PRODUCT BRIEF

ROLM MIL-SPEC Computer Systems

ARTS/ ADVANCED REAL-TIME SYSTEM

FEATURES

- **Real-time operating system configurable to meet the needs of a wide variety of applications**
- **AOS system call compatible**
- **Multiprogramming—supports up to 32 real-time processes each with up to 64k bytes of directly addressable memory**
- **Multitasking—processes may be divided into 32 tasks which compete with other tasks and other processes for CPU time**
- **Memory-only operation**
- **Maximum scheduling latency time guaranteed**
- **Maximum interrupt response time guaranteed**
- **Flexible process/task scheduler—one of three scheduling algorithms can be selected at system generation time to best fit the needs of the application**
- **Extremely modular system—operating system size is minimized by detailed system generation procedure**
- **Shared program areas—a process may share pages of program code or data with other processes**
- **Overlays and shared routines—allows the effective address space for a process to be extended beyond 64k bytes**
- **Memory resident files, as well as disk resident files—allows memory-only user to use features normally associated only with disk systems**
- **Buffered and nonbuffered file I/O**
- **Device independent I/O**
- **Interprocess communication facility—allows processes to efficiently synchronize program execution**
- **High-order languages supported—programs written in FORTRAN, PL/I, CMS-2M, and the DG/L™ system programming language are supported**
- **Flexible user device support—user device drivers may reside in the operating system address space or user address space**
- **Command Language for ARTS (CLA)—user level program that allows the user to control and inspect system environment from console terminal**
- **System Utility Programs—includes an interactive System Generation Program (ARTSGEN), Memory Loader, User Debugger, and System Debugger**

DESCRIPTION

ARTS (Advanced Real-Time System) is a real-time operating system for ROLM® Mil-Spec ECLIPSE® processors. It contains the appropriate advanced features of AOS, incorporating them into a compact, modular, and configurable system. ARTS is capable of supporting a wide range of system hardware configurations—from small to large memory capacity and with or without disk storage. It is designed to handle environments encountered in modern military systems without compromising performance. ARTS is compact, powerful, and a first for real-time processing.

Multiple processes with multiple tasks are managed by ARTS with a full complement of real-time capabilities.

ARTS is a subset of the Data General Advanced Operating System (AOS), which provides operating system compatibility for the development of application software under AOS.

AOS COMPATIBILITY

Data General's Advanced Operating System provides the development tools necessary to efficiently develop real-time ARTS application software. An AOS software development facility provides the dynamic environment required for source file editing, for assembly or compilation, and for binding of object files into a program file for execution.

ARTS is a compatible subset of AOS, with an emphasis on real-time response. It supports all of the basic features of AOS, except those which are unnecessary in a real-time system. This compatibility becomes very important during application development since checkout can be done in either the AOS or ARTS operating environment.

APPLICATIONS

ARTS can be used in a wide variety of applications using various memory and peripheral device combinations. The ARTS application environment is characterized by a relatively small number of processes (1 to 32) that assume a known environment. The system can be tailored precisely to an application's requirements ranging from a small memory only configuration to a 2-megabyte system with multiple disk units.

The ARTS system generation program minimizes memory space required for the operating system. The sophistication of this process allows the user to specify system generation parameters ranging from physical memory size to the particular scheduling algorithm to be used. The user selects what system routines are required, what devices are to be supported, and what buffer spaces are to be allocated.

™DG/L is a trademark of Data General Corporation.

®ROLM is the registered trademark of ROLM Corporation.

®ECLIPSE is a registered trademark of Data General Corporation.

Simple memory-only configurations with a single process and a few tasks requiring 32k bytes of user space may need only 64k bytes of memory. A very large system with eight processes, using multitasking, supporting a disk and a variety of peripherals may require a memory capacity of 320k bytes. Of major importance is the configurability of ARTS to meet the precise needs of the application.

MULTIPROGRAMMING

Although the Mil-Spec ECLIPSE processor can execute only one instruction at a time, under ARTS many of the features and advantages of multiple processors can be obtained. This is accomplished by sharing the MSE's CPU and peripherals among several independent programs to best utilize the system resources. ARTS has two programming entities that can be used to achieve the illusion of parallel processing: tasks and processes.

Up to 32 processes can execute in apparent independence of each other under ARTS control. Each process can have up to 64k bytes of its own directly addressable memory, some of which can be shared by other processes. The memory is subdivided into pages of 2048 bytes each (i.e., 32 pages for any one process). The memory pages are allocated as unshared memory which is unique to a particular process, and shared memory which can be used by more than one process according to its needs.

