MICROPROCESSOR-BASED DIGITAL CONTROLLER FOR THE ADVANCED TELEMETRY TRACKING SYSTEM

MARCELO V. ROSALES
INSTRUMENTATION DIRECTORATE
WHITE SANDS MISSILE RANGE, NM

ABSTRACT

This paper discusses the design and implementation of a microcomputer system that functions as the central processing unit for performing servo system control, tracking mode determination, operator interface, switching, and logic operations. The computer hardware consists of VMEbus compatible boards that include a Motorola 32-bit MC68020 microprocessor-based CPU board, and a variety of interface boards. The computer is connected to the Radio Frequency system, Antenna Control Unit, azimuth and elevation servo systems, and other systems of the Advanced Transportable Telemetry Acquisition System (TTAS-A) through extensive serial, analog, and digital input/output interfacing. The software platform consists of a commercially-acquired real-time multi-tasking operating system, and in-house developed device drivers and tracking system software. The operating system kernel is written in assembly language, while the application software is written using the C programming language.

To enhance the operation of the TTAS-A, software was also developed to provide color graphics, CRT menus, printer listings, interactive real-time hardware/software diagnostics, and a GPIB (IEEE-488 bus) interface for Automated Testing System support.

KEY WORDS

Microprocessor, Real-Time Computing System, Tracking System Controller

INTRODUCTION

The Instrumentation Directorate has recently implemented a microprocessor-based digital controller (MBDC) that provides the means for performing telemetry signal acquisition and tracking. The MBDC performs real-time computations and logic functions, interfaces to a variety of subsystems, and communicates to the outside world via a serial data link. This paper describes the functional requirements, developmental approach, hardware design, software design, and implementation of this microcomputer system.

DEFINITION OF REQUIREMENTS

The MBDC provides the logic and control for the TTAS-A to acquire, track, and receive telemetry data from airborne targets. Also, extensive diagnostic software is included to provide information about system performance and facilitate maintenance of the TTAS-A. The main functional requirements established for the MBDC are listed below:

a. Interactive Operator Interface Through the Antenna Control Unit (ACU) and the System Console. The ACU houses the displays, indicators, switches, etc., that are used by the TTAS-A operator for operating the TTAS-A antenna pedestal. The MBDC contains the software to read and validate the status of the input devices on the ACU; it also outputs display information for the readouts, meters, and indicators located on the ACU front panel. The system console is used to display menus and options, allowing the operator to interact with the MBDC.

b. Interface to the Radio Frequency (RF) Sub-system. The tracking error signals from the RF tracking demodulator are used by the MBDC to generate a drive voltage for the servo system when the operator has selected the AUTOTRACK mode. The AGC signals from the RF receivers are used to determine the validity of the AUTOTRACK signals.

c. Interface to the WSMR Precise Acquisition System (PAS). The PAS consists of instrument pointing data provided by the WSMR Range Control Center. The PAS is implemented as a serial message that contains the site identification, synchronization bits, and pointing information for tracking instruments. Along with the serial data, there is a 20 pulse-per-second clock signal that is used to synchronize the MBDC to the PAS data message.

Also, there is a “return” PAS data message called the Instrument Data Message (IDM). The IDM is generated at the tracking instrument and is transmitted to the Range Control Center. It contains the site identification number of the tracking instrument (in this case the TTAS-A-A) and the azimuth and elevation pointing angles of the antenna pedestal. Information about the status of the TTAS-A is also included in the IDM; these include: tracking status, servo bandwidth, and receiver status, etc.

d. Implementation of the Automated Testing System (ATS). The ATS is a collection of hardware and software that the operator uses to test the RF subsystem and the azimuth and elevation control systems under computer control. This includes ACU, antenna pedestal, video monitor, system console, printer, power wattmeter, sweep oscillator, frequency counter, telemetry receivers, and the Signal Distribution Panel.

