THE MICROPROGRAMMABLE MULTI-PROCESSOR (MMP) SYSTEM FOR SIMULTANEOUS EMULATION OF INTEROPERATING COMPUTER SYSTEMS

Roy Mattson
US Army Electronics Command
Fort Monmouth, New Jersey

Alan Salisbury
Office of the Project Manager,
Army Tactical Data Systems
Fort Monmouth, New Jersey

INTRODUCTION

A Teleprocessing Design Center (TDC) has been established within the Communications/Automatic Laboratory of the US Army Electronics Command (ECOM) at Fort Monmouth, New Jersey, for the purpose of supporting experimentation in developing and validating Army Tactical Data Systems (ARTADS) configurations. Facilities of the center include a powerful microprogrammable multiprocessor (MMP) system as well as a full complement of militarized computers and peripherals and special purpose devices planned for fielding with ARTAD systems.

The TDC is collocated within the Communications/Automatic Data Processing Laboratory with the Communications System Design Facility which includes circuit and message switching systems, systems and technical control facilities, and transmission media. Together, the two facilities provide a broad range of capabilities for testing and evaluating ARTAD systems and components in various operating environments. The heart of the TDC is the MMP system which will provide the capability of simultaneously emulating up to two interoperating ARTAD systems. This paper describes that system, including the hardware and key software modules, and provides a detailed look at its use in emulating one of the ARTAD systems, namely the Tactical Fire Direction System (TACFIRE). Of particular interest is the system nature of this emulation, since not only the processor itself, but each and every one of the diverse peripheral units will be emulated.

MMP SYSTEM

The MMP System is intended to provide a facility to be used both as a tool and test bed for implementation of the following functions:

- Emulation and evaluation of systems.
- Quick system assembly for evaluation of engineering approaches and the solution of specific problems.
- Emulation of new computers or systems prior to development or in advance of delivery.
- Evaluation of candidate hardware/software replacements.
- Evaluation of computer peripheral devices.
- System integration and interface testing.
- Experimentation with novel software and hardware.
- Test bed for the isolation of field anomalies and the solution of specific computer system problems.
- Test bed for application software development, testing, and post deployment redesign.

Based upon an analysis of the tests and evaluations, recommendations for changes, improvements, and restrictions can be made concerning the particular devices or systems. Because these functions are diverse and vary in complexity, the MMP system consists of a versatile set of reconfigurable and interchangeable physical equipments, together with a system structure of logical resources, allowing addition of new processes and interactive changes in the configuration.

HARDWARE

A block diagram of the overall MMP system is shown in Figure 1. All hardware is from the 5600 series of Control Data Corporation. A total of seven processors are included with the capability of adding up to three additional units. Five large plane memory units are also in the present system with expansion provisions for up to eight. Five of the microprogrammable processors (MPP's) in the system are Model 5605 32-bit processors, while the remaining two are 5601 units or 16-bit processors. (5600 processors can be configured in 4-bit increments from 8 to 32 bits.) These processors utilize what could be considered a "quasi" horizontal microinstruction of 32-bits with an 80 nanosecond control store of from 256 to 8K 64-bit words (each word contains two microinstructions). Both read/write and ROM control store are available. For emulation purposes, hardware "transform modules" are utilized to facilitate target instruction field extraction and decode; special boards must be designed for each processor emulated if standard transforms are not sufficient. One 32-bit processor acts as the overall MMP system controller. This processor emulates the CDC MP-60 architecture which in turn runs the MPX/RT operating system and System Nucleus to control all tactical emulations run on the system. Three 32-bit processors are available as general purpose emulators, and the final one is dedicated to the function of Mass Storage Controller (MSC), controlling an 869 M bit disk. Other than this one dedicated processor, any of the 32-bit processors can serve as the MP-60 while the remaining three operate under its control. The two 16-bit processors are dedicated as I/O Controllers, one handling unit

