Book Reviews

D. Mange: Microprogrammed Systems. An Introduction to Firmware Theory

Chapman & Hall, London, 1992, pp. x, 393, ISBN 0-412-40800-7; original French language edition: *Systèmes microprogrammés: une introduction au magiciel*, Presses polytechniques et universitaires romandes et CNET-ENST, Lausanne, 1990, ISBN 2-88074-173-4

As is it is usually the case for a manuscript of an author himself being a university professor, this book is basically fulfilling two roles: of a textbook, and of a scientific treatise. As a textbook, it provides subject matter for an introductory course on microprogramming, thus supporting the analysis and design of such systems. As a scientific treatise, it elaborates on the foundations of the same subject, thus providing for a new and fresh view over it. Of course, by doing so the book reveals the author's standpoint which tends *"to achieve a synthesis of two theories that correspond to the two divisions of computer science, that of logic systems (the hardware division) and that of sequential programming (the software division)"* [quoted from the introduction]. This point of view is being exposed in an original way, by using a particular vehicle for algorithm description – binary decision trees (BDTs) and binary description diagrams (BDDs), respectively. Hardware implementations are derived quite directly from such a description, while software ones, i.e. (micro)programs, are obtained by reinterpreting BDTs/BDDs as flowcharts.

In the textbook role, *Microprogrammed Systems* furnishes an introduction to microprogramming which this reviewer would call a gentle one, albeit being highly systematic and with an accentuated sense for accurateness. The subject matter is introduced in a number of refinement steps, basing itself on previously defined notions, and stressing what is the already mentioned conceptual invariant of the text, namely the equivalence of hardwired logic systems and microprogrammed, i.e. software based, ones. The author succeeds to seamlessly incorporate present bit-slice components into his approach, so building a bridge between the basics proper to introductory courses and commercially available building blocks to be used in real-life design. Throughout the book a consistent number of illustrative examples, discussed and commented in detail, corroborate the exposition, with additional exercises being provided for homework practicing. I liked at most the comment paragraphs being added regularly at the end of each section, as well as within it where the need arose to emphasize some new result, provide additional explanation or express the author's opinion.

The book consists of eight chapters, the last one including an appendix, and is supplemented by a thorough bibliography comprising roughly 150 reference items, and a well-compiled index.

In the introductory Chapter 1 the author reviews the usual logic primitives, both combinational and sequential, employed in conventional hardware design, as well as memories, playing a particular role in microprogrammed systems. In doing so, as it is common when introducing the subject matter, he develops a description notation to be applied in later text to formalize the exposition.

The second chapter is fully devoted to combinational systems analysis and synthesis based on a representation using BDTs and BDDs, respectively. Pseudo-boolean (discrete) functions are introduced for compacting multi-output logic functions which are subsequently expressed as BDTs/BDDs to be implemented as hardware or software. Hardware implementations are derived directly, replacing demultiplexers (or multiplexers) for the test elements, while software ones reduce to associated microprograms written in a language consisting of a test and an assignment (micro)instruction and interpreted by an adequate binary decision machine (BDM).

Microinstruction procedures are studied in Chapter 3, with the aim to introduce the basics for structured microprogramming to be exposed at a later stage. The concept is elaborated on examples of decomposable circuits like counters and combinational systems, thus leading to definitions of iterative components. The introduction of procedures implies the extension of the microinstruction set with a procedure call and a return. This of course requires a modified interpreter - the stack-based BDM, to be in turn extended in order to support parameter passing through parameterized procedures.

Chapter 4 approaches the microprogram interpreter itself. In order to minimize it, the address path is assumed to be incremental for the case of normal execution (no branching), so leading to sequential microprograms. Microinstruction definitions are modified accordingly, and an unconditional jump is added to harmonize this new language with the one so far used. The introduction of sequential microprograms allows the adoption of a particular stack-based BDM – the sequencer – otherwise commercially available as an off-the-shelf component, which is then discussed in a greater detail.

In the fifth chapter the author approaches the structuring of microprograms in order to provide a better, i.e. more efficient and easier to use, methodology of writing them. Several methods of microprogram structuring ("transformations") are introduced, as well as three new structured languages – a low-level one to be directly interpreted on sequencer based BDMs (the objective language), a high-level one to support ease of programming (the source language), and a symbolic one to express microprograms in a more compact way. With this formal apparatus microprogram implementations of sequential systems – with asynchronous and with synchronous behaviour, respectively – are addressed in the text to follow. Three methods for designing such systems are then elaborated, leading to microprogram implementations of a different degree of structuring. The chapter ends with the application of this approach to iterative and non-iterative combinational systems design.

The topic covered by Chapter 6 is migration between software and hardware in the framework of microprogrammed systems, meaning the relationship between software-dominated architectures and hardware-dominated ones, and elaborated on the case study of specialized processors. The main issue in such processors is the implementation of specialized complex instructions. The first approach takes the view of decomposing them into a sequence of primitive ones, provided with the interpreter. Thus a process of successive refinement is used to derive the microprogram. The solution thus obtained consists of a minimal processing unit and a control unit with a relatively large microprogram. The second approach concentrates on hardware realizations of complex instructions extending the microinstruction language. This specialized hardware is further factored into its combinational part realizing (specialized) control, and its sequential part realizing action instructions. Such a solution shows control units with minimal programs and relatively complex processing units. The chapter includes the explication of a method for synthesizing specialized processors.

Chapter 7 is devoted to universal processor and respective microprogram design, this being the last stage in the author's classification of architectures so far discussed in the book. This classification is performed according to so-called primitive specifications – internal states or instructions – describing the information processed by hardware during a single clock time. Thus sequential systems, BDMs, specialized processors, and universal processors are mentioned. The universal processor is elaborated through the analysis of its (universal) processing unit, implemented using standard RALUs (slices), and the already described sequencer based control unit. The compiler for the high-level structured language of Chapter 5 is subsequently developed as

well as the respective microprogram to be run on the universal processor, serving as a case study of universal processors usage.

The last chapter contains an appendix comprising several methods for manipulating discrete functions, BDTs and BDDS, and for structuring (micro)programs.

In conclusion, the author's effort in writing the book should be stressed which resulted in a methodical, refined, and thorough exposition of the subject, very well backed by a bibliography which is analytically commented in the text, and by a great deal of well-chosen examples and exercices. In my opinion the book *Microprogrammed Systems* is well-conceived. It is also systematically written, and great care is used in developing the exposition. As a textbook it surely deserves to be recommended. At the same time it provides an original point of view upon the field of microprogramming, helping to sistematize it.

Vlado Glavinić