of CPU's Sharing Switched Memory Bank with Redundant Memory.

Figure 20 shows a memory configuration which has all the memory modules connected all the time. Each of the shared modules would need an enable switch on each port. This is normally provided on each port in a ported memory in order to allow unused ports to be disabled and to isolate failures during troubleshooting. Disabling a memory port essentially disconnects that port from the computer connected to that port. This enable function has very little effect upon reliability, because most memory failures are caused by failures of the memory chips. The sharing would be (1,2), (3,4), (5,6) if the Level 1 memory banks were enabled, and (2,3), (4,5), if the Level 2 banks were enabled. This system works only with an even number of CPU's, and no spare CPU is available.

Figure 21 overcomes the handicap of an even number of CPU's. CPU 11 is a "spare". However, examination of the interconnections will reveal that any adjacent pair of CPU's can share a memory bank, and that any designated CPU can be isolated.

Figures 22 and 23 are equivalents of Figures 18 and 19 with three processors sharing a memory bank instead of two. Figure 24 is the equivalent of Figure 20 again with three processors sharing a memory bank instead of two. Note that in this case it takes 3 levels of shared memory banks to allow the CPU's to be connected together in any configuration. This configuration is for $3N$ CPU's, and has a spare CPU available if $3N-1$ CPU's are required by the simulation.

The configuration of Figure 24 has another interesting property. The CPU's can be connected in either pairs or triples depending on which memory ports are enabled. For example, CPU's 1, 2, and 3 can be connected by enabling the Level 1 memory bank. CPU's 4 and 5 can be connected using the Level 1 memory bank. CPU's 6 and 7 can be connected using the Level 2 memory bank. CPU's 8 and 9 can be connected using the Level 1 memory bank, and CPU's 10 and 11 can be connected using the Level 1 memory bank. CPU 12 becomes a spare in this configuration. Examination of the interconnection diagram will reveal that any CPU can be taken as a spare with the remaining CPU's connected either as pairs or triples.

Figure 25 and 26 are an extension of Figure 24 to $3N+1$ and $3N+2$ CPU's. Once again the CPU's can be connected as pairs or triples in any configuration.

The scheme described in Figures 24, 25, and 26 can be extended to CPU's connected as quads, triples, or pairs, or any higher interconnection level. The only restriction on the level of interconnection is the number of memory parts available. Note that the number of memory banks required is independent of the interconnection level--i.e., computers connected as triples and pairs require no more memory than computers connected as pairs only.

Any of the schemes shown in Figures 24, 25, or 26 would be satisfactory for the VTXTS simulator.

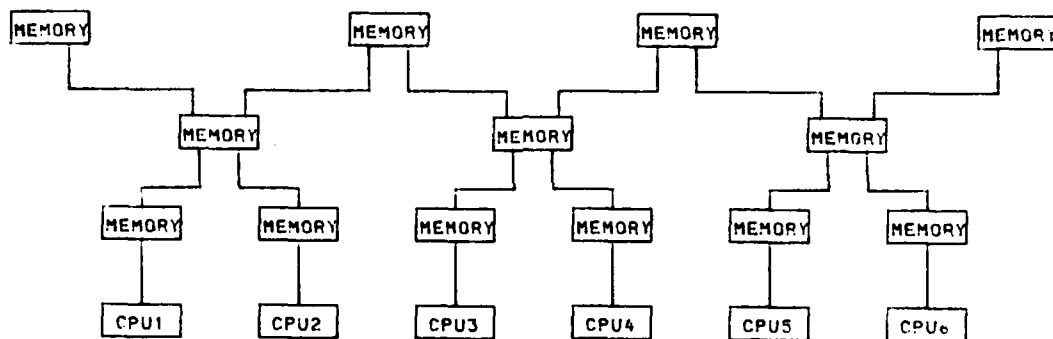
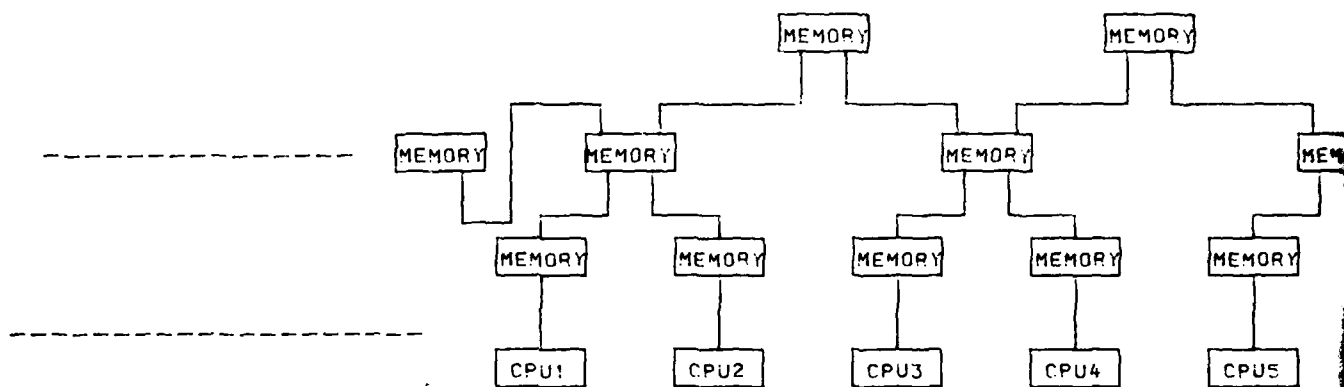
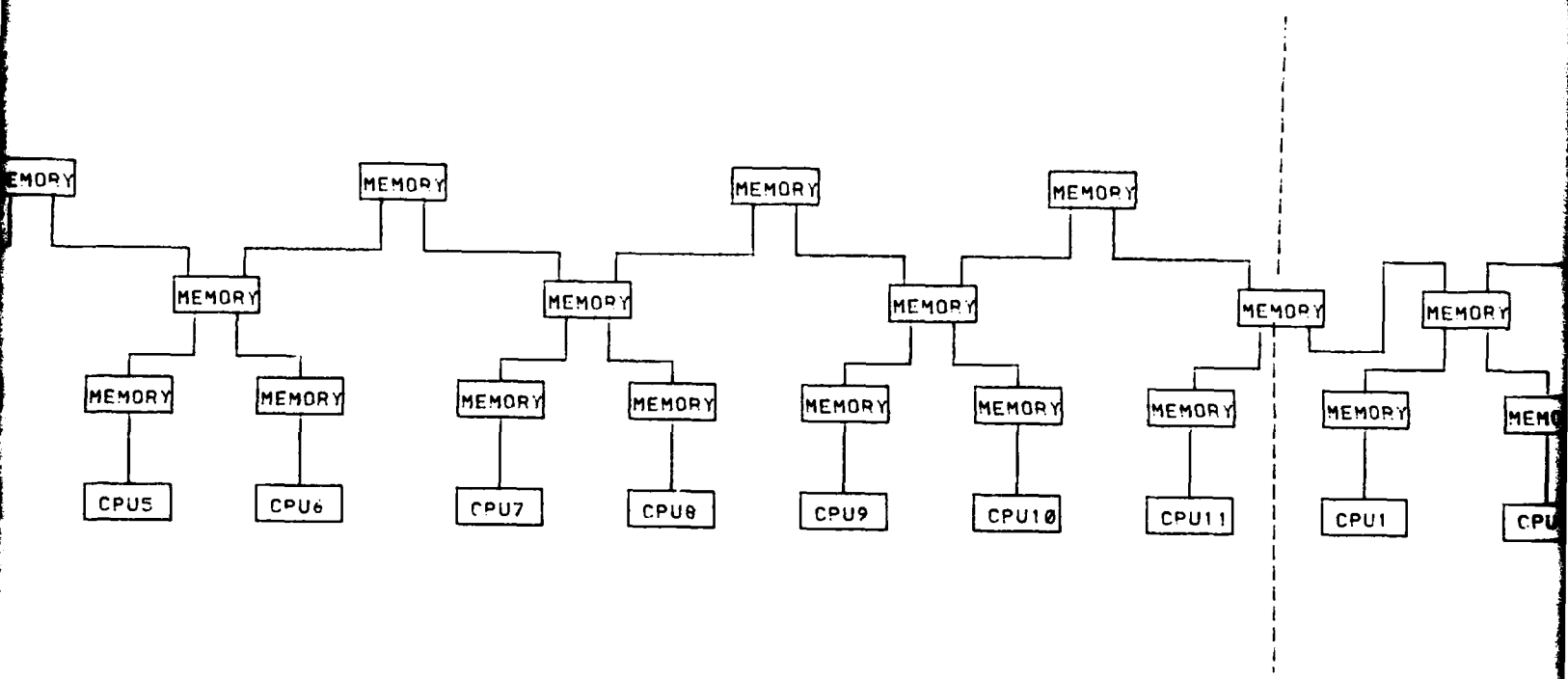


Figure 20. Pairs of CPU's Sharing Wired Memory Bank.





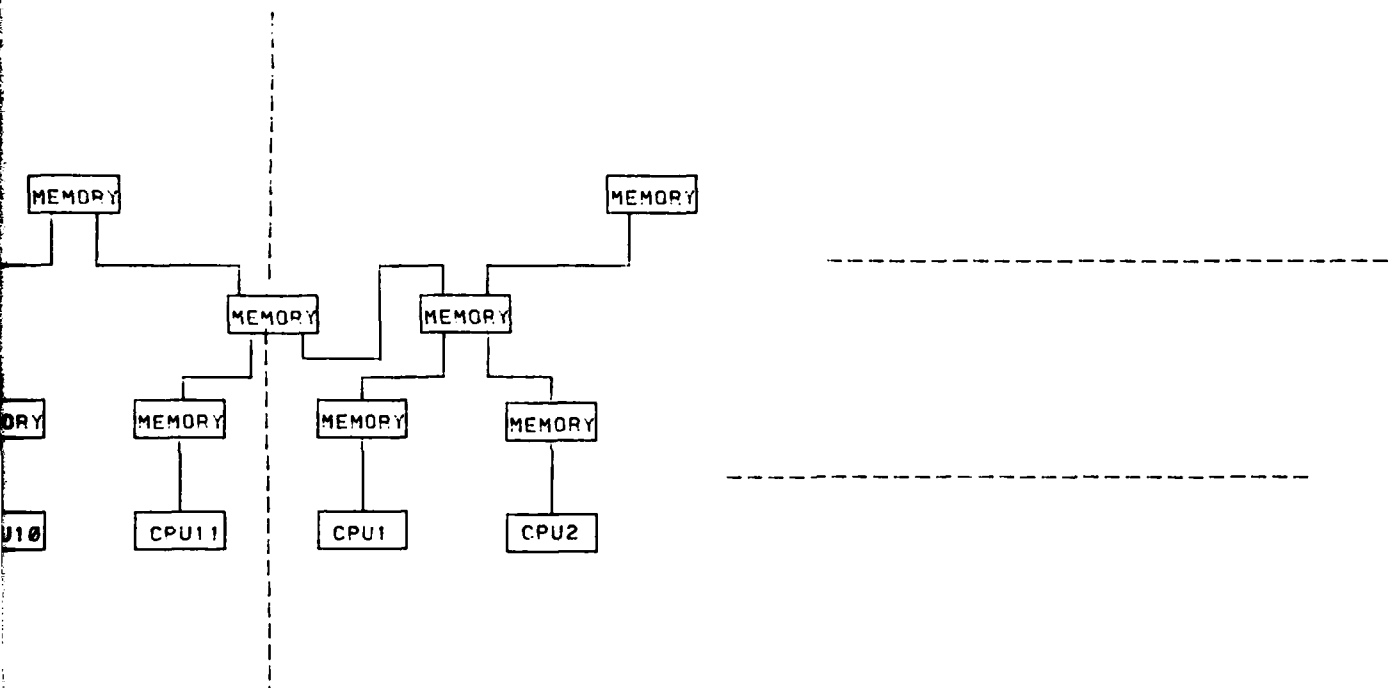
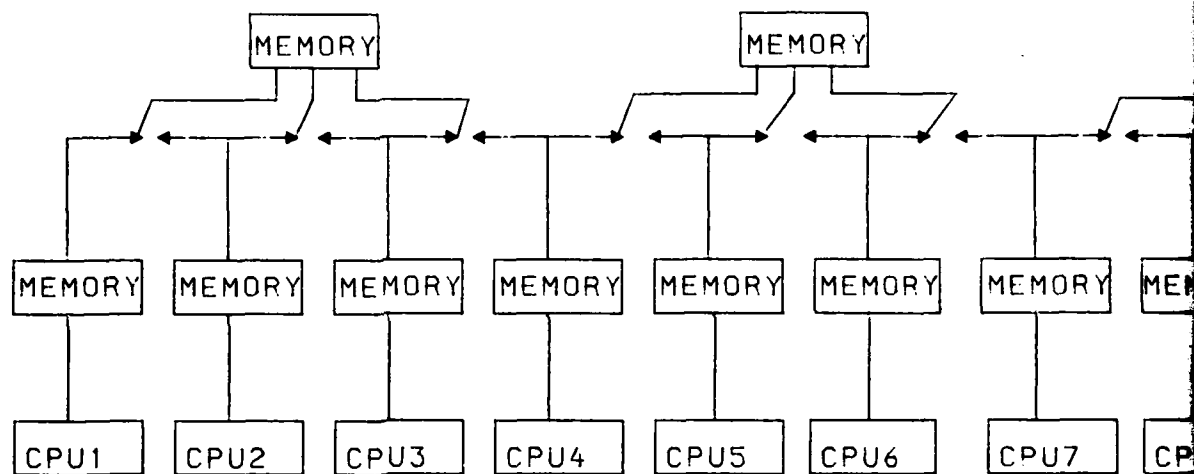


Figure 21. Pairs of CPU's Sharing Wired Memory Bank with Redundant Memory.



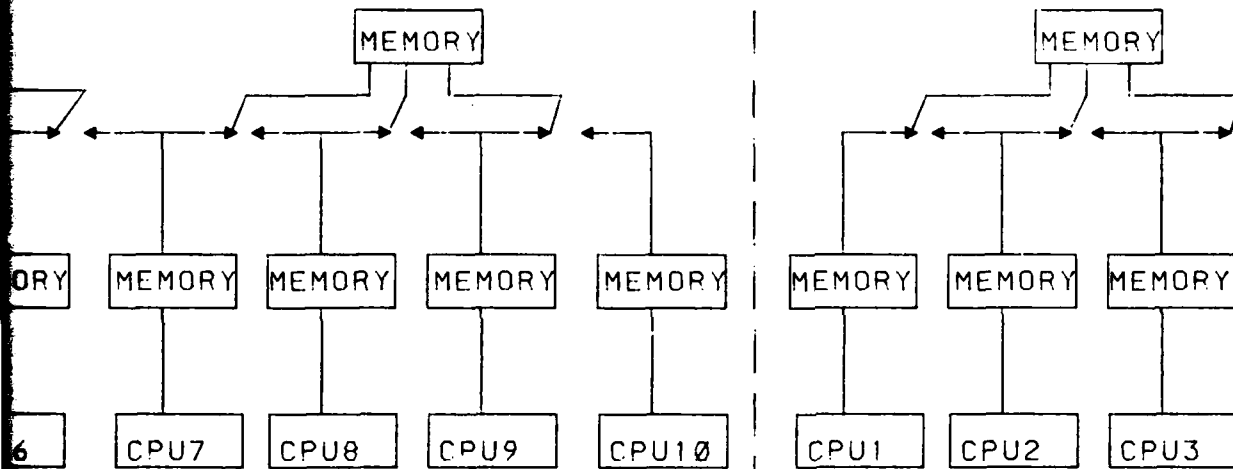


Figure 22. Triple CPU's Sharing Switched Memory Bank.

