THE COMMERCIALIZATION OF REGISTER TRANSFER MODULES*

John Grason
Assistant Professor of Electrical Engineering
Carnegie-Mellon University

C. Gordon Bell*
Digital Equipment Corporation

and

John Eggert
Digital Equipment Corporation

Introduction

Unlike the other modules described in this issue, Register Transfer Modules (RTM's)** have been produced and marketed commercially (by Digital Equipment Corporation). This gives RTM's an interest all their own; it is one thing to conceive of a set of register transfer level digital modules, but quite another to market them.

In this article we shall explore this issue of the commercialization of RTM's, rather than their logical and physical structure, which is described elsewhere.[1,2,3]

We shall proceed by tracing the significant events in the history of RTMs. Many issues will emerge, but the "marketing concept" is perhaps the most important. That is: just what is the product? Is it a set of modules? Can the modules be used on the design of systems within DEC? Are they the basis for a service to the customer, in which a system is custom-designed (and installed) for him? Is the product a fixed configuration sold as a subminicomputer? RTM's have in fact appeared in all these forms, as well as others.

As we explore the various marketing concepts for RTM's, we shall focus on how they appeared, from both the producer and the customer points of view.

The History of RTM's

RTM's are Born

In 1968, one of the authors (GB), with the encouragement of the late E.M. Williams, then Head of the Department of Electrical Engineering at Carnegie-Mellon University, perceived the need for a new, higher level approach to teaching digital systems design, to focus on algorithms and the register transfer level of design. The development of integrated circuit technology was making more and more logically complex digital circuits, such as adders, decoders, and registers, available in single, compact, nonsubdividable packages. It was clear that logic designers could soon be using these as their basic building blocks, rather than the traditional gates and flip flops.

*On leave from a position as Professor of Electrical Engineering and Computer Science at Carnegie-Mellon University.

**Register Transfer Module (RTM) is a registered trademark of Digital Equipment Corporation, Maynard, Massachusetts. The product is also known, in a special form, as the PDP-16.

So Professors Bell and Williams applied for and received a National Science Foundation grant† to furnish a digital systems laboratory with the facility to design and build digital systems with components of this scale, referred to as the "register transfer" (RT) level of design. Ideally, the components were to be secured in the form of a well-defined set of digital modules for RT level design. However, the anticipated RT level modules had not yet appeared on the commercial scene. The Macromodules of Clark had already been announced,[4] but working models had not yet been built, and it was not clear that they would lead to a commercial version in the near future. Thus, drawing on a number of sources, including some earlier work with control modules in the PDP-6, a module computer at Lehigh University,[5] the asynchronous modules of Dennis and Patil,[6] and Macromodules, Professor Bell proceeded to design a set of RT level modules, which he called Register Transfer Modules. A working paper on the modules was prepared in late 1969.

† Grant #GY-5160 from June 1968-1970 (and extended to June 1971.)

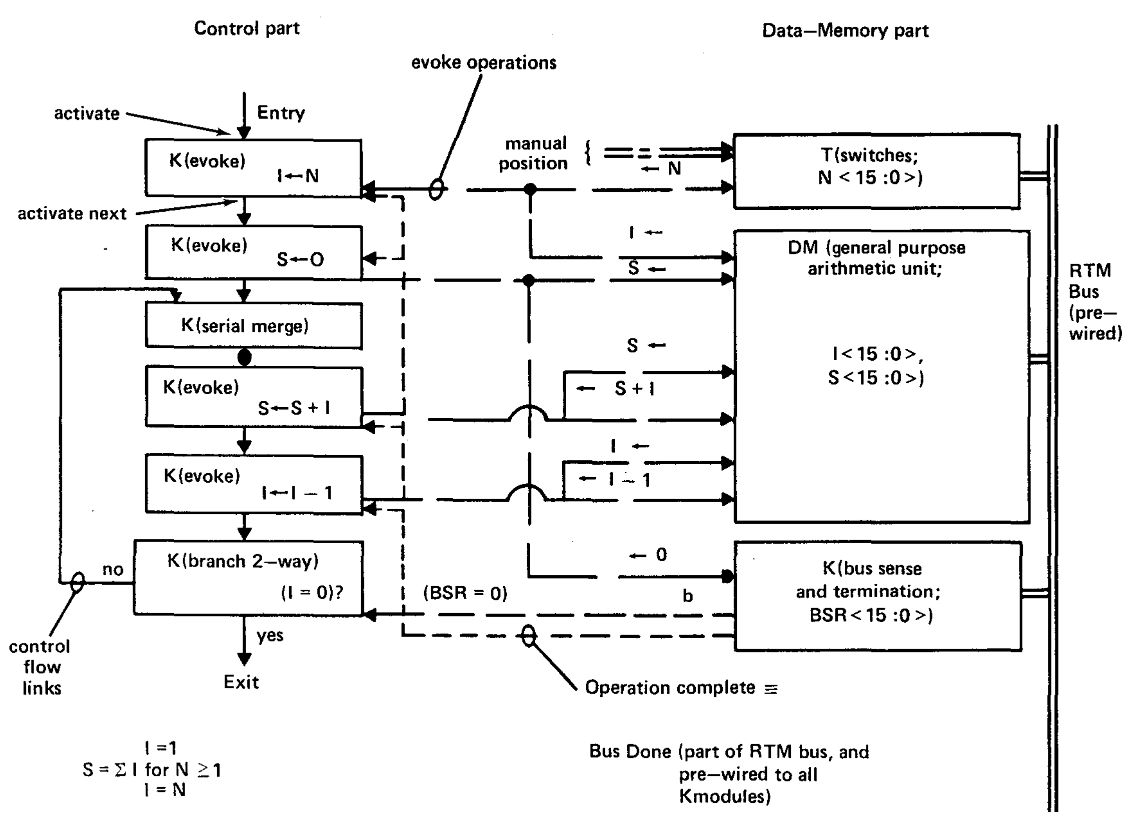
"The designer did not need to know anything about voltage levels, or gates, or flip-flops. All of those details were taken care of inside the modules, and the user just connected them together."

