Latency Analysis for Sequential Circuits

Alexander Finder André Sülflow Görschwin Fey
University of Bremen,
28359 Bremen, Germany
{final, suelflow, fey}@informatik.uni-bremen.de

***Abstract*—Verification is a major bottleneck in today's circuit and system design. This includes the tasks of error detection, error localization, and error correction in an implemented design as well as the analysis and avoidance of transient faults. For all those tasks, knowing for how long values of signals influence the system is important.**

In this paper, we propose a minimal and maximal latency measure for sequential circuits. This measure explains how long a circuit's state and outputs depend on input stimuli. Exact and heuristic algorithms are proposed to determine the measure. Experiments show that the measure provides insight into the behavior of circuit designs.

I. Introduction

Verification has become a major bottleneck in circuit and system design. Up to 80% of the overall design cost are due to verification. While there exists a lot of work for formal verification, there are fewer approaches dedicated to debugging (error localization and error correction) [1], [2], [3] and understanding a counterexample often requires as much as effort as coding a design. For debugging such erroneous behavior, it is useful to know for how many time cycles a sequential circuit has to be considered at least and at the maximum to locate the source of erroneous output. By this, localization and correction of bugs are supported.

Another upcoming application area of verification tools is the analysis of transient faults [4], [5], [6], [7], [8]. Similarly, for analyzing transient faults, it is of interest to know for how many time cycles the fault may affect the output of the circuit at the maximum. This is equal to the number of future time steps a circuit's behavior depends on.

In this paper we formally define the new terms *minimal latency* and *maximal latency* of a sequential synchronous circuit represented by a *Finite State Machine* (FSM). Our approach investigates when an input assignment to a sequential circuit is observed at the outputs and for how long the input assignment influences the internal state. Three approaches to compute the latency for synchronous sequential circuits are proposed: (1) an exact approach based on *Sequential Equivalence Checking* (SEC), (2) an under-approximation based on 2-valued simulation, and (3) an over-approximation based on 3-valued analysis. By this, if only the two approximate approaches are applied, the difference between the results indicates how accurate the results are. In many cases the maximal latency of a circuit may be infinite because the design under analysis contains loops. However, this is also decidable by the SEC-approach. While we define latency with respect to primary inputs, the technique can also be applied selectively to parts of a circuit, like e.g., memory elements or certain signals. This further supports a designer in understanding observed behavior.

The remaining part of this paper is organized as follows: In Section II we review related work. FSMs and sequential circuits are introduced in Section III. Section IV defines the term *latency* and explains the principle of the proposed latency analysis. Experimental results on ITC'99 benchmarks are presented in Section V. The paper is concluded in Section VI.

This work was supported in part by the European Union (Project DIAMOND, FP7-2009-IST-4-248613).

II. Related Work

In the context of *Bounded Model Checking* (BMC) already several measures have been proposed on synchronous sequential circuits represented by FSMs. Significant effort has been spent to calculate the *diameter* [9], and the *sequential depth* of circuits [10], [11], [12]. In [13], [14] further the *recurrence diameter* is introduced.

The diameter d of circuits, i.e., the maximum length of longest-shortest paths where no states are repeated, is required to decide whether a proof in BMC is complete. In [9] the analysis for the diameter is re-formulated as a *Quantified Boolean Formula* (QBF) and reduced to a *Boolean Satisfiability Problem* (SAT Problem).

The sequential depth s, which is the longest loop-free path is an over-approximation $s \geq d$ of the diameter, since every shortest path is a loop-free path. In [10] the sequential depth of FSMs has been automatically determined by a simulation-based approach using evolutionary algorithms while in [12] the maximal length of potential counterexamples is determined which depends on the sequential depth of the FSMs. The authors further establish tighter bounds for the maximal length of counterexamples by defining a recurrence diameter r which differs from the sequential depth by considering *initialized* loop-free paths in an FSM. Analogously to the sequential depth it over-approximates the diameter $r \geq d$. Kroening and Strichman discuss various techniques to compute the recurrence diameter efficiently in [14].

In [11] the robustness of BMC is increased by computing lower bounds on the reachable states of the circuit. The lower bounds are determined by BDD-based reachability analysis. The bounds are then used to shorten the counterexample.

However, these approaches are different from our approach in so far as the sequential depth and the recurrence diameter are considered. In this paper we define and analyze the latency of signals with respect to all (reachable) states as defined in Section IV.