e. Interface, Switching and Control of the Azimuth and Elevation Servo systems. This includes: selectable servo bandwidth, interface with the pulse-width-modulated servo amplifier, the tachometer, and the synchro-system. The servo system (azimuth and elevation) consists of the following items: power amplifier, tachometer, tachometer attenuator, motor, synchro system, and synchro-to-digital convertor. A PID (proportionality-integrator-differentiator) controller algorithm is executed in the MBDC for position control.

f. Interface to the TTAS-A antenna pedestal. The antenna pedestal includes the motor, gearbox, cable wrap potentiometer, limit switches, interlock switches, tachometers, motors, blower fans, stowlocks, electromechanical brakes, and other items.

g. Input 20 pulse-per-second (pps) strobe signal from the Basic Timing Unit. This signal provides the event trigger that is used to execute one iteration of the real-time servo control software. The 20 pps signal can be time synchronized to IRIG timing, if required.

DESIGN PHILOSOPHY

Based on the requirements discussed above, and past experience, we decided to design the MBDC using off-the-shelf products. This was accomplished by using commercially-available hardware (based on the VMEbus specification), a commercially-available real-time operating system (PDOS - Eyring Inc.), a C compiler, and a VMEbus-based computer (FORCE Computers) that was used as the development system and the target system. Using this approach, we were able to concentrate on developing the application software.

HARDWARE DESIGN

The MBDC consists of different boards that comply with the VMEbus standard. The following VMEbus boards are used in the computer system:

a. VMEbus system controller - bus arbiter, real-time clock with on-board battery backup.

b. CPU board - MC68020 CPU, 25 MHZ clock frequency; floating point coprocessor, 25 MHZ clock frequency; data transfer size: 8, 16, 24, and 32 bits; two serial RS232 interface ports; 1 Mbyte of static random-access memory (RAM); status indication light-emitting-diodes; VMEbus compatible.

c. Memory board - read-only memory (ROM); RAM; on-board programming logic for erasable-programmable ROM, on-board battery backup for static RAM's; VMEbus compatible.

d. ISCSI (intelligent small computer serial interface) - Hard disk controller, floppy disk controller; VMEbus compatible. This board is used for software development purposes only.

e. Mass storage - 175 Mbyte Winchester disk, 1 Mbyte floppy disk. This item is used for software developmental purposes only.

f. Intelligent serial input/output (ISIO) - MC68010 microprocessor-based serial data communications controller, dual-ported RAM. Used to receive and transmit PAS data serial messages. Includes firmware that was developed in-house.

g. Analog input/output - 16 input channels, 16 output channels; VMEbus compatible.

h. Digital input/output - 148 programmable input/output channels; VMEbus compatible.

I. Relay output - 32 relays; VMEbus compatible.

j. Optically-isolated input; 32 channels, VMEbus compatible.

k. Dual-channel synchro-to-digital convertor - converts three-phase synchro input to 14-bit circular binary output; VMEbus compatible.

l. VME to IEEE488 bus interface - contains IEEE-488 talker/listener/ controller; manufacturer provides software drivers; VMEbus compatible.

m. Graphics controller - Manufacturer provides software drivers in the form of graphic primitives compatible with the PDOS; resolution: 480 x 640 pixels; RGB video outputs; VMEbus compatible.

n. Prototype board - voltage level conversion: RS-232 to TTL, TTL to RS-232. Used to interface (PAS data) modem to ISIO board; developed in-house.

The VMEbus boards reside in a standard 19-inch rack; a power supply, and cooling fans are included.

SYSTEM INTERFACE

Figure 1, Digital Control Unit and Interface shows the MBDC, other subsystems, and the operator interface equipment. The drawing shows the different types of input and output signals for the system; and it shows the configuration of the VMEbus boards that comprise the MBDC. Since the computer is the focal point of system, it provides a powerful tool for monitoring physical system parameters, handling input/output devices, performing servo control, automated testing, and data display capabilities.

