

# RTX 4000

*Philip Koopman  
Rick VanNorman*

*Harris Semiconductor*

## Introduction

Harris Semiconductor has begun design of the RTX 4000, a 32-bit member of the Real Time Express (RTX) product family. The RTX 4000 will be a 32-bit stack processor optimized for real time control applications, with similar design objectives as the RTX 2000.

## History

The RTX 4000 project has its roots in the work of WISC Technologies, and the WISC CPU/32. The CPU/32, introduced at the Rochester Forth Conference in 1987, was a discrete TTL design that used six printed circuit boards and ran at 6 MHz.

After the introduction to the CPU/32, Harris Semiconductor licensed patent rights to the design, and produced the RTX 32P. The RTX 32P is an exact duplication of the CPU/32, only using a pair of 2.5 micron semi-custom integrated circuits to fit the design onto a single printed circuit board. The RTX 32P executes programs at 8 MHz in typical operating conditions.

The RTX 4000 will be a single-chip production version of the RTX 32P, with substantial design improvements. These improvements will allow an operating speed of at least 18 MHz across the commercial operating environment on 2.0 micron technology, and 25 MHz on 1.2 micron technology (soon to be available). The actual chips may well be significantly faster, depending on the success of detailed design optimizations now in progress.

## RTX 4000 Block Diagram

The RTX 4000 is based on a 2-stack, 0-operand computation model, very similar to the Forth virtual machine. It is capable of executing instructions that correspond to all the RTX 2000 instructions with an appropriate set of microcode, and can go beyond the