Prabhakar and Hsiao presented in [15] a debugging algorithm using a logic implication-based learning approach [16] to intelligently select trace signals. This is important because an effective silicon debug technique uses a trace buffer to monitor and capture a portion of the behavior during its post-silicon operation. Due to the limited space, the selection of critical trace signals plays an important role. By this, signals are prefered which cover implications that are not implied by other signals. This method is explicitly used for post-silicon debug and concentrates on single signal traces. Our approach can be used as a supplement to the debugging and post-silicon debugging techniques discussed above as it restricts the analysis to a certain time interval. Especially, for the investigation of soft errors and their effects, considering the latency of signals may reduce the effort for making circuits robust.

III. Preliminaries

This section provides the terminology used in the remainder of this paper. FSMs model synchronous sequential circuits and are defined in the following. We use the terms FSM and circuit interchangeably.

Definition 1 (Finite State Machine (FSM)). *An FSM is a 5-tuple $M = (I, O, S, \delta, \omega)$, where I is the* input alphabet*, O is the* output alphabet*, S is a finite set of states, $\delta : S \times I \to S$ is the state-transition function, and $\omega : S \times I \to O$ is the output function.*

The state-transition function $\delta : S \times I \to S$ determines the next state of the machine based on its current state and inputs. The output function $\omega : S \times I \to O$ determines the output of the machine based on its current state and inputs.

A synchronous sequential circuit modeled by an FSM has a finite number n of primary inputs $(pi_0, \ldots, pi_n)$, a finite number m of primary outputs $(po_0, \ldots, po_m)$, and a finite number k of state or memory elements $(s_0, \ldots, s_k)$. Since synchronous sequential circuits are considered, each application of δ transfers the state of the circuit from one time cycle to the next time cycle. To indicate the time step we use the superscript t, like e.g., $\boldsymbol{pi^t}$ denotes the primary input signals at time step t. The functions δ and ω are computed by the combinational part of the circuit.

$$\begin{aligned} \boldsymbol{s^{t+1}} &= \delta(\boldsymbol{s^t}, \boldsymbol{pi^t}) \\ \boldsymbol{po^t} &= \omega(\boldsymbol{s^t}, \boldsymbol{pi^t}) \end{aligned}$$

where $\boldsymbol{pi^t} \equiv (pi_0^t, \ldots, pi_n^t)$, $\boldsymbol{s} \equiv (s_0^t, \ldots, s_k^t)$, $\boldsymbol{s^{t+1}} \equiv (s_0^{t+1}, \ldots, s_k^{t+1})$, $\boldsymbol{po^t} \equiv (po_0^t, \ldots, po_m^t)$.

The transition relation τ of the circuit is derived from δ.

Definition 2 (State-transition relation τ). *Given an input $\boldsymbol{pi^t} \equiv (pi_1^t, \ldots, pi_n^t)$, a current state $\boldsymbol{s^t} \equiv (s_1^t, \ldots, s_k^t)$, a next state $\boldsymbol{s^{t+1}} \equiv (s_1^{t+1}, \ldots s_k^{t+1})$, and a state-transition function δ then the characteristic function of the transition relation τ of an FSM is defined as:*

$$\tau(\boldsymbol{s^t}, \boldsymbol{pi^t}, \boldsymbol{s^{t+1}}) = \boldsymbol{s^{t+1}} \equiv \delta(\boldsymbol{s^t}, \boldsymbol{pi^t})$$

If $\tau(\boldsymbol{s^t}, \boldsymbol{pi^t}, \boldsymbol{s^{t+1}})$ evaluates to 1, this implies that there exists a transition from state $\boldsymbol{s^t}$ to $\boldsymbol{s^{t+1}}$.

Based on the notations above we define the unrolling ρ_C^t of a circuit C for t time steps by

$$\rho_C^t = \bigwedge_{i=0}^{t} \tau(\boldsymbol{s^i}, \boldsymbol{pi^i}, \boldsymbol{s^{i+1}}) \wedge \boldsymbol{po^i} \equiv \boldsymbol{\omega(s^i, pi^i)}$$

We use symbols $\boldsymbol{\gamma^t}$, $\boldsymbol{\nu^t}$, $\boldsymbol{o^t}$ to denote elements of I, S, and O. These correspond to assignments to $\boldsymbol{pi^t}$, $\boldsymbol{s^t}$, and $\boldsymbol{po^t}$, respectively. A path π of an FSM is defined as follows.

