The CPM-X—A systems approach to performance measurement

by RICHARD J. RUUD

Allied Computer Technology, Inc.
Santa Monica, California

INTRODUCTION

Hardware monitors of one kind or another have been employed in the instrumentation of general purpose data processing equipment since the early 1960's. Until 1969, however, the production of hardware monitors was, in the main, confined to computer manufacturers and research projects. In 1969, the performance measurement industry was created through the introduction of relatively standard commercially available hardware measurement products.

With the advance of the state of the art of measurement and the increase in sophistication and skill of the user from 1969 to the present, it became necessary to design and build a new generation hardware monitor. This monitor is known as the CPM-X.

This paper is divided into two parts. The first section will discuss the goals and rationale involved in specifying the CPM-X. The second part will discuss the architecture and implementation of the CPM-X in light of the goals previously discussed.

SYSTEM DESIGN GOALS

The basic monitor

The CPM-X's minimum design goal was to provide the user all of the standard features normally expected on a commercially available performance monitor. Such a system is illustrated in Figure 1.

The basic features provided with this typical system consist of a number of probes and probe receivers to transmit the data from the host computer to the monitor. The data are then routed through a plugboard containing a complement of combinatorial logic and then routed to a set of counters for event counting and timing. Periodically, the data contained in the counters, together with a real time clock and other optional data for message stamping, would be written to an output media which is almost universally magnetic tape. Display capability is provided through a luminescent binary or digital readout to allow the operator limited real time monitoring of measurement experiments. The display is also useful for checking out plugboard wiring and in diagnosing monitor failures. The last essential part of this typical system is a data reduction program which takes the magnetic tape output and condenses and formats it, customarily on the data processing system which is being measured, for the user's analysis. These functions constitute the minimum acceptable capability in hardware monitor implementation.

Parallel input

A majority of monitors in use today have an additional input source to the serial inputs first described above. This is a parallel input source accepting from 18 to 36 bits of data, customarily from a host computer register such as the instruction counter. In some monitors, this data is presented to a comparator for comparison against preset, user-entered values. The results of the comparisons are then presented to standard time/event counters.

The other common use of the parallel input feature is as input to a distribution unit. The distribution unit consists of an arithmetic and logical unit and a set of storage registers. These registers are normally maintained in a core storage unit with an average size of 1,024 16-bit words. The distribution unit is capable of operating in two modes. In distribute mode, the high order 9 bits of the parallel input stream are used as an index value to address the storage unit. When a word is selected by this index value, it is incremented by 1, thus providing a count of the number of incidences of

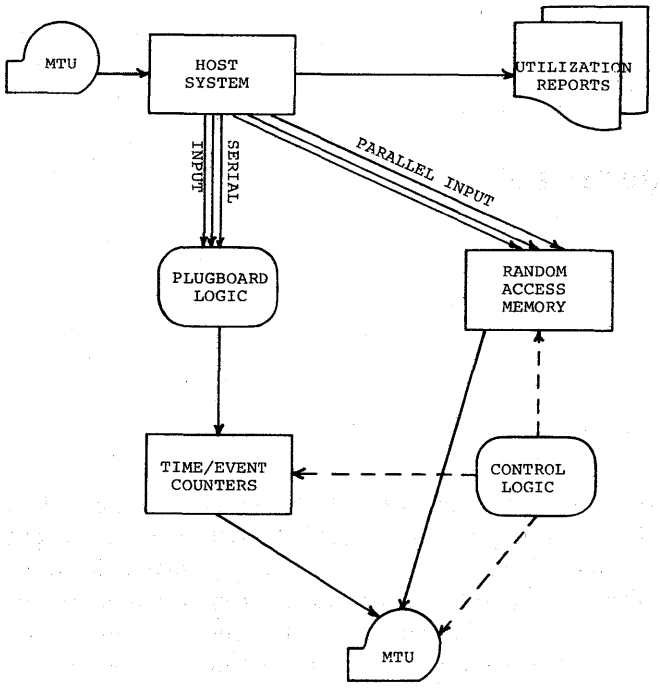

Figure 1—Representative commercial hardware monitor

each 9-bit number recorded by the monitor. In store mode, on the other hand, 16 bits of the parallel input are treated as data and stored directly into ascending word locations in the storage unit. In both modes, contents of the storage unit are periodically written to magnetic tape for off-line data reduction and report preparation. In the implementation described, 512 words are receiving data while the other 512 words are written to tape.

