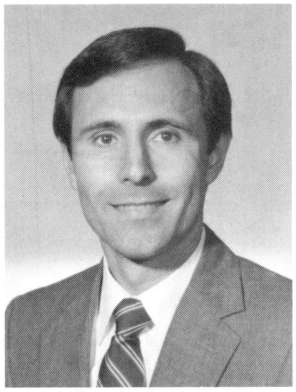
RTX PRESS

Vol. 1, No. 2 **The Latest On Harris Real Time Express™ and Real-time Applications** *September, 1989*

Future Expressions

- The field of real-time microprocessing is evolving to meet the increasing demands of new technology, and with the development and introduction of the RTX 2000™, Harris Semiconductor has taken major strides in shaping the future of that field.



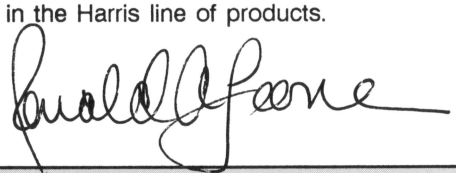
Ronald A. Leone
Vice President,
Microprocessor Product Line

With the RTX™, which is a hardware implementation of the Forth abstract stack engine, we have broken new ground to achieve customer acceptance world wide, with over one hundred design wins for the RTX 2000 product line already secured in North America, Europe, and the Far East. And with a look toward the future, we have begun aggressive development of a 32-bit version, the RTX 4000.

The RTX product portfolio is designed to suit our customers' requirements. These include commercial and industrial applications, military high reliability demands, and quick turn ASIC capability using the Harris FASTRACK™ IC design system.

The team of Harris Technical Support personnel who are dedicated to supporting these product lines has been augmented with additional factory and field engineers in our determination to meet the needs of our customers. With these added resources, we have taken an aggressive stance in our corporate commitment to the RTX product line, and will continue that commitment, in keeping with our long range plans.

The normal life cycle of Harris Military/Commercial products exceeds 15 years, due to the extended product life cycle required for military products. This life cycle factor, combined with the initial market response and the anticipated increase in both acceptance and demand, has made the RTX a permanent fixture in the Harris line of products.



TRACKING THE RTX PRESS

CHUCK MOORE. To many, the name is synonymous with "Forth". This isn't surprising, since his was the mind that conceived the concept and philosophy of Forth. But Chuck did not set out to create a new software language. He set out to find a solution to a problem dealing with scientific calculations. Forth is, simply, a tool that evolved as he worked toward his real goal.

Chuck Moore
President and Founder,
Computer Cowboys

When speaking about the use of Forth, Chuck Moore doesn't just talk about software. He talks about analyzing the basic problem first, and determining what is really wanted, and whether reaching that goal will actually solve the root problem (so often the two are not the same). Most emphatically, he talks about the need for simplicity. In his view, "The solution to a problem lies in factoring the problem - analyzing it and simplifying it." In this way, the simplest, least expensive solution to a problem can be found.

But there is a limit to how much improvement can be found through analyzing the problem and simplifying the software. Nine years ago, in his August 1980 article for **BYTE** Magazine, Chuck wrote, "I think that hardware today is in the same shape as software was 20 years ago. No offense, but it is time that the hardware people learned something about software. There is an order or two of magnitude improvement in performance possible with existing technology." Today, Chuck says that at the time he made that statement, he felt he'd taken the software about as far as it could go on its own. Changes were needed in the hardware.

It was that sense of restriction that motivated his early work to implement a Forth engine in silicon, resulting in the 1985 introduction of the Novix NC4000P (the forerunner of the RTX 2000). With this one exception, he sees little improvement in hardware since 1980. The differences between RISC machines and CISC machines are minor, he says, because the complexity which has been removed from the instruction sets of RISC machines has just been replaced with more complexity in the coprocessor interface. This situation makes for a very big future in closer integration of hardware and software. As the only hardware implementation which addresses the fundamental gaps between hardware and software design concepts the Novix/RTX 2000 design removes some of the transition layers that separate the programmer from the processor hardware. In this way, those parts of the system which make the operation more complex or which cannot be "seen" and therefore cannot be analyzed have been eliminated, preventing unknown factors which can introduce

Copyright (c) 1989 by Harris Corporation

problems in system predictability or repeatability. In fact, Chuck sees the close integration between the hardware and software in the RTX 2000 as a major advantage because of its simplicity and easily understood operation.

Flexibility is another key factor in developing the solution to a problem. According to Chuck, the approach to a problem should depend on that specific problem, and it is a mistake to allow previous methods of problem solving to introduce limitations which hinder *this* solution. To define the most efficient solution to a problem, he says that there must be an underlying willingness to solve the problem in the optimum way (often not the case!). This must be true, even if doing so involves abandoning some or all of an existing method. He suggests that after the problem is analyzed, the previous methods of handling that type should be studied, then only those portions that work well *for this application* should be used.

Chuck says he likes to make changes, to keep moving ahead while doing as little work as possible. He has been known, after long usage, to make name changes that draw protests of "Why change?" His response is, "Why not, since this works better?" In his own approach to solving a problem, he likes to take a broad scale view, then he likes to sit down and write code, jumping back and forth between the broad scale view and coding. He likes Forth because it encourages this, allowing validation to begin very early, while allowing changes to be implemented simply as the solution evolves.

Despite its obvious advantages, this innovative approach can evoke difficulties. The fact is that the acceptance of Forth as a standard in the industry has been relatively slow, because the structured nature of most corporate environments discourages the use of something that cannot conform to established guidelines. The more highly structured the organization, the greater the resistance to using "maverick" techniques. This puts Forth in a difficult light, with its inherent encouragement of change and simplification. The resultant flexibility available to the individual software designer is too great to fit within the controls established through corporate standards. Typically, a Forth enthusiast runs the risk of "rocking the boat" by attempting to use Forth in the work place. On the other hand, the use of Forth in situations where a problem didn't require conformance probably accounts for the tenacity of Forth's presence, and the fierce support it receives in environments where its virtues have been established (who can knock success?). Ultimately, the acceptance of Forth within a corporate structure may depend to a large extent on the two-way communications between management and the software designers who perform the implementation.

With its introduction of the RTX 2000, Harris Semiconductor became the first large corporation to make a major commitment to support Forth as a viable tool in the field of real-time control. Based on early results, that commitment will prove to be a major force in reshaping that marketplace.

The RTX 2000 owes its origins to the inspiration and hard work of Chuck Moore, and those of us in the RTX Market Development group would like to offer him a salute and our thanks, both for taking the time to share his thoughts here, and for his role in making our work more interesting. ■

Would you like more information on the RTX Family of products? If so, fill in the attached reader response card and mail today.

If you would like to submit an article about a real-time application to RTX$_{PRESS}$, send it in. We welcome contributions from our readers.

The **RTX$_{PRESS}$** is published quarterly, and is copyrighted by Harris Semiconductor. Correspondence and submissions should be addressed to: **RTX$_{PRESS}$**, Harris Semiconductor, P.O. Box 883, MS 62A-021, Melbourne, Florida 32902-0883.

Editor:
Monnie Smith

Executive Board:
David P. Barath
Linda daCosta
Michael H. Graff
Ronald A. Leone
David G. Williams

Editorial Board:
Susan Alexander
Robert DeRobertis
Tim Dwyer
Alice Gills
Tom Hand
Ken Lyons
Mike Mellen
Rick VanNorman

RTX™, RTX 2000™, RTXFORTH™, FASTRACK™, and ASIC Bus™ are trademarks of Harris Semiconductor.
SUN™ is a trademark of Sun Microsystems
IBM™ is a registered trademark of International Business Machines Inc

News From SIGForth

The ACM Special Interest Group on FORTH, SIGForth, formed by the Association for Computing Machinery in recognition of the significant and growing impact of the FORTH programming language, will be holding its second annual Symposium on Real-time Software Development -- Tools, Techniques, and Environments, (SIGForth'90) in Dallas, Texas, February 16-18th, 1990, at the Colony Parke Hotel near Downtown Dallas.

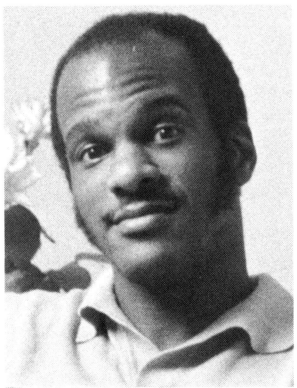
For further information on the SIGForth'90 conference, please contact the conference chairman, Howard Harkness, 3316 Vine Ridge, Bedford, Texas 76021, (817) 545-6767 (home/answering machine) (214) 580-1515 ext. 545 (work). If you would like to present a paper, please ask for author guidelines. The deadline for abstracts is November 1, 1989.

SIGForth membership is open to both ACM and non-ACM members, and includes a quarterly newsletter and substantial discounts on SIGForth publications and events (such as the upcoming SIGForth'90).

For membership information, contact ACM, PO Box 12115, Church Street Station, New York, NY 10249, (212) 869-7440.

RTX PRESS

HARD TRACKS - The National Aeronautics and Space Administration Instrument Division at the Goddard Space Flight Center in Greenbelt, Maryland, has a system under study which uses the RTX 2000. We spoke with **Douglas Ross** about their application, and about how the RTX is used.



Douglas Ross, NASA Goddard Space Flight Center

The Wideband Transport Frame Formatter (WTFF) is being developed as a proof of concept project. This system is intended for implementation in satellites and space stations, and will act as the Ku-band return link gateway for the Tracking and Data Relay Satellite (TDRS). This system is designed to implement the link layer protocol (layer 2) and the physical layer protocol (layer 1) of the Open System Interconnect model, and supports the Consultative Committee for Space Data Systems (CCSDS) for international applications.

The WTFF is a multiplexing device designed to frame and format up to eight user input channels into transport frames. For each user channel, the high rate data (10 to 140 Mbps) can be generated by each of a wide variety of user handling units, each of which may receive its input from a number of on-board experiments.

Data received by the WTFF via an optical fiber link is placed into a user channel block along with various control fields and an identifier which designates the originating user channel. Each transfer frame will contain the data from only one user input channel, which is multiplexed into two bit streams that are compatible with TDRS Ku I and Q band service. To ensure one of three choices for the quality of service, the WTFF provides telecommunication coding, as assigned for each user channel.

Audio data is also accepted by the WTFF system and inserted in the downlink. The user channel block is then encased in a transfer frame for downlink. Subsequent output includes the data, start/end message signal, and a message clock to the Virtual Channel Processor. Data can be processed and downlinked by the WTFF at a combined data rate of 300 Mbps.

Transfer frames from all data channels are multiplexed into a common data stream which is downlinked continuously over the TDRS I and Q channels at a constant output rate of 150 Mbps per channel. If no "true" data is available, an idle channel frame is transmitted.

Service access points into the WTFF are through an onboard local area network, via direct connection to the WTFF with high rate data or video data, or via an audio input channel. The figure opposite shows a simplified functional diagram for data handling in the WTFF system.

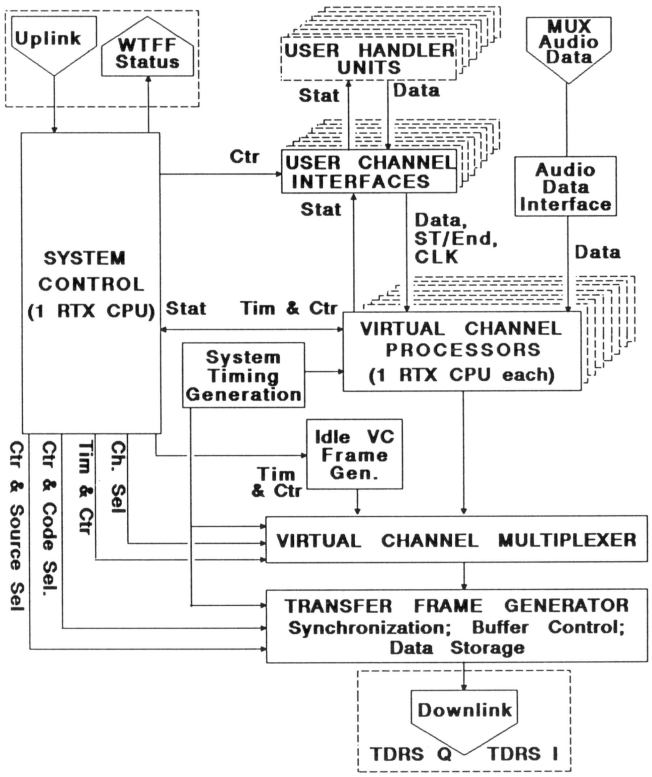
Doug Ross defined several hardware and software criteria that he felt were essential in the selection of a microprocessor for this application which explained his selection of the RTX 2000.

The system needed a software environment that provided simple, compact, and self-contained code, and Doug felt that Forth was best suited to provide those. The RTX was the only microcontroller available for which Forth was the resident language.

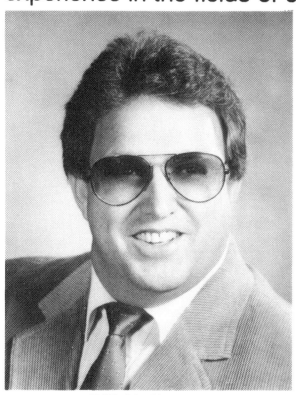
Because of the time critical, interrupt intensive nature of the system and its routines, it was essential to use a high performance microcontroller that could still provide totally predictable performance. The RTX was able to meet these criteria due to its stack-based, non-pipelined architecture.

Power and size are also factors. The use of a low power device is imperative in satellite applications, and this requirement is met by the RTX because it is a CMOS device. Size becomes important when the system is actually implemented in a satellite. The highly integrated nature of the RTX 2000 allows the final board design to become relatively small due to the reduced "glue" logic needed.

The System Controller and each of the Virtual Channel Processors is interrupt driven, and must be able to switch between states readily, to control data paths through the G-port, to route data/signals, and to give hardware control signals. One RTX 2000 is used in each subsystem to perform these operations, resulting in a total of nine RTX microcontrollers used in the full WTFF system. ■



Bill Holloway, President of VME Inc., has 25 years of experience in the fields of computers and computer controlled industrial test systems. He was one of the originators of the VMEbus in 1981 and, as an active participant in the standardization of the VMEbus, he has been a key figure in helping to establish the marketplace held by the VMEbus today.



Bill Holloway
President, VME Inc.

VME Inc., which Mr. Holloway founded in 1984, offers a range of products which fall into three basic categories. The Standard Products Line contains CPU and I/O-board products targeted at the VMEbus compatible marketplace. Their Custom Products are designed and built on a contract basis for use in customer systems. Their Support Products focus on new product development and prototyping tools.

We asked Bill to speak with us about his products and his experiences in using the RTX, and here are some of his responses.

Could you describe the types of applications in which your products are used? *Essentially, we develop products which are used in embedded, computer assisted machines. Most of the equipment which utilizes our products can be classified as "canned" applications. They are typically industrial grade automation equipment, data acquisition or test systems, and special purpose laboratory equipment. Our products also find their way into military support equipment, where VMEbus has become a de facto standard and system requirements include a long life cycle.*

Are there any special requirements for products used in this type of equipment? *The environments in which our products are used tend to be rugged and the applications performance sensitive. The need for dependability and high performance in an arduous environment places a critical demand for high reliability on each component. In addition, because of the automated nature of these systems, operation is often interrupt directed, where overall system performance is directly related to stimulus response times. To provide the level of predictability needed, it is necessary to know the system's response time, and to know with surety the worst case scenarios. Typically, our products are used in control systems that are very unforgiving if a response is late.*

To bring this concept down to a personal level, if you are sitting on a box of dynamite, where protection from detonation is one of several system tasks, you would have a personal interest in whether that portion of the controller responds too slowly. So you look very closely at the performance of the control system. A statistical model of the detonator controller response time is meaningless, because it simply is not enough to know that 90% of the time, the controller you're using has an acceptable response time. Not if the other 10% of the time your fate is up for grabs! Before you sit on that box, you want to know with absolute certainty the worst case controller scenario, and you need to know with certainty the maximum response time requirement. After you know what the worst case response is, you want to know that the controller you have selected will guarantee that level of response. The RTX provides the kind of predictability and precision which supports that level of deterministic behavior.

Could you tell us something about your products which use the RTX 2000? *At the present time, we have two products available which incorporate the RTX 2000. They are the V510 SCSI Channel Processor, and the V401 Single Height Processor Board for VMEbus. Size and performance were crucial factors in the development of the V401 CPU Board, while task management and data handling capabilities were critical in the V510 SCSI Board. Neither product would have been currently possible without the RTX 2000.*

Did using the RTX result in any new features in your products? *Yes. We have been able to solve some design problems in interesting new ways. For instance, in the SCSI peripheral interface area, many engineers are acquainted with the typical "Hurry up and wait" implementation of this protocol. The data transfer phase is quite fast, while everything else is, by comparison, pitifully slow. The problem is that most SCSI controller ICs include a low performance 8-bit microprocessor as the Data Link controller. By directly implementing the Data Link software in the RTX, we were able to obtain an order of magnitude faster channel communication.*

We have also used the RTX as a replacement for standard DMA controller ICs. Utilizing the RTX Streamed Instruction format in a broadside data transfer architecture, we have produced products which perform VMEbus transfers in excess of 33 megabytes per second.