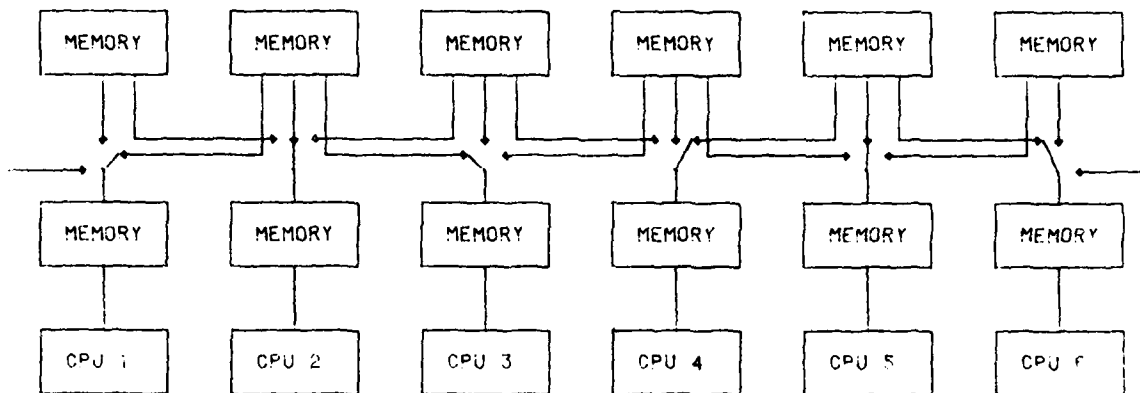
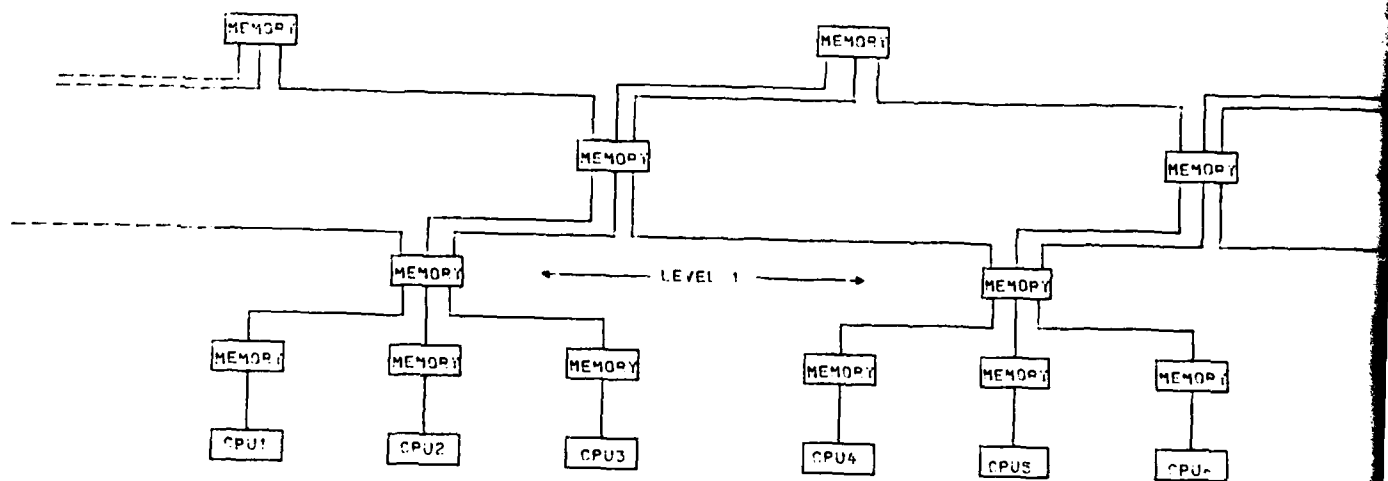


Figure 23. Triple CPU's Sharing Switched Memory Bank with Redundant Memory.



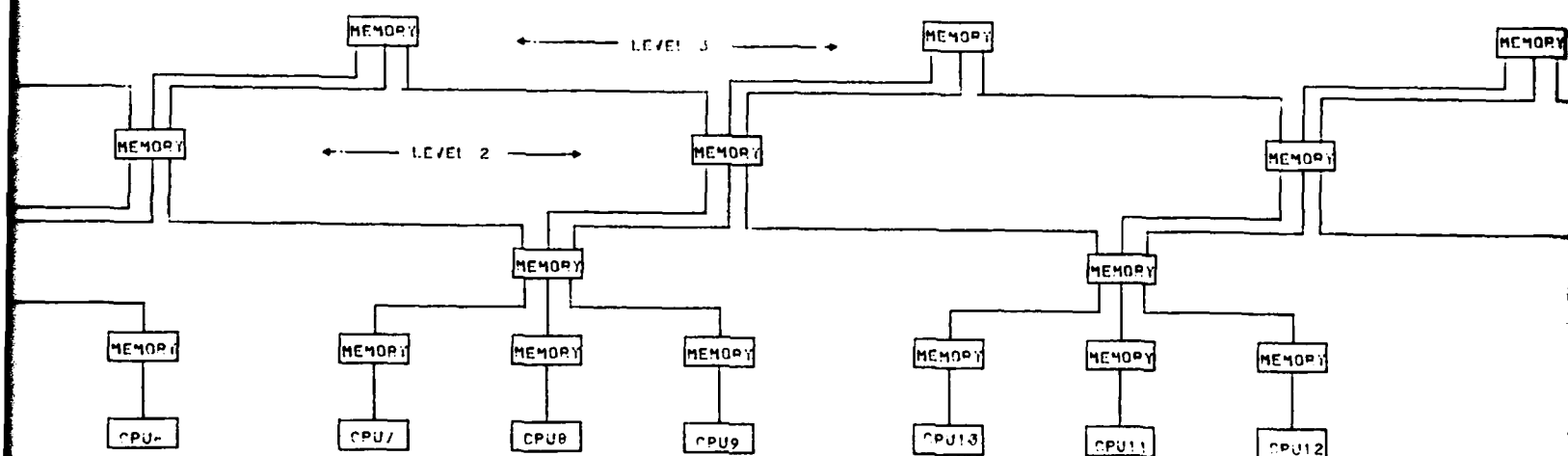


Figure 24. Triple C Memory,

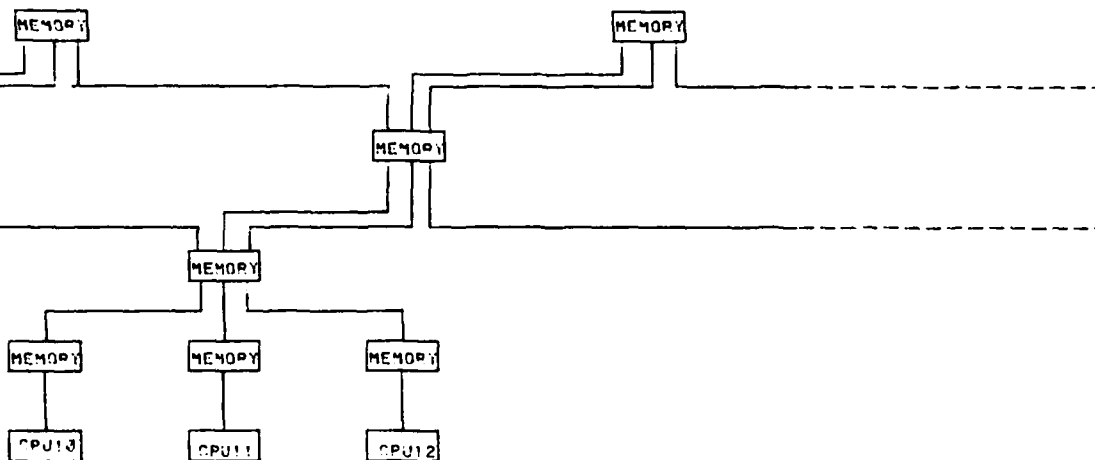
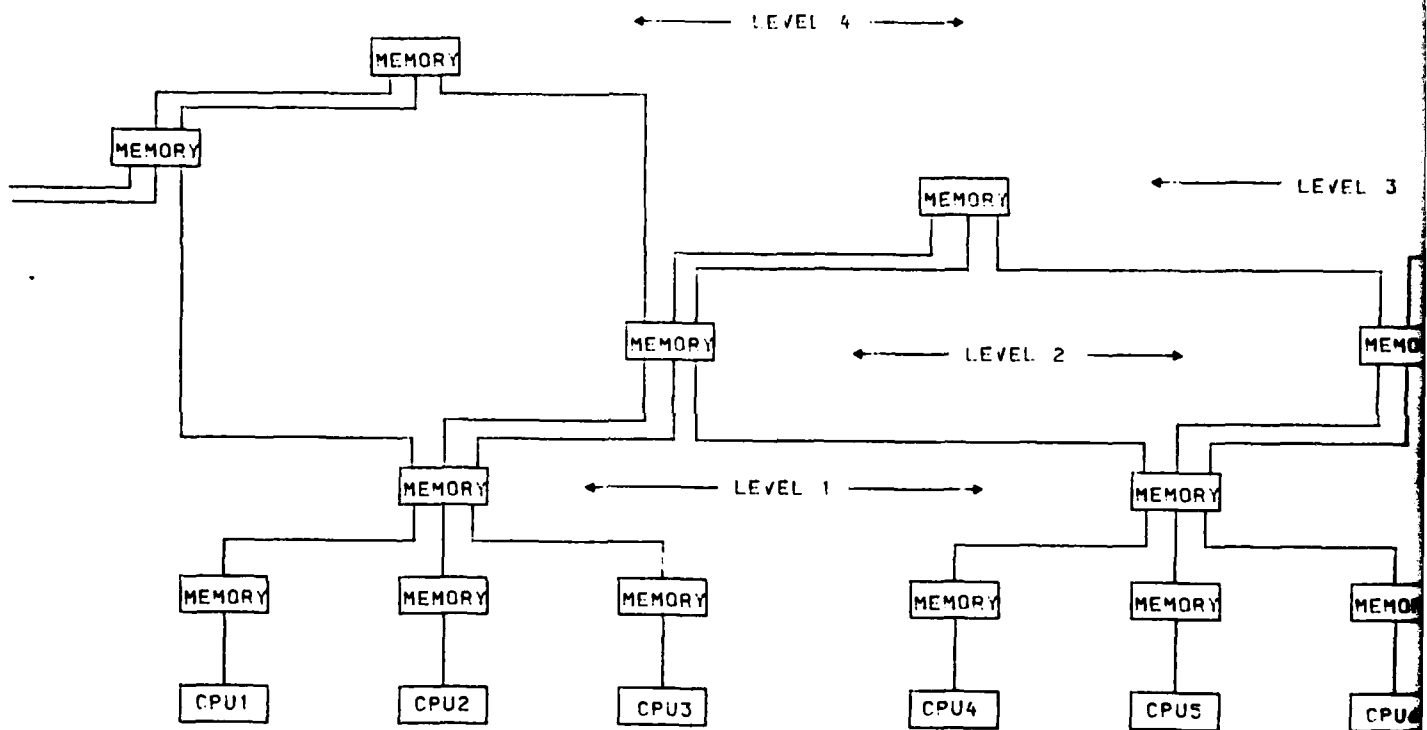


Figure 24. Triple CPU's Sharing Multiple-Level Memory, 3N CPU's.