Each process is assigned a priority which determines when the process will receive CPU time. ARTS maintains independent control of concurrently executing processes and protects the processes from one another's activities.

ARTS provides a full set of features to support the efficient operation of multiprocess applications. These include a memory resident file structure, mapped overlays, a cache-like collection of shared pages maintained on a least recently used basis, and interprocess communication.

MULTITASKING

ARTS supports multiple tasks within each process. Some of the advantages of multitasking are that tasks can share the same physical address space, tasks can synchronize and communicate efficiently, and data can be passed between tasks efficiently. Multitasking provides the capability to logically divide a process into independent entities. This subdivision of processes allows ARTS to provide the rapid asynchronous service required by device interrupt routines, time-out routines, and alarm routines which demand quick response. Multitasking provides a structured control of event-driven processes.

Each task has its own Task Control Block, which is a block of data maintained by the operating system with a memory image of the CPU registers and other context data for each task. Since each task has its own program counter, several tasks can appear to run simultaneously, either through a section of reentrant code (i.e., code that is not modified during execution), or through their own unique code paths. What in fact occurs is that the system schedules tasks and allows them to run alternately on the basis of their states and priorities.

A given task may be ready for execution, or suspended and awaiting the occurrence of an event to make it ready, or it may actually be in control of the CPU and executing. At any moment, only one task in one process may be executing. The system switches control from task to task depending on the relation of process and task priorities.

The system provides an elaborate body of task management calls to permit the control of a dynamically varying environment. Task states can also be modified at will, and tasks can be queued for periodic execution based upon elapsed time. Finally, a mechanism is provided for transmitting and receiving messages between tasks.

REAL-TIME CONSIDERATIONS

The execution priority associated with operating system code and user code provides an environment that ensures prompt real-time response to interrupts and a guaranteed maximum latency time for task scheduling. ARTS handles interrupt/device servicing and scheduling at the highest priority level, while system call processors execute at user levels. This allows much of the less critical system code to execute at the same level as user processes, which gives the user more control over scheduling.

All ARTS system code is written to guarantee absolute limits to the time in which interrupts are disabled. This is in support of guaranteeing interrupt response time, scheduling latency time, and orderly shutdown in the case of a power fail.

When processing a system call, ARTS runs at the priority of the process or task that issued the call. This means that the real-time activation of an event-driven task cannot be hindered by lower priority tasks executing complex system calls. The internal structure of ARTS, coupled with the guaranteed interrupts-off time, ensures that the task scheduling latency time is minimized.

A Real-Time Clock (RTC) is supported by ARTS and is used for keeping a time-of-day counter, timing out events, task queuing, user DELAY calls, and scheduler time slicing. The Programmable Interrupt Timer (PIT) is supported as a user device for additional real-time capability.

SCHEDULING

The ARTS scheduler for processes and tasks offers the flexibility to tailor the scheduling algorithm to the particular target application. At system generation time the user can select one of three choices:

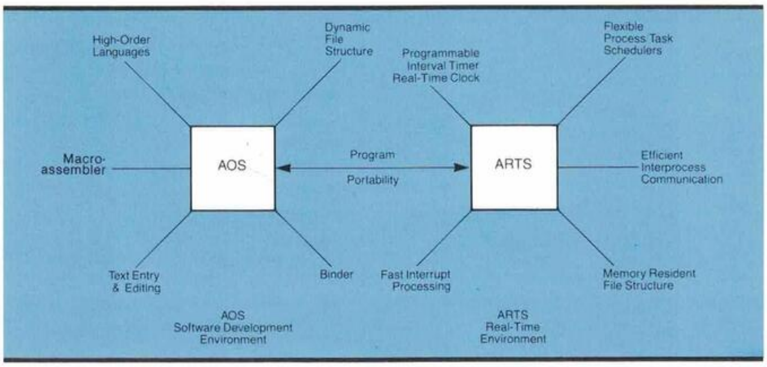
- Scheduling is based solely on process priority (used by AOS).
- Scheduling is based solely on task priorities.
- Scheduling is based on process priority except in the case where there are two processes with equal priority. In this case, the process with the highest priority ready task is executed.

All equal-priority processes and tasks are assured of CPU time to execute by time slicing the tasks. A task, after given control of the CPU, can execute until it gives up control or until a fixed time elapses. The task is then queued behind the other equal priority tasks. The time slice interval is established at system generation time, giving the user another dimension of flexibility in tailoring the scheduler.