SOFTWARE DESIGN AND THEORY OF OPERATION

Figure 2 shows the functional relationship between the physical subsystems, and the MBDC software. The MBDC software is comprised of real-time tasks and other programs. A task is an independent program that shares the processor with other tasks in the system. Tasks provide the means for dividing the MBDC software into independent, understandable, and manageable modules.

Since each PDOS task is an independent module, it is beneficial to have each task perform a specific interface function. The task can be used to perform input, output, data conversion and formatting, logical, mathematical and other functions.

In the MBDC, a task communication scheme is used that does not require an extensive amount of memory overhead; yet, it provides a means by which the independent tasks can communicate with each other. This was accomplished by defining and allocating a block of system memory that could be accessed by all the tasks. The global memory is a block of RAM.

The memory on the ISIO board is external to PDOS; therefore it can be used without conflicting with the PDOS operating system. A name was assigned to each signal that is input, output, or otherwise manipulated by the computer system. This name (or mnemonic) was used to define and allocate memory for that particular signal.

For example, *a_motor_i is the name of the floating point variable assigned to represent the value of the motor current for the azimuth servo. This analog signal is input to the computer, and stored at the memory referenced by the variable *a_motor_i. Read operations can then be performed to determine the value of that physical parameter.

Two PDOS files were created to define and allocate the global memory described above; these are “DEFINE:C” and “TEMP_MEM:C.” These two files are C language “include files” that are compiled with the target software that comprises each task and program. “DEFINE:C” is used to define and allocate memory for the input/output signals of the computer system. “TEMP_MEM:C” is similar to “DEFINE,” except that it is used to define variables that are used for data formatting, status indicators, etc., that are not directly related to a physical input/output signal.

The MBDC software consists of a start-up program that is automatically executed on power-up. The start-up file then creates tasks and executes a master program that operates through the PDOS command interpreter. In the PDOS environment, one program can call another program. This calling feature was used to develop a control path that permits several programs to be executed sequentially through the PDOS command interpreter. Control is always returned to the calling program so that the TTAS-A operator is always at a predefined state of the operating software.

The MBDC software can be categorized as follows: initialization programs, input/output programs (tasks), real-time servo system controller (task), TTAS-A operator interface programs, and special software.

IMPLEMENTATION

The development/target system is a VMEbus microcomputer manufactured by FORCE Computers, Inc. This system, along with the other VMEbus I/O boards, comprises the MBDC. It is a self-contained system that includes a CPU board, memory board, system controller board, floppy/disk controller board, and a mass storage system.

VMEbus boards that are considered external to the FORCE system are the analog I/O boards, digital I/O boards, relay boards, etc. Software modules were developed that are specific to each board and application.

In order to develop, test, and debug the MBDC a lab configuration was used. The FORCE computer was located in another rack, and was connected to six terminals and a printer for developmental purposes. Initially, the I/O boards were housed in a separate chassis from the basic computer system, since there were not enough slots to accommodate all the different I/O boards. A VMEbus repeater board was used during this phase of the development cycle. A later version of this configuration consisted of a single chassis containing the computer boards and the I/O boards, thus eliminating the need for the repeater board. The mass-storage system (floppy drive and hard disk) was located in a separate VME rack.

As each I/O board was configured and integrated into the basic computer system it became part of the MBDC. Since the MBDC requires input signals for operation, the only way to develop and test it was to provide physical input signals; these input signals were used to exercise the DCU hardware and software. This was accomplished by using data simulators and math modelling software. The lab configuration consisted of interfacing the MBDC to the ACU, the PAS data simulator, a servo simulator, and a hard-wired simulator of the TTAS-A pedestal sensors. In addition, logic state analyzers and oscilloscopes were used to test the MBDC interface capability, and to verify critical timing specifications for the servo control software. Using this platform, a major portion of the software was developed and tested in the laboratory under controlled conditions.