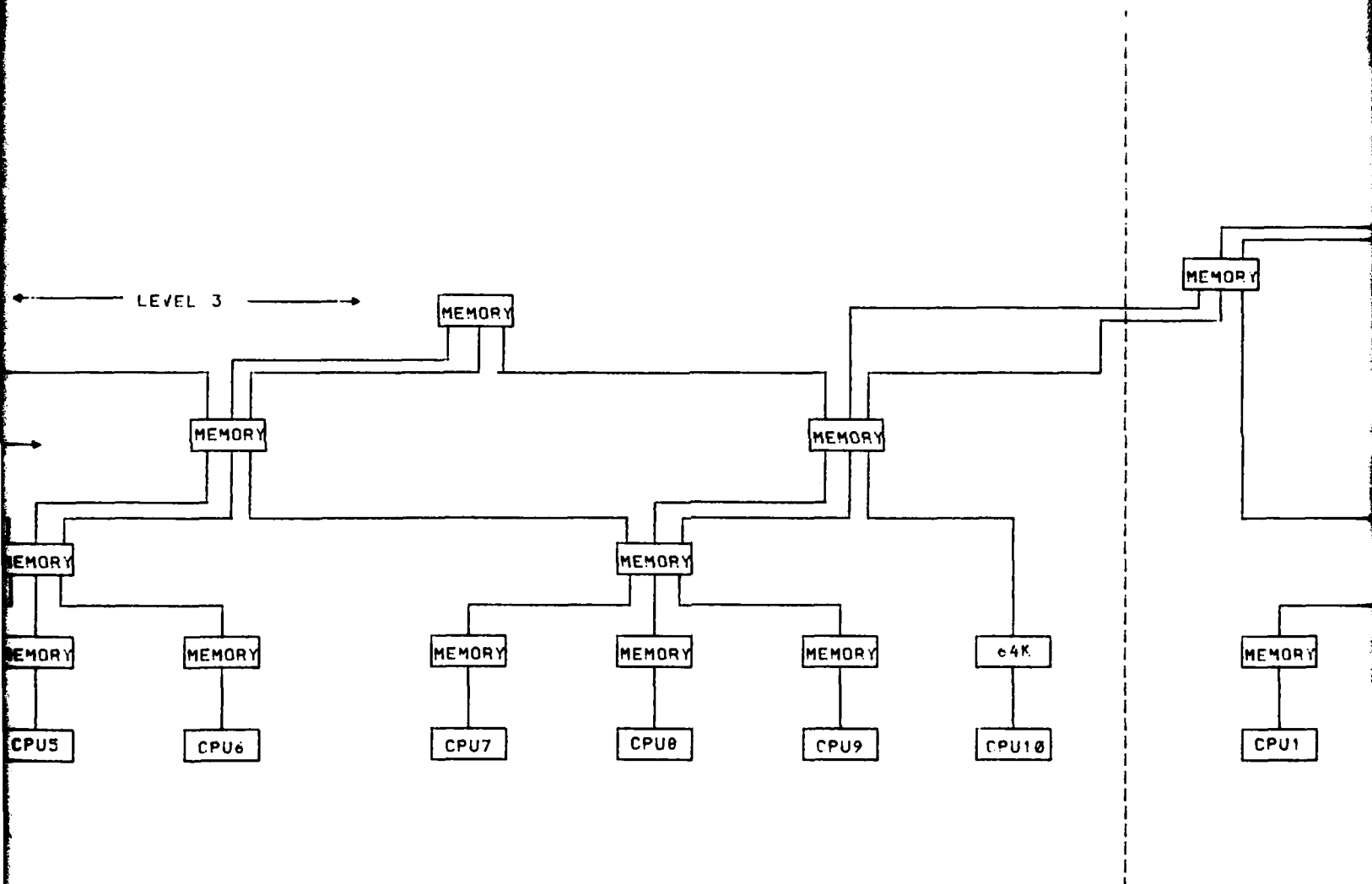


Figure 29

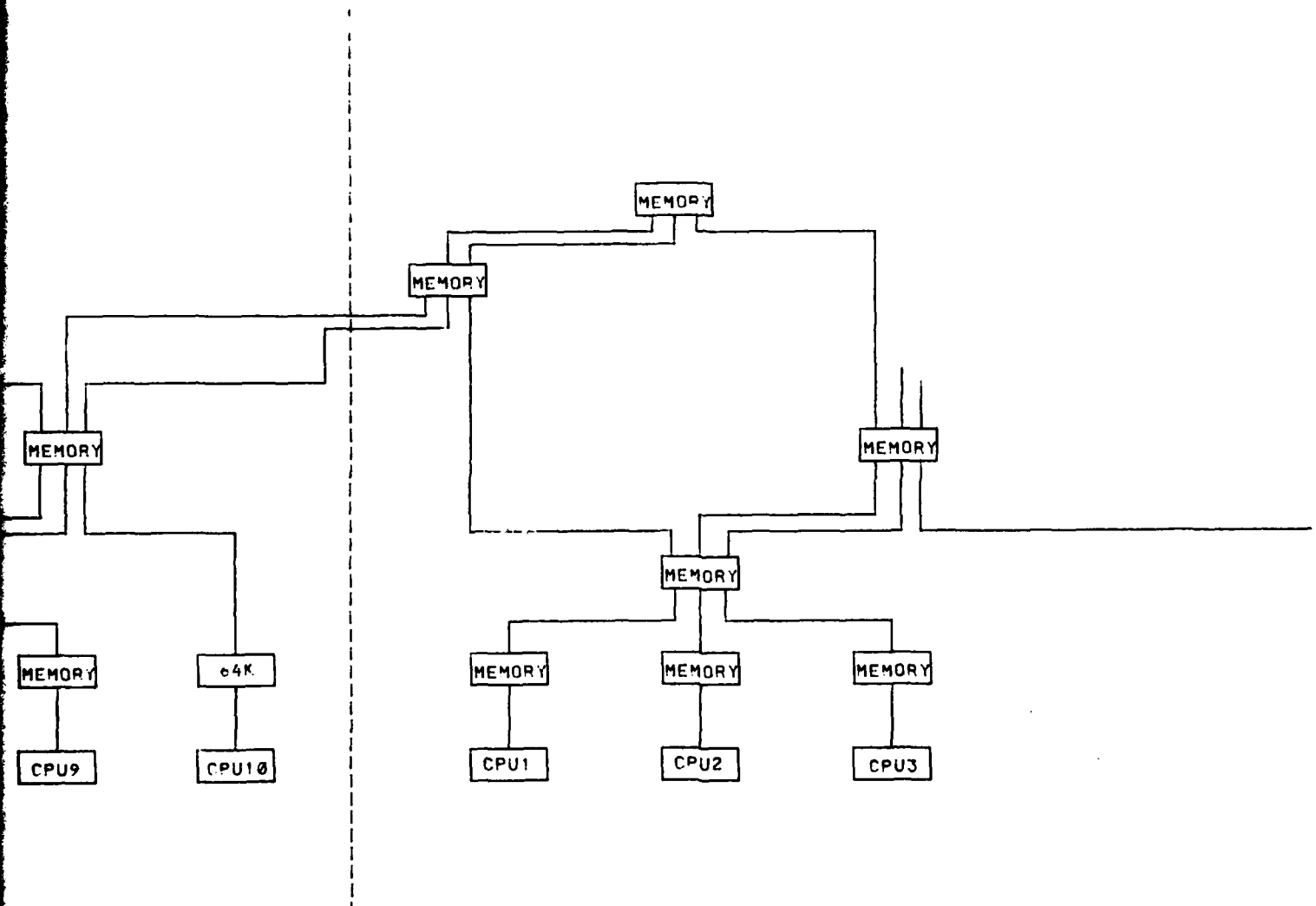
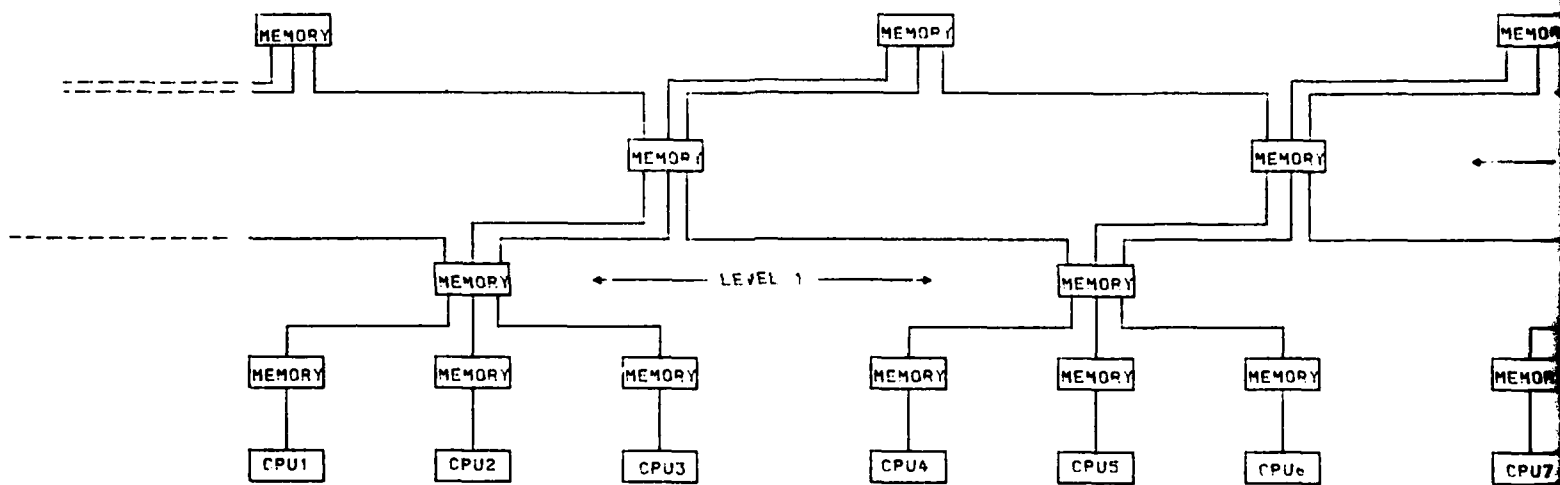


Figure 25. Triple CPU's Sharing Multiple-Level Memory, $3N+1$ CPU's.



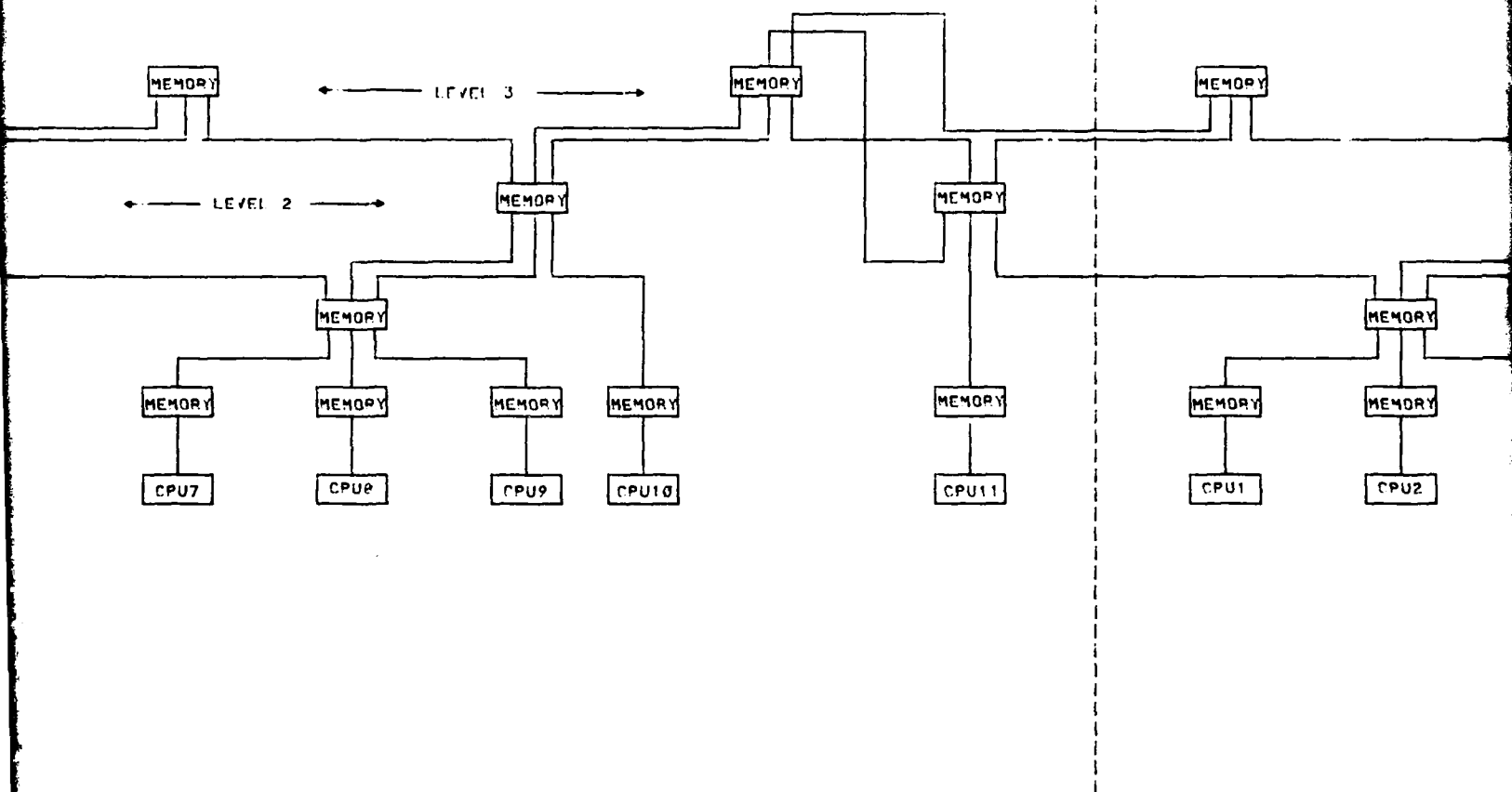


Figure 26. Triple
Memory

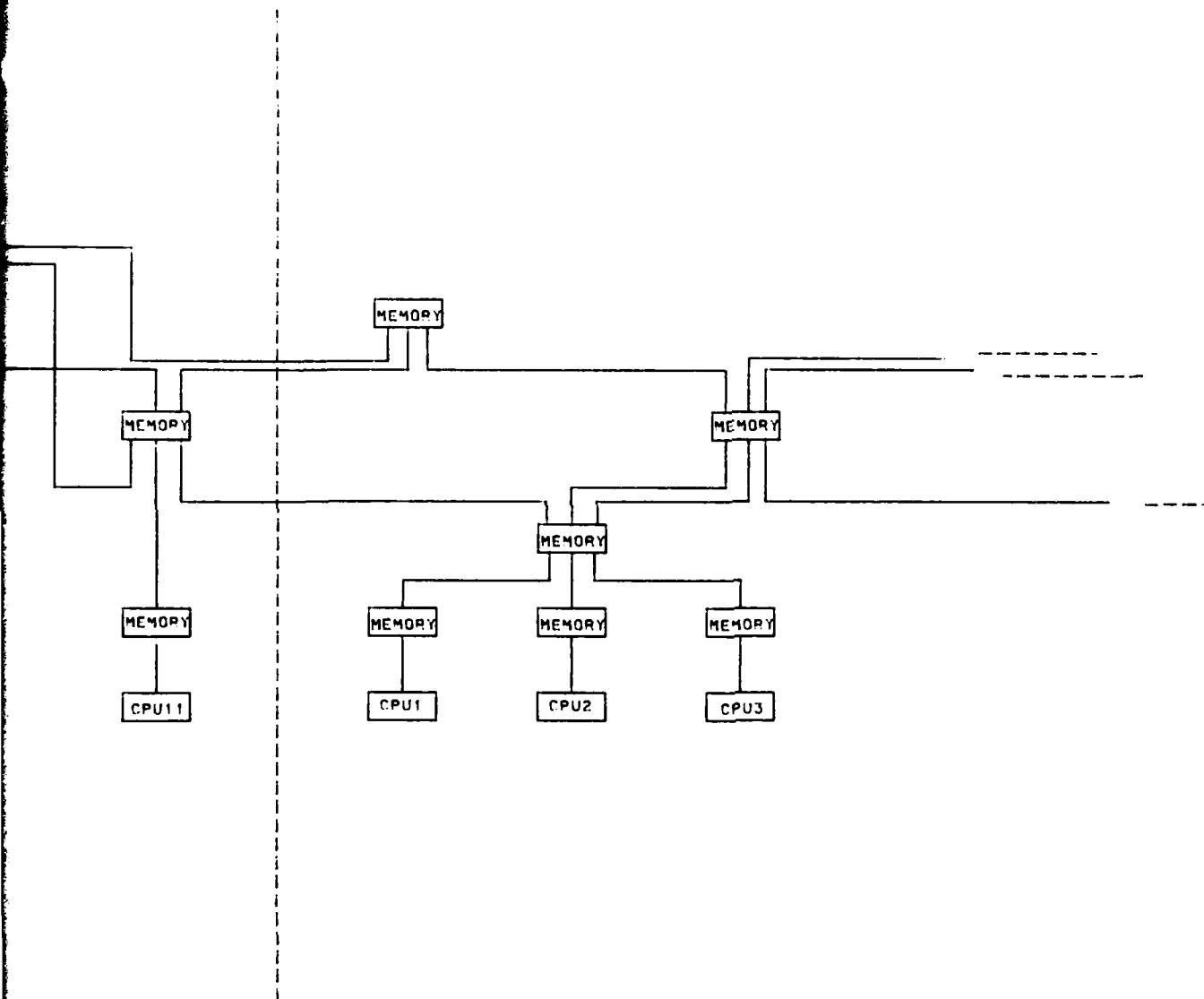


Figure 26. Triple CPU's Sharing Multiple-Level Memory, $3N+2$ CPU's.

INPUT/OUTPUT TO COCKPITS

Figure 27 shows an I/O scheme for multiple CPU's using a common I/O bus. Every CPU would have access to every cockpit by using this bus. The scheme would present a challenge to the designer of the bus interfaces, since determining bus priority without a bus controller may not be possible. Preventing lock-outs without communication between bus interfaces would also be a problem.

Figure 28 is an extension of Figure 27 to a multiple bus configuration. Although this scheme would have the same problems with bus control as the scheme shown in Figure 27 it would avoid delays of the I/O traffic to one cockpit caused by I/O traffic to another cockpit. The bandwidth of the multiple buses is much higher than the bandwidth of a single bus.

Figures 29 and 30 show schemes which use a multiplexer (MUX) to allow each CPU access to the cockpit I/O busses. In these cases the MUX solves the priority problem. It can do this in a straightforward manner since it is aware of all requests for access to the cockpit I/O busses. The MUX should contain a watchdog timer so that a request for a nonexistent device on the cockpit bus will eventually result in an answer back to the CPU. Normally the watchdog timer would send an error flag back to the CPU which had requested an illegal device.

Figures 29 and 30 differ only in the number of busses going to the cockpits. The limitations of a single bus for the cockpit interface have already been discussed.

Figure 31 shows a possible solution to the computer interconnection problem for VTXTS simulator which has 8 OFT's, 2 ACMT's, and 2 CPT's. The interconnection is done so that each cockpit or group of cockpits has access to one more computer than is normally required to perform the simulation. In the design shown, CPU 10 would normally be the redundant CPU. The first group of OFT's would use CPU's 1-3; the second group of OFT's would use CPU's 4-6; the first ACMT would use CPU's 7 and 8; the CPT's would use CPU 9; and the second ACMT would use CPU's 11 and 12. Should a CPU fail--e.g., CPU 5-- then the first group of OFT's would remain unchanged; the second group of OFT's would use CPU's 4, 6, and 7; the first ACMT would use CPU's 8 and 9; the CPT's would use CPU 10; and the second ACMT would use CPU's 11 and 12. CPU 5 would then be available to the maintenance crew for repair.

The MUX's for cockpit I/O are shown symbolically. Depending on speed requirements they could be designed either as shown, or could be designed as shown in Figure 30. This decision should be based on an analysis of the I/O speed requirements.

A configuration could be proposed where all the CPU's are connected to all the cockpits through a large MUX. However, there is no advantage to this arrangement, and the large MUX would, in fact, have a lower availability than the proposed system. Using one large MUX, any failure of the MUX causes the entire system to go down. With the 5 MUX's shown in Figure 31, loss of a MUX only affects the cockpits tied to that MUX. All the other cockpits are still up and running. Use of 5 MUX's in place of one also increases the I/O data rate that the system can handle.