Do you have any future RTX products planned that you're free to discuss? *There is one product that will be available soon. It is a complete 15 MIPS computing platform, including memory, about the size of a credit card, which consumes 250 mA. In addition to offering this product for sale, we plan to use it internally as a foundation module in our future Standard and Custom Products.*

With all of the choices available, why did you choose to use the RTX? *There were a number of reasons. The RTX is a very good high performance processor which offers the performance of a RISC machine without the inherent problems. In my opinion, pipelined architectures and cache memories are simply not compatible with most types of real-time applications. In addition, the short development schedule provided by the RTX could stack up against anything else on the market, plus the RTX product offered the lowest cost development vehicle available. Nothing else provided a faster, more cost effective development environment. Finally, with its high feature content in an 84-pin package, the RTX provided the characteristics we needed in a development tool.*

continued....

RTX PRESS

RTX Bulletin Board

The Harris RTX Electronic Bulletin Board is now on-line. This board is provided as a free service by Harris Semiconductor to support RTX product users. It provides a 24-hour a day center for users to ask questions of the Harris RTX experts, share ideas and tips with other RTX users, find out the latest RTX product information, and download product updates and application code. Some of the features available are:

Conferences - Users may join product-specific conferences where they can share ideas and questions with other product users, and download software updates and applications examples.

Applications - Application notes and programming examples from the RTX Applications Group and code uploaded by other users available for downloading.

Questions - Leave a message with your applications questions; you will receive an answer in 1-2 working days.

New products - Up-to-the-minute announcements on new RTX related products, both from Harris and from third party vendors.

Training - Schedules for RTX and Forth training classes available from Harris and outside parties.

MAKING CONNECTIONS

The RTX Bulletin Board is open to anyone with an interest in the RTX product line. First time callers will be able to browse through the latest product announcements and sample the files available for downloading. Callers who register will be given extended privileges, such as the ability to join product-specific conferences. Registered RTX product users (yes, it does pay to send in the registration card) will have access to even more services, such as the ability to download product updates, application examples and utility programs.

To access the RTX Board, dial 407-729-4949 with your modem set to 300, 1200, 2400 or 9600 baud, 8 bits, no parity. Enter your name, city, and state when prompted. After reading the new user message, select the [R]egister option and supply the information requested. You are now a temporary user, and may scan the public bulletins and messages.

If you wish to become a permanent subscriber, use the [C]omment option from the Main Menu to leave a message giving your name, company (if applicable), and phone number. If you have purchased an RTX hardware or software product, include its name and serial number to gain extra privileges.

MORE USER'S TRACKS

You mentioned the RTX development environment. Could you tell us more about that? The RTX development environment was unquestionably an important benefit during our product development phase. Stephen Pelc, of MicroProcessor Engineering Ltd. has a great presentation which addresses and quantifies this subject. Anyone who has budgetary responsibility for engineering should see one of Stephen's presentations. All it takes is one RTX product design to make you a believer. The time and cost savings are significant.

Are there any additional ways in which the RTX helps to enhance your products? In the case of products where there are size constraints and the amount of additional support hardware must be limited, the high level of integration offered by the RTX, and the subsequent reduction in "glue" logic needed, is a key factor because it negates the necessity for designing custom ASICs.

In conclusion, Bill mentioned that besides the product considerations in choosing a vendor, he feels that the technical support staff involved can make a big difference in how well a development project progresses. He says that the cooperation and willingness to help which the Harris RTX support personnel displayed was a major factor in the timely completion of each project.

It's nice to know when our efforts are appreciated! We'd like to thank Bill for his comments, and especially for sharing his experiences and insights with our readers. ■

Fast Polynomials LOCOMOTION

by Martin Tracy

Did you know that the fastest possible way to evaluate a generic polynomial is given by Horner's method, and that it is especially suited to Forth? For example, you can compute a cosine to ten digits of precision with the polynomial:

$$\cos(x) = c_4x^8 - c_3x^6 + c_2x^4 - c_1x^2 + 1$$

where $0 \le x < \pi/4$ and the coefficients are

$$c_1 = .3084251349 \qquad c_2 = .0158543239$$

$$c_3 = .0003259365 \qquad c_4 = .0000035288$$

Horner's method recasts the polynomial into

$$\cos(x) = (((c_4x^2 - c_3)x^2 + c_2)x^2 - c_1)x^2 + 1$$

which can be easily evaluated with the Forth definition

```
: COS ( r -- r2 )   FDUP FDUP F*  ( x squared )
   .0000035288 FOVER F*  .0003259365 F-
               FOVER F*  .0158543239 F+
               FOVER F*  .3084251349 F-  F*  1.0 F+ ;
```

for a total of five floating point multiplications and four additions.

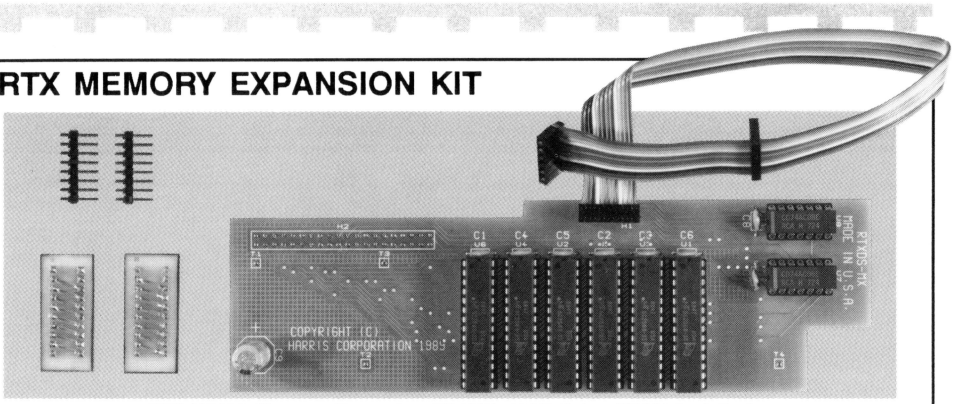
Martin Tracy is an independent consultant working in Southern California. He can be reached at: (213) 371-2499.

For information about the December RTX HW/SW Workshop schedule, call: 1-800-4-HARRIS, ext. 1990 (call 1/800-344-2444 in Canada). For Literature, call ext. 1288.

RTX PRESS

RTX MEMORY EXPANSION KIT

The standard RTX Development Board/System comes configured with 32K bytes of SRAM. Memory expansion kits are now available from Harris which allow easy expansion of that memory with high speed (zero wait state) static RAMs. The RTX-MX64 is a memory expansion board with 64K bytes of 35ns SRAM, and has sockets provided for an additional 128K bytes of SRAM that can be purchased from the memory vendor and installed by the customer. The RTX-MX192 is a memory expansion board with 192K bytes of 35ns SRAM.



RTX-MX192 Memory Expansion Kit With 192K Bytes SRAM Installed

Each expansion kit consists of a small PC board, connectors, ICs, and two 74245 transceiver adaptor boards. Full installation instructions are provided with each kit, as well as a test program. Development Systems with expanded memory already installed (RTXDS-10MX) are available for individuals who prefer not to perform the installation themselves.

RTX PRODUCT UPDATES

RTXDS VERSION 2.0: Will provide additional features such as support for full RTX memory; Single Step; Improved optimizing compiler; support for entire RTX 2000 family. Registered 1.0/1.1 users will receive free upgrade (1 yr limit).

The **RTX 2152** will be available in the second quarter of 1990.

RTX 2001A - Samples of the RTX 2001A will be available in the third quarter of 1989, and production quantities will be available in the first quarter of 1990. The RTX 2001A offers improved performance over the RTX 2001 through modified stack controllers, improved timing, and two added registers for user operations.

EXPRESS YOURSELF

Have there been price reductions for the RTX family of products? *Yes. Prices on the existing RTX 2000/2001 products reflect a decrease of 15% and the RTX 2001A will also reflect a significant reduction in price. This dramatic change is due to greatly increased yields produced by the new mask set as well as economy of scale due to the large number of design-ins of the RTX 2001 product.*

Is the RTX 2000 a RISC processor? *No. Though RISC-like in its hardwired instruction set and its lack of microcoding, the RTX is a stack machine with a highly parallel architecture. The RTX supports a large number of instructions while still achieving processor speeds that can match or beat those claimed by many RISC processors.*

RTX is on the FASTRACK™

In our Standard Products - *The FASTRACK IC design system has been instrumental in the success of the RTX 2000-based line of products, and further RTX 2000 and RTX 4000-based products are currently being developed with FASTRACK. Use of this system helps to insure quick turn around and first time success. An advantage of using the system internally is the ability to offer an RTX line of products in the form of ASIC megacells.*

In ASIC Products - *FASTRACK is a state-of-the-art ASIC design system for both analog and digital designs. Using this system, the customer can achieve improved performance through higher levels of integration and less interchip delay. Such improvements may include adding on-chip memory and peripheral functions around the RTX megacell, which permits a considerable amount of logic and memory to be included on-chip due to the relatively small size of the RTX core processor.*

FASTRACK offers compilable RAM and ROM for on-chip memory. The delays associated with on-chip and off-chip buffering are eliminated when more functionality is incorporated on-chip. Typical access time for a RAM is 18 ns (2K X 8 RAM), and only 12 ns for a 8K X 8 ROM. In addition, because primary current flow in a typical IC is in the I/O region, on-chip integration minimizes the overall power consumption and reduces system noise levels.

Direct access of the RTX memory bus and ASIC bus is a capability unique to the megacell. This allows extensibility to functions not located within the ASIC device, such as co-processors or EPROMS. This access simplifies the migration of existing RTX systems to systems with higher levels of integration. FASTRACK gives the flexibility to partition an RTX system at any level of integration.

Harris FASTRACK provides schematic capture, design verification, and place and route capability for Harris cell families, including the RTX-based HSC1000 standard cell library. The design package offers static timing analysis, logic optimization and back-annotation of layout delays. It is presently available for SUN™ 3 and SUN 4 workstations and is the first ASIC design package to support RTX megacells. For more information regarding FASTRACK and RTX megacells contact the Melbourne Design Center at (407) 729-5908.

Bruehl Entwicklungsgesellschaft mbH., Hegelstra_e 10, D-8500 Nuernberg 10, Phone: (49) 911-359088

Paul Elektronik, Erlenweg 18, 8901 Leitershofen, Phone: (49) 821-436796

MINIBEE-MB-RTX2000 This is an RTX 2000-based stand-alone system which features 128KB RAM, 64KB EPROM, Forth operating system on board (EPROMS), variable clock oscillator, serial interface to terminal or PC, with all ASIC and Memory Bus connections accessible. This board was developed by members of the German Forth User Group and the RTX 2000 fan club, and measures 100 by 100 mm.

FORTH-SYSTEME Angelika Flesch, P.O. Box 1103, D-7814 Breisach am Rhein, Phone: (49) 7667-551

The **WIESEL-2000 Entwicklungsboard** is a 100 by 160 mm Europe format RTX-based card. FORTH-system (Metacompiler) on board in EPROMs. Memory extension available up to 1 megabyte by sandwich type memory extension boards.

Delta t GmbH., Uhlenhorster Weg 3, D-2000 Hamburg 76, Phone: (49) 40-2296441

The **MUCK** consists of a complete Real-Time Control system featuring RTX 2000-based board with 64K byte RAM, 32K Byte ROM for Forth system, 32K byte EEPROM for storage of application software and another 32K byte EPROM for storage of application specific data tables, etc. Serial interface, 16 digital inputs, 16 digital outputs, 4 analog inputs with 12-bit resolution, programmable input sensitivity, 20 kHz maximal signal input rate. VME bus interface.

Innovative Integration, 4086 Little Hollow Place, Moorpark, CA 93021, Phone: (805) 529-7570

LMI FORTH for the FB2000 SBC offers a unique table driven optimizing compiler for the RTX 2000 and all of the functionality found in the FORTH-83 standard. In addition, it has been augmented with many useful screens for the FB2000 SBC.

Analog I/O Companion Card for the FB2000 - The analog I/O board is a companion to the FB2000 single board Forth language computer featuring 8 channels of 8-bit input and output, 16 TTL inputs, and 16 TTL outputs all on a 4.2" by 4.2" board. The Analog I/O board is ideal for real-time control, servo systems, and data collection where fast data conversion is required for use by the FB2000 SBC. The output DACs have a fast 1 μs settling time and may be configured for fixed or variable references. All of the DACs may be configured for unipolar or bipolar operation. Analog inputs are provided by eight input A/D converters which are 8-bit, 2.5 μs converts. The A/D converters may be configured to accept both unipolar and bipolar inputs and include an analog stage for gain adjustment and filtering. All of the A/D converters may be referenced to any of three precision voltage references. Program control of the FB2000 Analog I/O board from the FB2000 SBC is facilitated by using the predefined control words. More advanced programs may use the basic control words to build advanced applications.

FUTURE, INC., P.O. Box 10386, Blacksburg, VA 24062-0386 Phone: (703)552-1347

The **WARP-7** is an RTX 2001-based comprehensive environment for software development. This interactive development system includes a complete disassembler, a full FORTH 83 environment, local variables, macro-substitution, and case statements, as well as providing optimized object code and single pass compilation with forward referencing. Cross compilers for the RTX 2000 are available for the IBM™ PC/PS2 and the Apple Macintosh personal computers. Hardware includes an eight channel 16-bit I/O bus, 2 RS232 serial ports with handshaking, a centronics style parallel port, a real-time clock and 2K bytes of nonvolatile RAM, all on a half height Eurocard. 1M byte of extended memory is optional. This product is intended for applications requiring real-time data acquisition with on-line data analysis.

Laboratory Microsystems, Inc. P.O. Box 10430, Marina del Rey, CA 90295 Phone: (213) 306-7412 FAX: (213) 301-0761

The **LMI Metacompiler for the Harris RTX 2000** is a professional application development tool which compiles Forth source code into a stand-alone ROM- or RAM-based turnkey application, including a new interactive Forth interpreter/compiler. The Metacompiler is compatible with the Forth-83 Standard, and is also fully compatible with the Harris RTX 2000 Development System. The Metacompiler can generate a complete, interactive Forth interpreter/compiler that runs on the RTX 2000, allowing new code to be developed and tested interactively on the RTX. The Metacompiler also contains a table-driven optimizer that analyzes opcode sequences on the fly, for fast, compact code while remaining "invisible" to the programmer.

Offete Enterprises, Inc. (U.S. distributor) 1306 South B Street, San Mateo, CA 94402 Phone: (415) 574-8250

The **INDELKO FORTH KIT for the RTX 2000** measures 100 by 100 mm and runs cmForth on a Harris RTX 2000. Power, serial lines, and the ASIC-Bus are brought out to a EuroCard edge connector; the cmForth is ported from Chuck Moore's FORTHkit 4. This kit was designed and produced by Mr. Thor-Bjorn Bladh of INDELKO at Lund, Sweden, and includes the PC board described and two CMOS EPROMs with the modified cmForth. The user must provide the RTX 2000 chip, two CMOS SRAMs, a clock, and 4 other glue logic chips to complete the system. With 150 ns SRAMs, a 10 mHz clock is needed for 5 mHz operation.

Important Notice: RTX$_{PRESS}$ is an information service provided by Harris Semiconductor for readers involved in the field of real-time applications, and considerable effort has been expended to ensure that the information contained herein is accurate and complete. However, Harris RTX$_{PRESS}$ cannot assume responsibility for inaccuracies, omissions, manufacturers' claims, or their representations.

Optimization on the RTX 2000

by Robert DeRobertis

The RTX architecture provides the ability to repeat instructions with a minimum of overhead. These instructions, **FOR ... NEXT** and **OF(** , can be applied to such areas as DMA control, data acquisition and peripherals I/O. Utilizing these instructions will produce a significant performance enhancement.

The standard Forth **DO...LOOP** requires overhead to support the index tracking words, **I** and **J**, even if they are not used. **FOR...NEXT** takes advantage of the RTX hardware looping capability to significantly improve performance. Compare the following words:

```
: OFFSET 16 30 G@ SWAP - ;
: READ-BUS1 127 FOR OFFSET NEXT ;
: READ-BUS2 128 0 DO OFFSET LOOP ;
```

The word **OFFSET** will read a value from the ASIC bus at address 30 and subtract an offset of 16, taking two clock cycles. The two **READ-BUS** words are functionally equivalent where each will execute the word **OFFSET** 128 times. Executing the word **READ-BUS2** takes 2574 clock cycles and the word **READ-BUS1** takes 517 clock cycles a performance increase of almost 500%. Here you can see a significant enhancement with a minor change to your implementation.

If you need to utilize the loop index in a **FOR ... NEXT** loop, you can also do that. The **FOR ... NEXT** instruction uses the INDEX register (top of the RETURN stack) to hold the current count, access the current count by using: **n R@ -** , where n is the looping limit. Here only an overhead of three clock cycles is required.

