Calculating Digital Counters *

Walter Dosch

Institut für Informatik, Universität Augsburg, D-86135 Augsburg

Abstract. Deductive design characterizes a method where a system description is deduced from the functional specification of its behaviour applying formal transformations rules. Following this design methodology, we derive circuit descriptions for various combinational and sequential counters from a common functional specification using equational and inductive reasoning.

Keywords digital counters, deductive design, formal hardware description

1 Introduction

The deductive approach to system design summarizes a methodology where an algorithmic system description is systematically derived from a behavioural specification following formal transformation rules ([1]). Deductive design supports the "correct by construction" paradigm — in contrast to an "a posteriori" verification ([5]).

This paper presents algebraic methods for the deductive design of synchronous digital counters. We formally derive two combinational circuits — a parallel and a carry look-ahead successor function — and two sequential circuits — a parallel and a serial counter — from a common functional specification. Within a structured design methodology for modern VLSI technology ([14]) we abstract from layout and timing issues and concentrate on algorithmic design principles. The high-level derivations explicate the different realizations in terms of the design decisions leading to them. In particular, we disentangle the design decisions concerning

- the representation of natural numbers by digit sequences,
- the use of bounded and of fixed word length,
- the effects of binary coding,
- the tradeoff between space and time.

Important transformation steps consist in

- changing the representation from numbers to sequences,
- introducing suitable induction principles for recursive solutions,
- systematically reusing results of subcomputations,
- localizing data dependencies for parallel circuits.

* This research was partially supported by the ESPRIT working group 8533 NADA on New Hardware Design Methods and the DAAD project Deductive Design of Parallel Software and Hardware Systems.
This paper concentrates on deriving a representative family of digital counter circuits from a common specification by following different development lines. The single circuits cannot immediately be related on the implementation level although they all meet the same initial specification.

The derivations are based on simple calculational theories using equational and inductive reasoning combining conciseness and precision. They widely exploit the algebraic properties of sequences modelling the temporal or spatial succession of elements.

The digital circuits are uniformly described in a functional style both on the specification and the implementation level ([3]); this offers a coherent framework for algebraic reasoning. On the specification level, the functional style abstracts from the system's representation to the input/output function. On the implementation level, functional descriptions provide an algorithmic and a structural system model, compare [11] for the relational approach. The functional (de)composition also supports structured and hierarchical descriptions.

The paper is organized as follows: In Section 2 we introduce the data structures, viz. natural numbers and digit sequences, together with their algebraic properties. In Section 3 we specify the successor function as an operation induced by changing the representation from natural numbers to digit sequences. A first algorithmic solution is obtained by eliminating the abstraction and the representation functions. In Section 4 we study the deductive design of two combinational circuits, viz. a parallel and a carry look-ahead successor function. The recursion of the parallel successor function determines a regularly structured topology where the half-adder components are cascaded into the breadth of the network. For every input sequence of fixed length, we obtain a non-recursive network by successively unwinding the recursion. Isolating the carry computation and decomposing the recursion then leads to the carry look-ahead successor function. In Section 5 we formally derive two sequential circuits, viz. a parallel and a serial counter. First we transform the successor function into repetitive form. This allows to keep the topology static by reusing the elementary components. Then we fuse the input and the result parameters into a sequence of constant length. The combined input/result sequence models a register determining the state of the sequential circuit. Depending on the way of fusing, we end up with a parallel counter where the state is an array, or with a serial counter where a shift register determines the state. In Section 6 we outline the redesign of counter circuits by reusing the derivation.

The reader should be familiar with the standard notions of program transformations ([17]) and of digital circuits (for textbooks see, for example, [9, 13, 20]).

2 Data Structures and Representation

Digital counters work on representations of natural numbers as sequences of digits. Thus we first introduce the corresponding data structures together with their algebraic properties.