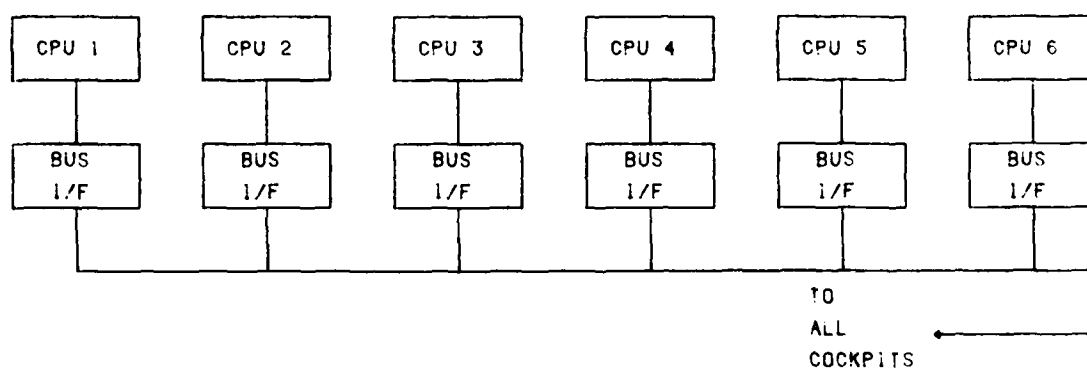


Figure 27. Single Bus Interface for All Cockpits.

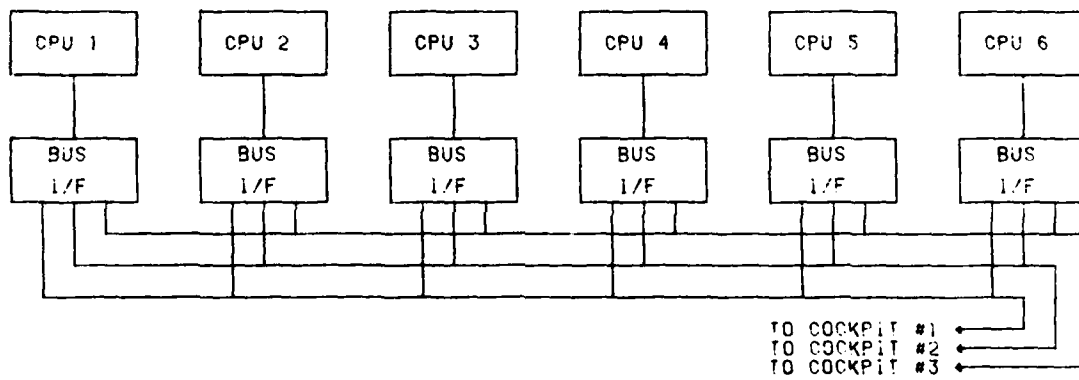


Figure 28. Multiple Bus Interfaces, One for Each Cockpit.

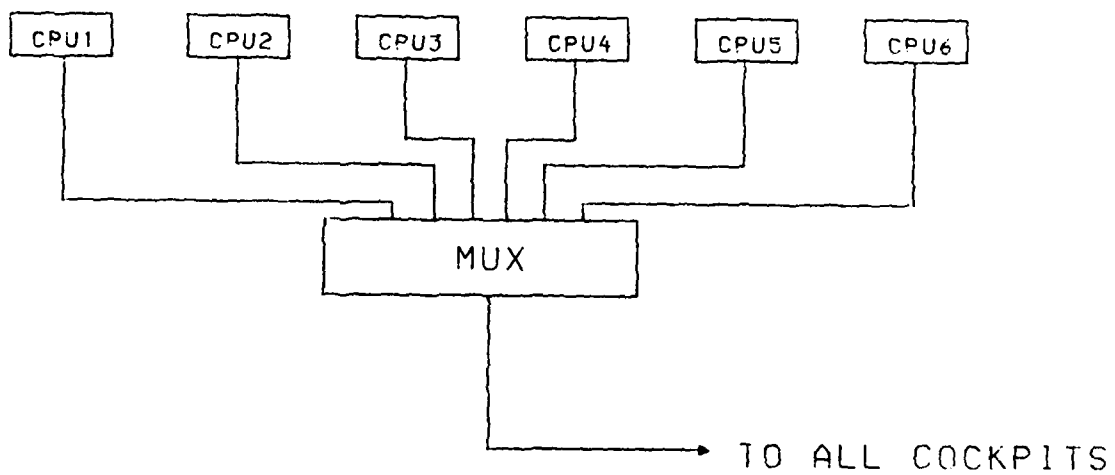


Figure 29. Multiplexer Interface with One Bus for All Cockpits.

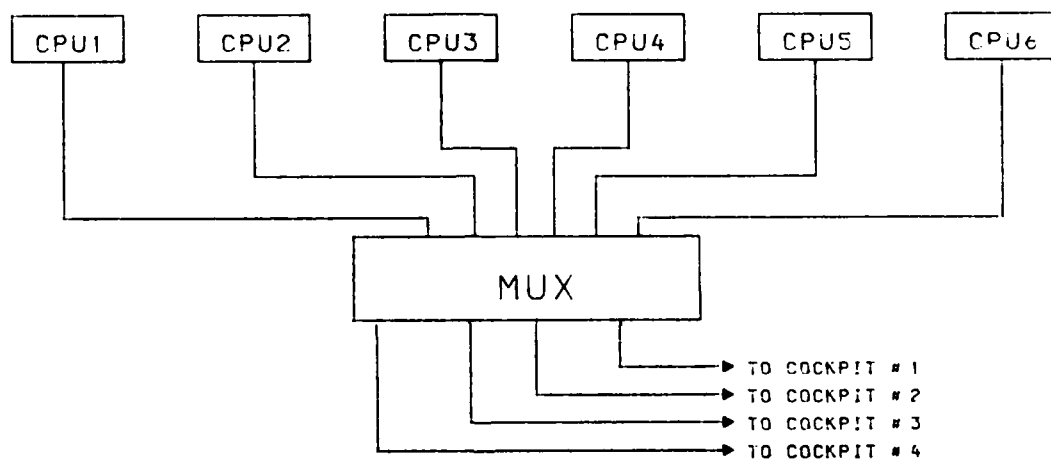
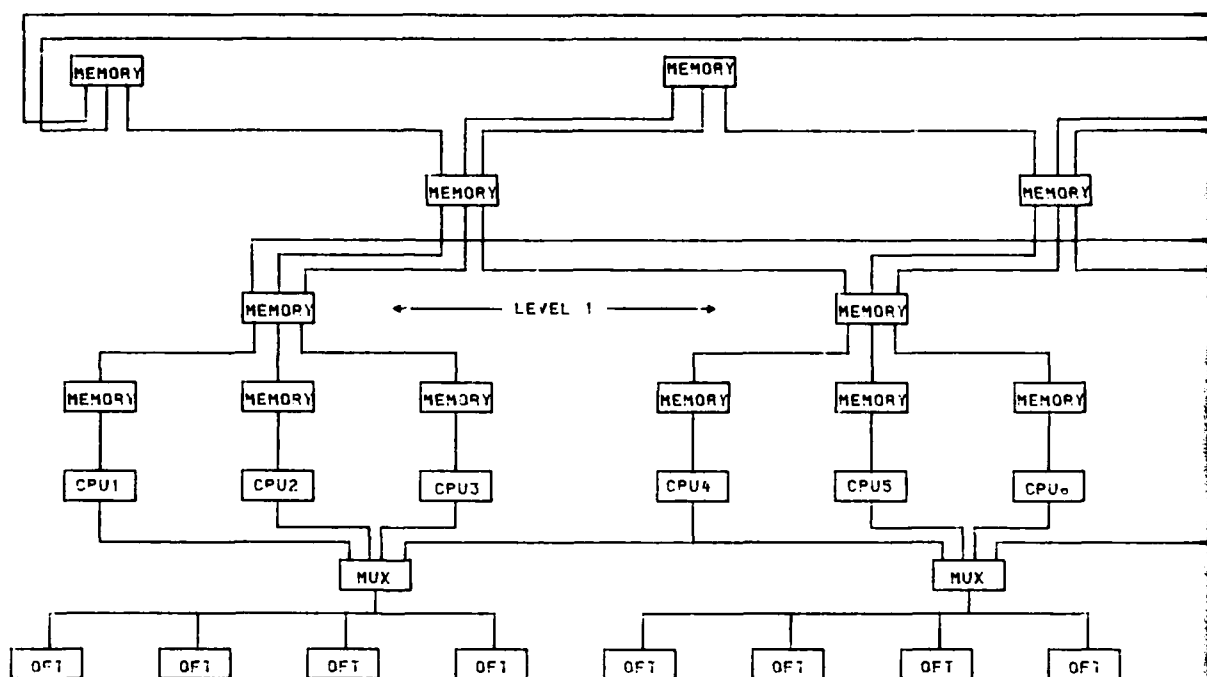


Figure 30. Multiplexer Interface with One Bus for Each Cockpit.



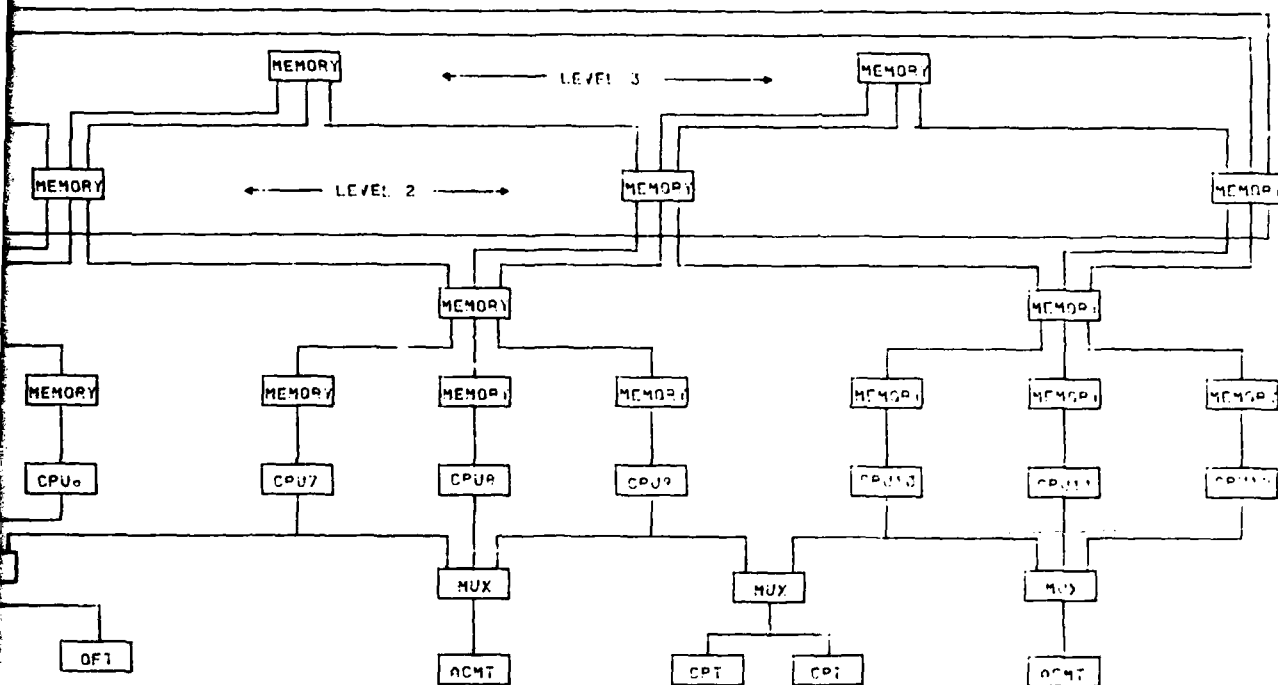


Figure 31. Multiple CPU Configuration for a Multiple Cockpit Trainer.

PERIPHERAL CONFIGURATIONS

Figure 32 shows a minimal peripheral configuration which might be workable for a system installed and working and not undergoing any significant updates. Printing anything from CPU's 3-6 requires writing an output file onto the disk, moving the disk to either CPU 1 or 2, and then printing the output file. Experience with this approach for development on a CPT shows it is definitely not a workable approach during development. Disk packs have to be moved frequently and the availability of only two line printers creates a serious bottleneck.

The lack of a common disk file for the data base is also a serious problem. Use of multiple small disk memories makes it extremely difficult to keep all the programs in all the disks up to date. It is possible to keep the programs up to date, but it can take an incredible amount of time.

Figure 33 is similar to Figure 32 except that each CPU has its own complete set of peripherals. While it avoids the line printer bottleneck problem, it still has the data base management problem.

Figure 34 shows an improvement over Figure 32 by adding a pair of 300 MB disks to hold the data base. Two disks are proposed for redundancy and back-up. This scheme has the added advantage that line printer output can be spooled to the large disk from CPU's 3-6, and then printed off the disk using CPU 1 or 2. This system has a small disk (a single cartridge disk) on each CPU for the operating system as was shown in Figure 15.

Figure 35 is a modification of Figure 34 with a line printer added to each CPU. This is the most desirable configuration for development, since it reduces to an absolute minimum the interaction between computer during program development.

REDUNDANT SYSTEMS

An investigation into the lost time due to computer failures reveals that a significant improvement in simulator availability can be achieved by adding some redundancy to the computer system. Several approaches to interconnecting computers so that a failed computer can be switched out of the system and the system reconfigured using a spare computer have already been described. The improvement in availability achieved by redundancy is shown in the next section.

Some further comments on the desirability of adding a redundant computer are in order. Anything that improves the simulator availability improves the image of the simulator as a training tool, particularly in the eyes of the pilots and their commanders. Pilot's time is generally at a premium, and a commander does not like his pilots to report to him that the training session in the simulator was a waste of time due to hardware malfunctions.

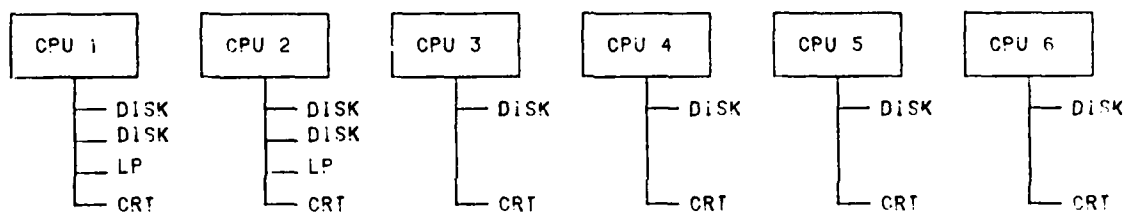


Figure 32. Minimum Peripheral Configuration.

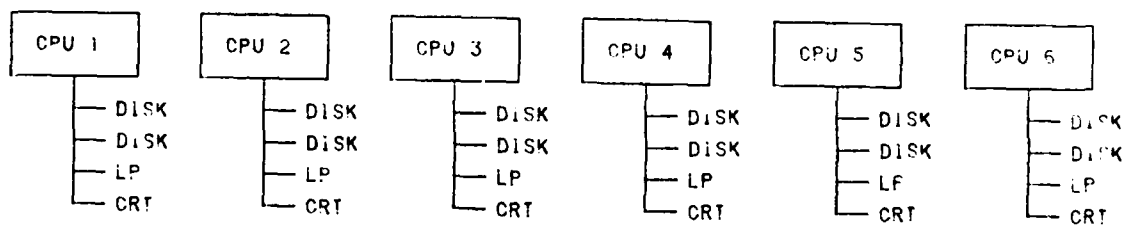


Figure 33. Configuration with No Shared Peripherals.