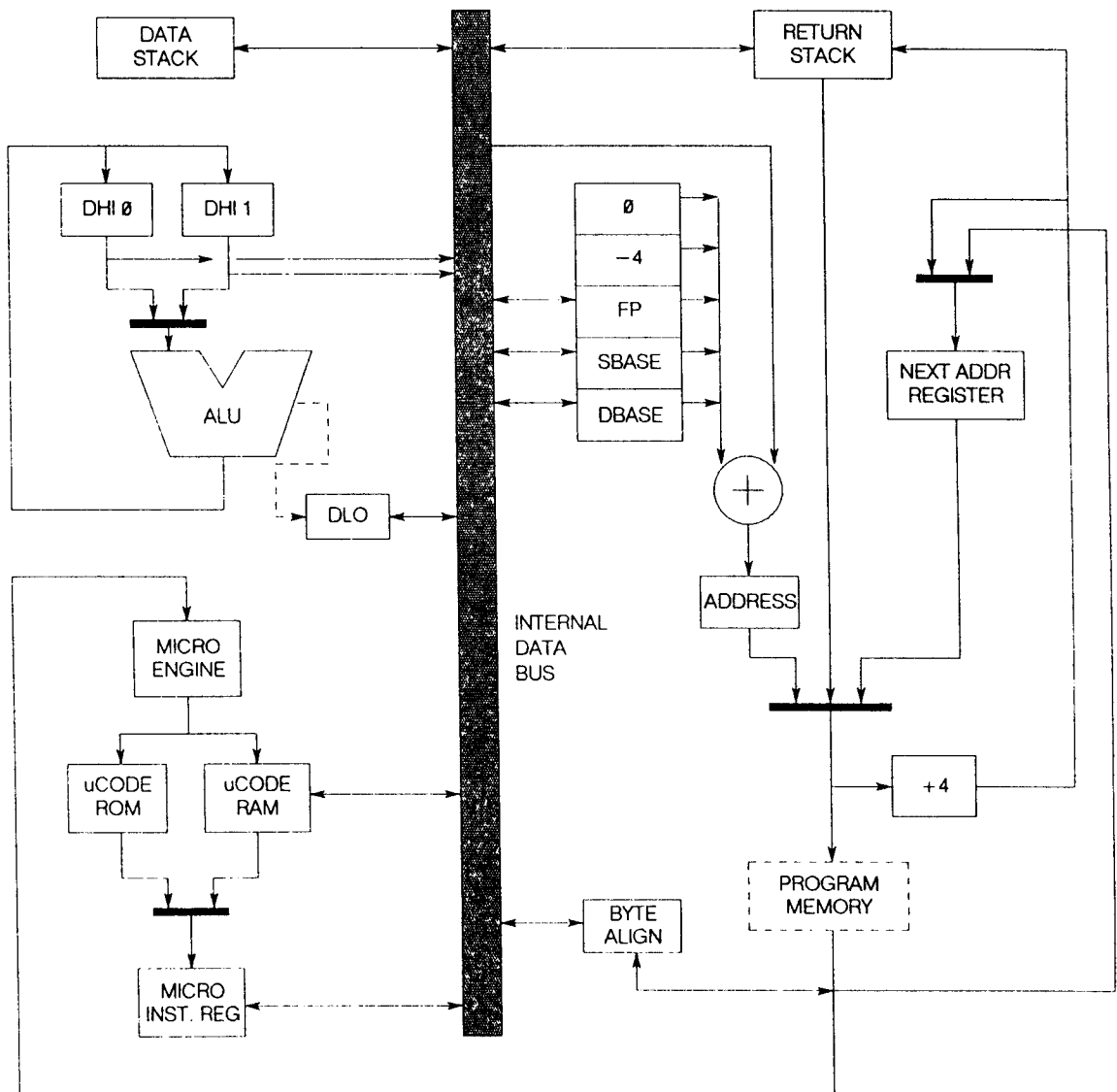


Figure 1. RTX 4000 Block Diagram

RTX 2000's capabilities with direct support for other high level languages such as C.

Figure 1 shows a simplified block diagram of the RTX 4000. The RTX 4000 has two independent on-chip stacks, of between 32 and 256 elements (depending on the particular implementation). The ALU contains integer logic and, in some versions, fast hardware floating point multiplication and addition logic. The DHI0 register is used as a top-of-stack buffer register, while DHI1 is used as a scratch-pad register to avoid congestion at the ALU. The DLO register is a scratch-pad register used to perform 64-bit shifting with the ALU.

The RTX 4000 has a Next Address Register and an incrementer that, when used together, can emulate a program counter. Memory addresses are supplied for instruction fetches either from the Next Address Register or the return stack. Data fetches and stores use the ADDR register, which is loaded with a base-plus-offset address from one of four base registers (register 0 is a constant 0 value). Program memory, most or all of which is off-chip, can be accessed as appropriately aligned words, half-words, or bytes.

A very simple micro-engine executes instructions from on-chip ROM and RAM, using a microinstruction register to control the chip. Typical ROM sizes will be from 512 words to 2K words, and RAM sizes from 64 words to 512 words. Eight microcode words are provided in support of each opcode. Microcode RAM is writable on-the-fly while executing programs.

**RTX 4000 Instruction Format**

The RTX 4000 is optimized for two-cycle program memory. This is in recognition of the fact that real time embedded systems often can not afford the exotic technology required to support single-cycle memory access at high clock speeds. In order to speed up program execution, the RTX 4000 allows processing two opcodes, or an opcode and a subroutine call/return for every instruction fetch. This gives an instruction execution rate of one instruction per clock cycle for many code sequences.

Figure 2 shows the instruction formats for the RTX 4000. There are four instruction types: call, return, sequential, and dual opcode. Figure 2a shows that three of the four instruction types use the traditional WISC instruction format of a 9-bit opcode, 21-bit address field, and 2-bit control field. For subroutine calls, the 9-bit opcode is executed in parallel with a subroutine call to a word-aligned address contained in bits 2-22. For subroutine returns and sequential program execution (i.e. next address is current address plus 4), bits 2-22 provide a 23-bit sign-extended literal value that may be used by the opcode. Subroutine returns proceed in parallel with opcode execution.

Figure 2b shows a new instruction format that was introduced after experimentation with the RTX 32P. It was found that there were many sequences of opcodes with no intervening subroutine calls. Thus, the RTX 4000 has an instruction type that packs two opcodes per instruction word, with each opcode having a 6-bit signed literal value. In the frequent case that opcodes can execute in a single clock cycle, the dual opcode instruction format permits executing two opcodes per memory bus cycle.