Additional functions

It was felt that in addition to the minimum capabilities first specified, and the parallel input capability, certain other features were desirable and necessary in the design of the CPM-X. These features had been implemented on certain experimental monitor systems, but had not been provided to the commercial monitor user.

First, the user must have the opportunity to reduce the monitor data stream to delineate points of interest and display them in real time from the CPM-X. This is not to say that there is no longer a necessity for logging of detailed data on magnetic tape for off-line report preparation. However, the user must be given an opportunity to react to his measurements in real time and to interact with the measurement system, thereby modifying his experiments as the environment changes.

Second, the user must have the ability to control, either directly or indirectly, through the results of the measurement, the logic of the monitor system. The semi-hard-wired approach of using only a plugboard to control the logic is no longer an acceptable answer.

Third, the monitor must have the ability to capture from the host computer data concerning programs in operation on the host computer in real time. The problem of matching software data to hardware data is a difficult one; yet, it must be accomplished. It is not enough to know merely how a system is performing, one must also know why it is performing the way it is. It is extremely difficult to make these judgments based solely on hardware data; just as it is extremely difficult to make them solely on software data. The CPM-X, therefore, was to be given the ability to interact with the host computer software.

Finally, the interrelated design goals of modularity and flexibility were extremely important to the design. If the wide variety of features specified above were to be made available, it would be necessary to design the CPM-X both for ease of configuration and potential growth without necessitating either major modifications to the system or that it be traded in. The flexibility requirement is mandated by the same set of facts as modularity, not to mention the need for attaching the CPM-X to a wide variety of host computer systems. It has been our experience, as suppliers of measurement equipment, that users discover new applications for hardware monitors every day. Therefore, to make it an effective tool, it was necessary to avoid being locked in, but rather to provide open-ended hardware and software solutions to measurement.

SYSTEM ARCHITECTURE

Overview

The basic system data flow of the CPM-X is depicted in Figure 2. The building blocks of the CPM-X are four basic modules: instrumentation, control, memory, and computation. The instrumentation module contains the probe receivers, plugboard, counters and parallel input interfaces. The core memory may range from 8K to 65K 8-bit bytes depending upon the configuration and needs of the user. The memory has three ports and therefore may be accessed by the control module, the computation module, or the direct memory access channel (DMA). The control module contains registers and an arithmetic and logical unit which are used to control the data flow between the instrumentation module and core memory and also to control the mag-

netic tape drives in the system. Similarly, the computation module has an arithmetic and logical unit and a set of registers which can be used for operator interaction in the on-line data reduction and display function. This module is also used to control all I/O devices other than the tapes, such as display terminals, teletypes, and remote terminals. It can also read and write certain registers contained in the instrumentation module to modify the set-up of the measurement. Although the DMA channel communicates directly with memory, the channel is under the control of the computation module. Interrupt lines are maintained between the computation module and the control module to synchronize their operation.

CPM-X is configured in three basic models. Model A consists of an instrumentation module, a control module, 4K of memory, and one magnetic tape unit. This configuration will provide the basic functions enumerated in the earlier discussion of the typical commercial hardware monitor. This excludes the parallel input function. The Model B consists of the same modules but the memory is expanded to provide for parallel input as described above. In addition, certain field modifications are made to the control module to enable it to perform the distributive control function. A Model C, which is the largest configuration available, adds the computation module to the above. The number of tape drives may then be increased from one to four, if desired. At least one operator's console is required which may be a Model 33 Teletype or a CRT display with a teletype compatible interface. Provision is also made for remote hard copy and CRT terminals in addition to sophisticated local CRT terminals with graphic capability.