Definition 3 (Path π). *A path π from a concrete state $\boldsymbol{\nu^0}$ to another concrete state $\boldsymbol{\nu^t}$ in an FSM M is given by*

$$\pi = (\boldsymbol{\nu^0}, \boldsymbol{\gamma^0}, \boldsymbol{\nu^1}, \boldsymbol{\gamma^1}, \ldots, \boldsymbol{\gamma^{t-1}}, \boldsymbol{\nu^t})$$

with state values $\boldsymbol{\nu^j} = (\nu_0^j, \nu_1^j, \ldots, \nu_k^j) \in S$,
primary input values $\boldsymbol{\gamma^i} = (\gamma_0^i, \gamma_1^i, \ldots, \gamma_n^i) \in I$,
and $\tau(\boldsymbol{\nu^i}, \boldsymbol{\gamma^i}, \boldsymbol{\nu^{i+1}}) = 1$ $(i = 0, \ldots, t-1$ and $j = 0, \ldots, t)$.
The length of the path π is given by t.

Lemma 1. *Given an FSM M. π is a path in M of length t iff ρ_C^t is satisfied when variables corresponding to input and state elements are restricted to the values given by π.*

In the following we consider two-valued simulations as well as two-valued and three-valued SAT-encodings of circuits. The input alphabet I, the output alphabet O, and the states S will either be based on $\mathbb{B} = \{0, 1\}$, or on $\mathbb{A} = \{0, 1, X\}$, where X denotes an unknown value.

IV. Latency Analysis

In this section we first describe the idea behind this paper and formally define *latencies* of a circuit. In this context the terms *minimal latency* and *maximal latency* for primary inputs with respect to the primary outputs are introduced. After this, we present three approaches to calculate latency values and show that the presented methods can be extended, for example, to analyze the latency of parts of a circuit.

A. Idea

If incorrect behavior is observed at the primary outputs of a circuit C, it is often not clear what triggered this misbehavior and what the reason of a faulty computation was. The idea of the proposed approach is to compute the latency of signals. That means, we investigate, how long signal assignments are affecting the computation of the circuit and when the assignment of values on the signal becomes observable at the primary outputs, respectively. In our presentation we concentrate on primary inputs but alternatively other signals may be considered.

We define the *maximal latency* λ of synchronous sequential circuits as follows.

Definition 4 (Maximal Latency λ). *Given an FSM M for a sequential circuit, the* maximal latency *is the maximal number λ with the following property:*
There exists a path

$$\pi = (\boldsymbol{\nu^0}, \boldsymbol{\gamma^0}, \boldsymbol{\nu^1}, \boldsymbol{\gamma^1}, \ldots, \boldsymbol{\gamma^{\lambda-1}}, \boldsymbol{\nu^\lambda})$$

and a path

$$\pi' = (\boldsymbol{\nu^0}, \boldsymbol{\gamma'^0}, \boldsymbol{\nu'^1}, \boldsymbol{\gamma^1}, \ldots, \boldsymbol{\gamma^{\lambda-1}}, \boldsymbol{\nu'^\lambda})$$

with different primary input values $\boldsymbol{\gamma^0} \neq \boldsymbol{\gamma'^0}$,
different state values $\boldsymbol{\nu^j} \neq \boldsymbol{\nu'^j}$ $(j = 1, \ldots, \lambda)$,
and equal state values $\boldsymbol{\nu^t} \equiv \boldsymbol{\nu'^t}$ with $t > \lambda$.

Conceptually, after more than λ time steps a value at the primary inputs cannot influence the internal state of the circuit. Consider the analysis of transient faults. A transient fault flips a value in the circuit. The fault effect leaves the system after at most λ cycles. This particularly holds, as our approach considers any configuration of state bits including potentially unreachable states that might only occur due to transient faults in practice. We do not use the primary outputs $\boldsymbol{po^j}$ and $\boldsymbol{po'^j}$ for measuring the maximal latency because of loops, a difference at the primary outputs may only be observable every few time cycles. On the other hand, a difference between state elements $\boldsymbol{s^j}$ and $\boldsymbol{s'^j}$ is measurable over all time cycles. By this, the maximal latency value may be infinite. More on that later.

Similarly, we define the *minimal latency* μ over the primary outputs of a circuit.