The concept of these modules was quite simple and patterned on principles similar to those of structured programming. Each system built from the modules would consist of two basic parts: a data-memory (DM) part, and a control (K) part. The data-memory part of the system was organized around a bus, the RTM bus. Any number of data-memory modules could be plugged into the RTM bus, which would also have a special bus control module attached. Systems could have more than one bus to achieve a higher degree of parallelism. DM modules contained registers for storing data, and data portions, for operating on the data contained in the registers (e.g., logical and arithmetic operations on binary words). All transfers of data between modules would be by way of the RTM bus. Memory (M) modules could also be plugged into the bus for mass storage of data, and transducer (T) modules could be plugged in for interfacing to the outside world.

The purpose of the control part of the system was to evoke the required sequence of data operations and register transfers in the DM part of the system. The unique feature of this K part was that physically it was isomorphic to the flowchart description of the computational algorithm to be executed. That is, for each individual operation to be evoked in the DM part of the system, there existed a single K module. The K modules for the various operations were wired together, just like the boxes of the algorithm flowchart, to provide the sequence of control for the system. Special modules were provided for branching and merging, and even for hardware "subroutining," just as one would expect to do in flowcharting. A complete system, including wiring, is shown in the figure below for the problem of summing the integers from 1 to N.

Thus, design with the modules was to be just like flowcharting an algorithm.

About this same time DEC was also investigating larger scale integrated circuits for use in modules. They had a traditional line of logic products that they marketed in modular form (typically a printed circuit board's worth of related logic would constitute a "module"), and they



RTM Diagram for Sum of Integers from 1 to N wanted to take advantage of the developing IC technology in more advanced products. Professor Bell was on leave from DEC at that time, and he persuaded them to investigate using RTM's for pedagogical as well as general module use.

RTM Feasibility Study (Strictly Internal Use)

The modules were started at DEC in early 1970. The first attempt did not really get off the ground, and the project was restarted in the late spring of 1970. To provide an impetus DEC decided to use the modules to build a special-purpose instrument computer they had contracted for a customer. Having the hard deadline of a commitment served to accelerate the development of the modules.

The deadline was met, and the customer's system was delivered in late summer, 1970. The RTM's proved to be as easy to work with as had been proposed. Except for a few problems with the logical design of the modules themselves, the design and debugging of the customer's system using the modules went smoothly and rapidly, and a stored program computer was designed and built in about three months – more rapidly than any other computer at DEC.

With this experience, the engineers now better understood the modules, and proceeded on a redesign. A major step that was taken was to change the method of passing control among the modules from pulse mode sequential to fully interlocked fundamental mode (i.e., level sensitive) sequential. This step had the effect of reducing logic complexity, removing some race and hazard problems, and making the modules highly reliable. Also, systems could be debugged without using an oscilloscope.

> "Many customers really didn't know precisely what it was they wanted. Then, if they eventually did settle on a design, they would come back a short time later with modifications."

The PDP-16 Functional Computer (Modified Internal Use)

In late 1970 and early 1971, RTM's were still being groomed for eventual release as modules that would be sold to customers (Carnegie-Mellon was one such customer), and from which the customers would build systems. A tentative target date for the formal announcement was the Fall Joint Computer Conference, 1971.

