

A COMPARATIVE HARDWARE-SOFTWARE DESIGN STUDY  
USING DEC\* REGISTER TRANSFER MODULES (RTM\*\*)

C. G. Bell  
Professor of Computer Science and Electrical Engineering  
Carnegie-Mellon University  
Pittsburgh, Pa.

J. Grason  
Assistant Professor of Electrical Engineering  
Carnegie-Mellon University  
Pittsburgh, Pa.

Summary

In this paper the subject of hardwired versus programmed implementations of algorithms is discussed. A set of register transfer level modules for digital system design is described. These modules are used to construct two versions of a controller for an industrial anvil; one a hardwired implementation of the control algorithm, and the other a special processor (an interpreter) for executing the control algorithm as a program. The implementations are compared and discussed.

Using RTMs for Comparative Studies

In designing small, special purpose, digital systems, such as process controllers or computer I/O controllers, one is usually faced with two alternatives:

1. Custom-design a system in which the operating algorithm is hardwired.
2. Program a small computer to execute the operating algorithm.

In the hardwired approach, the hope is that custom-designing the system will make it cheaper by avoiding unnecessary parts, but the custom-design process in itself is usually quite expensive. In the programmed approach, one must use already designed, general purpose, small computers. These are mass produced to make them cheaper, but since they are general purpose they will usually contain some features that are not needed for the task at hand. Hence, unnecessary costs will be incurred.

Thus, when one compares the cost and performance of these two systems, one is comparing a custom-designed, special purpose system with a suitably adapted general purpose system. What we would really like to compare, however, is the cost and performance of the hardwired approach versus the programmed approach. This is quite a different issue. The fact that the special purpose and the general purpose systems were probably designed using different technologies, and the fact that the general purpose system probably contains unnecessary features make the comparison very difficult.

\* Digital Equipment Corporation, Maynard, Mass.

\*\* Registered trademark, DEC.

Register Transfer Modules (RTMs), also called the PDP-16, conceived by the authors and DEC, provide a methodology for doing such a comparison of the hardwired and programmed approaches. These modules allow one to quickly and easily design systems following both approaches, yet using the same set of hardware building blocks. Furthermore, since the processor for the programmed approach can be itself custom-designed for the task at hand, the unnecessary features that caused the unfair comparison mentioned above will no longer be present.

In this paper RTMs will be used to design both a hardwired version and a programmed version of a controller for a movable industrial anvil. The performance and cost of these two implementations will be compared.

RTM System Design

A detailed treatment of RTMs is given elsewhere<sup>1</sup>. A brief description is given here for the reader who is unfamiliar with them. RTMs are a set of about 20 asynchronous modules for building digital systems at the register transfer level of design. Four general types of modules are used:

1. Memory modules (M-type) for storing data.
2. Data-memory modules (DM-type) for operating on data and storing intermediate results.
3. Control modules (K-type) for determining which register transfer operations are to take place and when.
4. Transducer modules (T-type) for providing interface to the outside world.

System design with RTMs is exceedingly easy. First the desired operating algorithm is drawn in flowchart form. This flowchart maps directly into the control part of the system, with the boxes on the flowchart corresponding to K modules, and the lines on the flowchart corresponding to control wires. The K modules send signals over to the data part of the system to evoke the various register transfer operations.

The data part of the system consists of selected M, DM, and T type modules connected to a

common data bus. The various register transfers evoked by the K modules take place via this bus.

The modules themselves are fabricated using MSI technology, and are compact, reasonably fast, and relatively inexpensive.

#### The Anvil Controller

The device that is to be controlled in this case study is an industrial anvil that travels up and down over a fixed bed by way of vertical screws. There are three modes of operation for the anvil: (1) it can be jogged up; (2) it can be jogged down; (3) it can be sent to a specific height above the fixed bed by inputting the desired height on a switch register, and then pressing a special button.

Various interface signals between the digital controller and the anvil machinery are provided.

#### The Hardwired Controller

Implementing the control algorithm for the anvil is a straightforward procedure using RTMs. Each control step in the algorithm maps directly into a single K-type module, and the modules are wired together with the same interconnections as the flowchart of the algorithm. In addition to the K-type modules, several DM and T-type modules are needed. These include eight Ts for the interface signals, a general purpose arithmetic unit (DMgpa) a field transfer register (DMtr), a constants generator (DMc), and a single word of non-volatile memory (M). The hardwired controller requires 62 K-type modules in all, and its total cost is 212 cost units.

#### The Programmed Controller

The register transfers that RTMs execute greatly resemble machine level computer instructions. Thus, there is a very straightforward way to obtain an instruction set for a processor to execute the control algorithm as a stored program. That is, provide an instruction within the processor for every distinct register transfer that occurs in the hardwired controller. There are 32 distinct register transfers in the hardwired controller, and these occur in 62 instances.

The processor for this instruction set was designed, and it utilizes 46 K modules, in contrast to the 62 needed for the hardwired version. However, the processor requires a few DM and M-type modules, in addition to those that the hardwired version used. These are two temporary storage registers (DMtr) and a 128 word, eight bit/word read only memory (Mrom). The total cost of the processor version is 223 cost units.

#### Comparison and Conclusions

Although the processor version requires fewer K modules than the hardwired version, it requires some extra DM and M modules. In the balance, the processor version costs slightly more.

However, in general, this factor depends almost solely on the number of steps in the control algorithm. As the number of control steps increases, the number of K's eliminated by going to the processor version will increase, and the processor version will become in fact cheaper. Depending on the system, this threshold is reached at a hundred or so control steps.

As for performance, since the same type modules are used for both versions, it is clear that the processor version is slower, by a factor of 5 or so - the time penalty for interpretation.

If a set of more complex instructions had been chosen as a basis, a third type of implementation would have been possible for the controller. This approach essentially first constructs a stored program computer. This approach would result in a greater savings in K modules, but would probably be still slower than the processor version.

#### Reference

1. Bell, C. G. and J. Grason, "The Register Transfer Module Design Concept", Computer Design (May 1971), p. 87-94.