MICROPROGRAMMABLE MULTI-PROCESSOR SYSTEM

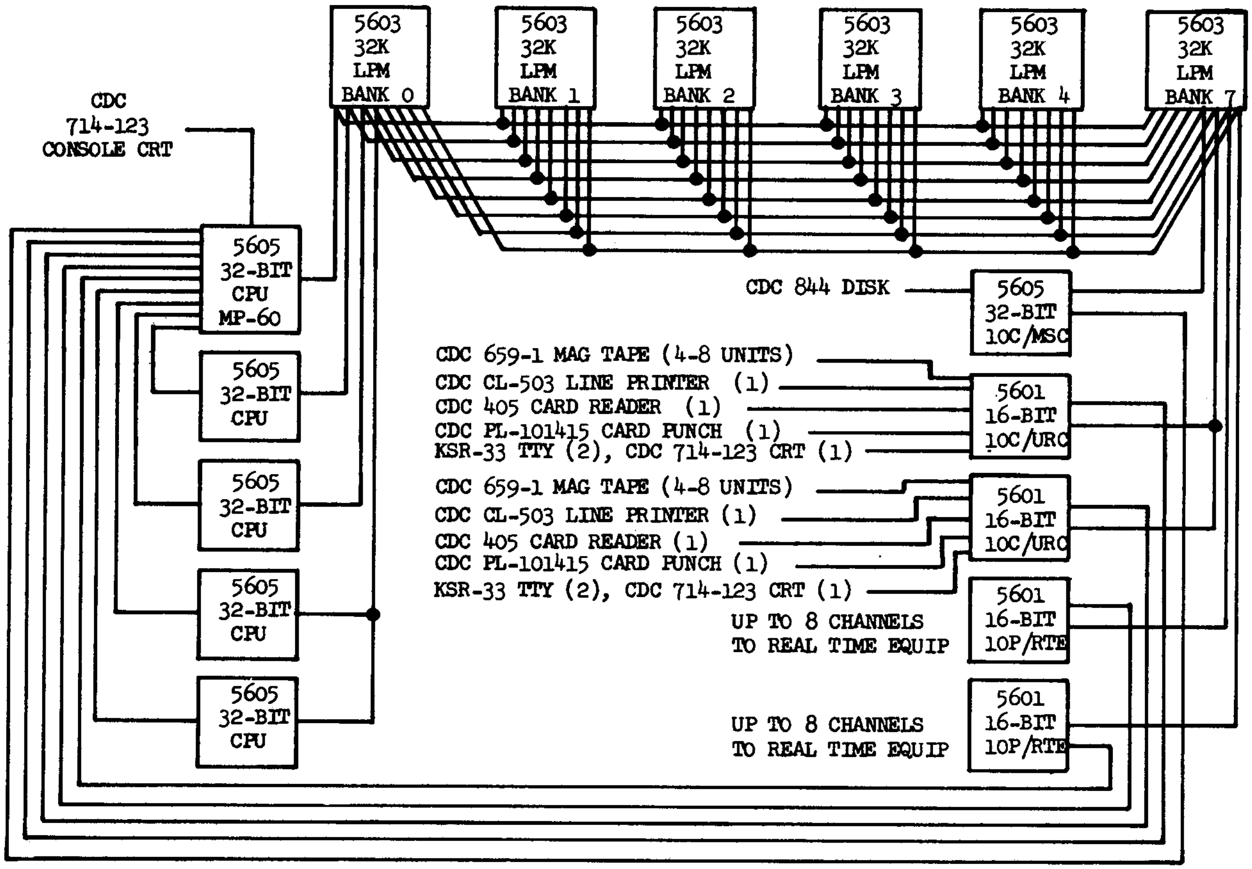

FIGURE 1

record equipment (TTY, CRT, card reader, card punch, magnetic tapes, and printer), and the other controlling real time tactical equipment peripheral devices.

Five Large Plane Memory (LPM) banks are available as system main memory, each bank holding 32K words of 36 bits (32 bits plus 4 byte parity bits). An interesting feature of the LPM banks is that they are 8 port banks with all processors having access to all memories. Thus, the LPM's also serve as a central exchange for data and commands among all units in the system. Direct communication between the system emulators and I/O controllers is restricted to interrupt signals; data and I/O commands are transferred to memory by sending units and read from memory by receiving units.