Since the results of the first custom designed systems proved successful, it was decided to enter the special systems business with RTM's in addition to that of modules. DEC would use RTM's internally to custom-build subminicomputer-sized systems for its customers, with one very important feature. Capitalizing on the ease with which RTM systems could be designed and built, DEC wrote a computer program, CHARTWARE, which accepted a description of the flowchart for the system (stated in terms of the easy-to-understand RTM primitives), and then calculated and printed out the engineering design, wiring, and cost of the system.[7] DEC and the customer would work together to enter his system requirements into the program, and then DEC would use the resulting design to fabricate the system for the customer, using RTM's. The systems produced in this way were to be called PDP-16 Functional Computers. Only finished systems were to be sold to the customer, not individual modules, so he didn't have to know that RTM's even existed. Thus, in addition to providing an entry into the subminicomputer market, this approach allowed DEC further internal experience with the modules before releasing them to the public.

> "Ninety percent of the time a newly built system was working perfectly within an hour or two of being turned on."

For greatest impact, the PDP-16 was announced at the March, 1971 IEEE International Convention. This gave the people who had been grooming RTM's as modules little time to prepare the PDP-16 marketing plan for the Convention. They met the deadline, however, and although little documentation was available for customers or DEC's sales force, the PDP-16 generated much attention, with many quotations made on the convention floor, using Teletypes running the CHARTWARE program.

Many orders for PDP-16 systems did come in, with applications in the area of monitoring and control. However, a pattern soon started to develop that was eventually to make the PDP-16 Functional Computer marketing concept difficult.

Many customers really didn't know precisely what it was they wanted. Then, if they eventually did settle on a design, they would come back a short time later with modifications. This required application engineers to study the customer's needs and decide on the final specifications. As a result, the PDP-16 engineers found themselves solving customer design problems rather than working on the basic product. As a consequence of this, the user documentation was neglected, thereby postponing the introduction of the module set.

Nevertheless, the modules performed admirably once the system specifications were properly known. Design and debugging of the systems were straightforward and rapid. A fairly complex system that required five 19" x 5" mounting panels worth of modules took only three weeks to be designed, built, and debugged. Ninety percent of the time a newly built system was working perfectly within an hour or two of being turned on.

It might be stated in summarizing this phase of the history of RTM's that the PDP-16 Functional Computers did successfully achieve their initial goal of providing DEC with an entry in the special systems marketplace. However, it would seem that the sale of products and engineering service is a tricky thing to attempt in one operation, and should be done only if the buyer and seller have an understanding of the problem solving process.

PDP-16's Also Sold as Modules

The PDP-16 Functional Computer design experiences with RTM's were so favorable that in the fall of 1971 DEC decided to release the modules themselves for sale separately. This was announced at the 1971 Fall Joint Computer Conference. Accompanying this announcement was the *PDP-16 Computer Designer's Handbook*,[8] which had been written in order to partially fill the user documentation vacuum on PDP-16's. This handbook gave the customer the option of buying PDP-16's in any of several modes: he could either let DEC do the complete design and fabrication of the system, or he could design his system using CHARTWARE and let DEC do the fabrication (these two were in essence the PDP-16 Functional Computer approach), or he could buy the modules and associated hardware and do the design and fabrication himself.

> "To design an RTM system, all one has to do is to prepare a flowchart for the desired computational algorithm, and then convert this flowchart *directly* into hardware on a one-for-one basis."

If one looks at the modules that were sold, along with the reactions of engineers and sales people within DEC to the modules, some very interesting observations can be made about register transfer level modules as a product.

To begin with, much to our surprise, there has always been an air of mystery about RTM's at DEC (including its sales force) and to users. How could this be so, when the concept of RTM's is so simple? To design an RTM system, all one has to do is to prepare a flowchart for the desired computational algorithm, and then convert this flowchart *directly* into hardware on a one-for-one basis. Compare this with *ad hoc* switching circuit level design, where one should first prepare a flowchart of the algorithm, convert it into a state diagram, then a state table, select from the myriad of flip-flop types and design styles, and then continue through three or four more steps to convert this into a gate and flip-flop implementation, all the while worrying about races, hazards, voltage levels, etc. Compared to this, how could RTM's be mysterious?