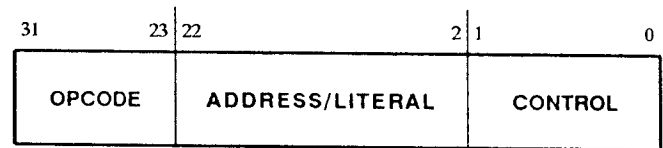
The RTX 4000 is a microcoded architecture. This allows using compact, 9-bit opcodes while providing for 512 possible opcodes. The usual complexities and speed penalties associated with microcoded complex instruction set computer architectures are

avoided by using the implied addressing inherent in a stack-based design. The use of microcode allows direct support for commonly occurring instructions that are too complicated to be handled in a single clock cycle. Of special importance are words that perform block memory transfers and complicated stack manipulations. We have found that the RTX 4000 takes approximately the same number of clock cycles to execute Forth programs as the RTX 2000, but uses only half the instruction fetch memory cycles (and, can execute at a faster clock speed as well). For special applications, customized microcode can provide significant performance increases.

**Testability**

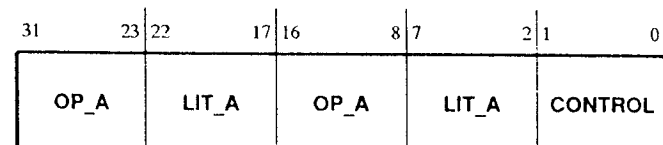
Much ado is made about testability on other microprocessors, but very little is actually done to promote complete and quick testing without significant sacrifice of chip resources. The RTX 4000, in contrast, is a very testable chip. The RTX 4000 uses a single test pin to allow direct access to the microinstruction register (using 32-bit parallel operations, NOT bit-serial operations). The RTX 4000 permits single-stepping microcode provided from an external source for testing, and allows reading and writing all major registers to and from an external test device via the data bus in a single clock cycle. All microcode RAM and ROM is directly accessible during program execution.

To make this testability available to developers, the first printed circuit board developed by Harris for the RTX 4000 will be an IBM PC plug-in card with software that supports single-stepping user-supplied microinstructions. This approach has already been used with the RTX 32P with great success.



CONTROL	ADDRESS / LITERAL
0 SUBROUTINE CALL	SUBROUTINE ADDRESS
1 SUBROUTINE RETURN	SHORT LITERAL (21 BITS)
3 SEQUENTIAL EXECUTION	SHORT LITERAL (21 BITS)

**OPCODE NUMBER**  
0 ... 511



CONTROL	ADDRESS / LITERAL
2 DUAL OPCODE / JUMP TO NEXT	

**OPCODE NUMBER**  
0 ... 511

*note: hex 80 in bits 2-7 specifies subroutine return*

Figure 2. Instruction Formats

### *The Future*

Envisioned applications for the RTX 4000 include any application that requires a very large memory space, fast floating point hardware support, or 32-bit integer precision calculations. Examples of these applications are: laser printer control engines, display control engines, image processing equipment, DSP applications, robotics controllers, telecommunications, and parallel processing. Special on-chip hardware support will be available in some RTX 4000 versions for DRAM control, counter/timers, and application-specific hardware support.

There will be several variants of the RTX 4000 family, including the RTX 4002 (a beta-test development chip, probably in 2.0 micron technology), RTX 4000 (a 1.2 micron chip with floating point unit and many on-chip peripherals), the RTX 4001 (a lower-cost 1.2 micron chip without a floating point unit). Since the RTX 4000 is being designed using standard-cell methods, new versions of the processor for specialized application areas are convenient and cost-effective. The RTX 4002 is scheduled to be available in late 1989. Other, commercially available members of the RTX 4000 family, will start becoming available in 1990.

Work continues on the software environment for the RTX family, and now includes a working C compiler for the RTX 2000 (which will be ported to the RTX 4000 when prototype silicon is available). Scheduled software releases available to developers will include a Forth-based microcode-level hardware simulator, a C-based instruction set simulator, and a C-based transportable host support environment.