PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 34, NO. 1, PP. 5-11 (1990)

SOME ASPECTS OF MICROPROGRAMMED CONTROL OF PROCESSORS

J. MÜLLER and A. MÖSCHWITZER

Technical University Dresden

Received January, 31, 1989.

Abstract

After a short review of general features of microprogrammed control units some design problems are treated, especially optimization of control store by means of compacting and coding.

Keywords: microprogramming, control store optimization.

Introduction

It is well known that a processor can be divided into two parts (see *Fig. 1*). There is at first a so called data path with a data input and a data output. The tasks of such a data path are any transformations of the data input signals into the data output signals. Secondly there is a so-called control path. The control path will send control signals for each clock period to the date path in order to determine in which way the next data transformation has to be done. On the other hand, the data path also will send signals of internal states to the control path, for example flags for overflow, negative etc. The control path can receive also external signals, for example HOLD, POWER-FAIL etc. The control tasks for a processor, for example the machine instructions are implemented in the control path. One important task for the designer of a digital system is to find an effective, flexible and low-cost realization for such a control path.

One successful way to do this is the design of a microprogrammed control unit. The first idea for this way was published by Wilkes in 1953 (WILKES, 1953). In *Figure 2* the basic structure of such a unit is represented. The basic idea is to place the necessary control signals systematically in a control store. Each column of the control store matches one elementary control signal. Some of such elementary control signals can produce a simple data path action, for example a register-transfer with an ALU-operation. Such actions are named microoperations. For each data path there is a set of possible microoperations.

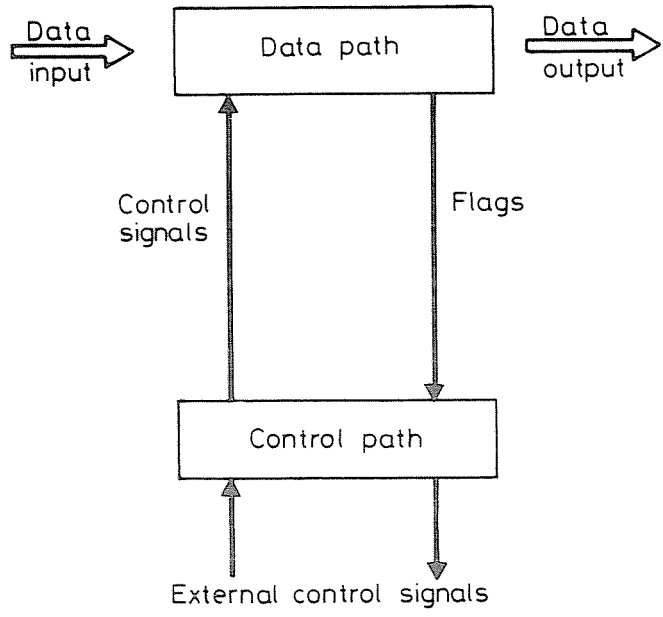

Fig. 1. Parting a digital system in a data path and control path

Each line in the control store represents a microinstruction. Such a microinstruction can have one or more microoperations. If two or more microoperations can be executed in parallel, they can be placed in one microinstruction. A sequence of microinstructions forms a microprogram. For example, for each machine instruction of a microprogrammed processor such a microprogram exists. That means, the execution of a machine instruction is the execution of a sequence of microinstructions.

Furthermore we can find in such a microprogrammed control unit a microinstruction address register, a microinstruction register and a subunit for realization of next-address-logic. Some of the next-address information can be part of the microinstruction. This structure is the simplest way to realize a microprogrammed control unit. The simplest microcycle works in the scheme:

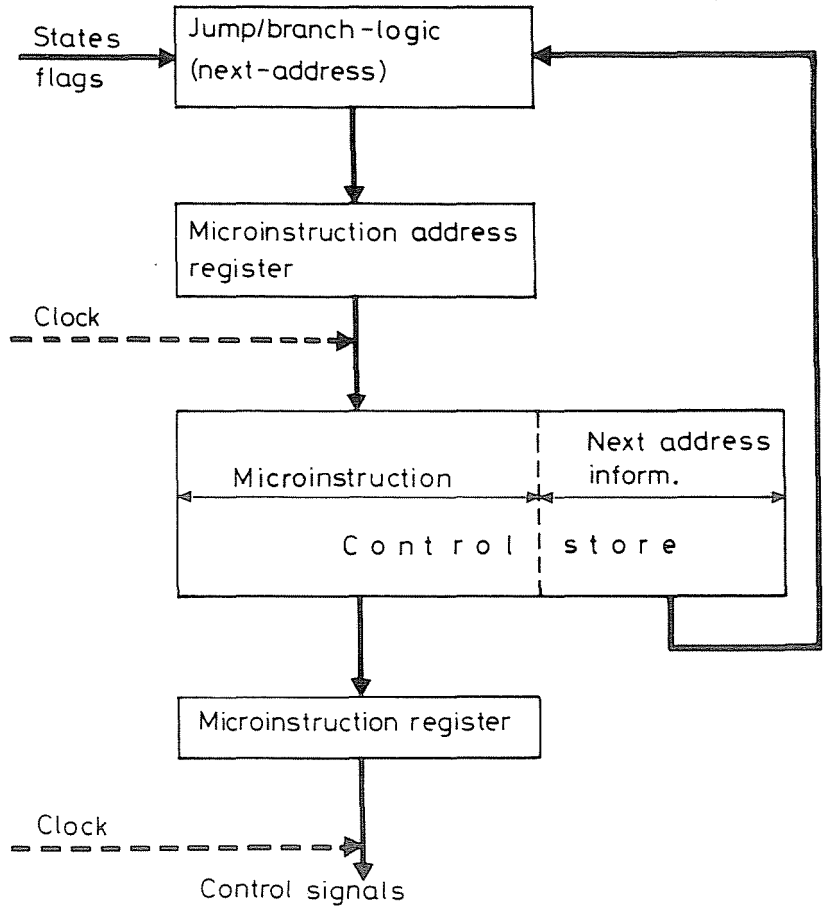

Fig. 2. Structure of a simple microprogrammed control unit

1. generate the next valid microprogram address;
2. read the next microinstruction from the control store into the microinstruction register;
3. execute microinstruction.

Such a control unit has some advantages. The control path is highly flexible, because another control task can be implemented by simply changing the content of the control store. Further advantages are low design costs or low costs in silicon area. Both together are difficult to realize, because a small silicon area (that means primarily a small control store) needs an excellent design with good and expensive design tools. The major problem

of using a microprogrammed control unit is the speed of the unit. In order to get a new microinstruction in the microinstruction register a memory access is needed, and so the memory access time will be a major part of the microcycle time. Another problem is the bit-level design. Bit-level design is very difficult for the designer and needs sophisticated design tools in order to get a useful design.

Some design problems

In *Fig.3* is represented an overview of a possible way to design a microprogrammed control unit.

There are two kinds of information in order to start the design process. First we need a description of the data path of the processor. The best may be a register-transfer description in a hardware description language (HARTENSTEIN, 1987). Secondly we need a functional specification of the processor. With the first information a designer is able to define the set of possible microoperations of the data path, and from the second information we get detailed task specifications of the microprograms. After this the designer can create a primary design of the necessary microprograms. There are many tools for this design step (so microprogramming languages, microprogramming compiler and assembler etc.). Generally such microprograms are sequences of single microoperations. The correctness of these microprograms should be verified by simulation. Now the compacting of the microprograms can be done. In this step the single microoperations will be arranged to microinstructions with several microoperations in order to get a maximum parallelism. By doing so we obtain a reduction of the length of the control store. This reduces the necessary silicon area, the number of microinstructions of a microprogram and the run time (see also *Fig. 4*).

The compacting process needs an analysis of the microprograms in order to detect data dependencies and resource conflicts. Only two microoperations without data dependencies and resource conflicts can be executed in parallel (MÜLLER, 1985).