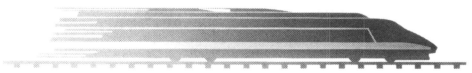
The stream instruction **n OF(** will repeat the first clock cycle of an instruction **n** times without any overhead. Note that all one-clock cycle instructions are streamed and the first clock cycle of a two-clock cycle instruction is streamed. Here memory or ASIC bus peripherals can be accessed at a rate of one word per clock cycle. An example of a memory access would be:

```
HEX
E942 UCODE @+2      \ DUP @ SWAP 2+    ( adr – data adr+2)
E5C2 UCODE !-2      \ SWAP OVER ! 2-   ( data adr – adr-2)
: READ-MEM  9000 7F OF( @+2  DROP ;
: WRITE-MEM A0FE 7F OF( !-2  DROP ;
: MEM-TRANSFER READ-MEM WRITE-MEM ;
```

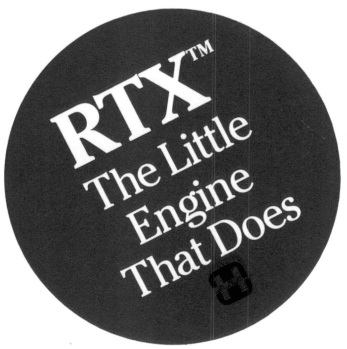
The **UCODE** definitions create Forth instructions which utilize the RTX's auto increment memory access instruction. The words **@+2** (fetch plus 2) and **!-2** (store minus two) read or write to the address on top of the stack, then increment or decrement the address by 2 to point to the next location to be accessed.

The next two word definitions read or write a block of memory at one clock per location. The word **READ-MEM** will read 128 words from memory starting at 9000h and place them on the stack. The word **WRITE-MEM** will write 128 values off the stack into memory starting at location A0FEh. Note that the data must be written into memory starting from the end of the block and decrementing because of the LIFO nature of the stack: the data on top of the stack represents the data read from the end of the memory block. **READ-MEM** and **WRITE-MEM** each take 134 clock cycles to execute: 5 clock cycles to set up the starting address and the count, 128 clock cycles to access memory locations, and 1 clock cycle to **DROP** the final memory address and do the subroutine return. A 128 word memory to memory transfer is accomplished with the word **MEM-TRANSFER** in 272 clock cycles.

The highly parallel nature of the processor provides many techniques for manipulating data or accessing peripherals. Utilizing these simple constructs will provide extensive application enhancements.



Harris Semiconductor
RTX™ Marketing
P.O. Box 883, MS 62A-021
Melbourne, FL 32902-0883



RTX PRESS℠

Vol. 1, No. 3 **The Latest On Harris Real Time Express™ and Real-Time Applications** *December, 1989*

Fred O. Hawkes, Director
RTX Marketing

The Harris Commitment - It takes a tremendous amount of confidence in the performance of a product for a company to offer that product to users, free of cost, on the assumption that "using is believing". Harris has that kind of confidence in the RTX family of products.

Starting on January 1, 1990, Harris will offer professionals in the technical community an opportunity to take part in a "Real-time Design" contest.

Winners in the first round will be selected from submitted proposals, and will receive a special edition RTX 2001A Evaluation Board. Those first round winners can then use their prizes to enter the second phase of the contest, from which the first prize winners will be chosen.

A grand prize of $10,000 will be awarded to the winner of the "Best Real-time Design". More details about the contest are described on page 3 inside this issue.

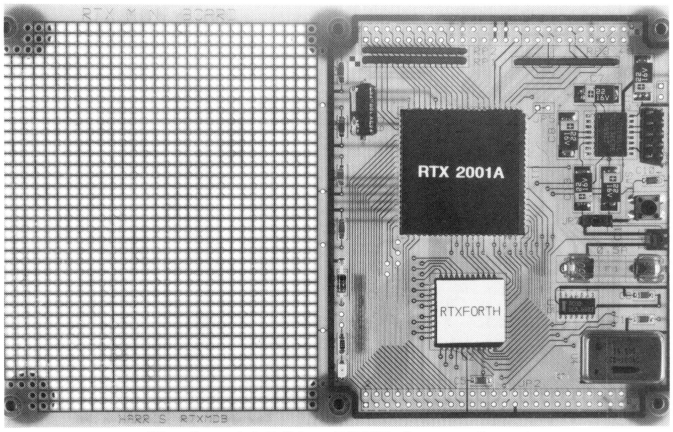
Some of the features about the Harris development tools that have been praised most highly by their users have been the ease and speed of development from start to finish. These are both features that this contest will help prove.

Another is the diversity of applications for which the RTX should be considered. No matter how simple or how complex, real-time processing and control applications have intense response requirements. Users of the RTX products have been very enthusiastic in their assessments of how well the RTX handles such requirements, as compared to the performance of other processors.

Harris expects this contest to affect a significant increase in the range of applications for which this product is a serious contender, and is confident that once considered, the RTX will become the microprocessor of choice in most real-time control applications. ■

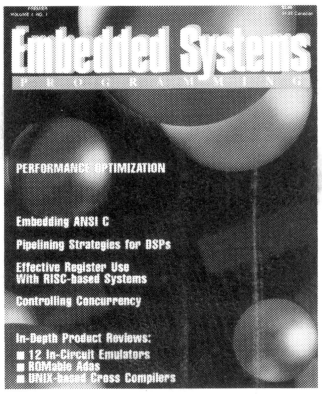
TRACKING THE RTXPRESS

RTX - A new technology for a new decade. Get a fast start with the Harris/ESP Real-time Design contest!



The Special Edition RTXEB Contest Board

Embedded Systems Programming - Ted Bahr is the publisher of **Embedded Systems Programming** magazine, which is co-sponsoring the Real-Time Design contest with Harris Semiconductor. We thought our readers would like to know more about his views.



Are there common threads that run through all embedded systems applications, or are the applications as diverse as they appear? There is tremendous diversity in terms of response time requirements, functional processing requirements, and memory requirements. The importance of software development and many of the tasks and tools involved in that development effort are the common thread around which we have based our magazine.

How is software for embedded systems unique? The requirements which must be met by software engineers in developing embedded applications are rigorous, and are very different from other fields of programming. Embedded systems software must be vitally sensitive to what is happening in real time, and each response must occur at exactly the time it is needed. "As fast as possible" is not adequate in real-time systems. In addition, software for embedded applications must be debugged to perfection. There is no release 2.0, and any bugs which exist become part of the final system. As if these constraints were not tough enough, embedded systems software engineers often must create their own operating systems to control the chip,

continued ....

HARRIS SEMICONDUCTOR
HARRIS · RCA · GE · INTERSIL

Copyright (c) 1989 by Harris Corporation

Ted Bahr, continued ....
because an existing kernel is not available. This is technically arduous and adds to the development time, increasing the risk of missing the market window.

Do you see any trends developing, and will they help to resolve some of these problems? There is a very definite trend among our readers toward use of larger, more powerful chips which offer the luxury of using commercial real-time kernels to speed up the time to market for a product. Coupled with this is a trend away from assembly language, with a resurgent interest in Forth (which was created to solve real-time problems) and use of high level languages such as C and ADA. Evidence of these trends can be seen in the market response to products like the RTX, with its emphasis on software support as well as hardware.

Where would you anticipate seeing the most future growth? We see no limit or boundary to where electronic products, expecially embedded systems, will go. They have invaded our lives, in our clock radios, our computers, our appliances, even in our cars and our children's toys. Enlightened companies like Harris, which are directly addressing the needs of software developers, are in the best position for future growth because they are already providing the support essential to growth in the embedded systems market. The potential for applications is almost everywhere, and is growing tremendously.

Why is Embedded Systems Programming co-sponsoring the Real-time Design contest with Harris, and what are some of the benefits which participants can expect to gain from entering this contest? Our organization is proud to be part of the embedded systems marketplace, and we welcome any opportunity to shine a spotlight on its unsung heroes. This includes both the applications engineers, and the companies who recognize the unique talents and challenges that exist within the field. One of our prime objectives is to raise software cognizance and to give engineers the tools and information to go wherever they want to go. We see **Embedded Systems Programming** and the technical information it provides as a launching pad for software imaginations. We are enthused to be involved in this contest because, in addition to the material gains to be made by the first place winner, the contest offers designers an opportunity to explore new products and services without high equipment costs or entry fees. ■

The **RTXPRESS** is published quarterly, and is copyrighted by Harris Semiconductor. Correspondence and submissions should be addressed to: **RTXPRESS**, Harris Semiconductor, P.O. Box 883, MS 62A-021, Melbourne, Florida 32902-0883.

Editor:
Monnie Smith

Executive Board:
David P. Barath
Linda daCosta
Michael H. Graff
Ronald A. Leone
David G. Williams
Fred O. Hawkes

Editorial Board:
Robert Bryant
Tim Dwyer
Alice Gills
Tom Hand
Philip J. Koopman
Ken Lyons
Susan Motes
John Reed
Rick VanNorman

Diversity in Real Time

Monnie Smith

What do military target recognition, medical imaging, physical therapy, and food packaging have in common? What about satellite communications, encryption circuits, navigation, speech processing and robotics? The answer is simple. RTX.

In these applications, development engineers have utilized RTX microcontrollers to produce systems that represent significant advancements in their particular fields of technology. At first glance, it is difficult to see how the same processor can be the optimum choice in each of these diverse applications.

A look at the architecture of computing machines holds the key to determining the common link in these applications.

In general purpose processors such as those using register-based architectures, operand sources and destinations (in either registers or memory), must be clearly specified for every operation. When an interrupt occurs and a context switch takes place, the configuration of the system must be saved, and a new configuration loaded. These steps take time. When the processor has finished handling an interrupt and returns to "normal" processing, the system's original configuration must be restored. This also takes time.

Some processors utilize pipelines and caches to increase processing speeds. But in interrupt driven systems, the overhead of managing pipelines actually makes the execution less predictable. Depending on the system configuration when an interrupt occurs, severe predictability problems may result, and data may be lost.

In a stack machine such as the RTX, access to operands is via fixed, predefined sources and destinations which remain the same for that processor, regardless of application. These sources and destinations are implicit for each instruction, and need not be defined explicitly. Internal data storage and retrieval is always via the stack, yielding faster and more predictable execution, even when a context switch occurs. Combined, these factors mean that "normal" execution speed can be faster, while the response delay incurred during servicing of interrupts is reduced significantly. In addition, since no pipeline flushes are performed, the risk of losing vital information is also greatly reduced.

The differences in architectures may be of little importance in applications where interrupt responses, context switching and predictability are not factors. But in applications where fast, reliable responses are critical, the architecture becomes a driving factor in the selection process. Such is the case in real-time processing and control.

The number of applications for which the RTX family of processors is being used continues to expand. So far, applications are limited more by the length of time it has been on the market than by the capabilities of the product line. It becomes evident that stack machines, and the RTX in particular, have an inherent adaptibility to real-time processing and controls. ■

*If you would like to submit an article about a real-time application to **RTXPRESS**, send it in. We welcome contributions from our readers.*

Real Time Design Contest

Real-Time Applications - Do you have a real-time application that needs just the right processor to make it happen? Well, Harris has your solution. The RTX 2001A microprocessor can put you on the right track to the future.

And to show you just how easy it is to develop your application on the RTX 2001A, we'll give you the opportunity to win a system of your own, by entering the **Real-Time Design Contest**, sponsored jointly by Harris Semiconductor and Embedded Systems Programming Magazine.

Once you've qualified to receive the Special Edition RTX 2001A Evaluation Board, you've already won a $500.00 value. But in this contest, real time is money. And there's lots more to win.

$10,000 GRAND PRIZE - The Harris/ESP Real-Time Design Competition concludes with a grand prize of $10,000. But it begins with ideas. So tell us how you would use this powerful real-time microcontroller to transform your vision of the future into a reality. If your entry qualifies, you'll win a free Special Edition RTX Evaluation Board - because we want to provide you with the right tools to make your vision real.

There's more. Besides the winner of the grand prize for best overall design, winners will be chosen from the "best hardware design" category, and "best software design" category, to receive awards of $2000 each. Runners-up in each of these categories will receive $500. Lots of other prizes will be awarded for the top 100 entries.

1000 FREE RTX2001A EVALUATION BOARDS - Up to 1000 free RTX 2001A Evaluation Boards will be given away to those entrants whose proposals qualify. These evaluation kits include a Euro-card sized development board which incorporates the RTX 2001A microcontroller, 16K bytes of PROM, 4K bytes of SRAM, a serial port, a large area for breadboarding and a Forth software development environment. (Unfamiliar with Forth? If you need help, we will provide you with training tools that can have you writing code the first day.)

Arrange For Your Entry Kit Today! - Timing is critical in the Harris/Embedded Systems Programming Real-Time Design Competition. Proposals must be received by April 16, 1990, so make arrangements now. To receive your contest information kit, fill out the reader response card at the back of this issue, and mail it today. Or call 1-800-4-HARRIS, ext. 1300.

Winning Designs - will be displayed at the Second Annual **Embedded Systems Conference**, held September 25-28, 1990, at the Hyatt Regency Hotel in Burlingame, California (near San Francisco). For more information about this conference, call Anastasia Mills at (415) 397-1881.

*Hop aboard the Harris **RTX DESIGN EXPRESS** to win your ticket into real time! Enter now!*

About the sponsors ...

Embedded Systems Programming is the first and only magazine written for design engineers, software engineers, and developers building real-time microprocessor and microcontroller-based systems. As co-sponsor of the Real-Time Design Contest, **Embedded Systems Programming** will provide two judges for the competition. Tentative plans include publication of an article about the winning design. For subscription information ($37.00 for 12 issues) write Kathy Shay, Miller Freeman Publications, 500 Howard Street, San Francisco, CA. 94105; or call (415) 995-2421.

Harris Semiconductor is a sector of Harris Corporation, and is the nation's 6th largest manufacturer of semiconductor products. **Harris Semiconductor** has a long history of innovative and advanced technology in both the commercial and military markets. Using a macrocell approach to circuit design, Harris developed a Standard Cell Library to meet customer requirements for advanced signal processing and control applications, particularly one-chip ASIC designs. This standard cell technology was used in development of the RTX product line for embedded real-time control applications.

Void where prohibited by law. In particular, we regret that certain international applicants may be excluded due to U.S. export laws.

For those who are reading about the RTX family of products for the first time, we thought that a brief description of the features offered by the RTX 2000 and RTX 2001A microcontrollers might be of interest.

The RTX 2000 offers a fast 100ns machine cycle; four-cycle interrupt latency; single-cycle subroutine call/return; fast multi-step arithmetic operations; a single cycle on-chip hardware multiplier; two 256-word deep on-chip stacks; an on-chip Interrupt Controller, three on-chip Timer/Counters; and Harris ASIC Bus™ for off-chip extension of architecture. This is a low power CMOS product.

The highly parallel stack architecture of the RTX microcontrollers allow execution of the equivalent of up to four RISC processor instructions within a single processor clock cycle, with a peak rating of 40 million Forth instructions per second and a sustained capacity of 10 to 15 million instructions per second. Unlike RISC processors, the RTX microcontrollers do not need to use pipelines or caches to achieve their speed, and therefore do not suffer the unpredictability inherent to RISC processors.

The RTX 2001A, is the processor on which the contest Evaluation Board is based, and was derived from the RTX 2000. This microcontroller is designed to meet the needs of customers with rigid cost requirements, but who do not need the intensive mathematical capabilities provided by an on-chip multiplier. The RTX 2001A has two on-chip 64-word stacks, and offers most of the same features, but with enhanced Interrupt Controller logic for multi-tasking operations.

if you would like more information about these products, mark the reader response card in back, and mail today.

A MULTITUDE OF USES

The RTX microprocessors have been incorporated into a number of application designs, in very different fields. Following are just a few examples of those applications.

AUDIO SIGNAL PROCESSING

Voice Compression - The RTX 2000 was used to implement a Continuously Variable Slope Delta (CVSD) Modulation algorithm in a voice compression application.

Professional Audio Control Equipment - The RTX development environment cut new product development time from two years to 6 months for a new remote control audio routing system.

Voice Identification for security purposes using neural network techniques.

COMMUNICATIONS

The **Wideband Transport Frame Formatter (WTFF)** is intended to act as the Ku-band return link gateway for the Tracking and Data Relay Satellite (TDRS). A software environment that would provide simple, compact, self-contained code, with a totally predictable, high level of performance was essential for this application. Power and size were also factors, due to the intended use in satellites.

SCSI Channel Processor and V401 application specific I/O processor - These two products were designed for use in unforgiving, performance sensitive applications in rugged environments, requiring very high reliability and fast, very predictable response.

MEDICAL SYSTEMS

Industrial and Medical Tomography Applications - needed a controller that could provide higher clarity images and better diagnostics through speed and performance by providing control pulses which were precise, conformed to a complex decay slope, and had a repeatable stability to a few parts per million.

An **Aerobic Exercise System** for the handicapped which takes on the normal stimulus and response functions of the human nervous system.

Bio-Medical Monitor - used to detect motion sickness.

NAVIGATIONAL SYSTEMS

The **Daylight Star Tracker** mounts on a guidance gimbal and uses a CCD sensor and the RTX 2000 to process a 900 pixel frame in 10.0 ms to update inertial reference platforms.

A **High precision LORAN Navigation Set** which can offer precision of at least 5 times that of the standard LORAN systems.

OTHER APPLICATIONS

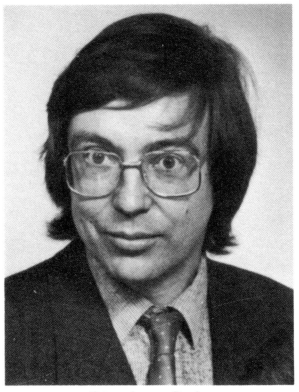
REMOTE SENSING - low power consumption of RTX extends life of monitoring devices.

ROBOTICS - Bi-pedal walking robot for operation in harsh or hostile environments.

HAND HELD COMPUTERS - For inventory tracking; analyzing system responses rapidly in the field.