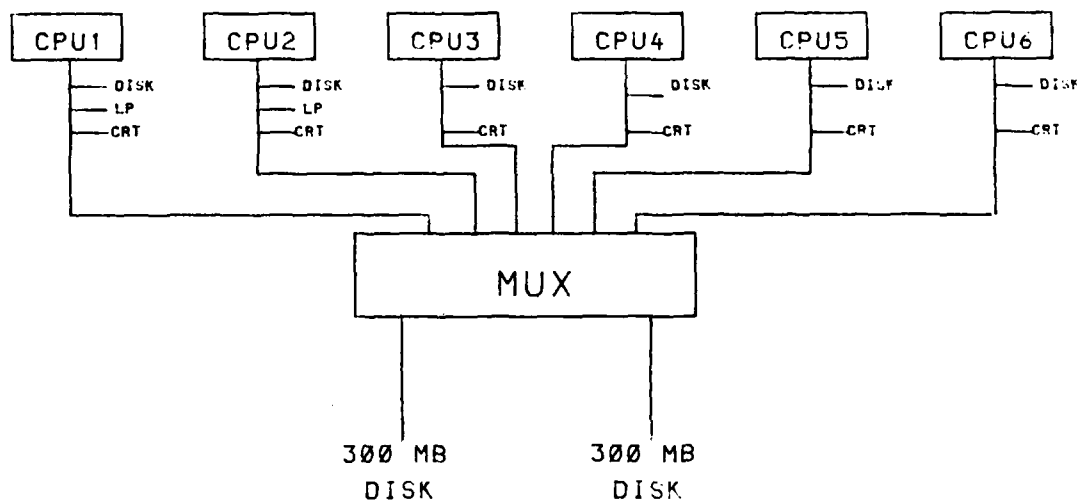


Figure 34. Shared Large Disk with Minimum Peripherals.

NAVTRAEQUIPCEN IH-336

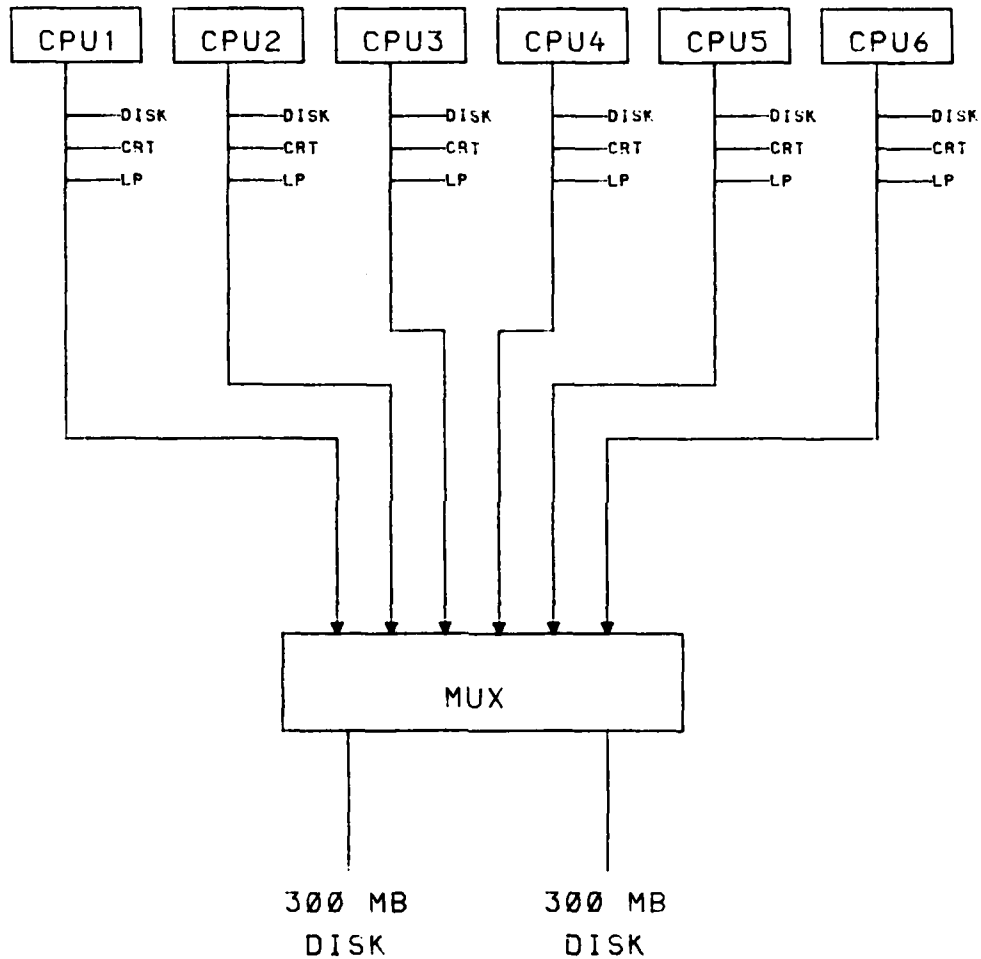


Figure 35. Shared Large Disk with Same Peripherals on Each CPU.

Adding a redundant computer will reduce the time required to diagnose problems. The quickest way to isolate a failure is by substitution. If a malfunction is detected and believed to be caused by a computer, substitution of another computer will quickly reveal whether or not the suspect computer has failed. Alternatives to this are normally the running of diagnostic programs (which are not always reliable in indicating failures).

Having a computer off-line for repair should result in more thorough testing after the repair since there will be no pressure on the service personnel to have the machine back on-line. Being able to troubleshoot the computer off-line should further reduce costs, since competent computer service personnel might not be required to cover all the simulator operation time.

SUPER MINICOMPUTERS

Since only two of these Super Minicomputers are required to perform all the computations for the simulators, one might conclude that no redundant computer is required. Thirteen conventional minicomputers will certainly have a higher failure rate than 2 superfast minicomputers. However, when one of the superfast minicomputers is down, more cockpits are affected. The result is that the cost of lost time for a simulator using 2 superfast minicomputers is the same as that for a simulator using 13 conventional minicomputers. The cost analysis in the next section applies to the super-fast minicomputer case as well as the conventional minicomputer case.

In Figure 36, no shared memory is shown. Since each computer has programs which are entirely self-contained, no communication is required between processors. This should minimize software costs since no multi-CPU operating system is required for this configuration.

NAVTRAEQUIPCEN IH-336

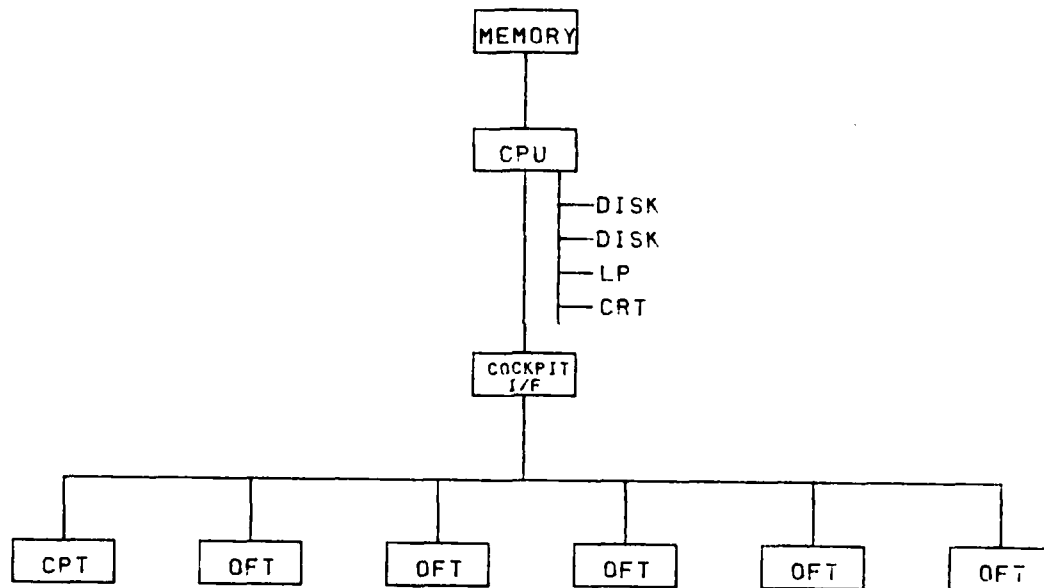


Figure 36. Super Minicomputer Configuration.

SECTION VI

EVALUATION OF ARCHITECTURES

The configurations introduced in Section IV show some of the choices in implementing a VTXTS complex. These will be analyzed to determine the effect upon: (1) availability, (2) maintenance, (3) expandability and (4) life cycle costs.

AVAILABILITY

The inherent availability of a system is the probability that the system, when properly used and adequately maintained, will operate satisfactorily at any point in time. This definition excludes preventative maintenance actions, logistics supply time and administrative downtime. The inherent availability is expressed as:

$$A_i = \frac{MTBF}{MTBF + MTTR}$$

where MTBF is the mean time between failures for the system considered, and MTTR is the mean time to repair a failure.

Availability of the simulators for training is a major consideration in the system design. For this analysis each simulator is assumed to consist of a cockpit, an instructor station and that portion of the total computer system required to make it work. The reliability and maintainability numbers used in this analysis are given in Table 12.

If the ratio MTTR/MTBF is small (0.02 or less), the availability to three decimal places can be computed by the relation:

$$A_i = 1 - \sum_i (\text{Failure Rate})_i (MTTR)_i$$

The expression above was used to compute the inherent availability of each type of trainer for each of the eleven configurations considered. The results are shown in Table 13.

NAVTRAEQUIPCEN IH-336

TABLE 12. RELIABILITY AND MAINTAINABILITY VALUES USED IN ANALYSIS

ELEMENT	FAILURES PER 1000 HOURS	MTBF (Hours)	MTTR (Hours)
Cockpit	2	500	1
Instructor Station	.5	2,000	1
Large Computer	20	50	2
Large Computer (with redundancy)	20	50	.25
Super Minicomputer	8	125	2
Super Minicomputer (with redundancy)	8	125	.25
Minicomputer	4	250	1.50
Minicomputer (with redundancy)	4	250	.25
Microcomputer	1	1000	.5

NAVTRAEQUIPCEN IH-336

TABLE 13. SIMULATOR AVAILABILITY

CONFIGURATION	AVAILABILITY		
	CPT	OFT	ACMT
Separate Computer Systems Each Cockpit	.9915	.9915	.9855
Separate Computer System Each Cockpit (W/Redun)	.9965	.9965	.9955
Shared Computers for Each Trainer Class	.9915	.9795	.9855
Shared Computers for Each Trainer Class (W/Redun)	.9965	.9945	.9955
Shared Computers with Min. Hardware	.9915	.9915	.9855
Shared Computers with Min. Hardware (with redundancy)	.9965	.9965	.9955
Super Mini-Computers	.9815	.9815	.9815
Super Mini-Computers (with redundancy)	.9955	.9955	.9955
Single Large Computer	.9575	.9575	.9575
Single Large Computer (with redundancy)	.9925	.9925	.9925
Microcomputers	.9970	.9960	.9945

MAINTENANCE

Maintenance of the computer system includes both hardware and software. At the level of this analysis, the distinction between contract and in-house support in either area cannot be made. Such a determination will require further study after a computer configuration is determined. A hardware cost estimate of one per cent of the initial cost of the equipment per month is used for this analysis. This estimate is derived from a study of maintenance contract charges for several vendors.

An exception is made in the case of the microprocessor system, for which vendor maintenance is not available. The cost of maintenance for the microcomputer system (except for the peripherals) is estimated at two per cent of the hardware cost. Peripheral maintenance is estimated at one per cent, just as in the other configurations.

The cost of this software support is estimated to be ten per cent of the initial software cost per year. This estimate is based loosely on the work of Putnam and Wolverton[11], which finds that support costs are 150 per cent of the acquisition cost for most software systems. For the simulator application, which should require less support than a data processing application, this cost factor is estimated to be less than the normal amount. Therefore, the software estimate used in this analysis is 100 per cent of the acquisition cost, or ten per cent of the acquisition cost per year.

EXPANDABILITY

Expandability will be considered in three parts: (1) an increase in the capability of each individual simulator, (2) the addition of one complete cockpit to the system, and (3) the addition of two cockpits to the system.

MINICOMPUTERS. A reasonable increase in capability for each cockpit requires no change in the minicomputer implementations considered, because a spare capability is built into the computer requirements. An increase that exceeds the spare capability provided presents a severe problem. In the implementation shown in Figure 2, such an increase can lead to doubling the number of computers required. Those configurations that use multiple computers to drive multiple cockpits are much less sensitive to increases in requirements. In a multiple computer configuration, the entire spare capacity of the computer system can be used for a single change. If a change does require addition of a computer, the entire spare capacity added is available to all simulators as needed.

An increase in the number of cockpits is a simple change for the configuration shown in Figure 2. An added cockpit requires the addition of a new cockpit and an associated computer, an addition that is totally independent of the existing system. Addition of another cockpit is the same. For multiple computer configurations, the addition of a cockpit can be

[11] Putnam, Lawrence H., and Ray W. Wolverton, Quantative Management: Software Cost Estimating. A tutorial for COMSAC '77, The IEEE Computer Society's First International Computer Software and Applications Conference, Chicago, IL, 8-10 Nov 1977.

achieved by adding another computer to the network and making the appropriate programming changes. The major difference is that the total system software must change and that some lost time will be encountered in introducing a new system on a shared computer complex.

LARGE COMPUTER. For the system using a single large computer, any addition up to the capacity of the computer is a relatively simple change. Since all spare capacity resides in one computer, a change may make use of the entire spare capability. When an addition exceeds the capacity of the computer, a serious problem is encountered. A choice must be made to add a second very expensive large computer or to add a stand-alone minicomputer. The use of a minicomputer may reduce the cost of the addition, but it adds to the maintenance problem by introducing a different type of computer with associated problems in spares parts and in training requirements. It complicates the software maintenance problem, even though the computer program is written in a high-level language. The software personnel must learn a new operating system and I/O handlers and the idiosyncrasies of the high-level language compiler for the new machine.

SUPER-MINICOMPUTER. The expandability for the super-minicomputer configuration is very much like the situation with a large computer. A single computer handles several cockpits, making an addition in the computation load up to half the spare capacity of the system relatively easy. However, any expansion greater than that requires addition of new, relatively large computer.

MICROPROCESSOR. The microcomputer implementation offers the easiest expansion. The modular nature of the system allows any change to be made with a minimum of additional computers.

LIFE CYCLE COST

The life cycle cost model considers only those elements associated with the computer configuration. The items used in the simplified life cycle cost model are derived below.

HARDWARE COST. Hardware costs used in the cost model are shown in Table 14. The computer prices are based upon nominal prices for each class of computer with memory and peripherals. These costs are based on list prices contained in a standard reference[4] and on information supplied by various vendors. Costs for an actual system might vary considerably because of Original Equipment Manufacturer (OEM) agreements that a simulator vendor might have with the computer manufacturer. However, the costs are considered reasonable for the purpose of comparing the different configurations.

Table 15 shows the life cycle cost for each major item for each configuration considered. The hardware cost is taken directly from Table 14, with the following exceptions:

- a. Cost for a second large computer is reduced by 20 per cent, because a complete set of peripherals is not required.

[4] GML Corporation, ob. cit.