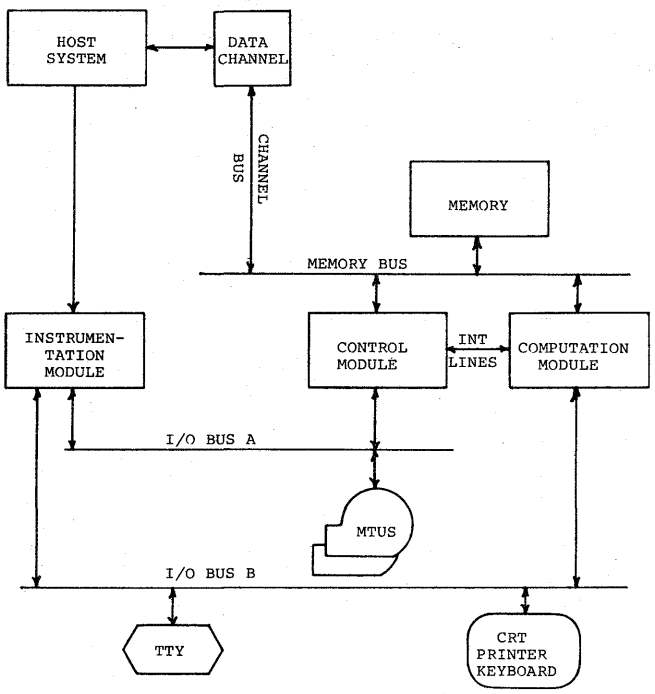

Figure 2—System data flow—CPM-X

The micro-processor

It became readily apparent that the control and computational functions of the CPM-X could most readily be implemented, and at the lowest cost, through the use of minicomputers, since both required arithmetic and logical capability and high speed register storage. It was also necessary to find a mini-computer which employed a three-ported memory because of the design considerations of the CPM-X.

The mini-computer selected was the Micro 1600-D dual processor system manufactured by Microdata Corporation which incorporated all of the required features. The 1600-D system consists of dual processors sharing a core memory which can vary in size from 4K to 65K 8-bit bytes. The system also had provisions for attaching a DMA channel to its memory bus. Each processor had a separate party-line, byte I/O bus communicating directly with its data flow.

An extremely important, perhaps over-riding, consideration in the selection of the Micro 1600-D was the ability for the implementers of the monitor to easily micro-program the processors, rather than depending on the manufacturer of the mini to perform that task. Between 250 and 4,096 16-bit words of ROM may be attached to each processor. Although cost considerations dictated that permanently written read only memory be utilized in field versions of the CPM-X, a writeable control store, exhibiting the same timing characteristics as the read-only store used in the field version, was available on the engineering model and used extensively for debugging the firmware; i.e., microprogrammed algorithms. This writeable control store together with a micro-assembler and micro-simulator greatly aided in the development of the CPM-X.

The basic firmware development strategy was as follows. Processor A, as identified in Figure 3, served the function of the control module. Various algorithms such as counter update, counter concatenate, and distribute, were developed in firmware. Additionally, a comprehensive set of microdiagnostics was developed to test the control module, the instrumentation module, and the tape units.

Processor B, the computation module, on the other hand, was primarily implemented to use the manufacturer's standard, general purpose instructions. How-

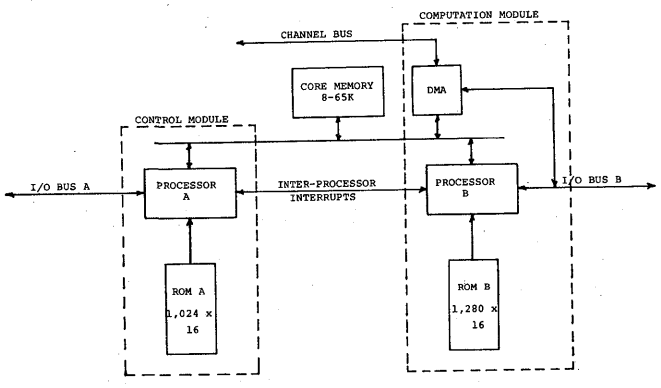

Figure 3—Application of the Microdata 1600-D multi-processor to the CPM-X

ever, this instruction set was modified to provide additional instructions, such as 32-bit multiply divide, an interrupt scheme more responsive to the environment in which the CPM-X would be used, and the various firmware routines for controlling the DMA channel interface with various manufacturers' host computers.