After testing the prototype MBDC with the ACU and the simulators, the lab setup was then dismantled and reconfigured in the TTAS-A. The following equipment was installed: MBDC, ACU, TTAS-A Interface Unit (TIU), Signal Distribution Panel, automated test equipment, color video monitor, console, printer, and the required cabling and wiring.

The MBDC was thoroughly tested in the TTAS-A, to verify that it was working properly. This process was rigorous and exacting. Every signal, hardware connection, computer board, and software program was checked out and debugged. The diagnostic software was used extensively during this phase to insure that the computer and interface were working properly.

For example, to check the ACU interface, the operator selects “HARDWARE DIAGNOSTIC” from the menu displayed on the terminal screen; he then selects “ANTENNA CONTROL UNIT” ; at this point he may select from the following: “AZIMUTH,” “ELEVATION,” “OTHER,” OR “QUIT.” Now the operator can press a switch, move a handwheel, or other input device on the ACU panel, and the “raw” representative reading of that device will be displayed on the console screen.

This diagnostic capability is extremely useful in testing the operation of the input device itself, the cabling in the ACU, TIU, and the interconnecting wiring between the

ACU and the DCU. In addition, this also allows the operator to exercise the input/output VMEbus board. the device driver software, RAM and the CPU board.

The refinement process was accomplished by operating the TTAS-A, observing its operation, and then modifying the software as needed to “fine-tune” the system. The programs were left on the hard disk so that modifications could be made on-line. Since the computer also functions as a development system (as long as the hard disk is connected, and the editor and compiler are installed), software programs can be edited, compiled, and executed -- even in a field environment. Without this capability, software modifications would have to be done in the laboratory and then programmed onto ROM IC’s.

The software was evaluated by operating the TTAS-A in the ATS mode and the tracking mode. Preliminary tests performed so far indicate that the system is functioning properly.

CONCLUSION

The Microprocessor-Based Digital Controller, and its associated hardware and software, has proven to be a successful realization of applying present day microcomputer technology.

The software development cycle was relatively short for a project this size, due to the fact that off-the-shelf hardware was used along with a commercially available real-time multi-user operating system. ID engineers were able to focus on developing application software specific to this project by building on proven technology and development tools, e.g., PDOS, and the VMEbus.

This MBDC development/target platform allows engineers to test, edit, compile, and retest software in the field environment; this capability (for embedded systems) had never been available at WSMR prior to this task.

By concentrating the logic and control functions in software rather than in hardware, there exists a highly flexible means of maintaining and modifying the software if such a need arises.

ACKNOWLEDGEMENTS

The author wishes to acknowledge the support provided by Henry B. Stephenson and Moises Pedroza during this effort. In addition, the following persons contributed to the development of the computer system and the interface: James E. Tillett, Ted A. Marsh, Daniel C. Bissell, Mark T. Ehlers, James W. Cutler and Michael J. Hart; Helene Essary provided editorial assistance.