TRAINING TRACKS - For information about the RTX training workshops directed toward learning optimized design techniques for embedded real-time systems, call 1-800-4-Harris, ext. 1990.

USER'S TRACKS - In our last issue, Bill Holloway (VME Inc.) suggested that we speak with Stephen Pelc, President of MicroProcessor Engineering, Limited, for another point of view on both Forth and on the "RTX experience". Stephen has been generous enough to share his thoughts here.



Stephen Pelc
MicroProcessor Engineering, Ltd.

How did your company's use of RTX products come about? Our specialty is in cross support software and hardware. For this field of applications, we required a high level language that could be ported over a wide range of processors, while still allowing us complete control over the language implementation. Our customers range from assembly language programmers who are working toward better software productivity, to hardware engineers who require something fast enough to do their job. The only language that fits this whole range of needs is Forth. Since the RTX 2000 is a hardware implementation of a two stack Forth language architecture, it is a tool that can be used to meet the needs of our entire spectrum of users.

Can you offer us some insights into the reasoning a manager might use in selection of his company's development tools? In a competitive market, the final decisions must be based on a balance between cost and performance. In making these decisions, there are three primary factors in determining what processor should be used in a real-time processing application.

Processor performance is the most critical of these factors. The processor chosen must be able to perform all required operations within the timing constraints defined by the application.

Development time is another crucial consideration. The amount of time it will take to go from initial concept to final implementation must be within the limits set by the market for that application. If implementation is too slow, the requirements may be eliminated or met by other means.

Development and production costs must also be considered. To be marketable, the final product must be economically practical. Though a $3000.00 mousetrap may be "better", there will be very few that actually sell. The length of development time can have a severe impact on final system cost, as can the number of components and amount of glue logic used in the final design. A fast processor which permits hardware to be replaced by software can therefore offer a significant cost advantage.

The impact of development time on final cost is particularly critical in software development. If money must be spent to streamline the development cycle, it makes most sense to invest it in the area that offers the biggest target for improvement. An analysis of software development costs shows that specification and coding each represent about 20-25% of the total, while debugging and integration exceed 50%. Since the debug/integration phase accounts for the lion's share of the software development expenses, this is the phase which best rewards an investment in tools. For this reason, a dependable, interactive debug environment is essential and can yield a significant reduction in the cost of this critical development phase. Our company has found Forth to be excellent in providing just such an environment, while helping to make the job easier for both hardware and software users.

Could you be more specific? In the case of hardware engineers, they must have components which can provide the fast response times required in real-time applications, while having software tools that do not require them to become programming experts to perform a task. "Painless" software tools are essential - tools which allow a quick initial learning curve while being "forgiving" enough to allow use with limited proficiency. Though optimization of code in the final product is desirable, the first priority for a hardware engineer is to have code (even if it's inefficient) that allows him to test the hardware operation.

Even experts in the use of complicated software tools can abhor the painstaking, non-portable and tedious use of assembler. For these engineers a portable high level language offers significant advantages. Software engineers also need hardware tools that provide easy, inexpensive and reliable support for their software development, without being forced to become hardware experts.

By designing our products around Forth and the RTX, we are able to provide our customers with a development platform that meets these needs.

Can you give us an idea of how some of your customers use your RTX-based products? Our main software tool for RTX is a PC or UNIX hosted Forth Optimising Cross Compiler. One customer used this to develop the controller for a driverless mass transit system. In conjunction with an EPROM emulator, he created an interactive Forth system during development for rapid testing. For field trials, the application is in EPROM. When the application is finished, the software is compiled one more time to produce a minimum sized turnkey application - the product. The mass transit application is also interesting because the power of the RTX 2000 allowed three computers of the original design to be implemented by one RTX board.

continued ....

Fast Tracking - Stephen Pelc, ***continued ....***

Our RTX2000 PowerBoard is used for television signal analysis, and for goods inward inspection from a line scan camera. In both cases, the onboard SCSI controller can be used for direct data logging to a disc, and the on-board RS485 link can be used for local networking. A major advantage of this board is that the RTX 2000 PC Adaptor can be used to turn this board into a full length PC plug-in card using dual port RAM communication and bi-directional interrupt handling. This allows development with very high speed host to target communications, or the boards can be used as a very high speed I/O processors for the PC.

Our new RTX2001 STE PowerBoard uses the STE-bus structure in the popular 160x100mm single Eurocard format. This one is used by Rutherford Appleton Laboratories for beam focusing of a 50Hz synchrotron accelerator. The board is exciting because, apart from the RS232 connection, it also has a 20Mb/s serial link which allows the boards to be networked. An optical crosspoint switch card allows clusters of up to 16 boards to be dynamically configured.

Thank you, Stephen, for your comments, and for providing us with this brief look at the diversity of applications which utilize your products (and subsequently, ours). We look forward to hearing more as information is made available.■

LOOK AHEAD for these features in RTXPRESS:

Hand-held computers - the latest on how they are being implemented.

The latest information on a disassembler package being developed for the Fluke PM3655 Logic analyzer.

From Micro Amps Limited, information about their G-Port Page Switch, which was developed to provide additional I/O address space.

Dr. Bertram Ezenwa, Director of Technical Development at Wright State University, will share information about the aerobic exercise system being developed for use by individuals who are physically handicapped.

Important Notice: **RTXPRESS** is an information service provided by Harris Semiconductor for readers involved in the field of real-time applications, and considerable effort has been expended to ensure that the information contained herein is accurate and complete. However, Harris **RTXPRESS** cannot assume responsibility for inaccuracies, omissions, manufacturer's claims, or their representations.

Good Things In Small Packages

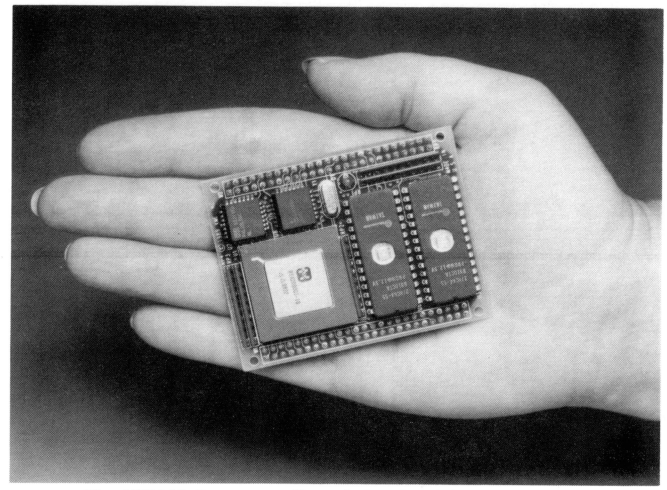
VME INC.
542 Valley Way
Milpitas, CA. 95035 (408) 946-3833

VT2000 Processor Foundation Module - This "credit card" size module offers a simple plug-in solution for the CPU portion of prototyping or embedded processor requirements. Measuring just 2.3" x 3.0" x .5", this module uses only 270 mA at 5 VDC, while providing performance of 15 MIPS at 10 MHz, with Interrupt responses of 400 ns and multiplication operations in 100 ns.

In addition to the 10 MHz RTX 2000, this fully operational processor module features 64K bytes of high speed CMOS SRAM, 64K bytes of EPROM, a serial I/O interface (DC to 1.25M baud), DC power fail detection, a watchdog timer, and the capability for automatic switchover to external battery backup.

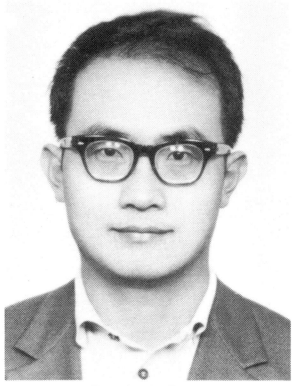
Connections are simple, via three wires which attach to a Host computer serial port to receive and transmit data and to provide the ground connection, and two wires for the 5 volt DC connection. Dual 50-pin male connectors provide a functionally complete interface which can be plugged directly into standard wire-wrap prototyping boards for expansion.

The VT2000 is intended for use in R & D prototyping environments and limited volume production for high speed data I/O, data analysis, machine control and peripheral control applications.



The VT2000 Processor Foundation Module from VME Inc.

RTX™, RTX 2000™, RTX 2001A™, RTS™, RTXDB™, RTXEB™, RTXDS™, and ASIC Bus™ are trademarks of Harris Semiconductor. GEnie™ is a trademark of G.E.



Dr. C.H. Ting
Offete Enterprises, Inc.

Those in the Forth community need no introduction to Dr. C. H. Ting. His work is well known through his publications, particularly **More On NC4000** (renamed **More On Forth Engines**).

In a recent paper presented at the 1989 Rochester Forth Conference, Dr. Ting's use of the RTX 2000 in implementing Continuous Fourier Transform (CFT) techniques is described. THE CFT is a modified Discrete Fourier Transform (DFT) designed for continuous processing of linearly digitized input signals in real time and uses unscaled integer arithmetic. The key to the differences between CFT and DFT is in how the sampling window is used in a CFT and how the resultant data is processed.

In CFT, the sampling window is shifted in the sampling domain (Figure 1), whereby the majority of data points in each successive sample window are the same as for the previous sample window (this eliminates the need for foldover, used in DFT). Each new data point generates an added error factor (+ Δ e), but at the same time the error factor generated for each data point which has been deleted from the sample window drops out (- Δ e) when the transform is performed. Statistically, the error that would otherwise accumulate with each new data point is compensated by the error that "drops out". In other words, error introduced at f(k) will be compensated N cycles later when f(k+N) is processed.

COMPARING CFT AND DFT

Advantages of CFT over DFT and FFT are:

- Faster update
- Does not require floating point hardware
- Errors do not accumulate
- Easily adaptable to parallel processing

Parallelism: Well adapted; every term T(k,n) is evaluated independently of the other terms.

DFT Equations:

To compute a set of N Fourier transform terms:

$$(1) \qquad F(k,n) = \sum_{m=k}^{k+N-1} f(m)\ \exp\{-2j_{\pi}(m-k)n/N\}$$

The DFT has a recursive relationship:

$$(2) \qquad F(k+1,n) = \exp\{2j_{\pi}n/N\}\ (\ F(k,n)-f(k)+f(k+N)\ )$$

CFT Equations:

$$(3) \qquad T(k,n) = \sum_{m=k}^{k+N-1} f(m)\ \exp\{-2j_{\pi}mn/N\}$$

The CFT has a simpler recursive relationship:

$$(4) \qquad T(k+1,n) = T(k,n) + (f(k+N)-f(k))\exp\{-2j_{\pi}kn/N\}$$

The CFT differs from the DFT by a simple phase factor:

$$(5) \qquad F(k,n) = T(k,n)\ \exp\{2j_{\pi}kn/N\}$$

The ability to evaluate every term T(k,n) independently of the other terms means that there is no limit to the number of sample windows which can be used when performing a transform. This is the key to Dr. Ting's proposed use of Fourier analysis in real-time applications.

Using his technique, the base "sample window" (Figure 2) and sample points can be defined, sampled, and the transform performed by a processor. At the same time, a different set of sample points for that window can be defined, and a second transform performed by a second processor which is synchronized with the first processor. The resultant combined transform can be produced within the same time period as one set, and the number of sample sets possible is limited only by the number of processors used.

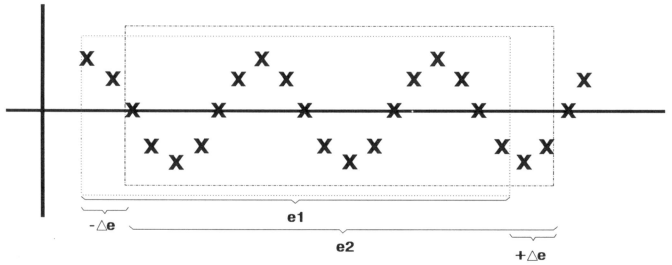


Figure 1. CFT - Sample Windows

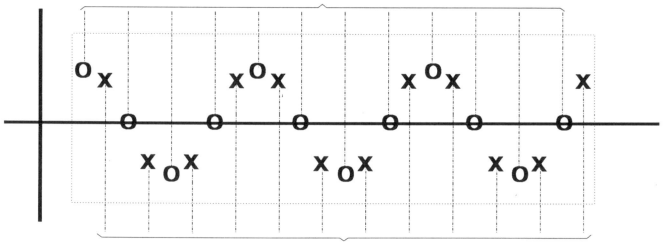


Figure 2. CFT - Parallel Processing

It is Dr. Ting's assessment that the Harris RTX 2000 microcontroller is well suited for this algorithm because of its integral one-cycle, 32-bit hardware multiplier and its powerful instruction set. The implementation of the CFT can calculate the 1024 point CFT of a real-time data stream at an update rate of 2.4 milliseconds on a 10 MHz RTX 2000. This performance is comparable to that of the special purpose DSP chips which have hardware multiplier-accumulator facility. By combining several RTX processors in parallel, it is possible to out-perform other available methods by several orders of magnitude.

Applications for which this technique would be well suited include:

- General Signal Processing
- Voice Processing
- Real-time Image Processing
- Radar Signature Processing
- Sonar Signal Processing

RTX ON TRACK

RTX 2000™ Real Time Express 16-Bit Microcontroller - A high performance microcontroller with on-chip timers, interrupt controller, multiplier, two 256-word stacks, and an ASIC Bus.

RTXDS™-10 HW/SW Development System - A complete hardware and software development environment based on the RTX 2000/2001A microcontrollers. The **RTXDS** combines the Harris **RTXDB™** and **RTS™-DS** for interactive development, debugging, and evaluation of real-time applications.

RTXDB-10 or -8 Development Boards - A ready-made microcontroller system, configured for immediate operation of RTX hardware when used with Harris RTS Development System software (not included) or with user-defined software.

RTS-DS Real Time Development System Software - A set of integrated software tools used for development and debugging of RTX 2000/2001A software. A PC-based "Host", and development board-based "Target" are provided, with editor, Forth compiler, and real-time interactive debugger. This software development environment is designed for use with the **RTXDB** or with user-defined RTX-based hardware.

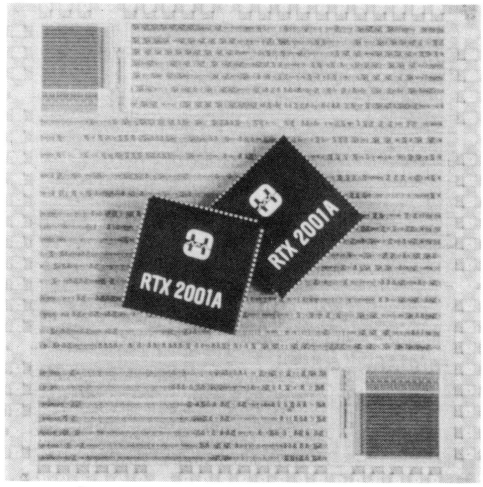
RTX Memory Expansion Kits - The **RTX-MX64** Memory Expansion Kit has 64K bytes of 35ns SRAM, and sockets for an additional 128K bytes of SRAM. The **RTX-MX192** has 192K bytes of 35ns SRAM. Development Systems with expanded memory already installed (**RTXDS-10MX**) are also available.

NEW ARRIVALS

MILITARY: The **RTX2000GM-8/883** is the first member of the Military/Aerospace RTX family which has been optimized for MIL/AERO embedded control applications. This part is fully compliant to MIL-STD-883C, paragraph 1.2.1.

Order entry is now open for this product, data sheets are in progress and sample quantities are available immediately. This product design has already been incorporated into applications such as the Star Tracker by Ball Aerospace, and the Wideband Transport Frame Formatter at the Goddard Space Flight Center (see issues 1 and 2 for information about these applications).

Additional applications for which this product will be well suited include Electronic Warfare/Electronic Counter Measures; RADAR tracking; Motion Control; Avionics Instrumentation and Flight Control; Missile and Weapon Navigation and Guidance; Target Acquisition and Identification; Space and Ground Systems Command and Telemetry; and Secure Data Communications Encryption and Decryption.



The RTX 2001A 16-Bit Microcontroller

COMMERCIAL: **RTS-C Cross Compiler** - The C Compiler for the RTX 2000 series of microcontrollers is a very efficient cross compiler that is designed to meet the proposed ANSI Standard for the C Programming Language. The compiler is available for those developers of embedded real-time systems who prefer to use the C programming language.

This product compiles high-level C code to executable RTX 2000 code. The cross compiler environment has a state-of-the-art window/menu user interface with an embedded help facility. In addition, the compiler package contains an excellent set of programmers' tools, including:

- a C statement profiler
- a debug session recorder
- user selectable text editor
- a MS-DOS command line
- a make utility
- in-line Forth directives
- a C source statement symbolic debugger

The debugger supports break points, view variables and watch points. With the step, display, break, watch, view, assign, and find options, the user can develop RTX 2000 applications with state-of-the-art debugging facilities.

For information about availability, call 407-724-7660, or 1-800-4-Harris. Full customer support is provided worldwide by Harris. In Europe, the C compiler is available through CESYS Gmbh, Henkestrasse 8, 8520 Erlangen, West Germany (49-9131-21098). CESYS provides all user support in Europe.

COMMERCIAL: **RTX 2001A 16-Bit Microcontroller** - Offering improved performance through modified stack controllers, improved timing, and two added registers for user operations, the **RTX 2001A** has 64-word stacks and is pin compatible to the **RTX 2000**. This new product has features similar to those of the **RTX 2000**, but no on-chip multiplier. Samples are available now, and production quantities will be available in the first quarter of 1990.