Counter hardware

A CPM-X may contain from 16 to 64 counters. Most counters have but a single function which is to accumulate time or event data from serial probe inputs. Certain counters, however, can perform other functions in addition to that of time/event measurement. The additional functions that these multi-function counters can perform are: Parallel data input to the CPM-X, distributions, and comparisons. The packing of the CPM-X counter groups is such that a standard group contains eight single-function counters. A multi-function counter group, on the other hand, contains four counters, one of which is a standard, single-function counter; one may be used as a 24-bit parallel input and shift register; while two may function as 24-bit comparators.

Figure 4 illustrated the data flow of a multi-function counter group. Since Counter A, in the diagram, functions in exactly the same manner as any other single-function counter, a description of the multi-function counter group, with references to the counters shown in Figure 4, will suffice to describe all possible counter combinations in the CPM-X.

All counters can receive data from the time or count serial input hubs on the plugboard. The count hub merely causes the input signal to be sent directly to the low order bit input of the counter; thereby accumulating a value in the counter equivalent to the number of times the signal has changed state from a logical 0 to a logical 1. The time hub takes the same input signal and uses it to gate the output of an internal or external clock. Therefore, these clock pulses will accumulate in the counter so long as the input signal is in the logical 1 state; effectively turning the counter into a timer.

An additional set of inputs is available for Counters B, C, and D, the multi-function counters shown in Figure 4. This is a 24-bit parallel input bus driven by 24 probes. Counters B and C may also obtain input parallel by bit, serial by byte from the 8-bit I/O output bus of the control processor.

The buffer outputs are available to the control processor parallel by bit, serial by byte on its I/O input bus. In addition, the results of comparisons, when multi-function counters are used in the compare mode, are available at their respective high-low-equal hubs on the plugboard.

In addition to the logic shown in Figure 4, there is control logic which is not illustrated. The control output logic is conditioned by the decoding of the control processor's I/O control register. This logic differentiates between address and data cycles and sets read or write status for subsequent data cycles. When a counter needs service, either because there has been an overflow from the high order bit position or a strobe pulse has been encountered in the case of the parallel input counter, the address of that counter is generated by hardware logic; and an interrupt line is raised to the control processor. This enables the control processor to either perform a counter update

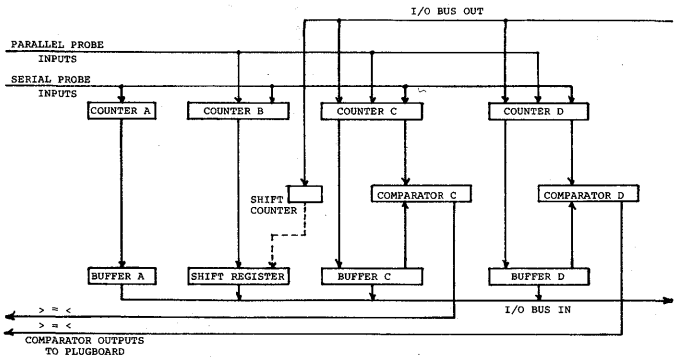

Figure 4—Multi-function counter group

sequence, a distribute sequence, or transfer the information contained in the parallel counter to storage.