AD-A115 006

NAVAL TRAINING EQUIPMENT CENTER ORLANDO FL
COMPUTER ARCHITECTURE STUDY FOR VTXTS SIMULATORS.(U)
APR 82 L D HEALY, G A WYNDE, B BAKER
NAVTRAEQUIPC-IH-336

F/G 9/2

UNCLASSIFIED

NL

2 of 2

3 of 3



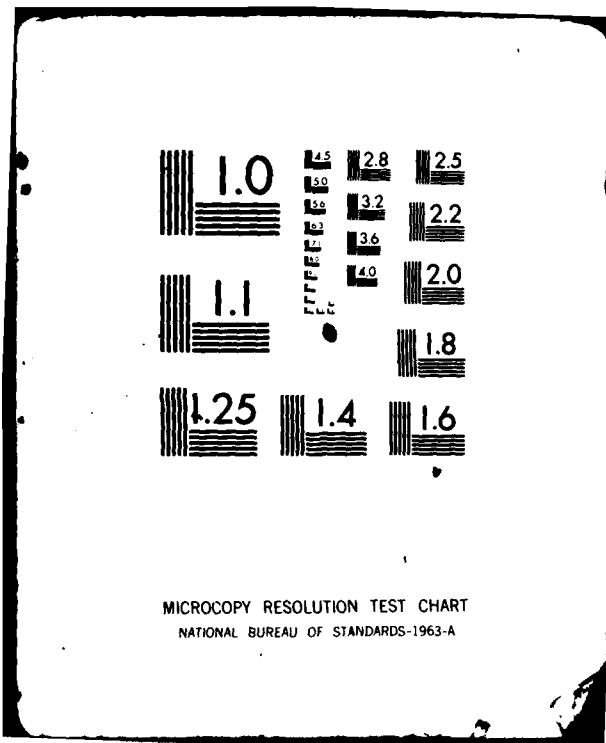
END

DATE

FILED

7-82

DTIC



NAVTRAEQUIPCEN IH-336

TABLE 14. HARDWARE COST ESTIMATES USED IN ANALYSIS

COMPUTER TYPE	COST (1000'S)
Microcomputer	4
Minicomputer	140
Super minicomputer	350
Large Computer	2,350
Memory Bank	50

TABLE 15. LIFE CYCLE COST OF VARIOUS COMPUTER CONFIGURATIONS

CONFIGURATION	COST (\$1000)				
	HARDWARE	SOFTWARE	ANNUAL	ANNUAL	10 YEAR
	ACQUI-	ACQUI-	SOFTWARE	HARDWARE	TOTAL
	SITION	SITION	SUPPORT	MAINT.	3 SYSTEMS
Separate Computer Systems	1,920	2,297	230	230	17,266
Each Cockpit					
Separate Computer System	2,060	2,297	230	247	18,190
Each Cockpit (W/Redun)					
Shared Computers for Each	1,740	2,614	261	209	16,712
Trainer Class					
Shared Computers for Each	1,880	2,614	261	226	17,636
Trainer Class (W/Redun)					
Shared Computers with Min.	1,370	2,545	255	164	14,132
Hardware					
Shared Computers with Min.	1,510	2,545	255	181	15,056
Hardware (with redundancy)					
Separate Computers for All	700	2,070	207	84	8,760
Separate Computers for All	1,050	2,070	207	126	11,070
(with redundancy)					
Single Large Computer	2,300	2,070	207	276	19,320
Single Large Computer	4,140	2,070	207	497	31,464
(with redundancy)					
Microcomputers	592	3,737	374	89	9,517

NAVTRAEQUIPCEN IH-336

- b. Cost of a set of peripherals is added to each microcomputer system. This cost is estimated to be \$40K per cockpit.

SOFTWARE COSTS. Software costs are more difficult to estimate. The basic software cost is estimated by determining the program size and applying a cost factor based only on the number of instructions. The software cost for each different configuration is determined by modifying this basic software cost to account for any additional complexity in the software due to partitioning the program to be executed so that it can be run in a multiprocessor environment.

The following assumptions are made in determining the basic software cost for each type of trainer:

- a. FORTRAN is the source language.
- b. The FORTRAN expansion ratio is 4 machine instructions/FORTRAN statement.
- c. Programmers will program at a rate of 2000 statements/year including design, code, and checkout.
- d. Labor costs are \$90,000 per man-year.
- e. A CPT has a 24K word program or 6K FORTRAN statements.
- f. An OFT has a 64K word program or 16K FORTRAN statements.
- g. An ACMT has a 96K word program or 24K FORTRAN statements.

The basic software cost computed using these assumptions is given in Table 16. This gives the cost of programming for each type of simulator when the program is executed entirely within one computer.

Many of the configurations considered require some partitioning of the problem to allow it to be solved in a multiprocessor system. The following assumptions are made about the increase in software complexity due to partitioning:

- a. The program size increases by ten percent for each state computer system over which the simulator software is distributed.
- b. The software development effort for a partitioned program grows as the square of the increase in program size.

The first assumption is an estimate of the growth in program size due to increased need for control and communication between program modules. The second is based upon the added complexity of the program because of the partitioning. The penalty for partitioning may appear to conflict with the advantages generally attributed to modular programming. However, recall that the partitioning is not simply dividing the computation into manageable parts as done in modular programming. The partitioning of the simulation program to allow sharing among several computers requires concurrent program execution and the use of shared storage.

NAVTRAEQUIPCEN IH-336

TABLE 16. SOFTWARE COST ESTIMATES USED IN ANALYSIS

TYPE	EFFORT	COST (1000'S)
CPT	3 MY	270
OFT	8 MY	720
ACMT	12 MY	1080

The method used to estimate the added complexity due to partitioning the program is to assume that the number of intercommunication elements grows linearly with program size and that program complexity (and the associated cost of developing the program) is proportional to the number of intercommunication paths available. Assume the program without partitioning have some number of intercommunication elements N . Then the number of paths between elements is $N(N-1)/2$. If the program size, and thus the number of elements increases by a factor of k because of partitioning, the number of paths becomes $kN(kN-1)/2$. Thus, the number of intercommunication paths, and therefore the complexity, grows as the square of k . This is the factor used as a measure of the programming effort required when a program is partitioned among several processors.

Table 17 shows the number of partitions used in each configuration and the resulting software cost. The partitions were determined by dividing the computation problem among computers such that the computation rate limits were met and the number of partitions was minimized.

SUPPORT COST. Annual support costs are shown in Table 15. The cost for hardware support is estimated as one per cent of the initial hardware cost per month per system. The annual cost for software support is estimated as ten per cent of the initial software cost. The reason for these estimates was explained above in the discussion of maintenance.

RISK

Risk in the development of the software is an important factor in the choice of an architecture for a simulator complex. The single large computer presents the lowest risk (provided the initial estimate of the computer requirement is not so low that it results in the need for an additional computer). The risk is low because the problem requires no partitioning and the computer has the highest flexibility.

The assignment of risk to partitioning a problem requires some explanation of the distinction between the partitioning of a problem to achieve a modular structure and the partitioning of a problem among several processors for execution. The partitioning among several processors adds complexity because the variables updated by one processor must be used by other processors. Steps must be taken to insure that (1) the variable used by another processor corresponds to the correct frame time in the computation and (2) that the variables are not updated while being used in computation. It is this relative timing problem that adds the risk to partitioning a problem among several processors.

The single minicomputer driving one or more cockpits presents the next lowest risk. This implementation does not have the flexibility of the large single computer, but it requires no partitioning of the problem and does not require several computers to work together. Multiple computers performing the job result in the highest risk. The increased risk in a multicomputer implementation of the problem is the result of the need for partitioning the program among the computers and the concurrent operation of the programs with

TABLE 17. SOFTWARE COST FOR VARIOUS CONFIGURATIONS

CONFIGURATION	SOFTWARE PARTITIONS			SOFTWARE COST (\$1000)
	CPT	OFT	ACMT	
Separate Computer Systems Each Cockpit	1	1	2	2,297
Separate Computer System Each Cockpit (W/Redun)	1	1	2	2,297
Shared Computers for Each Trainer Class	1	3	2	2,614
Shared Computers for Each Trainer Class (W/Redun)	1	3	2	2,614
Shared Computers with Min. Hardware	1	1	3	2,545
Shared Computers with Min. Hardware (with redundancy)	1	1	3	2,545
Super Mini-Computers	1	1	1	2,070
Super Mini-Computers (with redundancy)	1	1	1	2,070
Single Large Computer	1	1	1	2,070
Single Large Computer (with redundancy)	1	1	1	2,070
Microcomputers	1	3	6	3,737

its associated timing and synchronization problems. Thus the configurations shown in Figures 2 through 4 and Figure 12 present successively higher risk factors.

The microcomputer implementation provides the highest risk, so high that such an implementation is not recommended. Neither the partitioning problem nor the control problem have been solved--this alone is enough to eliminate the microcomputer from consideration. In addition, the level of support software available for microprocessors is far below what is available for minicomputers or large computers.

COST EFFECTIVENESS OF ADDING A REDUNDANT COMPUTER

The VTXTS computer complex is large enough to warrant consideration of a redundant computer to increase availability. All systems considered in the previous section offer a redundant implementation. The value of this redundancy is shown in the availability results of Table 13. However, the determination of whether a redundant computer is cost effective requires the derivation of costs for simulator downtime.

In order to calculate the value of adding a redundant computer, a cost must be established for an hour of lost time. Assume the simulator system costs shown in the second column of Table 18. The total acquisition cost for three simulator complexes is then \$90 million. If the life cycle maintenance cost is assumed to be one per cent of the initial acquisition cost per month as it was in the computer system analysis (a very conservative estimate when the whole complex is considered), the life cycle cost for the complexes is \$198 million. Since this is an order of magnitude greater than the life cycle cost of the computer configurations being considered, a single cost for an hour of lost time applicable to any of the configurations can be established.

If the simulators are used 52 weeks per year, 6 days per week, 12 hours per day over a 10 year life, one hour of time on the simulators based on first cost is the amount shown in the third column of Table 18. The cost of the lost personnel time should also be included. On the average, each cockpit will have one pilot and one instructor. If manpower costs are assumed to be \$20 per hour, this gives the personnel cost shown in the fourth column of Table 18. The fifth column shows the total cost per hour of downtime for each type of simulator.

The lost time caused by computer failure for each system is calculated from the data used in computing the availability. Column two of Table 19 shows the result of applying the lost time to the cost data given in Table 18. This is the cost of lost time for three configurations of computers. Column three of Table 19 shows the saving achieved by adding a redundant computer to four of the configurations. No redundancy is considered in the microcomputer implementation, because the microcomputer is essentially redundant by design. Its one-card configuration allows changing of the whole computer as the normal mode of maintenance.

Column four of Table 19 shows the life cycle cost of adding a redundant computer to each system configuration. The saving achieved by adding redundancy is greatest for the minicomputer systems and not as great for the

NAVTRAEQUIPCEN IH-336

TABLE 18. COST OF DOWNTIME

TYPE	COST (1000'S)	HARDWARE COST PER HOUR	LABOR COST PER HOUR	TOTAL PER HOUR
CPT	800	47	40	87
OFT WO/VISUAL	1500	88	40	128
OFT W/VISUAL	3000	176	40	216
ACMT	6000	352	80	432

NAVTRAEQUIPCEN IH-336

TABLE 19. SAVING BY USE OF A REDUNDANT COMPUTER

CONFIGURATION	LIFE CYCLE COST OF DWN TM MILLIONS	SAVING BY USING REDUN	COST OF BY USING REDUN
Separate Computer Systems Each Cockpit	2149	----	----
Separate Computer System Each Cockpit (W/Redun)	358	1,791	924
Shared Computers for Each Trainer Class	3,886	----	----
Shared Computers for Each Trainer Class (W/Redun)	1,716	2,170	924
Shared Computers with Min. Hardware	2,149	----	----
Shared Computers with Min. Hardware (with redundancy)	358	1,791	924
Super Mini-Computers	4,180	----	----
Super Mini-Computers (with redundancy)	523	3,657	2,310
Single Large Computer	10,450	----	----
Single Large Computer (with redundancy)	1,306	9,144	12,144
Microcomputers	1,036	----	----

TABLE 20. TRADEOFF SUMMARY TABLE

DESCRIPTION	MINI-COMPUTER SYSTEM							
	SEPARATE	SEPARATE	SEPARATE	SEPARATE	SEPARATE	SEPARATE	SEPARATE	SEPARATE
	EACH	EACH	EACH	EACH	EACH	EACH	MIN.	MIN.
	COCKPIT	COCKPIT	COCKPIT	COCKPIT	COCKPIT	COCKPIT	HDWR	HDWR
		W/REDUN	TRNR	TRNR	TRNR	W/REDUN		W/REDUN
HARDWARE COST	MEDIUM	MEDIUM	HIGH	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM
SOFTWARE COST	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM
EXPANSION	MAJOR	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
	MINOR	FAIR	FAIR	FAIR	FAIR	FAIR	FAIR	FAIR
SYSTEM AVAILABILITY	FAIR	GOOD	FAIR	FAIR	GOOD	GOOD	FAIR	GOOD
MAINTENANCE/LOGISTIC COST	MEDIUM	MEDIUM	HIGH	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM
LIFE CYCLE COST	MEDIUM	MEDIUM	HIGH	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM
RISK	LOW	LOW	LOW	LOW	LOW	LOW	LOW	LOW

TABLE 20.(CONTINUED) TRADEOFF SUMMARY TABLE

DESCRIPTION	SUPER MINI COMPUTER	SUPER MINI COMPUTER W/REDUN	SINGLE LARGE COMPUTER	SINGLE LARGE COMPUTER W/REDUN	MICRO- COMPUTER SYSTEM
HARDWARE COST	LOW	LOW	HIGH	VERY HIGH	LOWEST
SOFTWARE COST	LOW	LOW	LOW	LOW	HIGHEST
EXPANSION	MAJOR	MEDIUM	POOR	POOR	GOOD
	MINOR	FAIR	POOR	GOOD	GOOD
SYSTEM AVAILABILITY	FAIR	GOOD	POOR	GOOD	GOOD
MAINTENANCE/LOGISTIC COST	LOW	LOW	HIGH	VERY HIGH	LOWEST
LIFE CYCLE COST	LOW	LOW	MEDIUM HIGH	VERY HIGH	LOWEST
RISK	LOW	LOW	LOW	LOW	HIGH

super minicomputer systems. In the case of the large computer, the cost of redundancy exceeds the cost of the downtime that can be saved. For the systems likely to be used in implementing the VTXTS, redundancy offers a cost advantage over a system without redundancy.

RELATIVE ADVANTAGES AND DISADVANTAGES

The relative ratings of each configuration considered in several important areas are given in Table 20. Availability and cost comparisons are based upon data shown in Tables 13 and 15. The assessment of expansion capability and risk is based upon the discussion given above.