RTX PRODUCT UPDATES (continued)

RTX DOWN THE LINE

RTS-FORTH - A Forth operating system/development environment, supplied in ROM for the RTXDB. Available in the second quarter of 1990.

RTX 2152 2MBaud UART - A single-chip, high performance, programmable Universal Asynchronous Receiver/Transmitter and Baud Rate Generator (BRG) which will support data rates from DC to 2.06 MBaud asynchronously with a 16X clock (0 to 33 MHz). This product provides a direct RTX interface with no wait states, and is fully operational after programming only three 8-bit registers (up to 72 different baud rates using a single source frequency). The **RTX 2152 2MBaud UART** will be available in the second quarter of 1990.

RTS-Tasker - A real-time multitasking executive for the RTX 2000/2001A which is a pre-emptive, priority based, interrupt driven system. Available in the second quarter of 1990.

HARRIS RTX LITERATURE WHICH IS AVAILABLE:

RTX 2000 Data Sheet, May, 1989
RTX 2001A Data Sheet, October, 1989
RTXPRESS: Vol. 1, No. 1; May, 1989
RTXPRESS: Vol. 1, No. 2; September, 1989
RTX 2152 Data Sheet, March 1989
RTX Article Reprint Series #1
App.Note #117 - Implementing a Software UART / RTX 2000
App.Note #118 - Getting Started / RTX Development System

To request copies of this literature, call 1-800-4-Harris, ext. 1288.

IN REVIEW

STACK COMPUTERS The New Wave

By Philip J. Koopman, Jr.

Stack Computers: the new wave is the first book dedicated to the analysis of modern stack-based computer architecture. The use of stacks in conventional computing techniques has traditionally been relegated to the sidelines, to be utilized only when no other method will satisfy a particular set of requirements. Most books on computer architecture mention stacks as a good method for handling recursive subroutine calls. A few of the books might briefly discuss a machine with no registers that uses a stack for operand storage. The focus of this book reveals stacks as a fundamental tool, required for achieving results necessary for specific types of applications such as embedded real-time control.

Dr. Koopman presents an objective look at the functional aspects of such requirements as reentrancy and recursion, leaving the reader with a deeper understanding of how conventional techniques are inadequate to handle them. (Even conventional methods implement some sort of stack if they must handle recursion or reentrancy, unless they use fixed, dedicated memory.)

The discussion of stack architecture begins with an excellent explanation of what a stack is, how it can be implemented, and how stacks can be used. Dr. Koopman then provides a taxonomy that allows classification of different architectures based on three attributes: number of stacks; size of dedicated stack buffer memory; and number of operand fields in the instruction format. This taxonomy is invaluable when considering the design trade-offs of different architectures, and should be remembered by the reader when making design decisions in the future.

System requirements which influence design decisions are incorporated into the discussions. Requirements covered include: small program size; low complexity; high performance; good performance consistency; software selection and use of conventional languages; and hardware/software performance trade-offs.

A model for a generic stack machine design is described, for use as a base during the discussion of other designs. Several Forth-based 16- and 32-bit stack computer designs are analyzed, looking at their advantages and disadvantages.

The effects of a stack architecture on such hardware issues as stack size, context switching, interrupt response, and processor performance are discussed in detail. In addition, a full chapter is devoted to software issues such as program size, subroutine calls, and support for Forth and for conventional languages like C.

A discussion of typical applications of stack architectures is included, then Dr. Koopman wraps up with his projections on the future of stack machines. An appendix that contains a thorough survey of computers which provide some hardware stack support is also included.

In conclusion, the reviewers found that this book contained a wealth of information on stack architectures. It was very well written, and its discussions were easy to understand. Those who stand to gain the most significant insights from reading this book include those interested in computer architecture, Forth programming, hardware support of Forth, and architectures for real-time control applications.

ORDERING INFORMATION:

When ordering from a bookstore, order book number ISBN 0-470-21467-8. Direct phone orders may be placed using Mastercard or VISA by calling the Eastern Distribution Center of John Wiley & Sons, in Summerset, NJ. at 201-469-4400, ext 2499.

ABSTRACTS

Changing Processor Mid-stream, New Electronics, March 1989
by Simon Parry
The critical hardware and software requirements which must be met in development of industrial and medical tomography applications are described, along with the reasoning used by SMIS in choosing the RTX to meet those requirements.

The First Microcontroller Optimised For Real Time,
ICS & Semiconductors, Components In Electronics, February 1989
by Chris Evans Pughe, Editor
A brief description of the history and operation of the RTX 2000 is given. The rationale used by SMIS in choosing the RTX over other microprocessors for their applications is covered, summarizing the speed, performance and design advantages. Use of the RTX by Audix in its products is also discussed.

Signal Processing Using RTX Processors and DSP Accelerator ICs, Electronic Engineering, June 1989
by David Landeta, Chris W. Malinowski, and Clay Olmstead
Harris Semiconductor, Melbourne, Florida
The close-coupling of a microprocessor with dedicated DSP accelerator ICs is discussed. Applications in image processing for filtering, feature extraction, and image transformation are discussed.

Hardwired Language Processors Enjoy Renewed Attention, Computer Design, June 1, 1989
by Ernest Meyer, Contributing Editor
The predictability and performance advantages of hardwiring high-level language instructions into embedded processor designs is discussed, along with software requirements and options. Representatives from companies which have evaluated the RTX for their applications are quoted.

High-Performance Microcontroller Offers Unique Alternative for Real-time Embedded Control, Chilton's ECN, Volume 33, No. 7, July 1989
by Chris Malinowski, Senior Scientist
Harris Semiconductor, Melbourne, FL
System response times, interrupt responses, and deterministic behavior requirements of high-speed embedded real-time systems are discussed, along with processor architectures (RISC, CISC and RTX). A comprehensive overview of the RTX hardware and software environment is provided.

Embedded Controls Improve Exercise For Handicapped, Design News, 8-21-89
by Gail M. Robinson, Associate Editor
Utilization of the RTX 2000 microprocessor to take on the normal stimulus and response functions of the human nervous system is described as part of a prototype aerobic exercise system for the handicapped under development at Wright State University.

Forth Chips Target Embedded Applications, Microprocessor Report, Aug 1989
by Brian Case, Independent Consultant
This article begins with a description of how Forth-based chip designs evolved (Novix, RTX, SC32). The similarities and differences between RISC chips and the Forth chips which are currently available are then summarized.

Three RTX Development Systems, Embedded Systems Programming, Aug. 1989
by Tyler Sperry
Three development systems which are available for the RTX 2000 Microprocessor are described, along with the advantages and limitations of each.

High-Speed Microprocessor Makes Star Tracker Functional, Design News, 5-22-89
by Lyle H. McCarty, Western Editor
This article describes the Star Tracker navigational system from Ball Aerospace, which uses the Harris "Forth Engine" (the RTX 2000 microcontroller) to meet its tough processing requirements.

Stack-Based Processor Speeds Target Recognition, Design News, 9-4-89
by David J. Bak, East Coast Editor
This article describes system and high speed processor requirements for applications such as target recognition, and how the RTX 2000 addresses those needs while typical RISC processors cannot.

Stripped-down 16-bitter at $44, EE Times, October 6, 1989
by Martin Gold
The Harris RTX Microcontroller line is described, with special focus on the RTX 2001A. New pricing information is included.

CONFERENCE PAPERS: Proceedings of the 1989 Rochester Forth Conference, Institute for Applied Forth Research:

The Harris RTX 2000 C Compiler
by Tom Hand, Senior Scientist, Harris Semiconductor
The Harris RTX 2000 C Compiler and its features are discussed, including the window/menu user interface, embedded help facility, debugger, and profiler.

Performance of the Harris RTX 2000 C Compiler
by Tom Hand, Senior Scientist, Harris Semiconductor
The run-time performance of the RTX 2000 C Compiler is discussed, along with code generated and fine-tuning with in-line Forth code.

Continuous Fourier Transform
by C.H. Ting, Offete Enterprises, Inc., San Mateo, CA.
Rick VanNorman, Harris Semiconductor, Melbourne, FL
This paper presents both the theory and an implementation of the Continuous Fourier Transform (CFT), optimized for processing continuous input signals in real time using only integer arithmetic.

Cooperative Multitasking on the RTX 2000 Revisited
by Rick VanNorman, Harris Semiconductor, Melbourne, FL
The usefulness and limitations of a simple cooperative multitasker for the RTX 2000 are explored, with task switch implementation in high-level Forth for portability and optimization.

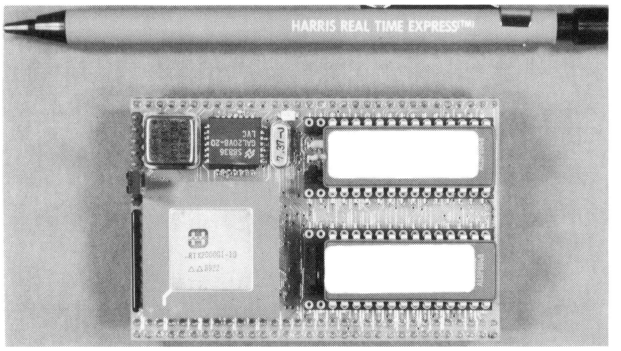
RTX 4000
by Philip Koopman and Rick VanNorman, Harris Semiconductor
This paper begins with the history behind the development of the RTX 4000, a 32-bit stack processor optimized for real-time control applications, and goes on to describe its capabilities, and features.

Beyond 64K on the Harris RTX 2000
by Mike Mellen, Harris Semiconductor, Melbourne, FL and
Harvey Glass, University of South Florida
Techniques for addressing the full memory and facilities available in the Harris TForth cross-compiler are discussed.

The publications in which these articles appear should be available through your local Technical Reference Library, or through the Publisher.
Complete Rochester Forth Conference Proceedings are available through the Institute for Applied Forth Research.
Individual copies of papers listed here may be requested through the Harris RTX Marketing Group.

FORTH-SYSTEME Angelika Flesch, Postfach 1103, D-7814 Breisach am Rhein; Telefon 49 - (0 76 67) 5 51

The **ModuNORM RTX-2001 Module** measures only 2.1" by 3.2" (about the size of a credit card). Features include 32 or 128 Kbyte high speed, 0 wait state SRAM; 64 to 256 Kbyte EPROM; a RTC72423 real-time clock; a watchdog timer, and a serial port. In addition to being available in either a 10 MHz or 8 MHz version, this board can also be configured with the RTX 2000 processor. Timing and decoding on the module is done by reprogrammable logic and up to three wait states can be inserted at various address ranges. Access to all processor, power, and UART connectors is provided at the bottom of the module.



The ModuNORM RTX-2001 Module

ARE YOU LOOKING FOR INFORMATION ABOUT FORTH?

If you are, one of the following organizations may be able to provide the information you need:

***SIGForth*, Association for Computing Machinery (ACM), P.O. Box 12115, Church Street Station, New York, New York 10249 (Telephone: 212-869-7440)** SIGForth is the ACM special interest group whose members are interested in applying Forth to solve hypothetical and real-world problems. The SIGForth newsletter is published quarterly.

In recognition of the significant and growing impact of the FORTH programming language, SIGForth will be holding its second annual Symposium on Real-time Software Development -- Tools, Techniques, and Environments, (SIGForth'90) in Dallas, Texas, February 16-18th, 1990, at the Colony Parke Hotel near Downtown Dallas.

For further information on the SIGForth'90 conference, please contact the conference chairman, Howard Harkness. His address is: 3316 Vine Ridge, Bedford, TX 76021. By phone, 817-545-6767 (home/answering machine) or 214-580-1515 ext. 545 (work). For electronic bulletin board access, use GEnie™ Email: H.Harkness).

SIGForth membership is open to both ACM and non-ACM members, and includes a quarterly newsletter and substantial discounts on SIGForth publications and events (such as the upcoming SIGForth'90). For membership information, contact ACM, PO Box 12115, Church Street Station, New York, NY 10249, (212) 869-7440.

BOOKS ON FORTH: Alternative Information Sources

Starting Forth, 2nd Edition - Leo Brody
Thinking Forth - Leo Brody
Stack Computers: the new wave - Philip J.Koopman,Jr.
Library Of Forth Routines And Utilities - James D.Terry
Object Oriented Forth - Dick Pountain
Toolbook Of Forth (Dr.Dobb's) ed. by Marlin Ouverson
Toolbook Of Forth, V.2 (Dr.Dobb's)

Forth Interest Group (FIG), P.O. Box 8231, San Jose, CA 95155 The Forth Interest Group is a world-wide, non-profit, member supported organization with over 4,000 members and 90 chapters. FIG membership includes a subscription to the bi-monthly publication, **FORTH Dimensions**, through which many of the available Forth publications may be ordered. Write for more information, or call (408) 277-0668.

***The Institute For Applied Forth Research* - 1990 Rochester Forth Conference on Embedded Systems** June 12-16, 1990: There is a call for papers on all aspects of Forth technology, applications and its implementation. Invited speakers will concentrate on Embedded Systems from satellite control to industrial controllers. Dr. Sergei Baranoff has been invited back.

Dr.Baranoff is from the Leningrad Institute for Information, and is a member of the USSR Academy of Sciences. His 1988 Russian textbook on Forth sold out in two weeks after a print run of 100,000 copies. A video featuring highlights of his 1989 conference presentations is available for purchase or rental.

Abstracts for the 1990 conference are due March 15, 1990 and final papers by May 15th. Please limit papers to 5 pages and abstracts to 100 words. The 10th conference will be held at the University of Rochester in cooperation with IEEE Computer Society and industrial sponsors. For more information, contact the Conference Chairman: Lawrence P. Forsley, Institute for Applied Forth Research.

Proceedings of the 1989 Rochester Forth Conference on Industrial Automation are now available for $25 plus $5 S/H. Included are 6 invited papers and 54 presented papers covering machine vision, object oriented programming, industrial applications, and several on the Harris RTX family. Subscriptions to the Journal of Forth Applications and Research are available.

RTX PRESS

We're Backing You Up With Products, Support, And Solutions!

Signal Processing
- Linear
- Custom Linear
- Data Conversion
- Interface
- Analog Switches
- Multiplexers
- Filters
- DSP
- Telecom

Digital
- CMOS Microprocessors and Peripherals
- CMOS Microcontrollers
- CMOS Logic
- CMOS Memories

ASICs
- Full-Custom
- Semicustom
  - Gate Arrays
  - Standard Cell
  - Cell Based
  - Core Processors
- ASIC Design Software

Power Products
- Power MOSFETs
- IGBTs
- Bipolar Discretes
- Transient Voltage Suppressors
- Opto Devices
- Power Rectifiers

Smart Power
- Power ICs
- Power ASICs
- Hybrid Programmable Switches
- Smart Power Modules
- Full-Custom High Voltage ICs

Microwave
- Gallium Arsenide FETs
- Standard MMICs
- Custom MMICs

Military/Aerospace Products
- Microprocessors and Peripherals
- Memories
- Analog ICs
- Digital ICs
- Discrete Power
  - Bipolar
  - MOSFET
- Rad-Hard ICs

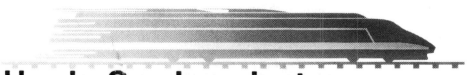
Military/Aerospace Programs
- COMSEC Programs
- Strategic and Space Programs
- Military ASIC Programs

Sales Headquarters

United States:
Melbourne, Florida (407) 724-3000

International:
Europe - Brussels, Belgium (32) 2-246-2330
Japan (Tokyo) 81-3-345-8911
Hong Kong 852-3-723-6339

Harris. What your vision of the future demands. Today.



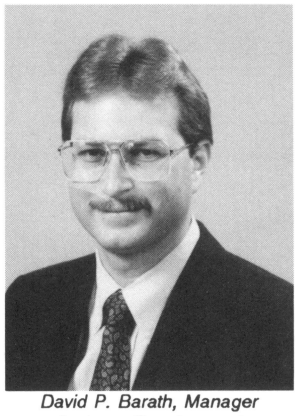
Harris Semiconductor
RTX™ Marketing
P.O. Box 883, MS 62A-021
Melbourne, FL 32902-0883

RTXPRESS℠

Vol. 2, No. 1 The Latest On Harris Real Time Express™ and Real-Time Applications June, 1990

Anniversary Issue - RTXPRESS Begins A New Year.

Real time flies when you're having real fun! The growing success of the RTX family of products has made many changes possible which will benefit everyone in the real-time system design field. This issue of RTXPRESS takes on a new look as it enters the second year of publication, and as Harris prepares some exciting new products for introduction. We are busy improving existing products and documentation - including the price list!



David P. Barath, Manager Systems Development

TRACKING THE RTXPRESS

$$$ Lower Prices On RTX Microcontrollers $$$ In March, substantial price reductions went into effect on all RTX 2000 and RTX 2001A Microcontrollers. This has allowed Harris to set new standards for real-time controller price/performance ratios. In high volumes, the RTX 2001A is under $30 for the PLCC, commercial, 8 MHz grade. Harris' policy is to maintain the RTX as the 16-bit price performance leader. New products will be very attractively priced also.

So, You Would Rather Program In C ... The RTX Marketing Group is proud to announce the availability of RTS-C™. In this issue we look at this powerful set of C development tools tailored for RTX Microcontrollers. See the brochure inside for details.