Standard counter operation

Standard counters, or multi-function counters operating in the standard mode, accumulate serial input data at a maximum rate of 20 million counts per second. Operating at this maximum rate, the counter will overflow in approximately 3.3 milliseconds. When an overflow occurs, the address generation logic presents an interrupt to the control processor together with the address of the overflowed counter. The processor firmware then utilizes this information to update a 16-bit counter extension contained in core memory; thus, a 32-bit counter is fashioned with the low order 16 bits being implemented in external hardware, and the high order 16 bits being implemented in core storage. The buffer follows the counter and always reflects the exact contents of the counter, except during a dump operation.

When a snapshot of counter contents is to be taken for manipulation, display or recording purposes, the dump sequence is entered. First, the connection from all counters to their associated buffers is broken at the same time by the control processor. This has the effect of stopping the input to all buffers while allowing the counters to continue to accumulate data. The buffers can now be unloaded without effecting data integrity and any problem of time skew is eliminated, since the buffers were all stopped at the same time. The control processor then unloads the two bytes of each buffer, concatenates these two bytes with the high order two bytes of the counter contained in core and stores these four bytes in another area of memory reserved for the purpose. When this update process is completed, the connection between the counters and the buffers is re-established allowing the buffer to be updated to the current counter value. The control processor then presents a microprogrammed interrupt to the computation processor in a Model C. This results in software interrupt being initiated in that processor. In the case of a Model A or B, the completion of a dump sequence initiates a firmware tape-write sequence, thereby recording the accumulated counter data on magnetic tape.

Comparator operations

Counters C and D shown in Figure 4 can also be used as comparators. In this mode, the width of the counter is expanded from 16 to 24 bits. The counter is first loaded by the control processor with a 24-bit comparand. The write sequence automatically causes this information to be transferred from the counter to the buffer where it is held. When the comparator receives a compare strobe signal from the plugboard, the information contained in the parallel probe inputs is gated to the counter; and the contents of the counter are then compared to the contents of the buffer. The high-low-equal result is then latched at the output of the comparator and sent to the plugboard. The result of the last compare is available until a new compare cycle is initiated. Comparators are extremely useful as filters in distribution and save and store mode, as well as being used solely as comparators.

Parallel input operations

Counter B, also 24 bits in width, may be used as a parallel input register to the CPM-X. As in the case of the comparators, 24 bits of data will be gated into the parallel input register upon receipt of a strobe signal from the plug-board. Two events then take place. The interrupt line is raised to the control processor and the address of the parallel input register is presented when the interrupt is acknowledged, allowing the processor to locate the register for unloading. At the same time the processor is being interrupted, the information contained in the counter portion is transferred to the buffer and concurrently shifted right under control of the shift counter. The shift counter is loaded from the processor's I/O output bus, and the count remains constant unless specifically changed. The truncated results are then read from the shift register portion of the counter into the control processor's data flow. The shift function is used to provide an appropriate window in distribution and data storage functions.

The channel interface

In order to close the loop between the host system software and the hardware monitor, the CPM-X, Model C, employs a data channel interface. The interface is unique to the host computer system to which it is attached and therefore requires changes to read-only memory and the hardware when moving from computer to computer.

The channel interface, quite simply, makes the CPM-X look like a standard peripheral device to the host system being monitored. As an example, the at-

tachment of a CPM-X to an IBM System/360 or System/370 may be considered. The CPM-X is assigned a control unit and device address by the installation and is attached to a system data channel through the standard I/O interface connector cables. The monitor's priority is dependent upon the physical position in which it is attached to the channel relative to other control units on that channel, and may be varied to suit the installation's requirements.

Data transfer operations are commenced when the host system issues a start I/O instruction. The CPM-X is selected and the command code is decoded by firmware routines contained in the read-only memory of the computation module. The DMA channel controls, indicating read or write status, beginning memory address, and length, are then set up. Once this is accomplished, the host processor is connected to the DMA channel for the data transfer operation. Since the data is transferred directly to the CPM-X's memory, the rate of transfer can be adjusted and data overruns will not occur.