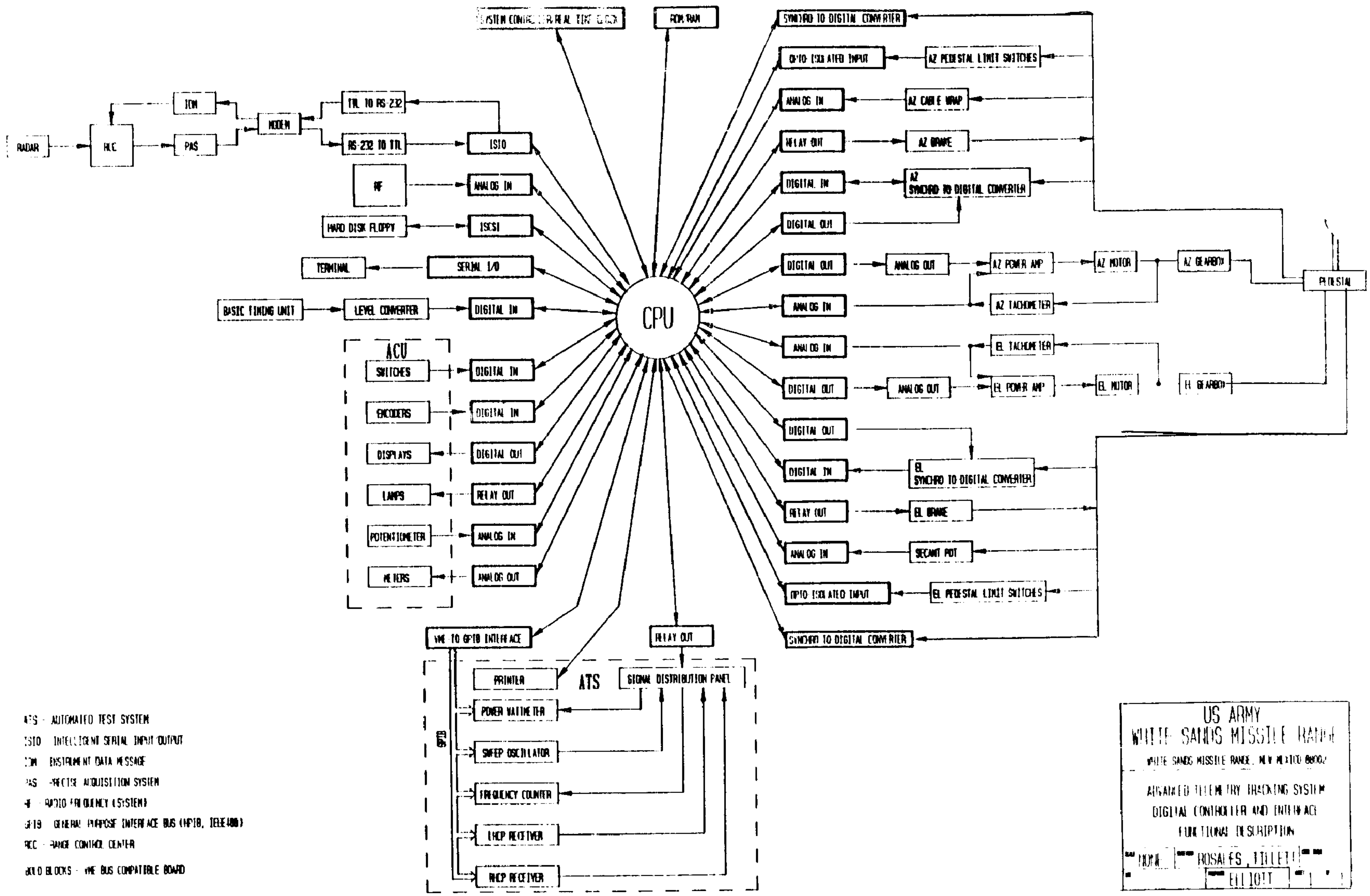


Figure 1. Digital Control Unit and Interface.