The C Development Environment Design Team Even as you read this, there is a dedicated group of individuals and third party developers working to enhance RTS-C to make it even better. We have included our "Harris C Family" photo so that you could meet some of them. We hope you'll share in our excitement over the introduction of RTS-C as a significant milestone in the refinement of the Harris RTX total solution approach to real-time design. See more on this exciting addition to Harris' product line inside.

David P. Barath

The Latest On The RTX Design Contest

from Lee Still, Manager of Microprocessor Applications What an eye opener! Even the gurus in RTX Central didn't realize how many ways a stack based microcontroller can meet embedded control system requirements.

To date, our contestants have shown us over 300 ways to apply the RTX product family to control precision equipment, help disabled persons, safeguard delicate instrumentation, communicate using multiple protocols, implement test equipment, process images, play music and on and on and on. When the contest is over we will publish the best entries, so be sure to keep reading the RTXPRESS in order to keep informed on when and how to get a copy.

In the USA, Phase I of the RTX Design Contest is completed and we are now chugging along in Phase II. By the time you read this, the Top Prize Finalists will have been picked by a group of distinguished judges at the Forth Conference in Rochester, New York. After the top prize winners are selected, we will verify their project boards

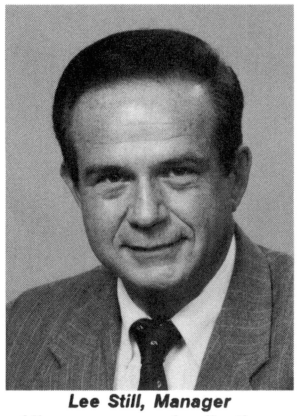
David Barath, Robert Bryant, Dan Carreira, Tom Hand, John Reed, Ted Dimbero Cinsy Krehbiel, Monnie Smith, Alice Gills

continued ...

HARRIS SEMICONDUCTOR
HARRIS · RCA · GE · INTERSIL

Copyright © 1990 by Harris Corporation

continued ...



Lee Still, Manager Microprocessor Applications

are working to be finished with final selection by August first.

The top prize winners will be notified by September 1 and their entries will be written up in the RTXPRESS. And to the grand prize winner, whoever you are, we hope to see you in September at the Embedded Systems Conference in California so that we can present your check in person.

In Northern Europe, the U.K. contest (covering England, Scotland, Ireland, and Wales) is picking up steam. **Electronic Times** and **Electronic Engineering** are both carrying contest advertisements. In cooperation with **Electronic Times**, 40,000 fliers are being distributed. In the U.K., Phase I will conclude on June 15 and Phase II will conclude on September 14. Winners will be notified in October.

The Scandanavian contest (in Norway, Sweden, Finland, and Denmark) is to be publicized in **Modern Elektronik**. Phase I will conclude on September 7, and Phase II will conclude on November 9. Scandanavian winners will be notified on or before the first of December.

In Central Europe, they've closed out Phase I and are rolling down the track to Phase II. Hard work and the cooperative efforts of **Elektronik Industrie Magazine** have resulted in making the RTX Contest well known. Over 40 excellent applications have been submitted which include simple and complex signal generation, signal analysis and medical applications. Phase II will be completed by July 20 and winners should be known by September.

In Southern Europe, the contest is in motion with entry forms available from the French or Italian office, published in French, Italian, and English. In France you can see the RTX Contest advertisements in the May 15 **Electronique Industreille** and June 1 **Electronique Hebdo**, while in Italy look out for the May and June copies of **Selezione**. Mailers have been sent out in Spain, Isreal, Benelux and France as well. Phase I will complete by June 15, and Phase II by August 31.

You can read more about the contest in the next issue of the RTXPRESS. So until the next time, good luck to all and keep the RTXPRESS rolling.

AppsCorner We know that providing high quality support is an important objective. To help meet the objective, we are starting a series of application notes to make embedded controller implementation even easier.

By the time you receive your next RTXPRESS, we expect to have the first four ready for you. The topics will include porting the RTX Development System Software to your target application or development system, choosing which RTX data bus to interface peripherals to and how to synchronize them to the RTX. How to use the stack controller to its maximum potential, and using real time interrupts at warp speed.

In another move to better meet your application needs, we have continued to augment our support teams in both Field Applications Engineering and in our internal Applications Group. If you need technical assistance, call 1-800-4-Harris. Our operator will provide you with the number of a Field Applications Engineer in your area.

The Harris RTX Electronic Bulletin Board has been discontinued so that we can provide better service directly through our Applications Engineers. In addition, we plan to establish a product support roundtable (conference) on one of the national networks. GEnie™ is currently under consideration. We'll let you know when that bulletin board has been established. So, keep reading the RTXPRESS for the latest on support and on new Applications Engineering Publications.

The **RTXPRESS** is published quarterly, and is copyrighted by Harris Semiconductor. Correspondence and submissions should be addressed to: **RTXPRESS**, Harris Semiconductor, P.O. Box 883, MS 62A-021, Melbourne, Florida 32902-0883.

Editor:
Monnie Smith

Executive Board:
David P. Barath
Linda daCosta
Michael H. Graff
Fred O. Hawkes
Ronald A. Leone
David G. Williams

Editorial Board:
Robert Bryant
Dan Carreira
Tim Dwyer
Alice Gills
Jim Gonos
Tom Hand
Philip J. Koopman
Susan Motes
John Reed
Rick VanNorman

RTX™ and RTS™ are trademarks of Harris Semiconductor. Microsoft™ C is a trademark of Microsoft Corporation; IBM™ is a trademark of IBM Corp.; GEnie™ is a trademark of General Electric.

If you would like to submit an article about a real-time application to RTXPRESS, send it in. We welcome contributions from our readers.

When Is C The Right Real-Time Tool? by *Dan Carreira*

A programming language is the vehicle through which a programmer interfaces with a computing machine. Selection of a language must be based on its ability to meet not only application requirements, but also the requirements of the machine and the human user. No software tool/language will always be the best fit for every application, because its selection must evolve from a compromise between speed/performance requirements of the application, and user interface requirements for development.

If we represent the user-machine relationship as shown in Figure 1, it can be seen that assembly/machine coding provides the interpretive interface to the machine. Languages which provide a higher level interface to the human user are comprised of assembly/machine code "building blocks".

In real-time microprocessing applications, opinions vary as to which language is the "best", but the C language usually ranks as a top contender. This is because C provides an interface between the machine and the user which offers a powerful set of software tools while minimizing the layering of interpretive code.

When performance requirements for an application are such that no interpretive layering can be tolerated, assembly language is the only solution. In the case of the RTX, Forth (RTX's native language) has produced some unprecedented results in performance.

In applications where that critical level of performance is required only during certain portions of the operation, other factors such as productivity and the ability to interface with existing development work become paramount. Use of a language like C becomes the most reasonable choice in such a case, particularly since it supports use of in-line assembly code for optimization of the speed-critical operations. We could liken C to a nail gun, while comparing assembly/machine language to a hammer. Both can drive a nail, but one will be more cost effective, depending on the application. If the tool is to be used for fine finishing work, a different set of requirements must be met than if it is needed for building structures, which would require more in the way of power and speed.

Like Forth, C supports recursion and modularity, which are critical to interrupt handling in real-time processing. But because C has a broad base of pre-defined tools, it is more readily used in large projects in which multiple programmers must work concurrently. In addition, C supports more complex parameter passing schemes and has features for a wider range of problem solving, including double precision and floating point arithmetic.

The ANSI C compatible RTS-C Cross Compiler is an optimizing C compiler for the RTX family of products. This development environment contains standard high level development tools which include the Linker, the Terminal Emulator, the RTX Assembler, and Standard C and RTX Specific Libraries. PROM programming capability is also provided. But the development tools which will be of special interest to a user are the Source Level Debugger and the Run-time Profiler.

The Source Level Debugger eliminates the need for interpretation of assembly language code by allowing debugging of application code written in C. In addition, it supports interactive display and modification of all variables, the setting of Watch points and Break points, and single step execution.

The Run-time Profiler helps in the location of run-time bottlenecks by maintaining an execution profile for a running program by reporting how many times each C statement has been executed.

So, when is C the right real-time tool? The answer depends on the requirements of each specific application, as defined by the designers. With the release of the RTS-C development environment, Harris has offered designers another choice in the set of tools available to meet their needs.

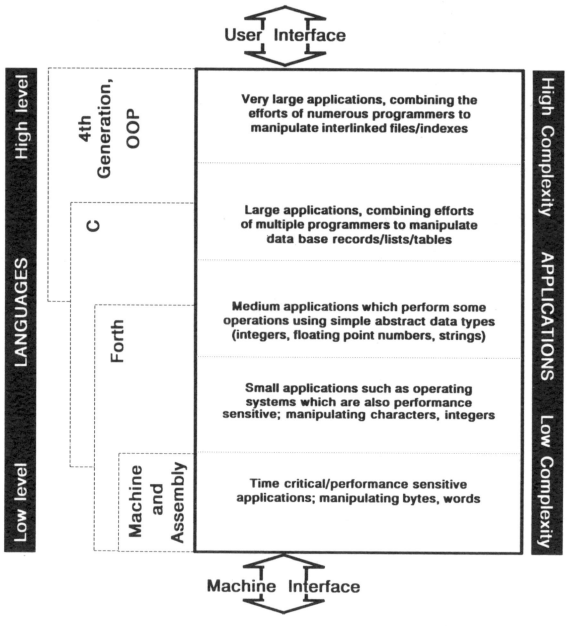


Figure 1: The User-Machine Relationship

Exterminate Your Bugs Fast With The dbgx

by Tom Hand Until recently, testing of C language applications was a complicated affair carried out by inserting numerous **printf** statements into the source code in order to display the values of important variables. This arduous process is unacceptable in today's rapid pace C language programming world.

The **dbgx** is a C high-level symbolic source language debugger, an integral part of **RTS-C**, which streamlines and quickens the testing process. With its use of pull-down menus and pop-up windows, debugging with this powerful and 'intuitive' tool is no longer a chore, but a fast and easy process. The **dbgx** provides many features not available with other debuggers; it simplifies many of the common debugging procedures, and saves developer's time in the bargain.

The abilities of the **dbgx** incorporate the features desired in an ideal testing situation to allow quick, efficient, and accurate testing. The key item allowing this streamlining is a user friendly interface through intelligently managed window selection menus. Other features that are available with the **dbgx** include:

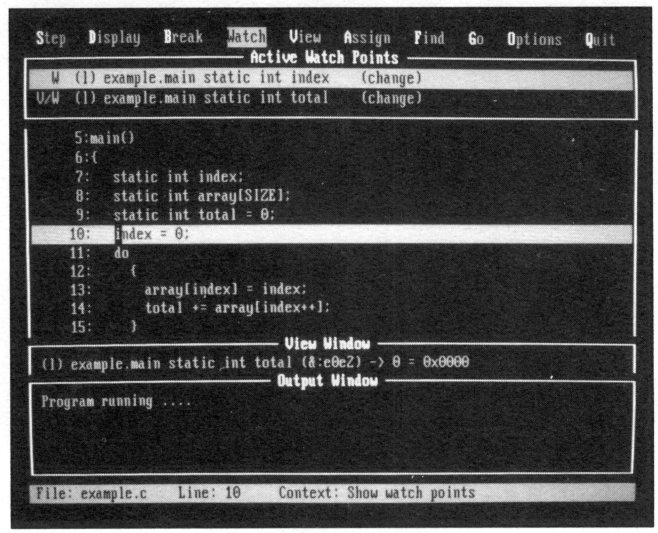
- Minimal key stokes for commands;
- Complete on-line context-sensitive HELP facility;
- Function keys to quickly perform repetitious commands;
- DOS command line which provides easy access to a familiar editor; and
- Ability to record in a file the history of selectable portions of debugging sessions.

The **dbgx** incorporates 25 'hot keys' for quick access to the most frequently used functions. Some of the more popular functions activated by hot keys include; 1) Help; 2) Show Break Points; 3) Single Step; 4) Examine Output; 5) Display Global Variables; 6) Display Local Variables; 7) Display Source Code Lines; and 8) Quit. Also, the **dbgx** has compiler switches available for specifying:

- Compilation from either the RTX 2000 or RTX 2001A Microcontroller.
- Execution with either the terminal emulator or debugger.
- Compilation with the profiler.
- The level of error messages to be displayed.

The **dbgx** has the ability to use, display, change, and assign values to view variables; locate and examine strings of source code and code generated by the cross compiler; and put assembly language code in-line with C code. The ability to take snapshots of various views of the program's execution, and scrolling within information windows allows developers to more effectively gain knowledge of their desired applications.

Screen displays found in the **dbgx** contain a menu bar of ten options across the top line as well as context-sensitive information across the bottom line. In the **Show Watch Points** screen display, shown in Screen 1, information about the two local watch points, **total** and **index**, is displayed in the **Active Watch Points Window**. Note that the **'W'** along the left-hand edge indicates that **total** and **index** are watch points while the **'V'** indicates that **total** is also a view variable.



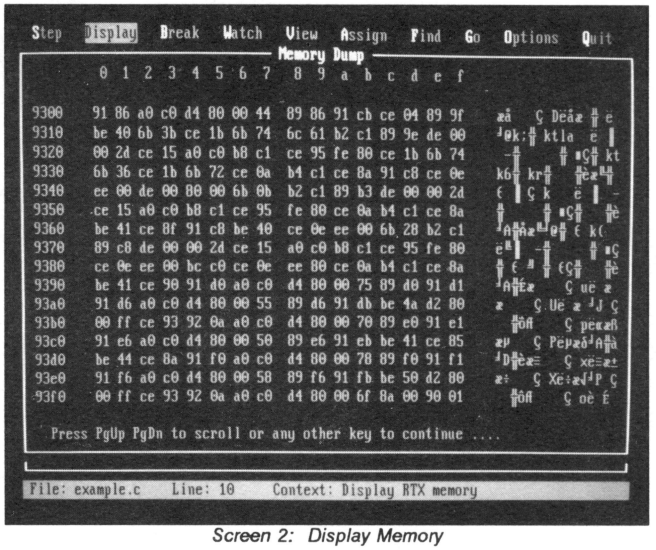
Screen 1: Show Watch Points

Screen 2 illustrates the output generated by the **Display Memory** option. Here the contents of the 256 successive bytes from memory, starting at location 9300H, are displayed in hex and ASCII format. Once displayed, three options are available. First, the **PgUp** and **PgDn** keys can be used to scroll through memory. Second, if the **Debug History** option is active, the contents of this display can be output to the file **example.log** by pressing **Cntrl-L**. The third option terminates the display memory option.

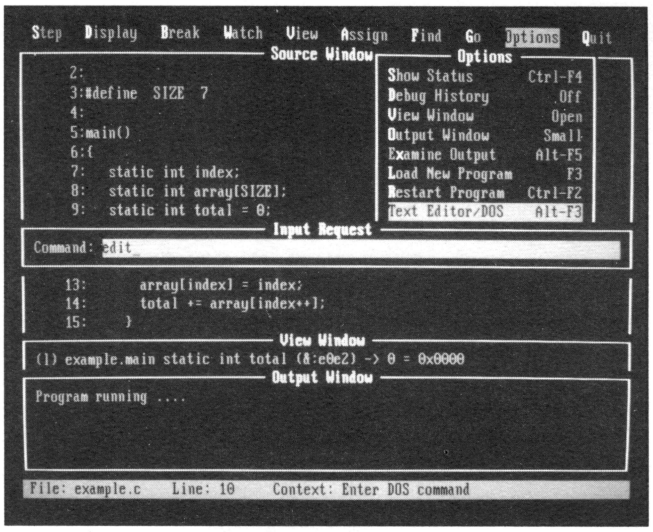
Screen 3 illustrates several capabilities of the debugger:

- The **Options Window** is open.
- The option **Text Editor/DOS** has been selected. The user-selectable default editor specified in the debug configuration file **dbgx.cfg** is **edit** and is shown in the **Input Request Window**. Any DOS command can be entered and executed if the command line in the **Input Request Window** is first cleared.
- Information about the static local variable **total** from **example.c** is shown in the **View Window**. The value is shown in both decimal and hex format and the address is given in hex format. The **View Window** has been automatically sized to contain those view variables that are appropriate for the present program position.

As applications become increasingly complex, developers must demand more sophisticated tools. The **dbgx** is a tool that can meet those needs.



Screen 2: Display Memory



Screen 3: Debugger Options

C Expression	Turbo C	C Expression	Turbo C
a = 2	4.2 X	a = 2 * a	2.8 X
a = i	2.8 X	a = 2 + a + 3	3.5 X
i++	3.0 X	a = 2 * (3 + 4)	4.6 X
a + 2 + a	2.5 X	b = (c * c) - (d * d)	3.2 X
a = a + i	2.1 X	b = a[2]	3.3 X
a = a - i	2.1 X	b = a[1] + a[2]	3.3 X
a = a * i	3.0 X	ptri = &i	3.8 X

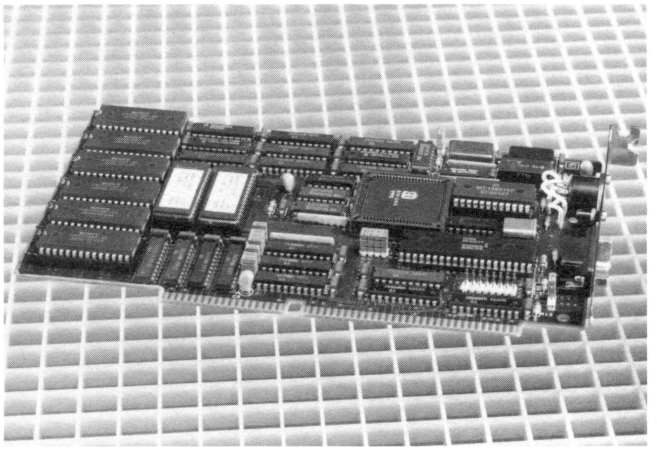
Block 1

Benchmark	Turbo C
Fibonacci	7.5 X
Queens	3.0 X
Sieve	2.4 X

Block 2

Run-time Comparisons The run-time speed of the RTS-C compiler running on a 10 MHz RTX 2000 was compared to that of the Turbo C Compiler running on a 10 MHz PC/AT. For simple C instructions, execution speeds of the Turbo C system were significantly slower than execution speeds using RTS-C. Block 1 shows how many times longer the Turbo C system took to execute these operations compared to RTS-C.