INTERPROCESS COMMUNICATION

ARTS provides a facility for processes to communicate with one another by the sending of free format messages of arbitrary length up to 2048 bytes. This allows the capability of sophisticated process synchronization and intercommunication, which is very important in certain process environments.

Interprocess communications are sent between ports that are full duplex communication paths. Each port is identified by a port number, and each process can have up to 127 ports. ARTS also allows each port to be given a name for ease of use. Messages can be sent by using high-level


High-Order Languages
Dynamic File Structure
Programmable Interval Timer Real-Time Clock
Flexible Process Task Schedulers
Macro-assembler
AOS
Program Portability
ARTS
Efficient Interprocess Communication
Text Entry & Editing
Binder
Fast Interrupt Processing
Memory Resident File Structure
AOS Software Development Environment
ARTS Real-Time Environment

calls that treat the interprocess communication like a standard peripheral device. Alternatively, more efficient communication can be achieved by using more primitive operating system calls.

MEMORY MANAGEMENT

ARTS provides a flexible and efficient means for utilizing main memory using the Memory Allocation and Protection feature (MAP), the hardware memory-mapping scheme for the Mil-Spec ECLIPSE computer. The MAP feature allows a user to reference a maximum of 2M bytes of main memory organized in fixed pages of 2048 8-bit bytes.

ARTS apportions total main memory dynamically between itself and user processes. System address space varies with the amount of user process activity, such as open channels and active processes, and with the amount of free (unused) memory available.

Each process is allocated unshared, dedicated memory space, and shared memory space. Unshared memory is that memory space unique to a particular process. Shared memory may be used in several ways: by defining a shared area to the binder, by using explicit shared page management via system calls, by using shared overlays, or by using shared routines.

The positive effects of efficient shared memory usage are twofold: better resource utilization and better system performance. Since memory is a system resource, its usage by more than one process lowers the effective memory required per process. System performance can be gained when shared routines or overlays are merely mapped into a process address space rather than requiring a block move or a disk read.

ARTS also maintains a chain that links, in least recently used order, all formerly shared memory pages residing in memory. This is very important in disk based systems for minimizing disk I/O on commonly used shared pages or files.

FILE MANAGEMENT

A file is any collection of related data treated as a unit. Files are identified by name during the system generation and can be specified as memory resident or device resident. The ARTS file system permits file references to be independent of where the file actually resides: in memory or on a storage device.

ARTS files are allocated as contiguous areas and must be defined at system generation time by name, size, and location. From this data a file directory is generated for use by ARTS.

A wide variety of devices is supported by ARTS, and there are two types of I/O that can be used in device independent file accesses: record I/O and block I/O. Record I/O causes transfers to occur in records of one of three formats:

Dynamic
Fixed length
Data sensitive

Since different devices have various physical characteristics, the particular record I/O that is most appropriate is used. The type of data being transferred may also affect the selection of record I/O. Block I/O is most applicable to disk and magnetic tape I/O when handling physical records. It also provides the user the capability of buffering data within the process address space rather than the system address space.

DEVICE-INDEPENDENT INPUT/OUTPUT

ARTS provides a flexible system for accessing files on peripheral devices. To execute an I/O transfer to any device, users simply open the file, read or write file data, and close the file. Users can read or write data by blocks or by logical groupings called records. This method is used to transfer data to and from all devices.

For all the file I/O operations, dynamic, fixed-length, and data-sensitive logical records are supported. Users can specify the logical record type when creating a file or when performing the input/output.

Block data transfers are supported on disk units, magnetic tape units, and multiprocessor communications adapters. Disk unit blocks are 512 bytes long; magnetic tape unit blocks vary in size; multiprocessor communications blocks can vary in length up to 8192 bytes.

HARDWARE DEVICE SUPPORT

ARTS supports and manages a wide variety of hardware devices that can be configured to meet the needs of a real-time application. Some of these devices are: floppy disks, fixed media moving-head disks, storage module moving-head disks, fixed-head disks, line printers, magnetic tape units, synchronous and asynchronous serial communication devices, and the multiprocessor communication adapter.

The Writable Control Store (WCS) option for the MSE/25 operates under ARTS. The WCS gives users access to the MSE/25 microprogrammed processor. User-defined instructions can execute at very high speed which is a very typical requirement for real-time applications. The WCS can be dynamically loaded with varying user instructions to accommodate change in the real-time environment.