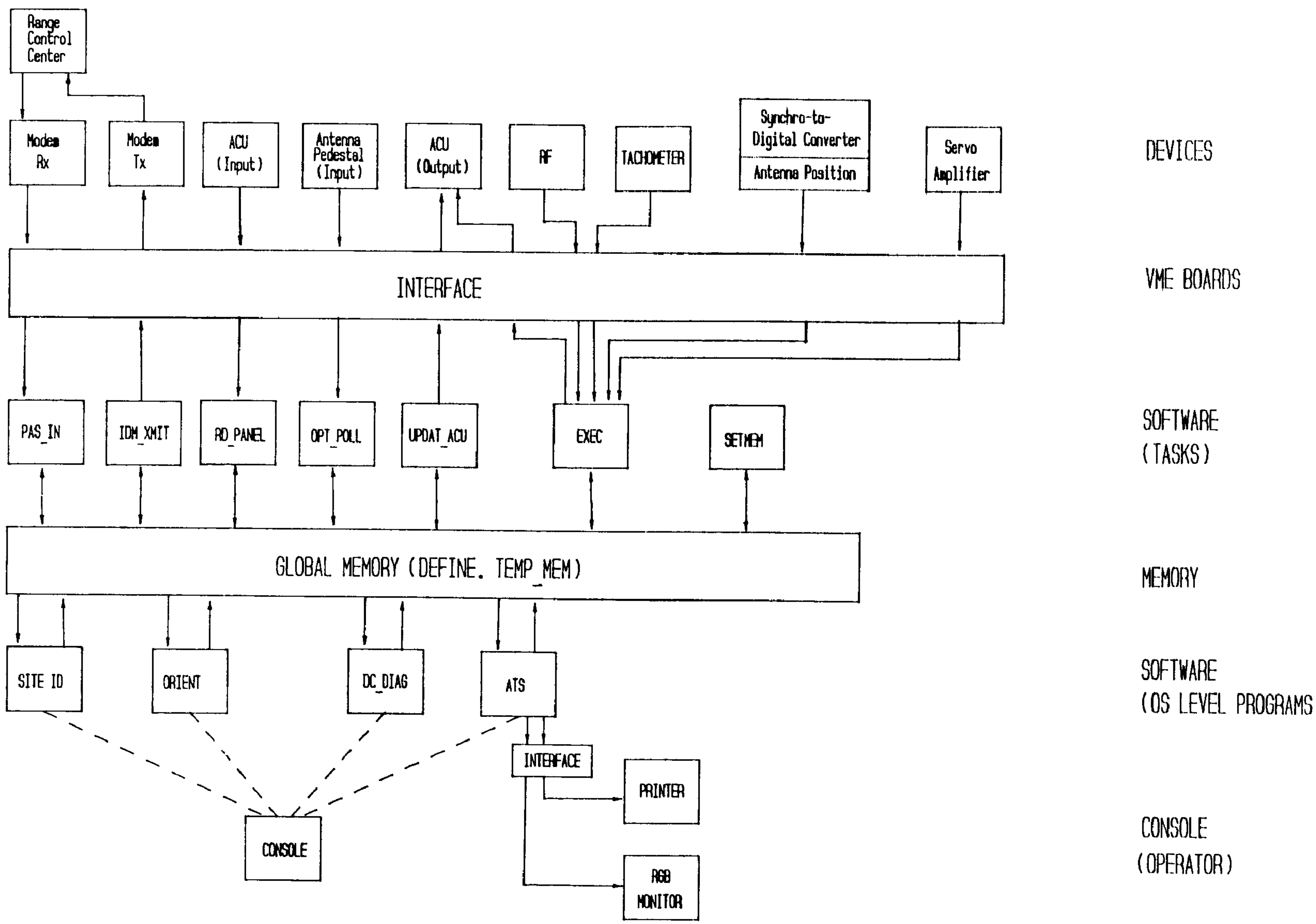
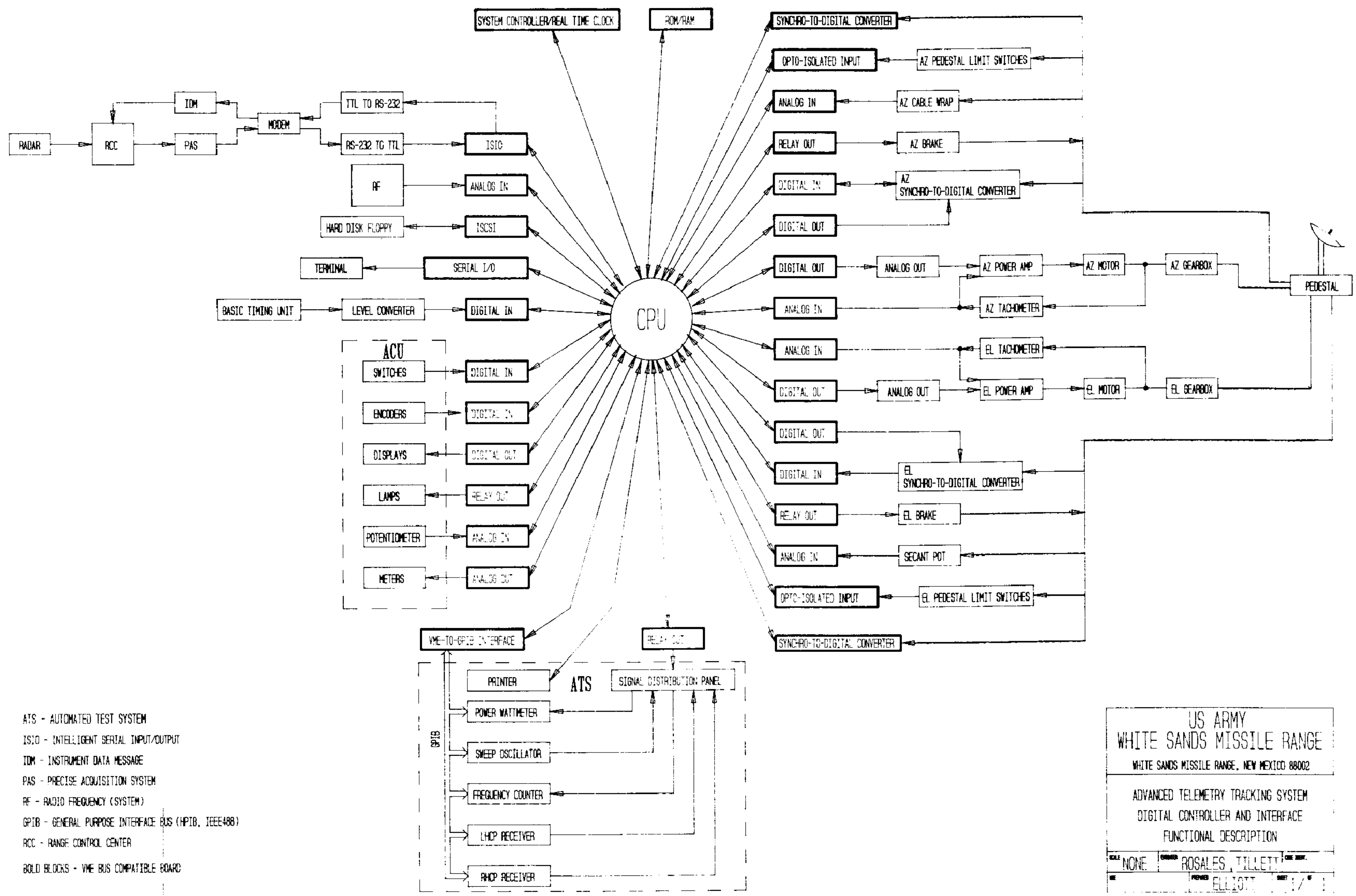


Figure 2. Hardware/Software Interaction



ATS - AUTOMATED TEST SYSTEM
ISIO - INTELLIGENT SERIAL INPUT/OUTPUT
IDM - INSTRUMENT DATA MESSAGE
PAS - PRECISE ACQUISITION SYSTEM
RF - RADIO FREQUENCY (SYSTEM)
GPIB - GENERAL PURPOSE INTERFACE BUS (HPIB, IEEE488)
RCC - RANGE CONTROL CENTER
BOLD BLOCKS - VME BUS COMPATIBLE BOARD

US ARMY
WHITE SANDS MISSILE RANGE
WHITE SANDS MISSILE RANGE, NEW MEXICO 88002

ADVANCED TELEMETRY TRACKING SYSTEM
DIGITAL CONTROLLER AND INTERFACE
FUNCTIONAL DESCRIPTION

SCALE NONE | CHECKED ROSALES, TILLETT | CODE IDENT.
PREPARED ELLIOTT | SHEET 1/1