To analyze these results for larger programs, standard benchmarks were used. The RTS-C system again outperformed the Turbo C system. Block 2 shows how many times longer the Turbo C system took than the RTS-C system to perform these standard benchmarks.



The AT2000 From VMEinc.

VMEinc., ***538A Valley Way, Milpitas, CA 95035; Telephone: 408-946-3833*** VMEinc has just announced the availability of the AT2000, which is a fully functional 15+ MIPS processor board capable of functioning as an IBM™ PC/AT Bus Master.

Integrating an AT Bus Controller function with the Harris RTX 2000/2001A Microcontroller has resulted in a product which has the ability to access AT Bus Memory and I/O directly, performing transfers at up to 10 Mbytes/second and responding to AT interrupts in as little as 400 ns.

The AT2000 Single Board Computer is targeted for use in data acquisition and analysis, test and support equipment, and embedded control applications.

continued ...

Backtracking Many of you may remember Doug Ross, with NASA. His Wideband Transport Frame Formatter application was described in the September 1989 issue of RTXPRESS. Lest anyone think the use of Forth has been abandoned in the excitement surrounding the release of RTS-C, Doug has provided a sample of how Forth for the RTX can be used to achieve faster performance than that possible using another language.

For this example, he presents a Pascal routine to permute an input data array for a Fast Hartley Transform (FHT). PERMUTE takes an index value and calculates its permuted value. By re-ordering the input data array into adjacent pairs of permuted input samples which represent butterfly inputs, faster transform computation is achieved.

The sample in Block 1 was taken from "Faster Than Fast Fourier," page 296 in the April 1988 issue of ***Byte*** magazine. Doug took this Pascal routine and performed a literal translation into Forth for the RTX 2000 (in Block 2). After some functional optimization (Block 3), he performed full functional optimization with the simple coding implementation shown in Block 4.

These versions of PERMUTE take index values from 1--N as inputs. This mimics the Pascal version. However, by deleting the 1 - in line 3 and 1 + in the last line, the routine will take index values from 0--(N-1). This better suits Forth's array indexing practice.

Another noteworthy aspect of these routines is that the code on each line between BEGIN and NEXT requires only one clock cycle to execute. In fact, the code in Block 4 executes in 3N + 7 cycles, where N = PWR_INDEX.

TEST-PERMUTE is used to show how an input data array is re-ordered. It is used as 9 512 TEST-PERMUTE, 5 32 TEST-PERMUTE, etc., where NPTS = 2^PWR-INDEX.

```
: TEST-PERMUTE ( PWR-INDEX  NPTS --) CR
  DUP 1 - FOR 2DUP R@ - SWAP PERMUTE 5 U.R NEXT DROP DROP ;
```

We'd like to thank Doug for sharing this example of how to improve performance through thoughtful selection of software tools.

```
1   function permute(index: integer): integer ;
2   var
3     i,j,s: integer
4   begin
5     j := 0 ;
6     index  := index - 1 ;
7     for 1  := to power_index do
8     begin
9       s  := index div 2 ;
10      j  := j + j index - s - s ;
11      index := s ;
12    end ;
13    permute := j + 1 ;
14  end ;
```

Block 1

```
1  : PERMUTE  (  INDEX  PWR_INDEX -- N)
2    1 - >R   (  SET LOOP COUNT)
3    1 -  0   (  INDEX-1        J=0   INITIAL STACK    )
4    BEGIN    (   INDX           J                     )
5      2*     (   INDX          2J                     )
6    OVER 2/  (   INDX          2J          S          )
7      2*     (   INDX          2J         2S          )
8      -      (   INDX        2J-2S                    )
9    OVER +   (   INDX      2J-2S+INDX                 )
10   SWAP 2/  (     J'       INDX/2                     )
11   SWAP     (     S          J'                       )
12   NEXT
13   NIP 1 + ; ( LEAVE PERMUTED INDEX J ON STACK     )
```

Block 2

```
1  : PERMUTE  (  INDEX  PWR_INDEX -- N)
2    1 - >R      (  SET LOOP COUNT)
3    1 -  0      (  INDEX-1     J=0 INITIAL STACK    )
4    BEGIN       (   INDX        J                   )
5      2*        (   INDX       2J                   )
6     OVER       (   INDX       2J       INDX        )
7    1 AND       (   INDX       2J        0¦1        )
8      +         (   INDX     2J¦2J+1                )
9    SWAP 2/     ( 2J¦2J+1   INDX/2                  )
10   SWAP        (   INDX/2  2J¦2J+1                 )
11   NEXT
12     NIP 1 + ; (  LEAVE PERMUTED INDEX J ON STACK )
```

Block 3

```
1 : PERMUTE  (  INDEX  PWR_INDEX -- N)
2    1 - >R  ( SET LOOP COUNT)
3    1 -  0  ( INDEX-1      J=0     INITIAL STACK      )
4   BEGIN    (  INDX         J                         )
5    SWAP c2/(    J        INDX/2    LSB --> c        )
6    SWAP 2*c( INDX/2       2J+c     LSB <-- c        )
7   NEXT
8    NIP 1 + ;  ( LEAVE PERMUTED INDEX J ON STACK )
```

Block 4

VMEinc continued ...

In addition to the Harris RTX processor, the AT2000 SBC includes an RS-232 serial port, an RS422/RS485 serial port with multidrop capability, and a 16-bit parallel port for external I/O at rates up to 20 Mbytes/second. Sockets are included to support up to 768 Kbytes of local RAM/ROM, a real-time calendar clock, and a keyboard port. Bus access for the entire AT I/O space, up to 448 Kbytes of AT Memory and up to 6 AT interrupt levels, is also provided.

Software support includes ANSI based Forth and C Languages with optimizing compilers, uncompilers, and real-time interactive debugger/monitors.

For more information, call Bill Holloway at 408-946-3833.

Adding I/O Address Space

by Richard Marriott, MICRO AMPS LTD., 233 Stoughton Road, Guildford, Surrey, U.K., GU2 6PG; Tel: (0483)/268999 The RTX processor has eight 16-bit locations on the external I/O bus, G-port addresses 18H to 1FH. In some applications, eight I/O locations are insufficient. For example, when adding serial and/or parallel ports to the system design, it may be desired to implement a page switch mechanism (where a page is a set of addresses).

The Micro AMPS "MACH2" board can provide additional I/O address space if a page switch mechanism is implemented on the board. The technique used is similar to that used within the RTX architecture itself for extended code and data pages.

With a page switch, the G-port address range can be extended by 16 address lines, because the RTX G-port is 16 bits wide, using a single G-port address location. The Micro AMPS MACH2 board extends the G-port address by 3 address lines from the page switch. The page switch can be addressed in any page by reading or writing to the G-port address 1FH, which is reserved in all pages for this purpose. On the MACH2 board the logic shown in Figure 1 was implemented in a programmable gate array. This mechanism allows access to different devices on separate pages. The page switch implemented gives a maximum of 56 locations in 8 pages.

To access the page switch, we use the following Forth definitions:

```
: >page ( n - )   1f g! ;
: page> ( - n )   1f g@ ;
```

Each function executes in two clock cycles taking 200 nsec at 10 MHz.

1. a) set the page switch to the required value
   b) read or write data to the appropriate location using the g@ and g! operators
   c) reset the page switch to a known state
2. a) read the page switch to the stack
   b) set the page switch to the required value
   c) read or write the data
   d) reset the page switch to the original value held on the stack
3. a) set the page switch to the required value
   b) read or write the data
   c) leave the page switch in the state set in a)

Page Switch Options

When programming an application using the page switch mechanism, the way in which the page switch will be used must be determined. Three options are possible (see insert).

The simplest is option (3), in which the page switch is set to the required value immediately before accessing the page. To read data from a device in page 2, the code shown below could be used, where **pio@** is an instruction that operates on a device in page 2.

```
2 > page pio@
```

Option (2) is required in multitasking systems because the state of the page switch must be saved before re-assigning the switch to a new page, then restored at the end of the routine. The following code could be used in a multitasking system:

```
page>  2 >page pio@ swap >page
```

The single **pio@** instruction can be replaced by several accesses to the same page to improve the efficiency of the page switch mechanism.

Adopting this convention of accessing different devices in separate pages is a convenient approach to hardware and software design. Hardware timing of the G-port decoding is eased because the page switch output is set up a minimum 50 nsec before the next GIO* signal becomes active and allows use of slower interface logic.

The efficiency of this page switch mechanism is largely governed by the way in which it is to be used and programmed. In most applications the RTX processor is fast enough for the page switch efficiency to be a low design priority.

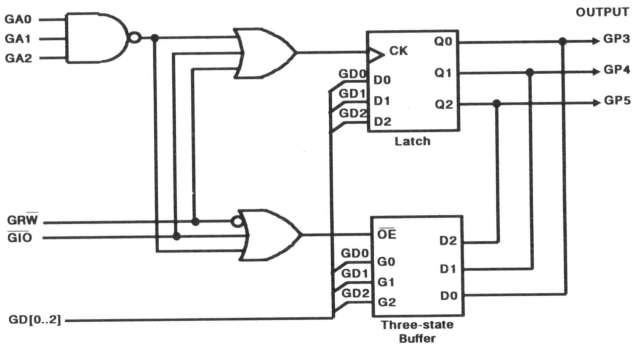


Figure 1: Page Switch Logic

STACK COMPUTERS the new wave by Philip J. Koopman, Jr. (reviewed in RTXPRESS, December 1989) is now available from Prentice Hall (Old Tappan, NJ). ISBN 0138379238, $61.50 in the United States. To place an order using a credit card, call 1-201-767-5937.

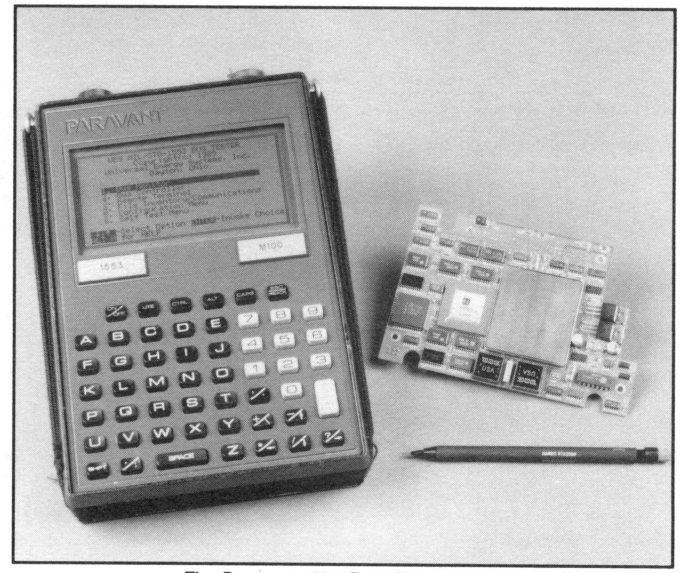
How Can A 1553 Data Bus Analyzer Be Used In Really Rough Terrain? This was a question posed by Paravant Computer Systems prior to the development of their 1553 Data Bus Analyzer. In fact, when it involves lugging over 40 pounds of power hungry test equipment out in rough terrain to troubleshoot a MIL-STD-1553 avionics bus system, it can be grueling, if not impossible to use the equipment in common use today.

Conversely, a handheld 1553 Data Bus Analyzer would make testing in even the worst field conditions simple, just the solution that Paravant Computers proposed. We spoke with Dave Carter, their 1553 Program Manager, to gain some insights into their experiences in developing this solution. Their Model 100 1553 Data Bus Analyzer was developed as a programmable option board for their RHC-88 handheld MS-DOS computer. The result is a handheld, full function 1553 tester. This product can perform high speed analysis of 1553 data bus traffic, collection of bus data in real-time conditions, and field identification of faulty Remote Terminal Units (RTU's) either on the flight line or in the field.

Production of a design with all these capabilities was not without its difficulties. There is a limit to the amount of hardware that can be packed into a box that measures only 9.4" by 6.4" by 2.6", and weighs only 5.0 pounds, including a battery pack. In addition, only a low power CMOS device was acceptable since this product required low power consumption during battery powered field test applications. But the most critical problem facing this development project was the need to find a processor that could provide the speed and performance required for intensive signal analysis during bus monitor applications, particularly when trying to filter address and subaddress information. The designers were forced to consider bit slice technology or going with a custom IC because of the lack of options available.

The problems looked insurmountable. The option board was nearly out of space, and the expense and time requirements needed to achieve a bit slice or custom IC solution looked exorbitant. The designers were faced with the unpalatable choices of either investing immense amounts of time and money in the project, or scrapping the design. Of course, neither solution was acceptable.

It was at this point that designers at Paravant learned about the Harris RTX 2001. They found that in addition to providing the high processing speeds needed, the RTX could also offer a high level of integration that eliminated some of the peripheral hardware which would have otherwise been included on the board to support a different type of processor. The net result was a design that included a 1553 Hybrid Module, a RHC/RLT processor



The Paravant 1553 Data Bus Analyzer

interface, additional memory, power input conversion hardware, a 16 MHz clock, "glue logic", and the RTX (see Figure 1).

The RTX: How does it make a difference? The processing requirements placed on a 1553 analyzer are rigorous. Every word transmitted must be analyzed to determine its type (see *"A brief look at 1553"*), then analyzed further for content and detection of errors. Control words (command and status) can easily be differentiated from data words due to the different sync patterns, but command words and status words have identical sync patterns. If an error occurs and sufficient analysis of each word is not

continued ...

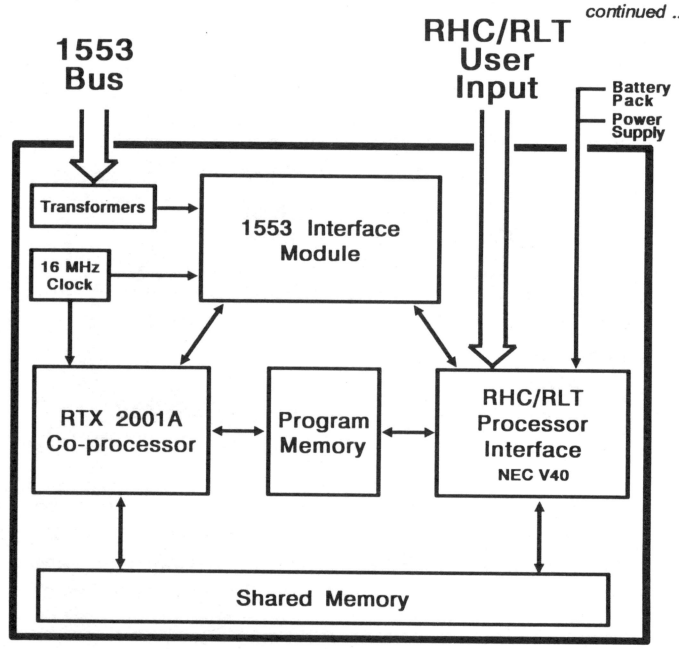


Figure 1: 1553 Bus Analyzer Option Board

continued ...

performed, the bus analyzer can mistake the next command sync pattern received to be the status sync pattern which should have been detected (see Figure 2). Since a status word may either preceed the data words or follow the data words, depending on the type of command issued, there are no safe shortcuts.

Thorough analysis of each word is essential to prevent erroneous results, and therefore becomes a task requiring high speed processing. This factor was crucial in the selection of the RTX for this application.

Why was the RTS-C Cross Compiler used during development? Concurrent development of the hardware and software was essential in order to complete the project in a timely manner. For this reason, early development work was performed and debugged on a PC using Microsoft™ C, then ported to RTS-C when the compiler and RTX hardware were made available by Harris. This provided a significant head start on the development cycle.

Another advantage provided by RTS-C became apparent when project development was complete. The memory used by the C development environment was found to be only 4K, and less than 2K of that was for code.

In addition, the portability of the C environment allows a change to a header file on "one side" to change both sides. This became essential in the final implementation because the RHC-88 (Microsoft C-based) and the RTX (RTS-C-based) utilize shared data structures and are required to pass data back and forth.

As a MS-DOS computer with a programmable option board, the Paravant 1553 Data Bus Analyzer interface provides a generic display of bus data and message errors. Using a PC development environment, a system integrator may write a custom user interface for any 1553 equipped vehicle.

The flexibility doesn't stop there, because the Harris RTX 2001's program can be downloaded. This approach allows Paravant customers to re-program the devices to their specific needs.