SOFTWARE

The System Nucleus (SN) is a modular extension to the interrupt driven MPX/RT real time operating system for the MP-60. It maintains control over the entire MMP system and provides operator communication facilities for the generation, loading, and operation of emulation systems. The portion of SN which is integrated with MPX/RT is called the SN Kernel. Executive facilities are included in the Kernel along with emulation control capabilities. SN utility routines facilitate the generation of an emulation system and an emulation main control program. User defined action and service routines can also be accommodated providing for a variable SN capability. Support software provided with the MMP system includes a micro assembler (Micro 71) for the microprogrammable processors, an MP-60 machine language assembler (Compass), a FORTRAN compiler, and maintenance and diagnostic routines, including microdiagnostics.

AN APPLICATION: "TACFIRE" EMULATION

The MMP will initially be used to assist in the evaluation of the TACFIRE system. The purpose of TACFIRE can best be summed up as a means to increase artillery efficiency, expand its effectiveness, and expedite command decisions through automation. It consists of ADP equipment used by operators to assist the commander in the delivery of fire. Among the functions that TACFIRE performs are Ammunition and Fire Unit Status, Meteorological Calculation, Fire Control, Fire Planning, Target Intelligence, and Target Analysis.

The TACFIRE system is made up of computer group equipment and display group equipment. The computer group equipment consists of a general purpose computer, the Litton L-3050, made up of one central processing unit, a dual 8K memory unit, a 131K mass core memory unit, and one input/output unit. Two devices used to augment the computer core memory are the Random Access Memory (RAM), consisting of two 256-track drums, and the Removable Media Memory Unit (RMMU), consisting of two cartridge tape units. The display group equipment includes devices normally associated with a computing complex and, also, devices unique to the particular applications. The Artillery Control Console (ACC) is composed of two display editors, an alphanumeric keyboard, and a switch panel assembly. The Digital Plotter Map (DPM) and Electronic Tactical Display (ETD) are used to provide graphic displays of a tactical situation, while the two Electronic Line Printers (ELP) are used for permanent record printing. Also associated with the local system are remote devices which are connected via communication lines and devices. These include Fixed Format Message Entry Devices (FFMED) and Variable Format Message Entry Devices (VFMED) (used as message entry devices by forward observers to enter fire mission and intelligence data into the computer) and the Battery Display Unit (BDU) at the firing batteries to receive fire commands and other messages.

The TACFIRE complex described is replicated at Army division level and at battalion levels forming a deployed area of computing centers. Communication links are also planned to the Tactical Operating System (TOS), now in the concept development stage for use by commanders and their staffs.

The L-3050 computer forms the nucleus of the TACFIRE system and is a high speed, general purpose digital computer. The characteristics of the computer are as follows:

(1) Thirty-two bit instruction word.
(2) One, eight, sixteen, thirty-two, or sixty-four bit data word.
(3) 2.0 microsecond memory read/write cycle.
(4) Memory expandable in 8K 32-bit word banks or 131K mass core memory units.
(5) Memory access control and protection and internal parity checking.
(6) One hundred basic instructions plus 50 extended mnemonic instructions.
(7) Nine addressing mode combinations.
(8) Sixty-four program levels.
(9) Sixteen 32-bit multipurpose registers and 18 additional special purpose registers per program level.
(10) Four additional special purpose registers.
(11) Up to 126 I/O channels each with program initiated, but independently operating, data transfers of up to 32 bits in parallel.
(12) I/O transfer rates of up to 400,000 32-bit words per second on one channel or 180,000 32-bit words per second when several channels are operating simultaneously. A queue word for each program level provides stacking of interrupts in automatic priority and high speed multiprogram switching in single or multiple processor configurations.

During the emulation of systems such as TACFIRE, a hierarchy of subsystems are built with each succeeding level using the facilities of the previous level. If components are considered at the lowest level, they are organized to provide the computer architecture of a processor such as the L-3050. Extending the processor with peripheral equipment now forms the architecture of an L-3050 hardware system that is visible to a software programmer. At this point, a system programmer extends the system with software such as an executive or operating system. The hardware organization has been extended with architectural features that can now be used by the next level. Usually, once the operating system has been defined, a translator (either a higher level language such as PL/1, FORTRAN, or TACPOL, or assembly languages, or both) is developed. The next extension or level in the hierarchy can then be considered to be (for example) the PL/1 features and not the organization below it. Finally, an application program is written such as the TACFIRE's ammunition status program, whereby the user (an artillery officer) sees only the features of that program's architecture. To him, all the hardware and software organization below the features he interfaces with are invisible.