RECOMMENDED SYSTEMS

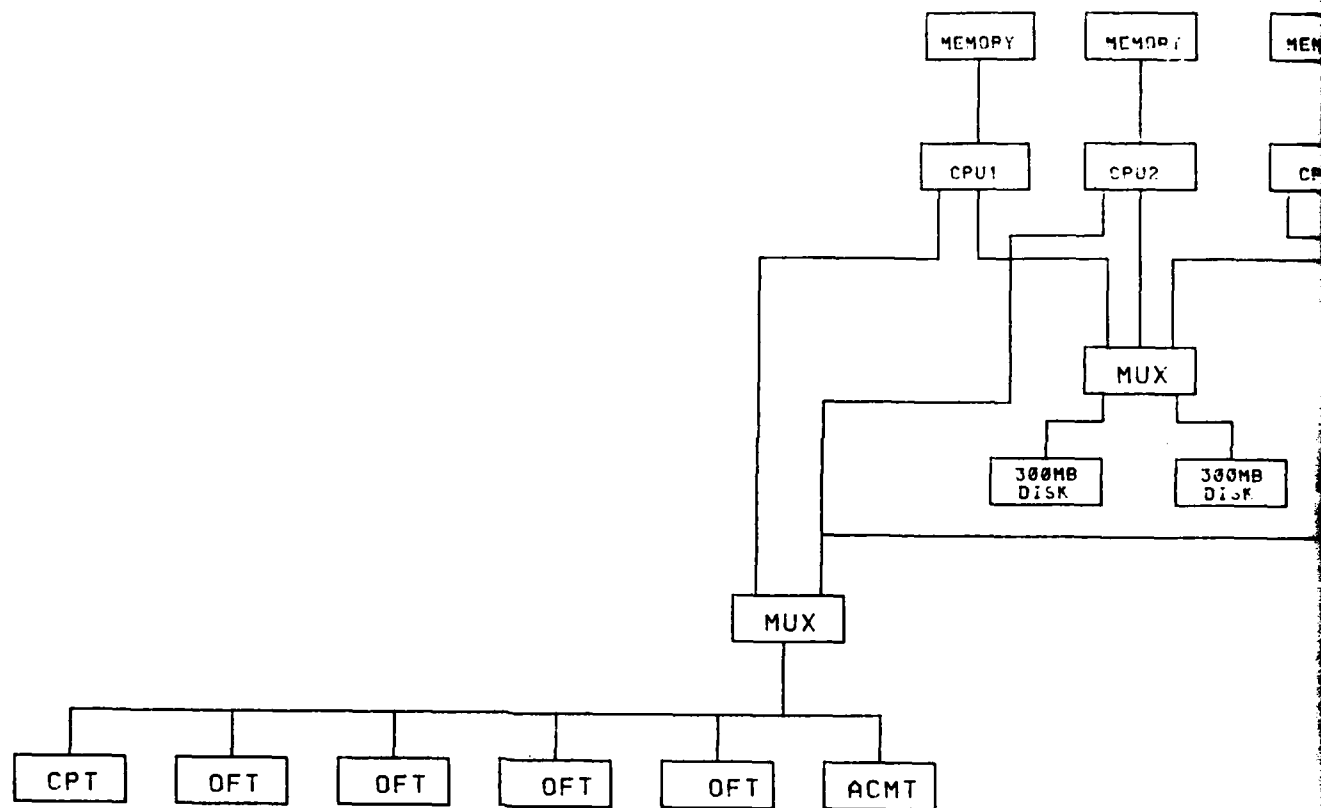
Figure 37 shows a recommended configuration using super-fast minicomputers. Only three computers are required, including a redundant computer. In normal operation, computers 1 and 3 support the trainer cockpits with computer 2 being the spare. This system presents some risk, because the super minicomputers are new on the market and no experience in the use of these computers is available. However, it is expected that the risk would be acceptable and this will be configuration chosen for the VTXTS.

This system has a cost advantage over a system using conventional minicomputers and provides a reasonable expansion capability. It is not as flexible in adding another cockpit to the system, because any addition beyond the capability of the two active computers proposed requires addition of a relatively expensive computer.

An alternative configuration using minicomputers is presented in case further study reveals that the risk in using the super minicomputers is unacceptable. Figure 38 shows the recommended configuration for a VTXTS simulator using conventional minicomputers. Enough computers are provided so that each OFT has its own computer, the ACMT's each have a pair of computers with shared memory, the two CPT's share a computer, and there is a redundant computer. The I/O to the cockpits is taken through multiplexers so that a failed computer can be taken off-line and the system reconfigured. Each group of cockpits has access through the multiplexer to one more computer than is required to perform the simulation computations.

The normal configuration would be the OFT's using computers 1-8, computer 9 would be a spare, the ACMT's would use computers 10-13, and the CPT's would use computer 14. Leaving computer 9 as a spare as opposed to computer 14 minimizes the reconfiguration problem if a computer goes down.

The configuration grows rather easily. Computers can be added one at a time if more processing speed is required. Memory can be expanded by increasing either the size of the memory modules or adding more modules. More cockpits could be added without disturbing the existing system.



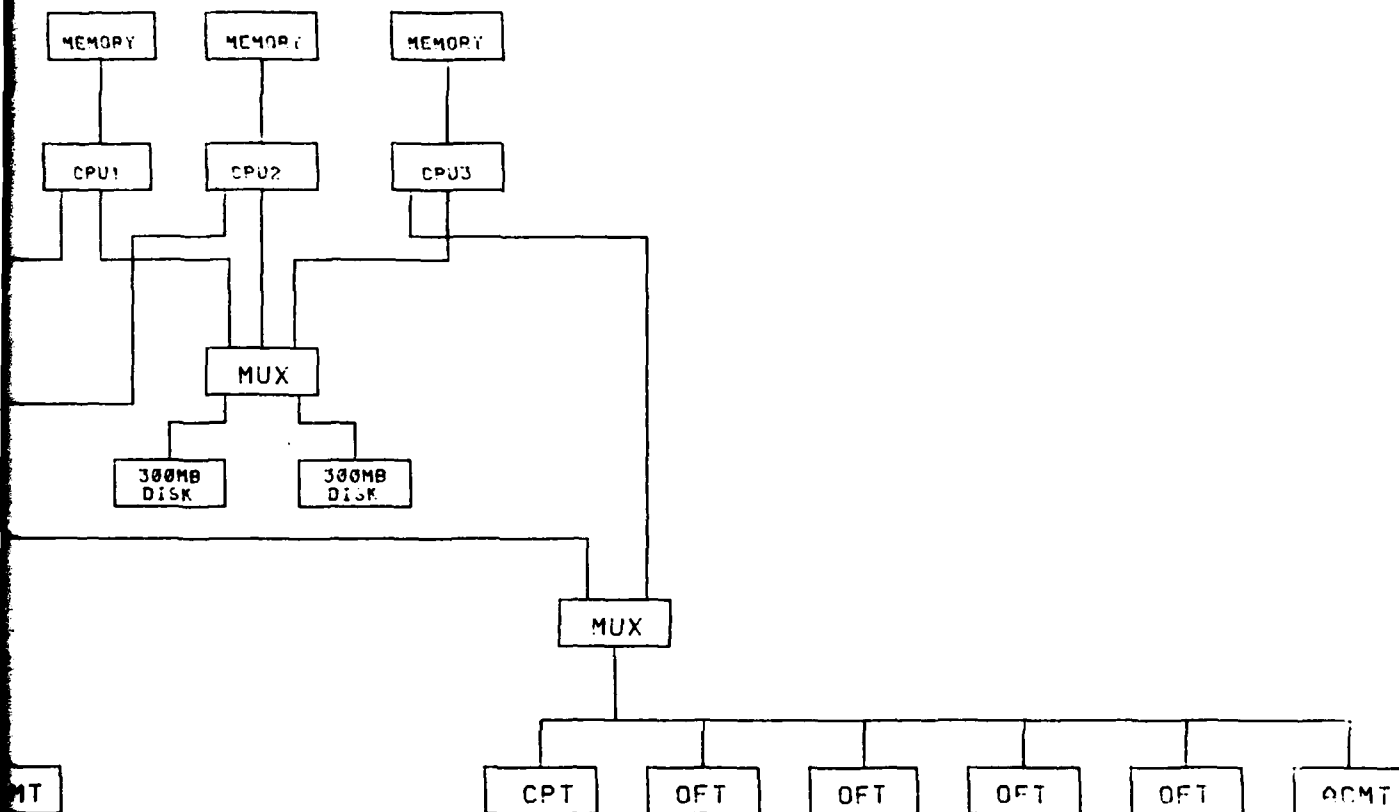
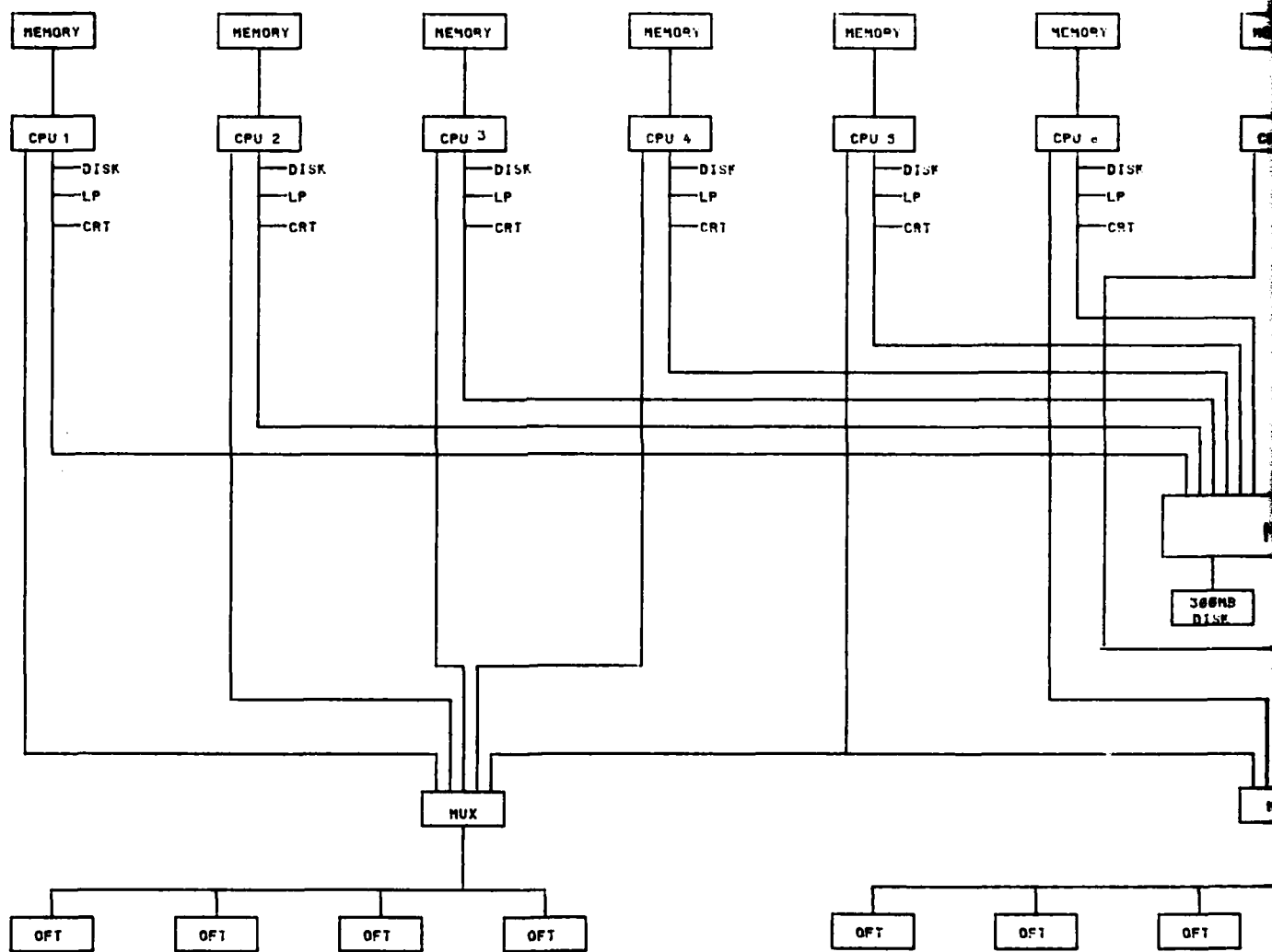


Figure 37. Recommended Super Minicomputer Configuration.