At the end of the data transfer phase of the operation, signalled either by the expiration of the count in the host computer or an end-of-buffer condition in the CPM-X, the appropriate status indications are generated by the firmware and presented to the host processor so that the I/O operation may be terminated.

In order to prevent the host processor from sending more data than the CPM-X can handle because of software or performance limitations within the computation module, the channel-end and device-end bits are not presented at the same time. When an operation is completed, the channel-end bit is routinely returned with ending status by the firmware. However, the software program must issue a specific instruction to present device-end to the host computer. This has the effect of making the CPM-X appear busy to the host computer should it wish to initiate additional data transfers before the CPM-X software is ready to accept them.

Should conditions occur asynchronously in the monitor which require that the host processor be alerted, CPM-X software can issue an attention interrupt. Upon recognition of this interrupt, the CPU can initiate a read operation to the CPM-X to determine its cause.

The channel interface provides a complete two-way communication path between the host processor and the CPM-X. Although the IBM 360/370 interface has been described, the principles enumerated are applicable to other manufacturers' hardware and channel interfaces are currently offered for both Univac and RCA computers.

Plugboard control

In order to provide communication between the hardware of the instrumentation module and the software of the computation module, the plugboard control interface was developed for the CPM-X. This gives the computation module the ability to modify hardware setups and also to recognize the occurrence of significant hardware events. There are two sources for output data and two sources for input data.

The command register is used as a source of pulse information from the computation module. Its 8 bits are set by command byte and remain set for a period of 200 nanoseconds. The outputs are used where momentary signals are required for such purposes as setting or resetting latches and resetting counters to zero.

The program register, on the other hand, while also 8 bits wide remains set to the bit configuration with which it was last loaded until reloaded by the computation module. Primary use of the program register is to modify the hardware setup by conditioning and deconditioning AND gates which in turn will alter the manner in which data or control signals are routed through the plugboard.

One source of input to the computation module is the multi-function register. This register is 16 bits wide and is read into the computational processor on demand. Inputs to the register may be probes for statistical sampling operations or any other signal available at the plugboard.

Finally, a variable number of interrupt hubs are provided which, when impulsed, cause the computational software to interrupt to a fixed location. These interrupts are used by the software to recognize exceptional hardware events occurring in the host processor for recording or monitor action.

Software support

The CPM-X, Models A and B, do not require internal software support. All algorithms required for operation of the control processor are contained in its ROM and core is only used for data storage. The Model C, however, will have a complete monitor operating system. All models of the CPM-X are supplied with versions of the Measurement Summary Report program to enable the user to perform data reduction and report writing functions on the host CPU.

The basic software design is that of an interactive interpreter. Through it, and some of the hardware features described above, the user will be able to exercise control over all aspects of the measurement

system. The software interface will either be through a teletypewriter or teletypewriter compatible CRT terminal.

The system is written in assembly language and occupies 3,000 bytes of storage. It is expandable and routines can be added by the user as required; however, full user capability for on-line data reduction and display is provided by this basic system.

The software support provided for the host computer to utilize the channel interface is limited. In the case of IBM OS support, the channel interface must be programmed at the EXCP level. Error routines for the CPM-X will be provided for inclusion in the operating system. The level of implementation for other manufacturers' host processors will be similar to that provided for IBM.

Parallel input applications

This family of applications will be described in some detail since they best illustrate the interaction of all elements of the CPM-X. These applications can be performed with a Model B or a Model C; however, limits and buffer size must be set manually in the Model B configuration. The applications described will therefore cover the Model C to illustrate the use of the computation module.

The basic strobe signal used to gate data into either the comparators or the parallel input register is usually obtained from a host computer signal which indicates that the register being inspected by the monitor has just been changed and that the new data is valid. Generally, comparators use this raw strobe signal so that every time a register is changing, a new comparison is being made; and the output latches are set accordingly. These latches may be used as direct inputs to counters to indicate the frequency with which certain data are occurring or their time duration. The outputs of multiple comparators can be wired together through plugboard logic to provide high and low limits and to indicate when data fall between these limits. This wiring scheme is usually used when comparators condition the strobe input to the parallel data register. Thus, limits are set by the computation module's software and the only time that data is accepted by the parallel input register is when that data falls within the range defined by those comparators.