A distinction has been made between the terms "architecture" and "organization" as applied to each level in the hierarchy. We have used the word "architecture" to refer to those aspects of a level which are visible to the user at that level. "Organization", on the other hand, is at a level below architecture and is concerned with many items which are transparent to the programmer. In short, we use the term "architecture" to describe what facilities are provided, and the term "organization" to describe how those facilities are implemented.

Via microprogramming, emulation of the architecture at the various levels of the described hierarchy can be performed. For instance, a directly executable PL/1 architecture can be provided. At a lower level, a machine can be emulated which includes the software extensions that the system software writer has provided. Here, portions of the operating system that are visible to application programs are now embedded in the organization of that processor. At a still lower level, the processor itself can be emulated architecturally, i.e., from descriptions available in a programmer's manual or a principles of operation manual. At this level, organization features such as memory look-ahead are not visible.

The level at which the TACFIRE system is emulated is predicated on the use that is foreseen for it. Evaluation of both TACFIRE's hardware and software is desired. All software that runs on an L-3050 system, whether stand-alone or supported by TACFIRE's OS, should operate on the emulation. A more stringent requirement is that all software that does not run on an L-3050 system also does not run on the emulator and should fail in the same way. The single most important problem for a tactical system is the requirement for it to respond in very short times to many inputs which arrive at unpredictable times and which may peak in numbers at any time. The most critical aspects include time tolerances associated with the scheduling, dispatching, and processing of tactical tasks. Within the emulation of the L-3050 and the TACFIRE system, preservation of real time placed an important constraint on the system. Recognizing that the achievable real-time may not correspond to elapsed wall clock time, a requirement is made that time be translatable to real-time. Another requirement placed on the emulation of the system was the desirability to acquire performance monitoring data on both the software and hardware.

Based upon these requirements, it was decided that the emulation be as accurate a replica of the L-3050 as possible. Starting with the architectural features visible to a programmer as a base, various organizational features were added such that "side effects" of instructions also exist in the emulation.

MAPPING TACFIRE ONTO THE MMP SYSTEM

Mapping of the TACFIRE system onto the MMP system is illustrated in Figure 2. The L-3050 CPU emulation is handled by one of the emulators that provides (1) all of the architectural features visible to a programmer, and (2) organizational features that influence the running of software. The IOU emulation is handled by SN routines which assign routines to process the L-3050 I/O instructions based on the device number and the mapping of tactical devices to available system devices. The memory management is handled primarily by the SN routine which provides the emulator with settings for the CDC 5600 page registers to map the L-3050 addresses into MMP assigned memory. The MCMU is handled by memory swapping between available core and the disk. The SN provides the emulator with the correct page settings for a swap. If an area of mass core is not in available memory, the 5600 page register will be set to cause a page fault which will then be passed on to the SN for handling of the swap.

The mapping of the tactical peripheral devices onto the system devices poses some problems. For tactical devices such as tapes and line printers, there exist similar devices in the MMP system. For the tactical drum, the comparable MMP device is the system disk, upon which the former can be simulated. Other devices such as the tactical plotter and the display have no direct or similar MMP system device.

The tactical tapes are mapped to either an existing tape drive or to the disk storage unit, depending on the availability of tape units. SN utility routines provide for copying a tape to the disk or copying the disk to a tape. Under operator control, the assignment of the tactical tape to either a physical tape or disk is made.

The tactical drum is mapped to the disk storage system of the MMP system. The L-3050 I/O instructions are used to generate the location of data to be transferred to or from the disk and are handled by SN routines.

The output from the L-3050 emulation to the printer is either directed to the system printer or stored on the disk for later display upon request from the operator.