The remaining steps belong to the bit-level design of microprogrammed control units. That means that the designer works with single bits and no more with mnemonics or similar higher level constructions. An important step is to minimize the width of the microinstructions. Originally there is exactly one elementary control signal mapped to one column. This is the horizontal format. With n bits in a format we can represent 2^n different microinstructions. The major part of this is not suggestive for the data path, another part is not used. And so k bits, with k less than n, are needed in order to represent the necessary microinstructions. For this

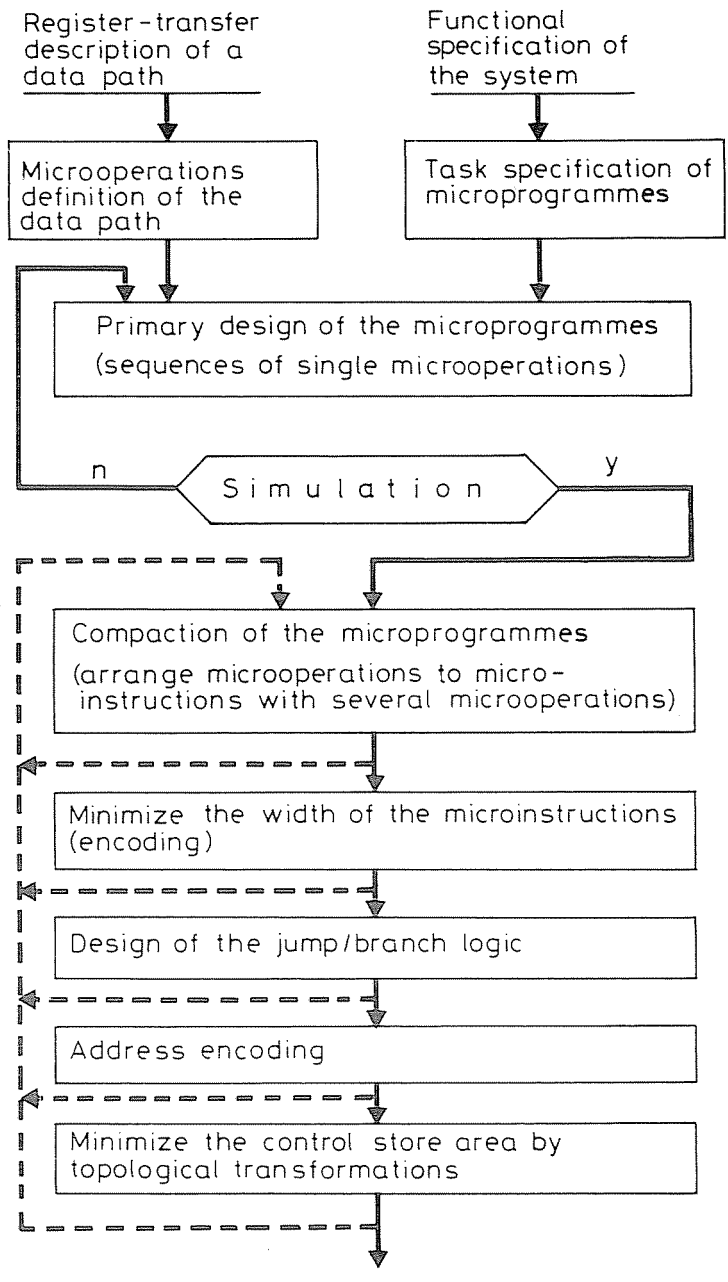

Fig. 5. A possible way to design a microprogrammed control unit

reason we can find a new code for each microinstruction, with a number of bits between k and n. With that we reduce the necessary silicon area. But

