Verification of the Generic Architecture of a Memory Circuit Using Parametric Timed Automata*

Remy Chevallier[1], Emmanuelle Encrenaz-Tiphène[2], Laurent Fribourg[2], and Weiwen Xu[2]

[1] STMicroelectronics, FTM, Central R&D, Crolles, France
[2] LSV - CNRS, ENS de Cachan, France

Abstract. Using a variant of Clariso-Cortadella's parametric method for verifying asynchronous circuits, we formally derive a set of linear constraints that ensure the correctness of some crucial timing behaviours of the architecture of SPSMALL memory. This allows us to check two different implementations of this architecture.

1 Introduction

In [10,9], Clariso and Cortadella propose a technique for verifying the timings of asynchronous circuits. The approach infers a set of sufficient linear constraints relating the delays of the internal gates of the circuit to the external delays of the circuit specification that guarantee the correct behavior of the circuit. The method is based on the reachability analysis of a timed model of the circuit (with additional abstract interpretation techniques [11]). As pointed out in [10], such parametric constraint sets are very informative for the designer, as they identify sensitive parts of the circuits (e.g., "critical paths") and interrelations between various data of the specification. Moreover, many technology mappings can be tested immediately (by mere instantiation of the parameters).

We follow here a similar approach for formally verifying some generic properties of a commercial memory designed by STMicroelectronics, called SPSMALL. Such a memory can either read or write a data (depending on the value of an input signal WEN). For the sake of brievity, we focus here on the write operation ($WEN = 0$). In this case, the memory stores the value of the input signal D into an internal memory point (located at address A), and propagates it to the output port Q. The circuit is made of a dozen of elementary components. Each component c_i is associated with an interval $[l_i^{\uparrow}, u_i^{\uparrow}]$ (resp. $[l_i^{\downarrow}, u_i^{\downarrow}]$), which gives lower and upper bounds of the component traversal delay when the input is rising (resp. falling)[1]. Such a circuit is specified by the manufacturer according to several "external" parameters (such as periods of a cyclic clock CK, time of stabilization of signal D, ...). Our timing analysis method derives a set of sufficient

* Partially supported by project MEDEA+ Blueberries.

[1] This is a straightforward generalization of "bi-bounded delay" model (see [6]), taking into account the rising or falling nature of input signal.

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 113–127, 2006.

linear constraints relating the external parameters to the internal gate delays that guarantee the correctness of the circuit's behavior. In particular, these constraints can be seen as sufficient conditions for certain paths of the circuit to be "critical" (i.e. those along which the propagation delay is the longest).

Using the model of parametric timed automata (see [3]) and tool HYTECH [14] for reachability analysis, we are able to generate a set of linear constraints that ensures that the correctness of some crucial timing behaviors of the memory: e.g., the result of a write or read command, transmits the value of input signal D to output port Q within one clock cycle. This method is applicable to several instances of SPSMALL memories implemented with different transistor technologies (corresponding to different sets of parameter values).

Comparison with Related Work. As pointed out above, our work is adapted from Clariso-Cortadella's method [10,9]. However, [10,9] focus on a particular form of linear constraints (linear inequalities with coefficients always equal to ± 1) and represent them as a particular form of convex polyhedra, called "octahedra" [9]. In contrast, we use here linear constraints and their classical form of convex polyhedra in their full generality [13]. Other differences with [10,9] are:

- a different level of modelling: the components of the memory are represented here at the "latch" level instead of gate level. At this level of representation, the flow of input signals traverses the circuit in a linear manner (without loops), while the flow is cyclic at the gate level (the ouput of one gate can be an input of another gate and the converse can be simultaneously true),
- the use of mere forward reachability analysis rather than techniques of fixpoint computation of abstract interpretation (using, e.g., widening operators),
- the use of a decompositional approach, which splits the global system into three smaller parts,
- the use of a *step-by-step refinement* of the reachability analysis process: we start with the most general form of constraints on parameters; we then refine them progressively, via *iterative* reachability analysis, detecting, at each run, some erroneous generated states until complete elimination.

Besides [10,9], our work is along the lines of [16,5,17] where timed automata have been used extensively to model and check timing properties of asynchronous circuits (cf. [12]). Reachability analysis is there performed via tool KRONOS [18]. Our work is also a continuation of [4,7] where the SPSMALL memory is modelled as a timed automaton and some of its timing properties are proven by reachability analysis (using tool UPPAAL [15]). The crucial difference here, with respect to these previous works, is that we use the model of *parametric* timed automata (performing reachability analysis with HYTECH [14]).

Plan of the paper. In Sect. 2, we present the general objectives of our verification process. In Sect. 3, we give a general description of our method. In Sect. 4, we explain how we apply it on SPSMALL memory after having split the model into 3 parts. Final remarks are given in Sect. 5.