There appear to be two main reasons why this is so. First, many current generation logic designers do *not* prepare behavioral flowcharts when they design. They regard the preparation of flowcharts as programming, and since they are not software engineers, they don't do it – nor in many cases do they know how to do it. Hence the very first (and only, in the case of RTM's) step which generates a formal specification is foreign to them. Second, the thought of working with a set of modules at the register transfer level is also unfamiliar because it is "top-down"-oriented. Since they are accustomed to working with gates and flip-flops and voltages and pulses, the idea of the algorithm with data transfers, control flow, branches, and merges seems unreal and abstract. While the whole concept of RT level modules is to remove switching circuit level details from the consideration of the designer, these designers refuse to work with any system in which they don't see what is going on at the switching circuit and electronic level. Third, engineers are traditionally taught to optimize the incremental cost of the parts. They see RTM's as producing non-optimal designs, because of the logic overhead required for modularity. However, the total cost of a system actually includes the design and incremental manufacturing costs (including parts and testing). For RTM systems the design cost is a factor of 10 lower than conventional design, and the incremental manufacturing cost is about the same (considering testing time).

In any case, it became clear that before RTM's could be marketed effectively, the customers would have to be reeducated in how to do RT level logic design this new, "revolutionary" way. Past experience with the modules had shown that most new users would experience a recognition threshold at some point, where they would say, "Aha! So that's all it is! Why, RTM's are really a simple and elegant idea." So the PDP-16 Product Line people started with a micro-training program to educate their sales force and other groups within DEC that would be potential users of the modules. In addition, the authors (GB and JG with Allen Newell at Carnegie-Mellon University) about this time started work on a book on RT level design, in which they were to use RTM's as illustrative building blocks.

The PDP-16/M (A Prepackaged PDP-16 System)

Then in the winter of 1971-72 another new marketing concept was originated that gave high promise of greater return. One of the problems with the product line until this time had been that of identity. Both the producers and the potential users of the PDP-16 weren't really sure whether it fit at the high end of the custom logic spectrum, or at the low end of the general-purpose minicomputer spectrum. The PDP-16 wasn't a computer, but rather a means of making digital systems, so in fact it could fit both these extremes. Thus, to give the product identity (within a computer company) and to ease the educational problems, a prepackaged PDP-16 system, called the PDP-16/M Sub-minicomputer, was developed.

The PDP-16/M is built around a preselected PDP-16 data-memory part, which consists of a representative, and generally useful, set of PDP-16 modules (RTM's) plugged into an RTM bus. The control part is innovative. Instead of using an individual K module to evoke each control step in the computational algorithm, all control steps are encoded into a specially programmed read-only memory (ROM). Then a small microprogrammed controller fetches these instructions from the ROM and decodes them to evoke operations in the DM part of the system, using the same signal format as the K modules had. A PDP-16 control part implemented with this mechanism, called the Programmed Control Sequencer (PCS), has a lower cost per control step than one built using K modules. Additional DM modules can be plugged into the RTM bus, at the customer's option, to increase the capability of the system.

Thus the PDP-16/M, which was to be sold at a package price below the minicomputer prices, took advantage of the already-developed PDP-16 technology, and at the same time gave the product line a unified identity. Yet, with the programmable ROM and the ability to add optional DM modules to the bus, the 16/M maintained almost all of the versatility of the PDP-16 Functional Computer.

Documentation for the 16/M was prepared, prior to its announcement at the spring 1972 IEEE International Convention. DEC felt that once customers became familiar with the PDP-16 concept through the PDP-16/M, sales of the rest of the PDP-16 product line would be stimulated. As of this writing several hundred PDP-16/M's have been delivered to customers. Because of their style of construction, DEC has found them to be perhaps the easiest to fabricate and the most reliable of the computers they have built.

A Book on RT-Level Design

In the fall of 1972 the book mentioned earlier, by Bell, Grason, and Newell, was published.[3] This was mainly intended to be a first book on the register transfer level design of digital systems. However, for a strong sense of hardware realism, RTM's were used as the fundamental building blocks for most of the example systems. The book was organized around a large number of worked-out designs to be used for self-teaching.

Several observations about the relation of the book to PDP-16's can be made. First, it stimulated a number of inquiries about the product, even though the product was not actively marketed during the first half-year the book was out. However, the same ambivalence exists with respect to the book that existed with respect to the product. Users with programming experience believe it to be worthwhile, easy to read, and *the* way to do logic design. The conventional, older logical designers find it difficult to comprehend, and "too academic." However, virtually all people who read the first three chapters experience the threshold of recognition that we described earlier. One of the authors (JG) has used the book in an introductory logic design course at Carnegie-Mellon University, accompanied by an RTM laboratory. The students have all had basic programming, and experienced no difficulty in mastering the modules.[9]

Conclusions

Technically, Register Transfer Modules achieved their design goals. The problem of educating the user was not recognized initially, but has since been addressed. Several markets and marketing methods have been explored: modules, systems designed to specification, and computers, each of which implies a different marketing strategy.

