A sequential circuit has a memorising ability to a certain degree. One distinguishes between two types of sequential circuits: asynchronous and synchronous. An asynchronous (sequential) circuit employs input signals that are continuously present. A synchronous (sequential) circuit accepts an input signal only during a usually brief time interval when a (periodic) clock signal is 1.

The theory of sequential circuits, as developed by Huffman (1954), is the theory of asynchronous circuits, and these, and only these, are the circuits we are concerned with. You can find excellent coverages of the theory in the literature, among the best being Krieger (1969), and Dietmeyer (1971). Despite its remarkable achievements, Huffman’s theory and its subsequent development (e.g., by Moore, Mealy, Unger, Tracey, to mention only a few of the pioneers), had and still has serious pitfalls.

The theory of asynchronous circuits presented in this division breaks with standard theory, the theory by Huffman and his disciples, except for the use of Huffman’s flow table. The backbone to describing, calculating, and verifying a sequential circuit is the so-called word-recognition tree discussed in great detail with the help of many examples in Chap. 13; it was conceived by Vingron (1983). Using the word-recognition tree to calculate circuits depends completely on the theory of latches put forth in Division Two, in particular in Chap. 10. Huffman’s flow table remains an indispensable tool. In Chap. 14 it is explained as a program governing the behaviour of a so-called sequential automaton. How to write such programs is discussed extensively. The prime and unsolved problem in Huffman’s theory is finding a so-called state assignment for the flow table. This is a binary encoding of the rows of the flow table enabling us to determine latches which, working in unison, realise the circuit’s memory. Chapter 15 presents an algorithmic solution to the state assignment problem—the algorithm being called iterative catenation. Chapter 17 contains a novel, algorithmic procedure for transforming a flow table into one with the least number of rows thereby making it possible to find the least number of latches (and the latches themselves) needed to realise the circuit—this process is called merging the flow table. The final chapter is concerned with the problem of verifying the input–output behaviour of sequential circuits, a
problem generally taken to be unsolved, and by some, considered to be (at least economically) unsolvable. As Chap. 18 shows, the problem is not only solvable, the results obtained are in fact very economically applicable.