ARTS provides the facilities for the integration of special user devices into the system at two levels of support. Depending on the interrupt response required or the number of processes requiring access to the special device, the device driver can be configured to best suit the situation.

For the fastest interrupt response or for multiple processes requiring the same device, the driver can reside in the unmapped operating system address space. This eliminates any remapping requirement and minimizes other overhead functions. Also, the driver, when residing within the operating system provides the commonality required if independent users require access to a single device.

ROLM MIL-SPEC Computer Systems

4900 Old Ironsides Drive
Santa Clara, CA 95050
(408) 988-2900 • TWX: 910-338-7350

Muehlheimer Str. 54
6052 Muehlheim/Main, West Germany
6108-60935, TWX: 4-189-063

California (714) 770-5606, (714) 741-2801, (714) 840-2275 • **Connecticut** (203) 265-6520 • **Florida** (305) 876-2280 • **Massachusetts** (617) 356-0321 • **New Jersey** (201) 446-3300 • **New York** (914) 739-7727, (914) 297-9533 • **Ohio** (513) 791-6136 • **Oklahoma** (918) 664-7769 • **Texas** (214) 661-8905 • **Washington** (206) 624-4911 • **Washington, D.C.** (703) 750-0300 • **Canada** (613) 234-2626

For situations where the interrupt response is not critical and the device is unique to one process, the driver can reside in the user address space. In either case, the process of adding the driver to the ARTS environment is well documented for the user's convenience.

SYSTEM UTILITIES

ARTS provides several utilities that support the user in all phases of development of the real-time application. The major utilities are:

SYSTEM GENERATION	This program executes under AOS at which time the user specifies all of the system parameters in an interactive mode. System information concerning memory size, devices, files, processes, and system capabilities are input. The ARTS disk file structure is also created by this program. An extensive editing capability which allows the user to conveniently modify the system is supported for system maintenance requirements.
LOADERS	ARTS provides a standalone program that loads the ARTS image from several program load devices.
CLA	The Command Language for ARTS is patterned after the AOS Command Line Interpreter. The CLA lets the user perform file maintenance and process management from the console.
DEBUGGER	ARTS supports both a System Debugger and a User Debugger. The System Debugger will support operating system and User program development, while the User Debugger supports only User program development. The System Debugger can be used in conjunction with the User Debugger to form a very powerful program development tool. The System Debugger takes control of the system, halting all processing while in use. The User Debugger takes control of only one User process and runs concurrently with other User processes.

APPLICATION DEVELOPMENT

Development of ARTS applications is accomplished by using the tools provided by an AOS software development facility. Program source files are generated by using the AOS Text Editing facility, which offers many powerful features. The source files can then be compiled or assembled into object (.OB) files before binding into program (.PR) files. AOS .PR files will run under ARTS.

ARTS has been engineered to be a compatible subset of AOS with an emphasis on real-time response. Typically, ARTS application programs can be developed and tested on an AOS system.

The ARTS system generation program and system build program execute under AOS to create an ARTS memory image. The memory image can then be loaded onto the AOS development system or the target real-time system for application check out.

HIGH-ORDER LANGUAGE SUPPORT

Programs written in a high-level language are supported by ARTS. FORTRAN 5, FORTRAN 77, PL/I, CMS-2M, and DG/L system programming languages provide very powerful development tools for complex applications. The results of using any of these languages include ease of implementing, faster development time, and lower software maintenance costs. As real-time applications become more complex these factors become increasingly important.

MINIMUM EQUIPMENT CONFIGURATION

A Mil-Spec ECLIPSE, 64k bytes of memory, a memory allocation and protection unit (MAP) and a 9-track magnetic tape*. An AOS software development facility running on a Data General ECLIPSE or a Mil-Spec ECLIPSE is required for application software development.

**Any peripheral can be substituted if the AOS development system has a compatible counterpart for program loading.*

SOFTWARE SERVICES

Software Subscription Service—The user is supplied with the latest revision of software and documentation on an as-required basis.

Software Trouble Report Service—Provides the user with service for reporting software difficulties.

Software Training—Training for ARTS is provided by ROLM's Mil-Spec Computer Training Department.

The materials contained herein are summary in nature, subject to change, and intended for general information only. Details and specifications concerning the use and operation of ROLM equipment and software are available in the applicable technical manuals.

Copyright © ROLM Corporation, Santa Clara, California 1981. All rights reserved. Printed in U.S.A.

105001-5/81