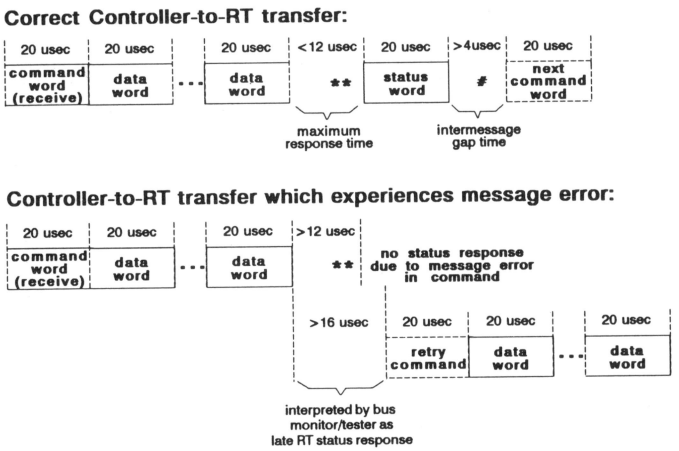


Figure 2: 1553 Transfer Error Sample

A Brief Look At 1553: MIL-STD-1553B is a military standard which describes the method of communication and the electrical interface requirements for subsystems connected to a 1553 Aircraft Internal Time Division Command/Response Multiplex Data Bus. This 1 Mbps serial communication bus is used to achieve aircraft avionics integration and may be used to extend the systems integration to flight controls, propulsion controls, and vehicle management systems. In addition to defining the specific characteristics for the twisted pair shielded cable which is used for the data bus, MIL-STD-1553B also defines specifications for the **Bus Controller**, embedded and stand-alone **Remote Terminals**, and the **Bus Monitor**.

The **Bus Controller** is the sole source of control on the data bus, and uses a command/response method directed at remote terminals to control data flow. A **Remote Terminal** is the interface to a sensor or subsystem on the data bus, and performs data transfer in and out of the subsystem, as controlled by the bus controller. The **Bus Monitor** listens to messages and collects data from the data bus and is typically used to perform diagnostics on a bus.

MIL-STD-1553B bus communications are based on message transmission, and follow carefully prescribed transfer formats which describe each of the 10 possible message types (see Figure 3). The format of each message in 1553 communications is made up of control words and data words based on a 20-bit word format. This word format is comprised of a sync pattern, 16 bits of data, and 1 parity bit.

Control words are used in system communication and data bus system control, and can be either **command** words or **status** words; **data words** are used to encode communication between system components.

A **command word** provides the definition of the message format to be transmitted. It can only be transmitted by the bus controller, and is used to initiate a sequence. The command word sync pattern is identical to the status word sync, and is a unique invalid Manchester waveform. This sync pattern cannot be duplicated for data words (see Figure 4) and therefore can always be distinguished from data word sync patterns.

continued ...

For more information about the Paravant Handheld 1553 Data Bus Analyzer, contact Dave Carter at Paravant (a UES Company), 305 East Drive, West Melbourne, Florida 32904; 407-727-3672.

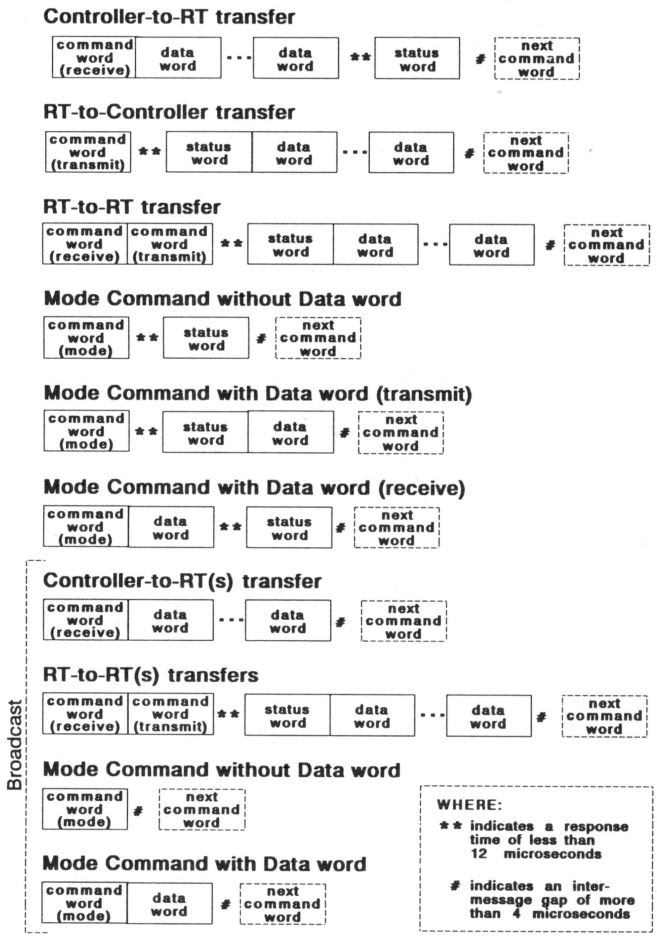


Figure 3: 1553 Transfer Formats

continued ...

A command word can be followed by data, another command word, or a response time gap prior to status word transmission by the remote terminal. Each command word is comprised of the sync pattern and 17 contiguous, valid Manchester II bit times. These 17 bit times are defined as a 5-bit remote terminal (RT) address for the RT being addressed, a transmit/receive (T/R) bit which indicates the direction of flow of data words to or from the RT, the subaddress/mode code definition field, a 5-bit field indicating word count or mode code, and a parity bit.

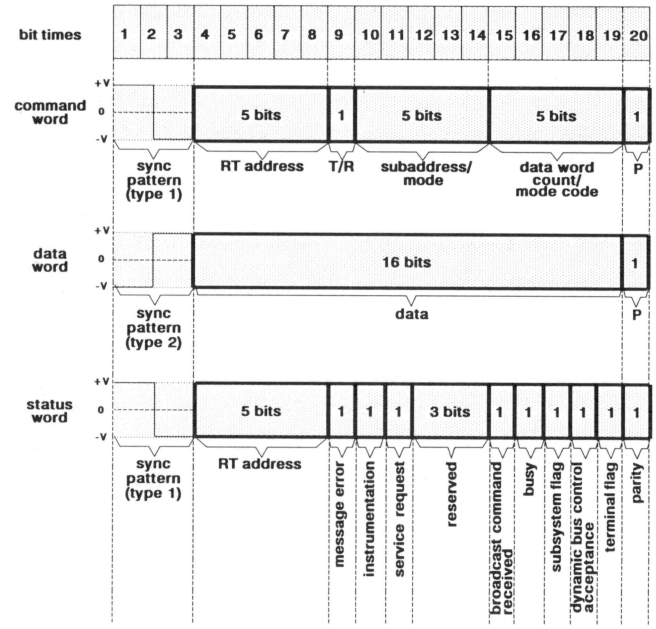


Figure 4: 1553 Sync Patterns/Word Formats

Each **status word** contains the sync pattern, an RT address so that the status word can be validated by the bus controller, a message error bit, ten additional message-specific status bits, and a parity bit (see Figure 4). The message error bit is used to identify messages which do not pass the word or message validation tests. Messages that are not error free will not have a corresponding status word. This allows the bus controller to time-out when no status response is received.

A **data word** contains a sync pattern (which is different from that used for command and status words), 16 data bits, and one parity bit. The only restriction currently placed on the encoding of the data field by MIL-STD-1553B is that the most significant bit must be transmitted first.

For more information, see MIL-STD-1553B.

Silicon Composers

208 California Avenue, Palo Alto, CA 94306

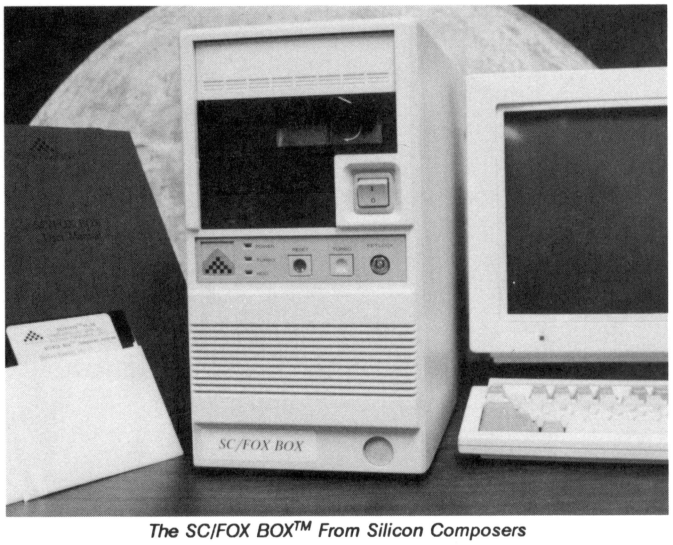
The **SC/FOX BOX™** is a stand-alone computer system designed to serve as both a real-time development platform and as an embedded computer for embedded system applications.

Utilizing the features and high speed performance of the RTX 2000, the SC/FOX BOX computer system environment allows signals to be acquired and conditioned from an external device at SCSI device recording speeds, making it ideal for real-time control, data acquisition, image and signal processing, factory automation, process control, and computation-intense applications.

The RTX 2000-based SC/FOX SBC single-board computer serves as the mother board for the SC/FOX BOX, for which the 8MHz standard configuration includes 128K bytes of 0-wait-state RAM, an RTX-based SCSI I/O Controller board, a 40 megabyte SCSI hard-disk drive, and two 5.25" floppy drives.

Two 56K baud RS-232 serial ports, a Centronics parallel printer port, a 16-bit bidirectional parallel port, and a SCSI port are brought to connectors on the back of the SC/FOX BOX mini-tower enclosure. SCSI transfer rates are 3M-bytes asynchronous and 5M-bytes synchronous.

continued ...



The SC/FOX BOX™ From Silicon Composers

Upgrades include 10 and 12 MHz operation with 256K bytes or 512K bytes of zero-wait-state static RAM and 100 or 384 megabyte SCSI hard disk drives. Custom configurations are also available.

The operating system and development software include the SC/Forth™ Language in EPROM, an interactive real-time multi-tasking Forth environment, operating system, and development system.

Only a serial cable and dumb terminal or PC are required to operate the SC/FOX BOX. Communication is through the 56K baud RS-232 serial terminal port.

For more information about this product, call George Nicol at 415-322-8763; SC/FOX, SC/FOX BOX, SC/FOX SBC and SC/Forth are trademarks of Silicon Composers, Inc.

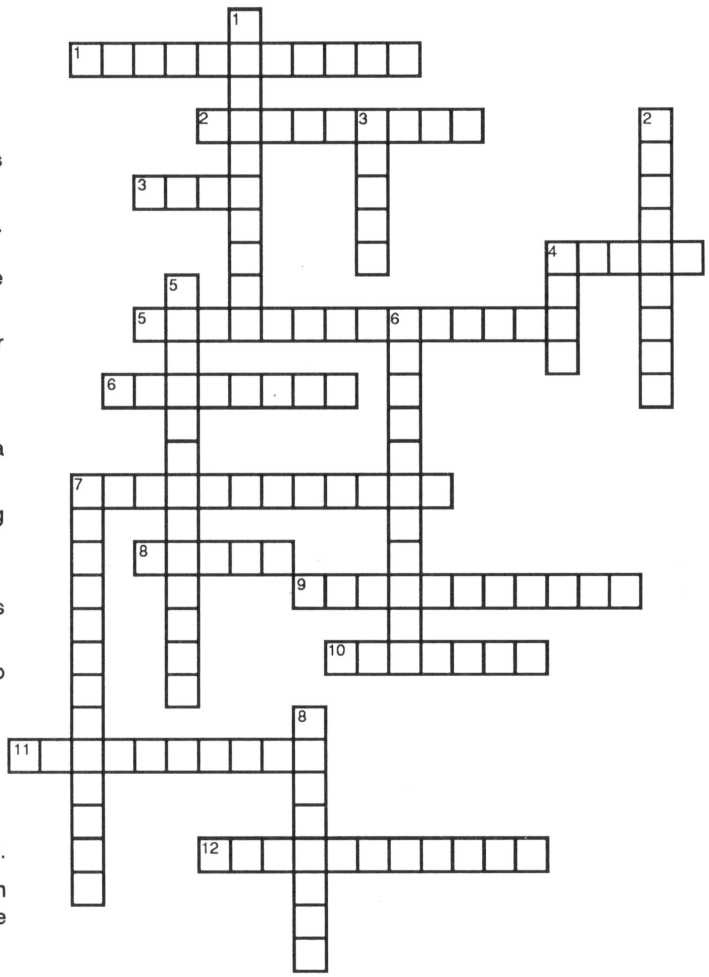
RTS-C Crossword Puzzle *(Look in the C Development Environment brochure insert if you need help with the answers!)*

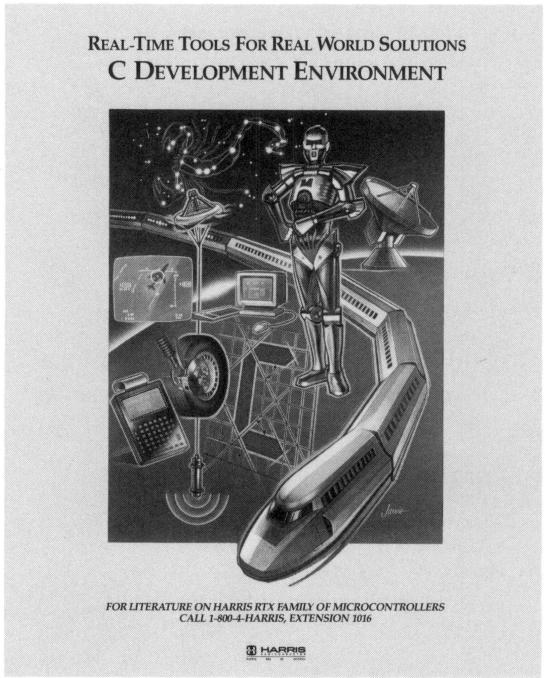
Across

1. When debugging, ___ ___ can be set to occur at entry of a function, at a source code line # or by moving the cursor.
2. These functions are found in Standard C and in the RTX-specific C Cross Development Environment.
3. ___ Standard (abbr.)
4. It's what you do when you solve program malfunctions.
5. ___ ___ allows execution of one C expression at a time.
6. With their ___ ___ consumption, the RTX microcontrollers are ideal for a variety of applications.
7. The ___ ___ displays a program that has been written in C.
8. A ___ point is used to detect when the value of a variable changes according to a predefined condition.
9. The **dbgx** provides an ___ symbolic C language debugger for testing code written with the **ccx** compiler.
10. ___ ___ give easy access to frequently used functions.
11. An ___ ___ allows selection of many functions from a screen generated list.
12. A performance characteristic of the RTX permitting quantifiable execution, resulting in a ___ system.

Down

1. High efficiency is attainable because Harris' **ccx** cross compiler is an ___ compiler.
2. The ___ profile generated by the Harris Profiler will help developers to determine locations of run-time bottlenecks.
3. Real Time Express Development Board (abbr.)
4. Harris' C language debugger.
5. ___ ___ are displayed when a break in execution occurs.
6. Collection of programs constituting a user software interface.
7. User developed, in-line assembly code permits optimization of ___ ___ code sections providing maximum performance from the RTX architecture.
8. ___ ___ menus descend from the top of the screen when prompted by the selection of a screen option.





If you've seen the latest RTX applications ad, you certainly won't want to miss the new RTX poster, which continues the theme. Just check the appropriate box on the attached reader response card, and we'll send you one!

Sales Office Headquarters

United States
Harris Semiconductor
1301 Woody Burke Road
Melbourne, Florida 32902
TEL: (407)724-3739
1-800-4-Harris, Ext. 1016

Europe
Harris Semiconductor
Mercure Centre
Rue de la Fusse 100
Brussels, Belgium 1130
TEL: (32)246-2201

North Asia
Harris K.K.
Shinjuku NS Bldg.
Box 6153
2-4-1 Nishi-Shinjuku
Shinjuku-Ku, Tokyo 163
Japan
TEL: 81-3-345-8911

South Asia
Harris Semiconductor
H.K. Ltd.
13/F Fourseas Building
208-212 Nathan Road
Tsimshatsui, Kowloon
Hong Kong
TEL: (852)3-723-6339

A Forth Kernel to Support RTX Products - Is Expensive Software Necessary? Look for a discussion of **AppForth** in the next issue of RTXPRESS. **AppForth** is the Harris RTX-Forth Application Note which provides stand-alone Forth for the RTX 2000 family of processors. This valuable tool will become available late this summer from Harris, at no charge. Look for more details about the features of **AppForth** in the next issue of RTXPRESS.

Is There A UART Fast Enough To Match Today's Microprocessor Speeds?

Yes. To learn more about the RTX 2152, make sure to read the next issue of RTXPRESS. In that issue we'll tell you all about this single-chip high performance programmable Universal Asynchronous Receiver/Transmitter and Baud Rate Generator, and how it supports data rates up to 2.06 MBaud asynchronously, with a 16X clock (0 to 33 MHz clock frequency). Many of the features of this low power CMOS product will be described, including the simple programming procedure, and 72 selectable baud rates.

Harris Semiconductor
RTX™ Marketing
P.O. Box 883, MS 62A-021
Melbourne, FL 32902-0883

HARRIS SM
RTX Real Time Express