DIGITEST: A Structural Language Based on Algebraical Models of the Logic Topology and the Time Behaviour of Digital Circuits.

Franz J. Rammig

Universität Dortmund

Abstract

The structural language DIGITEST will be described, which, besides other characteristics has the following features:

1) Detailed description of digital hardware on the IC-Level, based on an algebraical model called Quasi Real Boolean Function (QRBF). This description includes:

- variable delay, depending on the current values at the IC-inputs and the IC-outputs
- specefic high frequency absorption for different IC-inputs
- acceptance of transitions as arguments.

2) Recursive block-structure, i. e. structures described in DIGITEST can be used as primitives for structures to be described in DIGITEST.

3) Use of an arbitrary algorithmic language to describe the functional behaviour of the primitives.

4) Description of the logic topology of digital circuits directly based on a normal from of an algebraic model, called Formula System.

5) Possibility to translate the language using a simple precedence grammar.

Following a short introduction into the algebraic models used we will demonstrate some features of the language by an example. Finally we will show the use of DIGITEST as circuit description section of the input language of a simulator.

Introduction

While there are known many "Hardware Description Languages" on the register transfer level, for example CDL, DDL, APL, RTS, LOTIS, CASSANDRE, there are few languages mentioned to describe the lowest logical level.
(We hesitate as it is dangerous to use the terms "gate-level" and "register transfer level". Particulary the term "register transfer level" is very dangerous. It is an adequate level of abstraction for current technology, but it is in fact just a special case of a level that can be classified with terms like "relative control structure", "speed independence", "Petri Net representation of control structures", "explicit (visualized) control structure", while its counterpart, namely the lowest logical level, can be characterized with terms like "inherent (invisible) control structure", "absolute, time dependent control structure". Hence we believe that there is no need for a distinction among "Hardware Description Languages" and "Software Description Languages", but a distinction should be made among the levels of relative and absolute control structures, because there are a number of different problems involved in these levels).

Under "lowest logical level" we understand the following:

1) Design has come to a point where the designed data structures and particulary the control structures have to be implemented, i. e. have to be mapped into the interconnection of given modules with fixed logic and time behaviour.

2) The unit of interconnection is the bit.

3) The information about the time behaviour of the modules is available, either as a property of the used modules or as criterion for the selection of modules, for example to implement control structures.

Obviously a structural language is adequate for the description of such a level.

Description of the logical topology of digital circuits

Given an arbitrary set of modules, i. e. a set of transformations $t: \{0,1\}^n \rightarrow \{0,1\}$, boolean formulas are an adequate tool to describe tree-structured interconnections of these modules. We are speaking at this point only about the interconnection structure. Therefore the type of transformation, wether combinational or sequential is definitely of no interest.
To avoid the two strong restriction "tree-structure" and "one bit transformation result" of the basic modules, formula systems are introduced. We follow here the formalization as defined by Vogel (5).
The fundalental idea is to divide a digital circuit into parts which are describable by boolean formulas, i. e. parts without internal fan-out, feed-back and multiple bit result. These formulas use as variables either circuit-inputs or additional circuit-internal variables. The interconnection of these parts is expressed by a mapping while another mapping is used to show how single bit results are combined to a multiple bit output of a module.
Hence a formula system is quadrupel (F,Γ,Ψ,σ), where F is a finite set of formulas, Γ is a finite set of circuit-internal variables, $\Psi: \Gamma \rightarrow F$ is a totally defined function, showing for every internal variable the defining formula and $\sigma: F \rightarrow \overset{\circ}{\cup}(F)$ is a totally defined function that defines a partition of F. Every variables used within a formula system and not belonging to the set Γ is a circuit-input.
If there are as many internal variables as possible then we have the so called M-normal form of a formula system. That is exactly the model used in DIGITEST.

Description of the time behaviour of digital/modules

To describe the time behaviour of modules, a module should be thought of as a transformation of a set of signals into a set of signals, rather than as a transformation of a set of binary variables into a set of binary variables.

A digital signal can be described as $[0,1]^{\mathbb{R}}$ where $[0,1]$ is a closed interval from $\mathbb{R}$, the set of real numbers. Let $S_{\mathbb{R}}$ denote $[0,1]^{\mathbb{R}}$ and F denote $\{f \mid f: (S_{\mathbb{R}})^n \rightarrow (S_{\mathbb{R}})^m;\ n,m \in \mathbb{N}\}$ then we obtain a mathematical model $\Sigma := \langle S_{\mathbb{R}}; F \rangle$ called "Realistic Switching Model". Such a model can be described approximately by expanding the domain of an ordnary binary transformation to the k-th Cartesian product of the domain where one has the imagination of time cuts with fixed distance.

Hence a function $f: \{\{0,1^n\}^K\} \rightarrow \{0,1\}^m$ is obtained. If k is equal to the largest possible delay of the module to be described, the domain is large enough to describe delay, high frequency absorption and transition sensibility. The model is based on the assumption that a module is always computing and therefore there is a delay of every value instead of a delay of transitions. It is assumed that the delay of a module is variable, depending on the current values at the module inputs as well as the module outputs. In addition to above input and output dependent delay (IOD) the special cases of delay dependent on the inputs only (ID) and delay dependent on the output only (OD) are used. Transition sensibility and high frequency absorption (inertia) are treated as properties of the individual module inputs.