Acknowledgements

The authors gratefully acknowledge the organization and management contributions of F. Gould and S. Olsen (Digital Equipment Corporation), without whose goal-oriented commitment the RTM's could not have been built.

References

1. Bell, C.G. and J. Grason, "Register Transfer Modules (RTM) and their Design," *Computer Design,* May 1971.

2. Bell, C.G., J.L. Eggert, J. Grason, and P. Williams, "The Description and Use of Register Transfer Modules (RTMs)," *IEEE Trans. Comput.,* Vol. C-21, May 1972, pp. 495-500.

3. Bell, C.G., J. Grason, and A. Newell, *Designing Computers and Digital Systems (using PDP-16 Register Transfer Modules)*, Digital Press, 1972.

4. Clark, W.A., "Macromodular Computer Systems," *AFIPS Conference Proc.,* 1967 SJCC, Vol. 30, Washington, D.C.: Thompson, 1967, pp. 335-336. (Introduction to a set of six papers, pp. 337-400, in the same conference.)

5. Grason, J., and R. Ralsten, "Module control and timing specifications for Digital Data System," internal memo (based on work of Prof. William Hollabaugh), Electrical Engineering Dept., Lehigh University, October 1963.

6. Dennis, J.B. and S.S. Patil, *Computation Structures,* notes for subject 6.232, Massachusetts Institute of Technology, Department of Electrical Engineering, 1970.

7. Eggert, J., "Computerized Computer Design," to be published in *Computer Design,* late 1973.

8. Digital Equipment Corporation, *PDP-16 Computer Designer's Handbook,* Digital Equipment Corporation, Maynard, Mass., 1971.

9. Bell, C.G., J. Grason, and D.P. Siewiorek, "Pedagogical and other Applications of Register Transfer Modules (PDP-16)," First Texas Symposium on Computer Systems, Austin, Texas, June 1972.

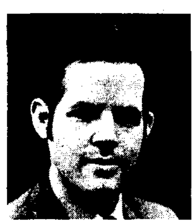
John Grason is an assistant professor of electrical engineering at Carnegie-Mellon University. His research areas include computer-aided design systems, digital system design, and fault diagnosis in digital systems. He has authored or co-authored more than ten technical papers since 1968, and recently co-authored a book with Gordon Bell and Allen Newell on register transfer level digital design, entitled *Designing Computers and Digital Systems (using PDP-16 register transfer modules)*, Digital Press, 1972.

A member of the ACM, IEEE, Design Methods Group, Phi Eta Sigma, Tau Beta Pi, Eta Kappa Nu, Phi Beta Kappa, and Sigma Xi, Dr. Grayson received his Ph.D. in electrical engineering (systems and communication science) from Carnegie-Mellon University in 1970. He did his undergraduate work in electrical engineering at Lehigh University, graduating in 1964, and received an MSEE from Carnegie Tech in 1965.

C. Gordon Bell is currently Vice President for Engineering at Digital Equipment Corporation in Maynard, Massachusetts. He is presently on leave from a position as professor of electrical engineering and computer science at Carnegie-Mellon University.

In 1959 he was with the Speech Communications Laboratory at the M.I.T. Division of Sponsored Research. From 1959 to 1960 he was a research engineer with the Electronic Systems Laboratory at Massachusetts Institute of Technology. From 1960 to 1966, as manager in charge of computer design at DEC, he was responsible for the PDP-4, -5, and -6 computers.

Mr. Bell received the BSEE in 1956 and the MSEE in 1957, both from MIT. He is a member of the Association for Computing Machinery and Eta Kappa Nu, and is a Senior Member of the IEEE. He has co-authored two books: *Computer Structures: Readings and Examples,* McGraw Hill, 1971, and *Designing Computers and Digital Systems (using PDP-16 register transfer modules)*, Digital Press, 1972. His research interests include design of computer systems.



John Eggert is an Engineering Manager at Digital Equipment Corporation. His principal area of responsibility is in the directing of all engineering activities associated with the PDP16/M computer.

He joined DEC in 1968 as a circuit design engineer. From 1968 to 1970 he designed over thirty products for the K-series module line. From 1970 to 1971 he designed, built, and programmed an automatic computer controlled module tester for the K-series module line. From 1971 to 1972 he designed the RTM control elements, developed the CHARTWARE automatic design program, and supervised the PDP16 engineering activities. Mr. Eggert is a member of Eta Kappa Nu and IEEE. His research interests include both software and hardware development of digital systems.

Mr. Eggert received the BEE in 1966 and the MEEE in 1968, both from Rensselaer Polytechnic Institute.