The ACC consists of three functional units, the Switch Assembly, a Receive Display (RD), and a Compose and Edit (C/E) display. Each display has seven 72 character lines. The simulation of the ACC is handled by an SN routine which uses the system CRT. The CRT provides for a 16 line by 80 character display which allows both of the RD & C/E displays to be on the screen at one time, along with two lines to indicate the switches and lights.

Two devices, the tactical display and the plotter map, have no devices with similar capability in the MMP system. In order to provide and obtain operator interaction, these devices would have to be provided as actual equipment in the system.

MICROPROGRAMMABLE MULTI-PROCESSOR SYSTEM
UNDER TACFIRE EMULATION

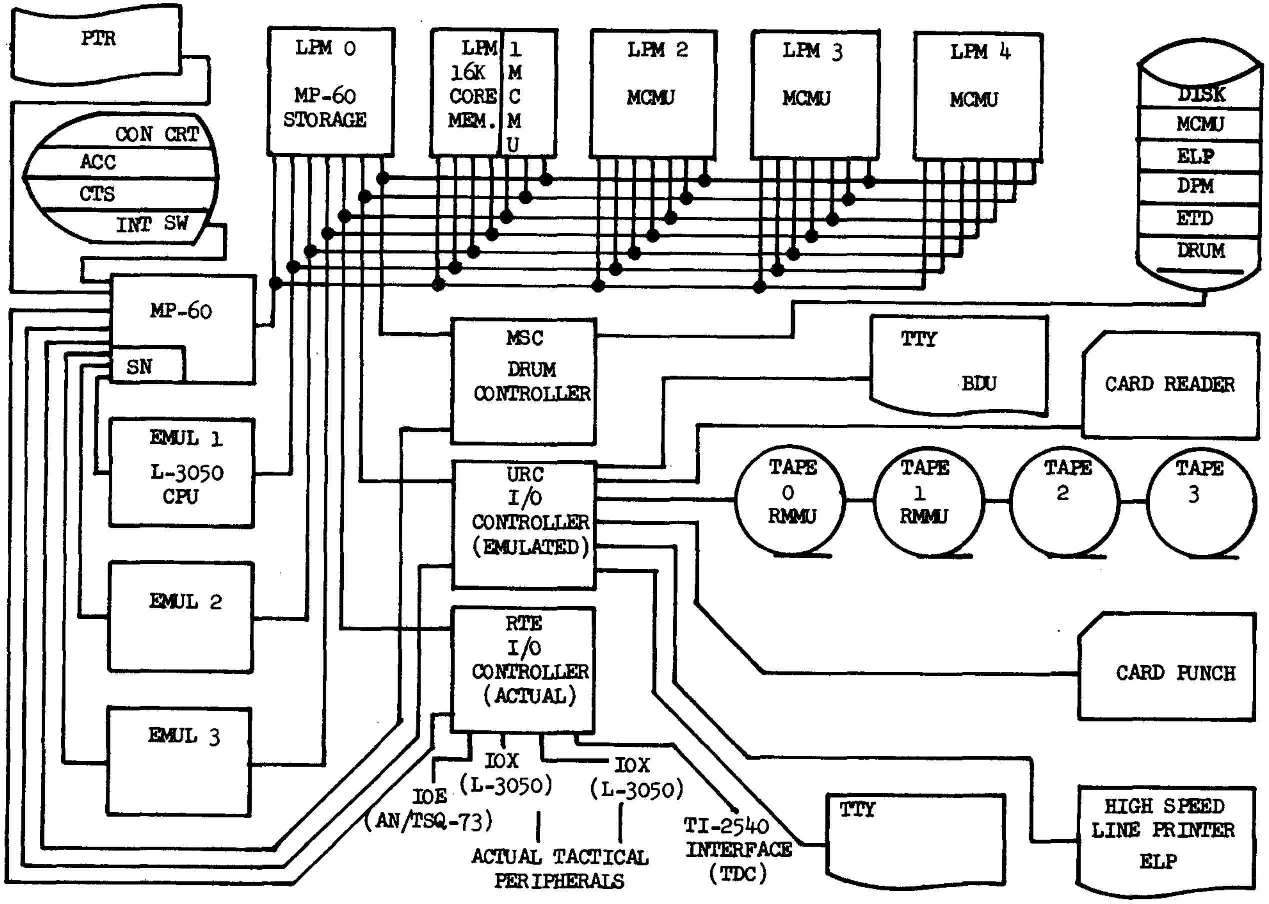

FIGURE 2

An alternative chosen is to provide for static storage of the display information on the disk and to use the printer as a plotter to provide an output of significant display changes.

