Ree,
AM SENDING ORIGINAL.
LET ME KNOW YOUR THOUGHTS
ABOUT THE RT/68
Rick

RT/68

REAL TIME OPERATING SYSTEM

MICROWARE SYSTEMS
CORPORATION

P.O. BOX 954 • DES MOINES, IOWA 50304 • (515) 279-9856

RT/68 is a mask-programmed ROM that is designed to serve as the primary console monitor/operating system for the M6800 microcomputer. The console monitor facilitates convenient program loading, dumping, modification and debugging. RT/68's real-time multiprogramming operating system features extraordinary versatility and efficiency in managing up to 16 concurrent tasks.

In addition, RT/68MR is completely software and hardware compatible with the popular Motorola MC6800-L7/L8 Mikbug* ROM used in many M6800 systems. Functional equivalents of every usable Mikbug* subroutine exist at identical addresses in the RT/68MR ROM. Even programs not readily adaptable to the RT/68 real-time executive may be executed properly in the Single Task Mode. In almost every function RT/68 represents a distinct improvement over Mikbug* in terms of convenience, versatility and sophistication.

RT/68 CONSOLE MONITOR

CONSOLE COMMANDS

The console monitor provides ten major functions. The monitor accepts commands from the system console device in the form of command strings consisting of a one letter command code and up to two parameters separated by commas. For example, a command to generate a formatted memory dump on the system console device would be entered as:

D,1ØØØ,1ØFF

All data in the console monitor mode is hexadecimal form. The monitor also provides seven numbered error messages in response to syntactical or functional errors.

The command codes are:

D,aaaa,bbbb	Formatted memory dump to console
P,aaaa,bbbb	Write formatted tape from memory
B,aaaa or B	Set breakpoint or remove breakpoint
M,aaaa	Open memory examine/change function
E,aaaa	Executive single task
G	Run single task from stack; return from breakpoint
L	Load formatted tape into memory
R	Print registers on stack
S	Activate real-time operating system
(ESC)	Patch to next ROM or user defined

CONSOLE I/O

The console routines operate with either a standard Mikbug* PIA-type interface at hex addresses 8004-8007 or an ACIA-type interface at hex addresses 8008 and 8009. The PIA interface must be present for other functions even though it may not be used to handle serial data to the terminal. Two jumpers on PIA inputs CB6 and CB5 determine the number of stop bits and the type of interface, respectively.

The tape I/O routines utilize the standard ASCII codes for tape device control. The tape write routine precedes each tape

write operation for the transmission of a series of null characters that create a leader with paper tape punches, or provide a tape startup delay for audio cassette interfaces. The tape write, tape read, and console dump subroutines are callable from user programs.

PROGRAM DEBUGGING FEATURES

An automatic breakpoint system permits easy insertion and deletion of breakpoints in any program in RAM storage. Upon reaching a breakpoint, the monitor will print the breakpoint address, the contents of all MPU registers, and return to the console mode. The Go command will resume program execution, automatically selecting the proper system mode.

With the addition of an optional debounced switch connected to the console interface PIA, a non-destructive program abort capability may be utilized. The result of this operation is identical to that of the breakpoint feature. This allows interrupting a program without resetting the system, which is particularly useful when debugging programs that use various types of hardware control registers (PIA's, ACIA's, etc.).

PROGRAM EXECUTION MODES

Two execution modes are provided, Single Task and Multi Task. In the Single Task Mode, the RT/68 executive is inactive and the system responds to interrupts with direct vectors as in the Mikbug* monitor. This mode assures that any program designed to run using Mikbug* will operate correctly, even though it may not be compatible with the RT/68 executive. This mode is entered by means of either the GO (restart or start from stack) or the Execute commands.

The system console command activates the Multi-Task Mode and the RT/68 real time executive begins execution, selecting the first "runnable" task and starting it. This process is described in detail below.

RT/68 REAL TIME OPERATING SYSTEM

MULTIPROGRAMMING

A task is some program module of any size or purpose. It may range from a simple peripheral device driver to a complete FORTRAN compiler. The only requirement for a module to be a task in RT/68 terms is that it posess an individual stack (a last-in, first-out type buffer used by the M6800 for subroutine and interrupt processing). Multiprogramming allows one or more tasks to be resident in memory at any one time, and the system's MPU divides it's time among the tasks accordingly. Though only one task is running at any instant, the MPU switches between tasks so rapidly that in terms of "outside world" time it (ideally) appears that all are running at once.

Particularly when peripheral devices have a slow response time relative to the time the task in question requires to service the peripherals, a very high throughput can be achieved in a multi-programming system.

For example, a BASIC interpreter waiting for input from a keyboard may be wasting 99.999% of the MPU time in a wait loop, when it might as well be executing several other programs at the same time. This example is analogous to many other common microcomputer applications that may be consolidated into a single system.

Processing of discrete applications is not the only virtue of multiprogramming. Even if only one application is considered, a multiprogramming operating system such as RT/68 can lend structure and modularity to otherwise long and complex programs by allowing the programmer to segment the problem into smaller, more managable tasks. In addition, RT/68 can handle the difficult "housekeeping" in real time systems, particularly where interrupts are involved.

RT/68 EXECUTIVE

The executive program controls which tasks run when, for how long, and how often. In addition, it "pre-services" interrupts and maintains vital system data structures.