Store mode

An area of main storage is allocated by the software as a data buffer. This storage area is divided in half to provide for double buffering. The word size of the buffers may be either one, two, or three bytes, depending upon the needs of the experiment. In a store mode operation, where a three byte word size is indicated, the total contents of the 24-bit parallel data register are transferred to the shift register and then to main storage. Each register transfer initiates a storage cycle whereby the data element is stored sequentially in the main storage buffer until that buffer is filled. Then an automatic buffer switch takes place and the software is alerted that a buffer has been filled. Store mode allows any sequence of data elements to be recorded for future analysis by the CPM-X. These elements may consist of instruction addresses for detailed trace operations, instructions themselves, data elements, or any other data stream the user might wish to specify.

In order to economize in both storage and speed, the user may wish to specify a basic data element of less than 24 bits. In this case, either the low order two bytes or the low order one byte of data is transferred to main storage. To enable any 8 or 16 bits out of the 24 to be recorded, a value is loaded into the shift register causing the specified number of bits on the right-hand side of the data element to be truncated; thus, any contiguous portion of the parallel input register can be recorded.

It is often desirable to record data only when it falls within defined limits. For this type of filtering, the comparators are employed. Two comparators are set with the upper and lower bounds, and the comparators' outputs are so wired as to only allow the parallel input register to be strobed when data falls between them.

A more complex variation of the store mode is that of sequencing, in which the experimenter is interested in retaining a previous event; but he only wishes to record it if one or more predefined, subsequent events take place. The common usage of this mode of operation would be in determining the location from which a sub-routine is being called. When a sub-routine of interest is loaded into core, the address of its entry point can be transmitted over the channel interface to the CPM-X; it is then loaded into a comparator. The plugboard is wired so that every address is recorded by both the parallel input register and the comparator. The address contained in the parallel input register is transmitted to the shift register and held there. When the next address in the stream arrives, it is compared to the contents of the comparator. If a match exists, the parallel input logic is allowed to interrupt the control processor and, thereby, transfer the contents of the shift register to main storage. This has the

effect of transferring the calling address to main storage, if that particular subroutine is entered. Multiple sequencing can be effected by using multiple comparators and cascading their output through latches.

Distribution

The second mode of use for the parallel input register is that of distribution. In this mode, a portion of the data ingated to the parallel register is used as a displacement in addressing main core. In this case, the buffer area in main core is divided into a number of logical accumulators. The accumulators may be 8, 16 or 32 bits in width. A buffer area is then defined by the user which must consist of a number of logical accumulators which is an integral power of two. Two such buffers are required to provide for on-line data reduction or transfer to tape without loss of data.

Distribution data are normally storage addresses although other data, such as operation codes may be used. Considering the case of storage addresses, it is often of greater interest to evaluate in detail the distribution of references to one or more sub-divisions of main storage than to make a gross evaluation of all of the storage available on the system. When it is desirable to sub-divide storage in this manner, the comparators are employed, as in the case of store mode, to define the boundaries of the storage area to be inspected. In the following discussion, L1 will indicate one boundary and L2 the other boundary of such a sub-division.

In many cases, it is impossible to allocate logical accumulators to individual data elements on a one-to-one basis because of the mis-match between the size of the memory area to be inspected and the amount of CPM-X core memory available for accumulators. When the number of discrete storage addresses exceeds the number of logical accumulators available, the occurrence of two, or more, adjoining data addresses must be summed in a single accumulator. A parameter known as the Resolution Factor has been defined as the number of data element occurrences which are summed in a single logical accumulator. A Resolution Factor of unity is ideal and the resolution of an experiment is inversely proportional to the Resolution Factor.