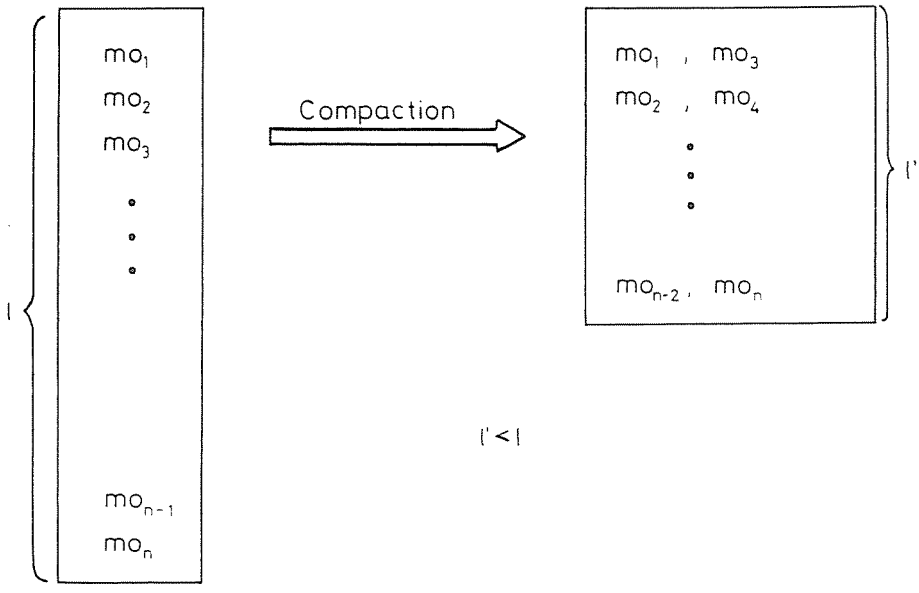

Fig. 4. Principles of microprogram compacting

we must have a decoder in order to generate the original format needed by the data path. The decoder leads to additional silicon area and run time. A maximal encoding with k bits is therefore not practical because of the expensive decoder. In practice we find so called 'quasi-horizontal' formats (see also *Fig. 5*). Here the original format is divided in suggestive groups and these groups will be encoded separately. This results in a good compromise between bit-width-reduction and decoder expense. Other steps in the bit-level design are for example the address encoding and also topological transformations, for example folding.

Conclusions

Especially for so called CISC-processors microprogrammed control units are an important way to realize the very complex control path. There must be an excellent CAD-process for these microprogrammed control units in order to get low cost and practicable designs. Control store optimization is one of the most important steps in this design process.

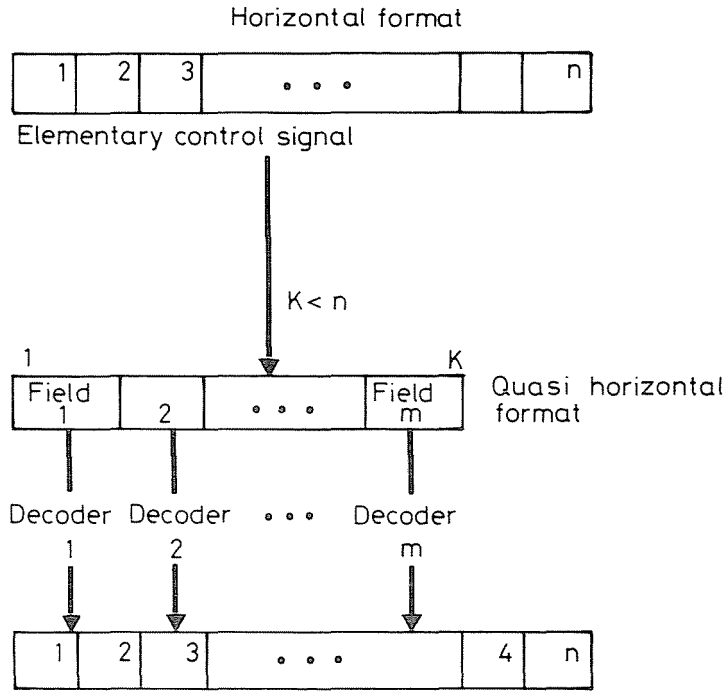

Fig. 5. The quasi-horizontal format of a microinstruction

References

HARTENSTEIN, R.W. (1987): Hardware Description Languages. In: Advances in CAD for VLSI, vol. 7, North-Holland, 1987.

MÜLLER, J. (1985): Ein Beitrag zur rechnergestützten Entwurf von Mikroprogrammsteuerwerken. Dissertation A, TU Dresden, Sektion Informationstechnik, 1985.

WILKES, M.V. — STRINGER, J.B. (1953): Microprogramming and the Design of the Control Circuits in an Electronic Digital Computer. in: Bell, C.G., Newell, A.: Computer Structures: Readings and Examples. McGraw-Hill, 1971.

Address:

Prof. Dr.-Ing. habil A. MÖSCHWITZER,
Dr.-Ing. J. MÜLLER
Technische Universität Dresden
Sektion Informationstechnik
Bereich Bauelemente und Systeme
Mommsenstr. 13
8027 Dresden
DDR