Remote devices such as the VFMED, FFMED, and BDU are connected to the L-3050 system via the communications net and are represented by the capability of transmitting and/or receiving ASCII messages via the Data Terminal Unit. These messages can be generated as part of the tactical environment generation for exercises and provided to the system by SN routines.

The term "emulation", as used here, means the generation of the exact behavior of a device or system by the use of another device or system. By "exact behavior" it is meant that the emulation produces exactly the same end results (normal emulation requirement) in exactly the same time (a unique requirement for our purposes) as would the original system. If it is desired to emulate the behavior of a system such as the TACFIRE system exactly, the interrelationship of results and timing must be accounted for, especially with asynchronous competing units such as CPU, memory, IOU, and peripherals. Depending on the emulation technique, the wall clock time for a CPU emulation may be faster or slower than the time required for the actual equipment. Peripheral transfers take place at a much slower rate than calculations, but, nevertheless, effect the exactness of results.

The solution selected for the TACFIRE emulation is to generate emulated (virtual) time or clocks. In this approach, an emulated clock will be incremented by the time required for the execution of instructions in the actual equipment, and, by taking a measurement of the emulated clock, the "same" time is read as in the actual equipment. The method of handling peripheral device transfers is to provide the time of initiation (emulated) to the peripheral device driver program in SN. This program may then calculate the time which would be necessary for completing the peripheral transfer and pass this time of completion back to the emulator. The emulator would then continue operation until the expected time of transfer completion, check back with the peripheral driver program, and stop operation until the transfer is actually completed. Since the emulated clock is then stopped, the peripheral transfer is obviously completed at the correct emulated time once the emulation is resumed.

The timing characteristics of the system being emulated are embedded in the microcode, and at the completion of each instruction, the virtual clock is incremented by the amount of time which would have been taken by the actual instruction in the emulated computer to the accuracy that the time of the instructions in the machine is known.

All peripheral device data transfers in the emulated L-3050 system are handled by devices other than the processor emulator. The general method of handling a data transfer is as follows:

(1) The emulator encounters a data transfer instruction.
(2) The information regarding the parameters of the instruction is transferred to common memory for communication with the MP-60.
(3) An interrupt is issued to the MP-60 to handle this instruction and the emulation stops operation waiting a reply from the MP-60.
(4) SN program hooks up the device assignment to the channel for the emulator and substitutes a physical device for the emulated device.
(5) SN program takes the parameters of the emulated device, such as access time, latency data transfer time, and other characteristics, and calculates the time which would be required for the transfer to take place and the I/O complete interrupt time. For periodic devices, such as drums or disks, the latency and access time could be exactly calculated, based on the last reference to the device.
(6) The time of completion is compared with other times of completion still outstanding for the emulator, and the closest time is passed back to the emulator which is informed to report back to the SN program when it gets to this time.
(7) The emulator takes the completion time and includes it in the timing of the machine, so when the completion time arrives, it will report back to the SN.
(8) At completion time, the emulator stops operating and reports back to SN. SN will wait until the actual transfer is completed before restarting the emulator.

The result of this scheme is that the emulation is always in time with data transfer completion and is sufficient for a programmed job which uses completion interrupts to allow continuation of processing. Programming practices such as attempting to process data as it is entered in the system by either checking key words or by seeing the data change will result in a loss of time fidelity. Because of the requirement of exactness, an expensive approach was taken to monitor instructions for references to the data area with a check back to the SN for a delay to bring data and time into place.

PERFORMANCE MONITORING

In order to assist in the evaluation of the TACFIRE system, performance monitoring data will be collected via the system emulation. The utilization of a software performance monitor is normally accompanied with a loading of the system due to the measurement techniques. Having preserved the characteristics of the L-3050 system, i.e., exact results and timing, it is relatively easy to collect performance monitoring data without additional loading of the software.

