Chapter 1
Hardwired Interpretation of Control Algorithms

Abstract. The chapter introduces such basic topics, as principles of microprogram control and specification of the control unit behavior using the graph-scheme of algorithm. Next, some methods of control algorithm interpretation, such as finite-state machines (FSM) and microprogram control units (MCU), are discussed. Last part of the chapter is devoted to the organization principles of compositional microprogram control units, which can be viewed as compositions of finite-state machine and microprogram control unit. These control units provide efficient interpretation of the so-called linear GSA, in which long sequences of operator vertices can be found. These sequences are called operational linear chains (OLC). Microinstructions corresponding to the components of OLC are addressed using the principle of natural microinstruction addressing. It permits to use the counter to keep microinstruction addresses and to simplify the combinational part of control unit, as compared with the classical Moore FSM. The Mealy FSM is used in CMCU to address microinstructions. It permits to calculate the transition address during one cycle of control unit’s operation. Due to this feature, performance of the CMCU (proportional to the number of cycles needed to execute the control algorithm) is better than performance of the equivalent MCU with natural microinstruction addressing.

1.1 Principle of Microprogram Control

The principle of microprogram control was proposed by M. Wilkes in 1951 [68,69] and was developed by V. Glushkov [1]. According to this principle, any complex operation executed by a digital system is represented as a sequence of elementary operations of information processing. These elementary operations are named microoperations. An ensemble of microoperations executed during one cycle of a digital system operation is named microinstruction. Special logical conditions (status signals or flags) are used to control the order of execution of microoperations. Their values are calculated as some Boolean functions depending on the values of operands. An algorithm of execution of some operation is represented in terms of microinstructions and logical conditions is named microprogram [22]. A digital
system with microprogram control is represented by an operational unit. The operational unit is the composition of operational automaton (OA), which is a data-path of the system, and control automaton (CA), which coordinates the interplay of all system blocks (Fig. 1.1) [1, 8, 9].

In the operational unit, CA analyses the code of operation together with values of logical conditions from the set \( X = \{x_1, \ldots, x_L\} \). Microinstructions \( Y_q \subseteq Y \) are executed on the base of this analysis, where \( Y = \{y_1, \ldots, y_N\} \) is a set of microoperations, which initialize operand processing and obtaining of intermediate and final results of operations, executed by the data-path. An algorithm of operational unit’s operation is represented using one of the formal methods [8]. In this book we use the language of graph- schemes of algorithm (GSA) which is very popular in design practice [8, 9].

Graph-scheme of algorithm \( \Gamma \) is the directed connected graph, characterized by a finite set of vertices, namely (Fig. 1.2): start (initial) vertex, end (final) vertex, operator and conditional vertices.

The start vertex, denoted here by the symbol \( b_0 \), corresponds to the beginning of control algorithm to be interpreted and has no input. The end vertex, denoted here by the symbol \( b_E \), corresponds to the end of control algorithm and has no output. The operator vertex \( b_t \in B_1 \), where \( B_1 \) is a finite set of operator vertices of GSA \( \Gamma \), contains a collection of microoperations \( Y_q \subseteq Y \) which are executed in parallel. The conditional vertex \( b_t \in B_2 \), where \( B_2 \) is a set of conditional vertices of GSA \( \Gamma \), contains single element \( x_l \in X \). It has two outputs, first corresponding to value "1" and second to value "0" of the logical condition to be checked. Thus, GSA \( \Gamma \) is characterized by a finite set of vertices \( B = B_1 \cup B_2 \cup \{b_0, b_E\} \). The vertices \( b_t \in B \) are connected by arcs from a finite set \( E = \{\langle b_t, b_q \rangle\} \), where \( b_t, b_q \in B \).