The boundaries of the storage area to be inspected are often dictated by factors beyond the experimenter's control, such as the size of a given user program or the executive. The experimenter is then generally interested in examining this area with the highest possible degree of resolution. The shift counter of the Parallel Input Register is employed to accomplish this. The value of C to be placed in the shift counter is calculated by the computation module based on the parameters of the experiment and is defined as the whole number less than or equal to:

$$C = 1 + \mathrm{LOG}_2 \left(\frac{|\,L1 - L2\,|}{\text{NUMBER OF ACCUMULATORS}} \right)$$

Once the shift count is computed, the Resolution Factor can be defined as: RESOLUTION FACTOR $= 2^c$.

In a typical application of distribution, the CPM-X might be used as follows: The experimenter would first enter the name of the job to be analyzed at the CPM-X console; this information would then be transferred to the executive of the host CPU via the channel interface. When the selected job is loaded into the host computer's memory, the boundaries of its core region would be returned to the CPM-X over the channel interface, together with an imperative to start address distribution. The computation module would then compute the shift count and load it, together with the boundary conditions into the shift counter and comparators of a multi-function counter group. The computation module would then order distribution to commence. At the termination of the job, the computation module would again be alerted by the host computer via the channel interface. The distribution would be wrapped up and required housekeeping performed.

CONCLUSIONS

The CPM-X design was based on three things: a survey of what was currently available commercially; techniques developed in experimental monitor systems; and the needs of the experimenter.

The key to the architecture of the CPM-X was the total integration of a duplex mini-computer into the hardware monitor. The Microdata 1600-D was selected primarily because of the ease with which it could be micro-programmed by the implementers. This decision has, in fact, materially reduced both time and cost in the development of the CPM-X. Furthermore, the utilization of interchangeable read-only memory modules, together with interchangeable hardware has greatly alleviated the problem of model changes and of interfacing the CPM-X to a variety of host computers.

Some of the more complex measurement applications have been described to illustrate the interaction between various elements of the CPM-X system.

ACKNOWLEDGMENTS

The author would like at this time to acknowledge the substantial contributions made by several individuals

in the design of the CPM-X. Messrs. Paul E. Chanel and Mark J. McGrew, of Allied Computer Technology, Inc.; both made numerous contributions to the original specifications of the CPM-X. Mr. Louis Gallenson, formerly of Allied Computer Technology, Inc., and currently a member of the Information Science Institute of the University of Southern California; contributed much to the early engineering design of the CPM-X, especially in the multi-function counter area. The overall design was carried out by Mr. Thomas O. Ellis, formerly of the Rand Corporation, and currently a member of the Information Science Institute of the University of Southern California. Finally, Mr. Frank M. Stepczyk, of Allied Computer Technology, Inc., was responsible for the definition and creation of the software routines for both the Microdata 1600-D and various host computer channel interfaces.

One of the experimental monitor systems analyzed on the CPM-X development is ADAM designed by Mr. James Hughes and his staff at Xerox Data Systems, Marina del Rey, California. Since no documentation has been published outside of XDS, the author can only thank Mr. Hughes for his discussions of ADAM without quoting a reference source for the reader.

REFERENCES

1 ALLIED COMPUTER TECHNOLOGY INC
Computer performance monitor II: systems summary manual
1969

2 R A ASCHENBRENNER et al
The neurotron monitor system
AFIPS Fall Joint Computer Conference 1971

3 G ESTRIN et al
SNUPER computer
AFIPS Spring Joint Computer Conference 1967

4 INTERNATIONAL BUSINESS MACHINES CORPORATION
IBM System/360 and System/370 I/O interface channel to control unit
Original Equipment Manufacturer's Information Form Number 6A22-6974-0 1971

5 MICRODATA CORPORATION
Micro 1600/21 Micro 821
Computer Reference Manual Form Number 71-1-821-001 1971

6 MICRODATA CORPORATION
Micro 1600
Computer Reference Manual Form Number 71-1-1600-001 1971

7 MICRODATA CORPORATION
Micro 1600-D dual processor system
Internal Document Number PS20002400 1972