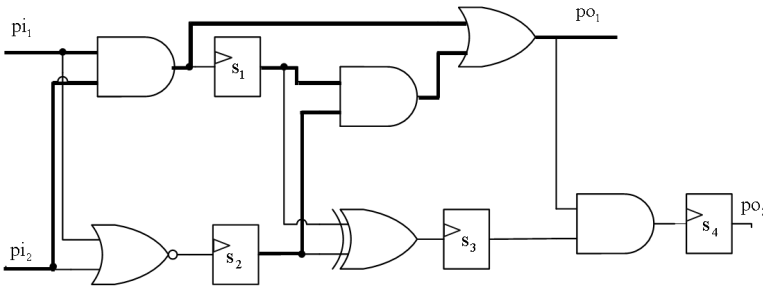
Definition 5 (Minimal Latency μ). *Given an FSM M for a sequential circuit, then the* minimal latency *is the minimal number μ with the following property:*
There exists a path

$$\pi = (\boldsymbol{\nu^0}, \boldsymbol{\gamma^0}, \boldsymbol{\nu^1}, \boldsymbol{\gamma^1}, \ldots, \boldsymbol{\gamma^{\mu-1}}, \boldsymbol{\nu^\mu})$$

and a path

$$\pi' = (\boldsymbol{\nu^0}, \boldsymbol{\gamma'^0}, \boldsymbol{\nu'^1}, \boldsymbol{\gamma^1}, \ldots, \boldsymbol{\gamma^{\mu-1}}, \boldsymbol{\nu'^\mu})$$

with different initial primary input values $\boldsymbol{\gamma^0} \neq \boldsymbol{\gamma'^0}$,
different state values $\boldsymbol{\nu^j} \neq \boldsymbol{\nu'^j}$ $(j = 1, \ldots, \mu)$,



t	0		1	2	3	4
pi_1pi_2	11		01	11	11	00
$pi'_1pi'_2$	01		01	11	11	00
po_1po_2	**10**	$\}\mu$	00	10	11	00
$po'_1po'_2$	**00**		00	10	10	00
$s_1s_2s_3s_4$	0000		1000	0010	**1001**	$\}\lambda$ 1010
$s'_1s'_2s'_3s'_4$	0000		0000	0000	**1000**	$\}\lambda$ 1010

Fig. 1. Example circuit

and the values of the primary outputs are different $\boldsymbol{o^{\mu} \neq o'^{\mu}}$ *for time cycle* μ*.*

The values of the primary outputs are computed by $\boldsymbol{o^{\mu}} = \omega(\boldsymbol{\nu^{\mu}, \gamma^{\mu}})$, $\boldsymbol{o'^{\mu}} = \omega(\boldsymbol{\nu'^{\mu}, \gamma^{\mu}})$, *with* $\boldsymbol{o^{\mu}} = (o_0^{\mu}, \ldots, o_m^{\mu}) \in O, \boldsymbol{o'^{\mu}} = (o_0'^{\mu}, \ldots, o_m'^{\mu}) \in O$.

In other words, μ represents the first influence of values at primary inputs on the values of primary outputs. When debugging, an erroneous output value must be explained and understood. In this case, the input assignments of at least μ time steps in the past must be considered, but at most λ time steps in the past are relevant for the explanation.

Example 1. *In Fig. 1 we illustrate a simple sequential circuit. If there is an erroneous output for a given input stimulus, the circuit has to be unrolled for* $\mu = 0$ *time cycles at the minimum and for* $\lambda = 3$ *time cycles at the maximum to reproduce the faulty output. The path for the minimal latency is in bold in the figure. In the table we give an example for the assignment of the primary inputs to retrieve the latency values. Assignments for 5 time cycles* $t = 0, \ldots, 4$ *are given. By definition, for* $t = 0$ *different assignments to the primary inputs are applied and the initial state values are equal. For this case, we get a minimal latency* $\mu = 0$*. In all following time cycles equal assignments are applied to the primary inputs, such that a difference is observable at the state elements. In time cycle* 4 *no longer a difference is observable. The entries in the table which explain the latency values are bold-faced.*

In this case, 0 *is also the number of flip flops on the shortest structural path from a primary input to a primary output and* 3 *is the number of flip flops on the longest structural path to a primary output plus one. This suggests a structural approach to determine the latency. But in general a structural approach is not exact as it neglects the functionality of the circuit.*

In comparison, the diameter of an FSM is defined as the longest of all shortest paths between the initial state and every other reachable state. In addition, no equal states on the path should occur. The minimal latency depends on the shortest among all shortest paths. In contrast, the sequential depth and the recurrence diameter are completely different from the minimal and the maximal latency, respectively. For example, consider a simple circuit implementing a 1-bit memory. Both the sequential depth and the recurrence diameter are 2 while the minimal and the maximal latency are 1. Extending this example to n state elements, i.e. an n-bit memory, we get 2^n as sequential depth while the maximal and the minimal latency are still 1. Thus, μ and λ can be exponentially smaller than r and d. Vice versa for other circuits, λ can be infinite while r and d are linear in size since both the definition of the minimal latency and the definition of the maximal latency do not have the restriction of being loop-free. Instead, there have to exist two different paths fulfilling the constraints.