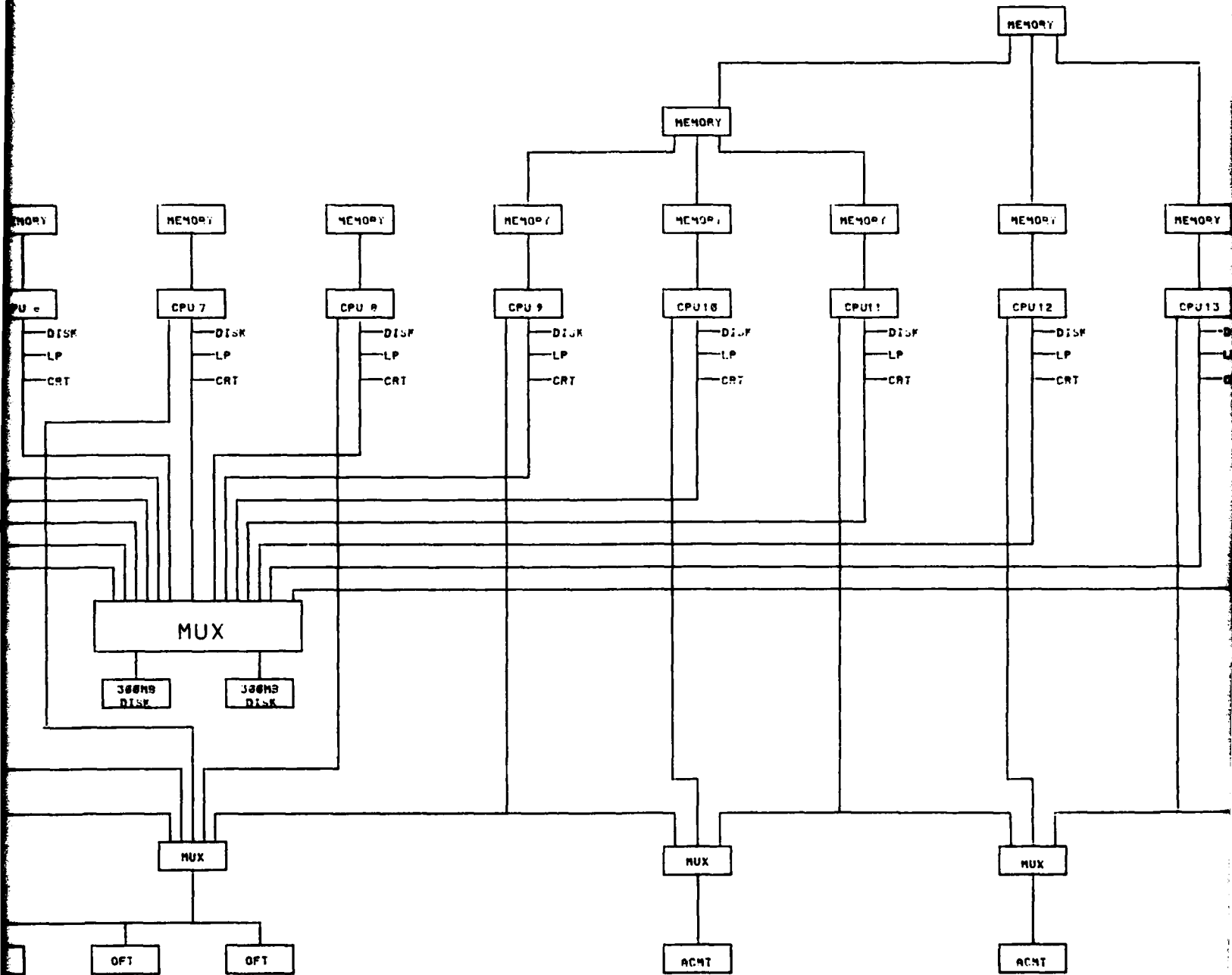


Figure 38. Altern
Using M

NAVTRAEQUIPCEN IH-336

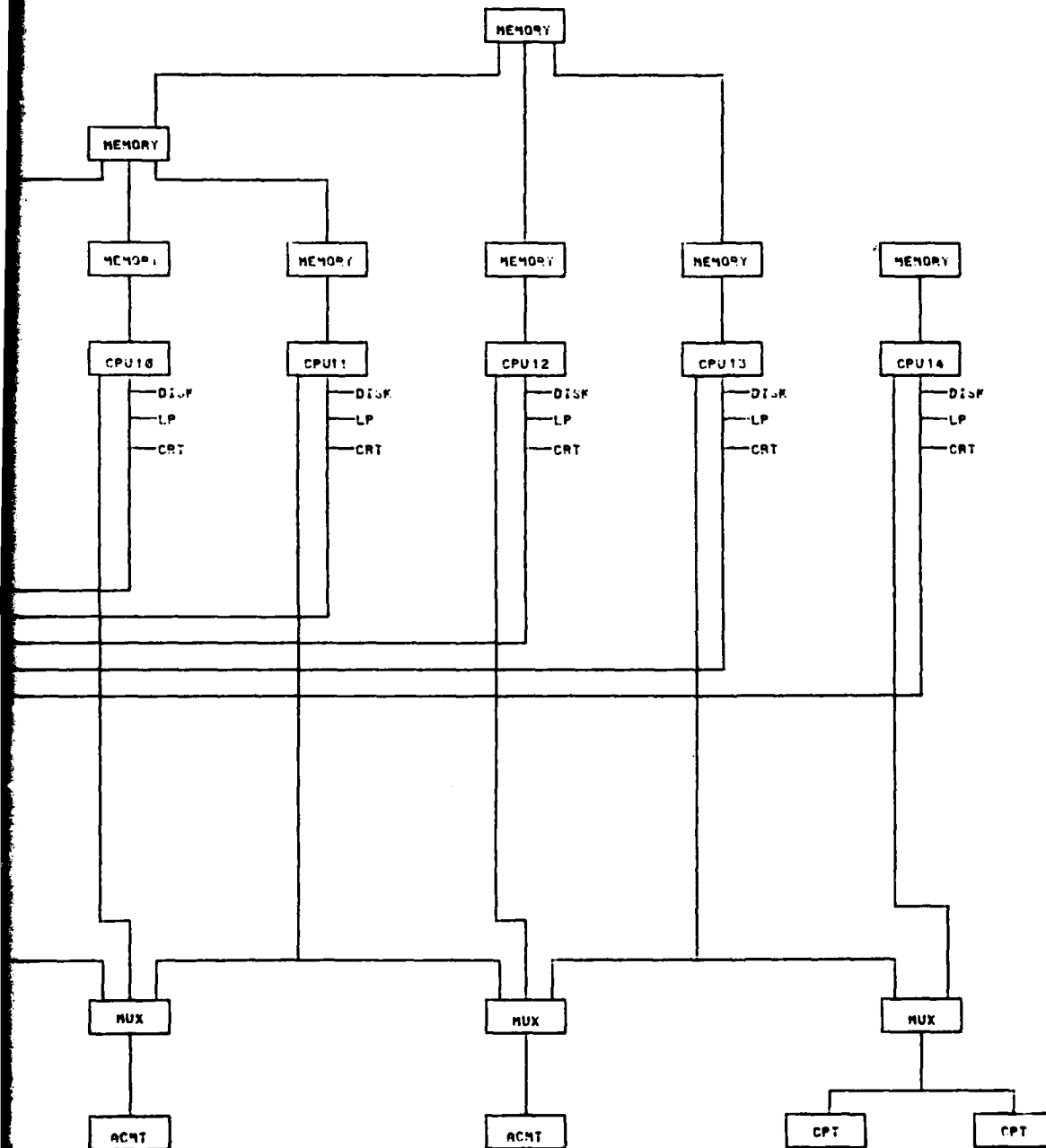


Figure 38. Alternate Recommended System Using Minicomputers.

SECTION VII

OTHER APPROACHES

The development of peripheral processors having a much greater computation capability than any minicomputer offers an alternative solution to the VTXTS computation problem. Three alternative approaches deserve discussion:

- a. Array Processors
- b. AD-10, made by Applied Dynamics, Inc.
- c. HEP, made by Denelcor, Inc.

These alternatives were considered for implementation of the VTXTS simulators but were found less attractive than the use of general-purpose digital computers.

ARRAY PROCESSORS

Array processors are extremely fast processors originally designed for calculation of discrete Fourier transforms. These processors were originally used by the seismic industry, but recently have found application to simulation problems, particularly problems involving aerodynamic function generation. However, these processors are not well suited to training simulators for the following reasons:

- a. An examination of the computer requirements for training simulators reveals that the software is largely logical instructions and not arithmetic calculations. The array processors are primarily designed to perform arithmetic calculations and not logic calculations.
- b. No optimized FORTRAN compilers yet exist for an array processor. At least one manufacturer has a FORTRAN compiler, but it is a subset of ANSI '66 FORTRAN and it is not an efficient compiler. Most or all of the programming would therefore need to be done in assembly language. Programming an array processor in assembly language is significantly more difficult than programming a conventional computer in assembly language.

Karplus and Cohen[12] provide further insight into the problem of using array processors for simulation for training.

[12] Karplus, Walter J., and Danny Cohen, "Architecture and Software Issues in the Design and Application of Peripheral Array Processors," Computer, vol 14 no 9, pp 11-17.

APPLIED DYNAMICS AD-10

The AD-10 is a 16 bit, fixed point computer specifically designed for simulation. It was originally designed as a function generator to be used with a digital, analog, or hybrid computer. It was extended to a full simulation computer essentially by the addition of an integrator module. The integrator module uses a 48 bit word to accumulate integrator outputs to avoid the limited range of 16 bit word used in the rest of the machine. The use of the AD-10 for this application is not recommended for the following reasons:

- a. Since the AD-10 is a fixed point computer, the programming is necessarily done in assembly language. Like the array processor, the AD-10 is a parallel machine, and programming it is significantly more difficult than programming a conventional mini-computer. In defense of the AD-10, it should be noted that ADI has developed a number of software packages to ease the programming problem with the AD-10.
- b. The AD-10 is primarily oriented toward arithmetic calculations. Examination of the computer software for training simulators reveals that a very small percentage of the software is actually used for arithmetic calculations. The majority of the computer software is devoted to logical calculations to determine the action to take as a function of various switch inputs. Since the AD-10 is primarily designed to perform arithmetic calculations, it is not well adapted to this job.

HETEROGENEOUS ELEMENT PROCESSOR

Heterogeneous Element Processor (HEP) is an extremely fast (10 MIPS - 160 MIPS) parallel processor designed primarily for simulation. It has a FORTRAN '66 compiler at the present time and a FORTRAN '77 compiler is scheduled to be finished in December 1981. Delivery of the machines is scheduled for January 1982. Although a 10 MIPS version of HEP would be adequate for the VTXTS application, use of the HEP should probably not be considered at the present time for the following reasons:

- a. HEP is very new. No field experience yet exists with the machine. Operating and software bugs surely exist which would hamper development of a simulator.
- b. HEP appears to be overkill for this problem. At \$1.5 M, it is not as cost effective as two or three of the superfast mini-computers at \$700K or \$1.0M. HEP is really designed for larger problems than this one where computer speed is of paramount importance and the problem does not partition easily.

SECTION VIII

SUMMARY AND CONCLUSIONS

This study provides an evaluation of computer configurations that might be applied to the VTXTS simulator complex. The computer configurations that have been considered give an insight into the effect of various computer arrangements in terms of cost, maintenance, availability, and risk. More important than the numbers themselves is the procedure used, because this procedure can be used to address other configurations and other numerical values for the variable involved. This procedure is recommended for use in evaluating more detailed implementations provided by the VTXTS contractors.

The treatment of computers was limited to consideration of the general characteristics of each generic class of computer, with no assessment of specific manufacturer's products. The classes considered are (1) large general-purpose computers, (2) super minicomputers, (3) minicomputers and (4) microcomputers. Eleven configurations were considered, two using large computers, two using super minicomputers, six using minicomputers, and one using microcomputers. A summary of the tradeoff among the various systems is given at the end of Section VI.

RECOMMENDED SYSTEMS

The analysis indicates that the best approach to implementation of the VTXTS simulators is use of super minicomputers. This type of computer, now becoming available, offers a significant cost advantage over the minicomputers used in the most recent simulators. The large computer and the microcomputer were found not suitable for the VTXTS application.

This is not a surprising result. It affirms the fact that the VTXTS application is not significantly different from flight simulators of recent vintage and should be implemented in much the same fashion. The difference in the recommended system for VTXTS implementation is the recent development of super minicomputers which provide several times the capability of the minicomputers of a few years ago. The recommended configuration consists of a super minicomputer performing the computations required for a CPT, four OFT's and an ACMT. Three computers are included in each complex, two computers to share the computation load and a redundant computer that can be used to replace either of the others in case of failure. This configuration provides a significant cost advantage over a system using conventional minicomputers.

The use of a large-scale, general-purpose computer is rejected, because that approach is more expensive than use of multiple minicomputers and results in a lower availability. Although the lower availability could be corrected by the use of a redundant system, the addition of a second computer makes the cost tradeoff even less attractive. It is instructive to look at the reason for this phenomenon in view of the use of large computer complexes for ticket reservation systems and other applications that seem to require much processing and the same high degree of reliability. The difference is that the applications where the large computers show an advantage are those where the problem is the processing of vast amounts of related data. The simulator

application is a problem where a dozen entirely separate problems are being solved. There is absolutely no advantage to be gained by solving unrelated problems on one computer.

If large computers are not the direction to go, it would seem that smaller computers must be the right answer. In the long range, this may be true. However, for the VTXTS application, the risk of entering a new technology far outweighs the advantage to be gained. The problems of control of multiple microcomputers in the simulator environment is being studied, but there is no suitable proven solution. Partitioning the problem for the microprocessor is another unanswered question. Software design aids for microprocessors is not as well developed as those for larger machines, and this would add a great risk to the overall project schedule and cost if microcomputers were selected. These problems should be solved in the R&D arena rather than on a major simulator procurement.

Because the super minicomputer is new to the market and there is no experience with its use, further study is needed before making a final choice. A configuration using minicomputers is presented as an alternative in case that study shows that the use of super minicomputers is unacceptable.

The choice of a minicomputer configuration for the VTXTS application presents a problem in risk assessment similar to that in applying microcomputers. The least expensive configuration is one that uses eight minicomputers to share the computation for all cockpits. The more conservative configuration uses separate computer systems for each cockpit, requiring a total of thirteen computers. The conservative approach, even though it costs more, is the one recommended.

This requires some explanation. Use of eight computers requires partitioning of the tasks, with associated problems in software complexity, intercommunication between computers and synchronizing of concurrent operations. Although the study shows that the saving in computer cost will more than pay for the software development, this approach is rejected because of its risk. The additional software development is likely to take six months to a year longer than the software development for the more conservative approach. This high-risk approach should be undertaken only with an understanding of the risk involved and with a plan to monitor the software development closely.

AREAS FOR FURTHER RESEARCH

This analysis gives an overview of some of the considerations in choosing a computer configuration for a simulator system. In preparing this study, the authors have been forced to rely upon estimates based upon experience and reasonable extrapolation of known results. Several areas touched upon in this study require further research to define the system better and to provide a firm basis for a VTXTS choice.

SUPER MINICOMPUTERS. The new class of super minicomputers appears to offer the most attractive solution to the VTXTS computation problem. However, the lack of experience in applying this kind of computer indicates a need for closer monitoring of contractor effort in evaluating this configuration.

COMPUTER BENCHMARKS. Use of the average instruction execution time in evaluating computer performance is one of the major weaknesses in this study and in the trainer procurement process. Use of modern components such as cache memory, pipeline processing, and floating-point processors and the use of high-level languages provide computer systems whose performance cannot be determined by such a simple measure. Instead of measuring performance by use of an assumed instruction mix, performance should be measured by executing computer programs typical of the application on the machine to be evaluated and measuring execution time and storage required. Developing and validating such a set of programs should be part of the VTXTS effort.

MULTIPROCESSOR SYSTEMS. Several of the configurations considered used multiple computers to solve a single simulation problem. For example, the implementation of VTXTS in minicomputers can be done using a greater number of processors and assigning a processor or two to each simulation task, or it can be done by using a minimum number of processors and spreading the computation task among these processors. The minimum-processor approach has not been recommended, because, although it is less costly in hardware, it presents a higher risk. The implementation using microcomputers spreads the computation problems over an even greater number of computers.

Use of multiple processors for real-time simulation requires further study to reduce the risks. Work in partitioning the problem, control algorithms, and configuration of processors and memory for this application should be continued. This research will apply to both minicomputer networks in the immediate future and to microcomputer networks now becoming a viable method for simulator implementation.

INTERRUPT-DRIVEN SYSTEMS. Simulators for training ordinarily use a fixed frame computation in which the total task to be performed is divided into subframes to be computed at various rates (e.g., 30, 15, 7.5 times per second). Research should be performed to determine the advantages and disadvantages in using an interrupt-driven system which allows subframes to be processed in the time left over after the mainframe computation. This seems to offer a more effective use of time available than the conventional method.

NAVTRAEQUIPCEN IH-336

REFERENCES

- [1] Naval Air Development Center Report 79065-20, Naval Air System Command, Analysis of VTXTS Technology Base Studies.
- [2] Edward J. Lias, "Tracking the Elusive KOPS," DATAMATION vol. 26, no. 12, Nov 1980, pp 99-105.
- [3] Summer, C. F., Jr., G. A. Wyndle and David E. Daniel, Computer System Requirements Analysis, Device 2F112, F-14 Weapon System Trainer. Orlando, FL 1977. Report No. NAVTRAEQUIPCEN IH-262.
- [4] GML Corporation, COMPUTER REVIEW. vol 21, no 1, p 5.
- [5] Hoo-min D. Toong and Amar Gupta, "An Architectural Comparison of Contemporary 16-Bit Microprocessors." IEEE MICRO. vol 1, no 2, pp 26-37.
- [6] B. A. Zempolich, Future Avionics Systems Architecture. Naval Air System Command, Report No. AIR-360, Oct 1977
- [7] W. J. Dejka, Microcomputers and System Design. Naval Electronics Laboratory Center, Report TD-507, Feb 1977
- [8] Summer, C. F., and G. A. Wyndle, An Analysis of Microcomputer Technology for Application to Real-Time Trainers. Orlando, FL, July, 1979, Report No. NAVTRAEQUIPCEN IH-313.
- [9] Pettus, R. O., R. D. Bonnell, M. N. Huhns, L. M. Stephens, and G. M. Wierzba, Multiple Microcomputer Control Algorithm. Orlando, FL, Sep, 1979, Report No. NAVTRAEQUIPCEN-78-C-0157-1.
- [10] Pettus, R. O., M. N. Huhns, L. M. Stephens, and M. O. Trask, Multiple Microcomputer Control Algorithm Feasibility Breadboard. Orlando, FL, Aug, 1981, Report No. NAVTRAEQUIPCEN-79-C-0096-1.
- [11] Putnam, Lawrence H., and Ray W. Wolverton, Quantative Management: Software Cost Estimating. A tutorial for COMSAC '77, The IEEE Computer Society's First International Computer Software and Applications Conference, Chicago, IL, 8-10 Nov 1977.
- [12] Karplus, Walter J., and Danny Cohen, "Architecture and Software Issues in the Design and Application of Peripheral Array Processors," Computer, vol 14 no 9, pp 11-17.

NAVTRAEQUIPCEN IH-336

DISTRIBUTION LIST

Naval Training Equipment Center (50)
Orlando, FL 32813

Defense Technical Info Center (12)
Cameron Station
Alexandria, VA 22314

PM TRADE, US ARMY (2)
Attn: SE
Orlando, FL 32813

ALL OTHERS RECEIVE ONE COPY

Chief of Naval Education & Training
Code N-331
Pensacola, FL 32508

Naval Air Systems Command
AIR 340D
Washington, DC 20350

Naval Air Systems Command
AIR 340F
Washington, DC 20350

Naval Air Systems Command
AIR 360B, Attn: B. Zempolich
Washington, DC 20350

Naval Air Systems Command
Library AIR 50174
Washington, DC 20350

Chief of Naval Material, Navy Dept.
Code MAT 08T
Washington, DC 20360

Chief of Naval Material, Navy Dept.
MAT 08Y
Washington, DC 20360

Chief of Naval Material, Navy Dept.
MAT 09Y
Washington, DC 20360

Chief of Naval Research
ONR 461
800 North Quincy Street
Arlington, VA 22217

Naval Air Systems Command
AIR 413
Washington, DC 20350

AF ASD/ENE
Attn: Mr. A. Doty
Wright-Patterson AFB, OH 45433

AF ASD/ENO
Attn: Mr. Phil Babel
Wright-Patterson AFB, OH 45433

Dept. of Electrical & Computer
Engineering
Attn: Dr. Harold Stone
University of Massachusetts
Amherst, MA 01003

Chief of Naval Operations
OP-115
Attn: LCDR J. Lawhon
Washington, DC 20350