The time interval during which a task runs is called a time slice. This time is variable, depending on a number of factors. The RT/68 executive may determine the order and time slice of each task on the basis of preset values, priority (8 levels), elapsed time, external events (interrupts), program requests, or any combination of the preceding.

RT/68 makes full use of the M6800's powerful stack and interrupt features. It is this characteristic of the M6800 that allows an executive as versatile, efficient, and compact as RT/68.

The executive is active briefly after an interrupt, and between tasks as required. In a typical system the executive consumes less than Ø.1% of the total MPU time.

REAL TIME FEATURES

RT/68, when used with an external timing source (optional) provides several useful features. The time reference is a clock signal, optimally in the range of 1Ø to 1ØØØ Hz (1ms.-1ØØms.). This may be obtained from the AC power supply, the system baud rate generator, or a simple 555 IC timer circuit. The reference clock is connected to the CA2 input of the control interface PIA.

RT/68 maintains a 16 bit counter in memory that is incremented every clock period or "tick". User programs may refer to the RT clock counter for relative or absolute time data. In addition, the clock reference is used to measure the length of each task's time slice.

RT/68 provides for a timed task function to allow precise time delay and measurement. When this feature is active, a 16 bit counter in memory is decremented at each tick. At the one to zero transition of this counter, an internal "interrupt" is generated that is processed identically to other types of interrupts.

An illustration of this feature is a program that implements a digital clock. A task is designated as the "timed task" and is programmed to: update counter variables for second, minutes, and hours; set the task timer to 1ØØ (assuming a 1ØØ Hz clock); and then turn itself off. Each 1ØØ ticks (one second) the task will run when the task timer reaches zero. The task updates the time counters and reinitializes itself. This method can keep time as accurate as the reference clock.

TASK SWITCHING AND SCHEDULING

It is a characteristic of the M6800 MPU that the complete MPU status (contents of all registers) is saved on the stack whenever an interrupt occurs. This is what makes a system, as compact, efficient, and versatile as RT/68, possible.

Tasks are always switched after a clock, software, or external interrupt. Thus, by merely storing and restoring the task pointer associated with each task, tasks may be changed in a "transparent" manner.

Each task's stack pointer and other status information is stored in a table called the System Status Table which is located at addresses AØ5Ø-AØ7F in the scratchpad RAM.

Other task status information in the table includes: an on (waiting to run) or off (inactive or non-existent) status flag; a priority of Ø to 7; and an initial time limit of 1 to 15 ticks, or unrestricted limit. This status data is packed into one byte for each possible task.

After a task is suspended, the executive searches the table for the next runnable task in a round-robin fashion, beginning with the task number following the previous task. A task is runnable if it's priority is equal to or higher than the current system priority and the task status flag is "on".

The systems programmer has much flexibility in using the task parameters to define system operation. In addition, tasks may alter their own, the system's, or other task's parameters. User program callable subroutines in the ROM facilitate these operations.

INTERRUPT HANDLING

Upon an NMI, IRQ, or timed task interrupt, the executive consults one of three interrupt status bytes to obtain interrupt handling information. One byte corresponds to each type of interrupt. The status byte associates a given task with that particular type of interrupt, indicates if the interrupt task is to run immediately or be scheduled, and gives an interrupt prioriy level. If the interrupt priority level is higher than the system priority, the system will assume the new priority level, and the interrupt service task will run (unless deferred by the state of the flag in the status byte). If the interrupt priority is lower than the system priority, the task will be deferred unconditionally.

ADDITIONAL INFORMATION

COMPATIBILITY

As mentioned previously, RT/68MR was designed with much attention paid to compatibility with Mikbug* software. Various editor/assemblers, BASIC interpreters and assorted other software designed for Mikbug* available from several sources has been successfully run without any modification in the RT/68 Single Task Mode. However, most of these programs are meant to be used in interrupt-free circumstances and cannot be run in the Multi Task Mode. The Microware Corporation will soon offer similar software that is compatible, in both firmware and tape media.

Most M6800 systems require only minor hardware modification to support the RT/68MR ROM. Changes or additions to the control interface PIA are for the reference clock input, the abort switch input, and the terminal type jumper. These are all optional but highly recommended, as they tremendously increase the versatility of the system.

Motorola Evaluation Module 1 (MEK6800D1) systems need to have the ROM enable jumper moved from point E2 to E1. SWTPC systems need one small foil cut on ROM address pin A9 and a 3" jumper soldered to enable the full ROM. These changes are necessary because the Mikbug* ROM only uses the lower $\frac{1}{2}$ of available ROM. (The other half of Mikbug* is a test pattern and a 256-byte Minibug program).

Complete details on the above changes/additions are contained in the RT/68MR manual.

MANUAL

Microware Corporation wants to provide the best possible documentation for all its products. The manual contains a complete commented source listing of RT/68MR along with keyed flowcharts and descriptions of subroutines. The rest of the manual covers every aspect of system operation and programming including many program examples.

FUTURE RELEASES

In the event of future releases of RT/68MR, purchasers of the current version will be given exchange credit. The extent and conditions of such credit will depend upon the nature of the changes, in an equitable manner.

WARRANTY/REFUND

If you are not satisfied with RT/68 for any reason, we will promptly refund your money if returned in good condition within 30 days of delivery.

PRICE - RT68MR $45.00 PPD.

DELIVERY - 3 WEEKS ARO

*Mikbug is a trademark of Motorola, Inc.