Key to all of the performance monitoring statistics is the time of occurrence of an event. Using the emulated clock approach, the emulator always has the precise time to a microsecond resolution. All performance monitoring information is passed to the SN of the MP-60 with the emulated time and an indication of the event occurring. Among the functions to be monitored are the following:

(1) Interrupt processing, accountability by event, CPU, program, etc.
  (a) Internal.
  (b) External.
(2) Execution times (e.g., overhead processing, time in loops).
(3) Priorities on I/O.
(4) Instruction usage.
(5) Core utilization.
(6) I/O device utilization.
(7) Overlap between processor and I/O.
(8) Time within a unique programming level (e.g., time spent in executive).
(9) Wait time in queues.
(10) Processing utilization.

The performance gathering facility is both within the firmware of the CPU and within the SN, depending upon where it can best be obtained. Several specific examples follow.

Interrupt Processing: Internal interrupts are executed in the L-3050 by an automatic transfer to program level 2 or by a trap to the instruction in register 15. This information is gathered from the emulation by reporting to the SN when any of these events occur. Information supplied to the SN includes

- emulated time
- type of action.

The SN program may request additional information from the emulator such as setting of instruction counter, contents of indicator register, level trapped from, etc.

External interrupts are indicated by an automatic change in program level. However, in this instance, the SN is the initiator of the external interrupt, so effectively, no information is required from the emulator. However, the emulator will be reporting level changes and the following information:

- emulated time
- type of action
- setting of instruction counter
- level trapped to and from.

Execution Times: The question of execution times depends on the fineness of detail required. The recording of level changes and the time of the changes will give an indication of the time spent in a level. How much time is spent in a particular portion of the program is provided by a breakpoint facility in the emulation where a number of breakpoints may be set up by the SN program in the emulator, and each time the instruction counter executes from one of the breakpoints, a report is made back to the SN.

Instruction Usage: This data is provided by using an op code frequency counter in the emulator. The SN program can request a dump of the cells.

Core Utilization: The amount of core actively utilized and the percentage of time used is determined by a jump trace capability. The emulator reports to the SN each time the instruction counter is changed due to a jump being taken or a program level change.

I/O Device Utilization: This information is developed in the SN since each I/O operation is processed by the SN, and the time of each I/O request, the time of completion, and the type of request is known. A report is then generated on the type of usage and the time in a wait status.

Time within a Unique Program Level: The level change report gives the time of entry into a level and the time of exit from the level. This raw data provides the necessary data for the report.

Processor Utilization: Within the L-3050, normally a unique level is utilized as an "idle" level which is used when no jobs are run or can proceed prior to some event. Monitoring of this level gives processor utilization data.

FUTURE PLANS

The TACFIRE example described (namely, emulating the total system and obtaining performance monitoring data) is but one of several applications of the MMP. Our immediate efforts are concerned with the "fine tuning" of both the microcode for the emulation and the performance monitoring data gathering. The example given does not describe the full capability of the entire MMP system in that only one emulation is taking place. Ultimately, we plan to simultaneously emulate two systems (e.g., TACFIRE and TOS) to analyze their interoperation.

NOTE: The material and references to commercial companies contained in this report should not be construed as policy of the US Government or the US Army Electronics Command in particular.

REFERENCES

1. MP-60 Computer System Reference Manual, Control Data Corporation, June 1973.

2. Control Data 5600 Series of Microprogrammable Processors Reference Manual, Control Data Corporation, August 1972.

3. A. Salisbury, A Study of General-Purpose Microprogrammable Computer Architectures, Stanford Electronics Laboratory Technical Report No. 59, July 1973.

4. J. Pederson, Control Data Corporation Internal Notes.

5. G. Chapin, "What is Different About Tactical Military Operational Programs" in 1973 National Computer Conference, AFIPS Conference Proceedings, pages 787-795.

6. W. Phillips, "What's Different About Tactical Executive Systems" in 1973 National Computer Conference, AFIPS Conference Proceedings, pages 811-816.