Let S_T be a restriction of S_{IR} where IR>T is on enumerable set of equidistant time points and $F' := \{f' \mid f' : (S_T)^n \rightarrow (S_T)^m;\ n,m \in IN\}$.

$f' \in F'$ is called "Quasi Real Boolean Function" (QRBF) iff it can be defined in an unique way by a function
$g: \{\{0,1^n\}^K\} \rightarrow \{0,1\}^m$ as described above.

Let Q be the set of the Quasi Real Boolean Functiones. Hence the final model $K := \langle S_T;\ Q \rangle$ is obtained.

A publication with a detailed definition in in preparation. It should be mentioned that the restriction to the domain {0,1} can be avoided ba a combination of interval-arithmetic and multivalued logic.

The language DIGITEST

The structural language DIGITEST has been defined as a formal language to describe digital circuits based on the M-normal form of the formula systems and Quasi Real Boolean Functions.
A circuit description, called "description section" (DS), consits of a number of DIGITEST description statements (DDS) in an arbitrary sequence. Every DDS consists of a formula system part and a QRBF-part. The formula system section, describes time independently the rule of the module described within the logical topology. It supplies information about the input-pinset, the output-pin set and the transformation used.

In DIGITEST this has the form:

<inputpinset> = <type of transformation> (<outputpinset>)

Example:

Q,NQ=DFF (DE,CLOCK,CLEAR,PRESET)

The interconnection of the modules is expressed by using indentical namens within the input-pin set of the "receiving" module and the output-pin set of the not necessary different "transmitting" module. According to the M-normal form of Formula Systems only output-pins get own names. Circuit inputs are treated as outputs of the module "environment".

In DIGITEST the description of transition sensibility is treated as part of the transformation algorithm. Hence the QRBF-part of a DDS consists of constructs for the description of delay, high frequency absorption and as an additional feature for the description of fan-in-fan-out restrictions.

An example of a DIGITEST description statement will now be used to show some special features of DIGITEST:

```
FF1:Q,NQ=DFF (P1-P4),FAN (Q,NQ:10/P1:1/P2-P4:2),
     DELAY (P1→'Q':UP10-12,DOWN11-13/P2,P3→'Q':8-10,9-11),
     MINWIDTH (P1:10/P2,P3:7);
```

a) As addition to the first example this statement has a label. This allows the use of mnemonic names at various places within the circuit description. The names "Q" and "NQ" can be used to designate pins as outputs of every flipflop occurence within the described circuit. To indentify the unique output pin Q of the flipflop occurence described in the DDS labeled with FF1, the double-name FF1.Q may be used.

b) P1-P4 is a so-called "pin sequence". It is equivalent to P1,P2,P3,P4.

c) The section "fanblock", entered with the word FAN, describes that the outputs may be connected with maximal 10 unit-inputs.

d) The delay block, entered with the word DELAY, is a so called "represented delay block". The output pin "Q" serves as representative of the output pin set (Q,NQ).There is different delay according to transitions at the inputs P1, P2, or P3. By default the influence of the input P4 is treated to be identical to the influence of the first occurring input pin within the delay block (i.e. P1).

e) At the input P1, pulses must have a duration of at least 10 time units, at the inputs P2, P3 of at least 7 time units.

The example shows a DDS of a medium degree of accurary. Delay, as well as the other parameters, can be described more or less accurately.
Writing parameter descriptions can be minimized by using default conventions, pointers to already used descriptions and a date base.
Special constructs exist for OD, ID and IOD, each allowing the use of intervals. (In the above example a special case of IOD with intervals has been used).
An arbitrary number ofDDS's can be combined as a description section (DS). This DS must be labeled. Its label can be referenced by the type of transformation of a DDS. In this way the structure description can be levelled up and a block-structure is obtained.
There are a set of transformation primitives in a data base and the user can add his own transformation primitives by describing them using an algorithmic language. This description can be placed within a DS in an arbitrary manner.

Use of the language DIGITEST

Using DIGITEST including the bisic models in the field of bit-time simulations is obvious. There are many bit-time simulators (1), some of them offering a more or less exact simulation of time behaviour (2), (4). References to basic algebraic models as well as input-languages from the user's point of view are made rarely. To demonstrate the above basic ideas the simple compiled-mode simulator DIGITEST (version 1) has been implemented. More information about this simulator can be obtained from the DIGITEST (version 1)-programmer's manual (3). It should be noted that this simulator was built for demonstration purpose. In particular, it is too slow for practical use.

At the University of Dortmund we are now implementing the table-driven simulator DIGITEST (version 2). It offers the following features:
"selective trace", "critical event simulation", parallel fault simulation" maximal 64 000 output-pins,DIGITEST as description -part of the input language, QRBF as basic model, "mixed level simulation", interactive use.
At the same time we are implementing a simple precedence compiler for the DIGITEST language.

References

1) M. A. Breuer (Ed.): Design Automation of Digital Systems 1972

2) S. G. Chappel, S. S. Yau: Simulation of large asynchronous logic circuits using an ambigous gate model. FJCC 1971 651-661

3) F. J. Rammig: DIGITEST (release 1) Programmierhinweise 1973, unpublished

4) S. A. Szygenda, D.M. Rouse: A model and implementation of a universal time delay simulator for large digital nets. SJCC 1970 pp207-216

5) A. Vogel: Fehlerdiagnose in kombinatorischen Schaltungen. 1973, unpublished

6) H. Weber: Ein Programmsystem zur Unterstützung der Rechnerentwicklung. ACM German Chapter : Lectures III 1973