B. Realization

We present three approaches to compute the latency of a circuit C. Depending on the approach either exact or approximate values for the minimal latency μ and maximal latency λ are returned. The first approach is based on SEC and the second approach is based on 3-valued BMC. For both approaches a SAT-instance is created and the circuit considered is unrolled incrementally. The third approach is simulation-based. The three procedures are compared and discussed at the end of this section.

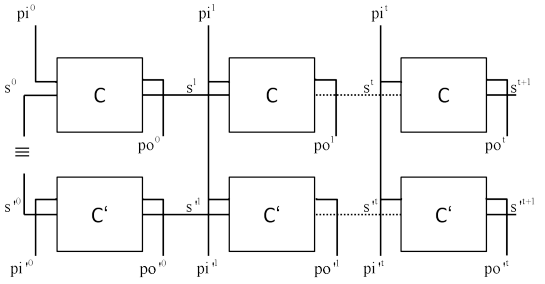


Fig. 2. Principle of latency analysis with equivalence check

Note, that we do not consider reachability issues in the following but each of the algorithms can be restricted to reachable states. In this case, both the values for the minimal latency μ and the maximal latency λ may vary. But, especially regarding transient faults a circuit also can get into a state which is actually unreachable during normal operation.

1) SEC-based Latency Analysis: The SEC approach is based on two-valued logic using an input alphabet $I = \mathbb{B}^n$, states $S = \mathbb{B}^k$, and an output alphabet $O = \mathbb{B}^m$, where $\mathbb{B} = \{0, 1\}$. The circuit C and a copy C' of C are unrolled for t time steps and an equivalence check is carried out between both circuits. This is formulated by $\rho_C^t \wedge \rho_{C'}^t$. The principle of this method is shown in Fig. 2 that illustrates the structure of the underlying SAT-instance. The SAT-instance is the basis to decide whether $\lambda \geq t$ or $\mu \leq t$. Additional constraints at primary inputs, primary outputs, and states are required to determine the latency.

For SEC a circuit C and the copy C' are considered. For the initial states $\boldsymbol{s^0}$ and $\boldsymbol{s'^0}$ identical assignments are assumed, such that:

$$\sigma_{SEC}^0 = \bigwedge_{i=0}^{k} s_i^0 \equiv s_i'^0$$

C and C' are unrolled incrementally as shown in Fig. 2. In all time cycles $t > 0$ equivalent signal assignments are assumed for the primary inputs of both circuits:

$$\iota_{SEC}^t = \bigwedge_{j=1}^{t} \bigwedge_{i=0}^{n} pi_i^j \equiv pi_i'^j.$$

Now, the minimal latency is given as $\mu = t$ by the first time cycle t where the following formula is satisfiable. The symbol $\oplus$ denotes the XOR function.

$$\phi_{SEC} = \rho_C^t \wedge \rho_{C'}^t \wedge \sigma_{SEC}^0 \wedge \iota_{SEC}^t \wedge \bigvee_{i=0}^{m} po_i^t \oplus po_i'^t$$

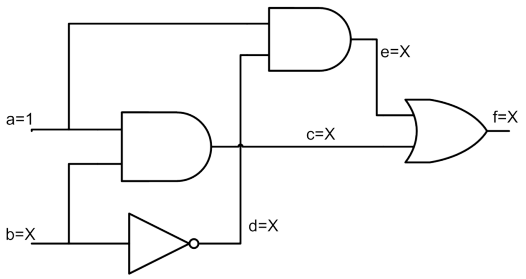


Fig. 3. Conservative X propagation

To apply incremental satisfiability [17], this SAT-instance is created starting at $t = 0$ to iteratively increase t until finding a satisfying assignment.

And corresponding to this, the maximal latency is given by $\lambda = r - 1$ if the following formula is satisfiable for $t \leq r - 1$ and unsatisfiable for $t = r$:

$$\sigma_{SEC} = \rho_C^t \wedge \rho_{C'}^t \wedge \sigma_{SEC}^0 \wedge \iota_{SEC}^t \wedge \bigvee_{i=0}^{k} s_i^r \oplus s_i'^r$$

With this method we exactly determine when an assignment at the primary inputs $\boldsymbol{pi^0}$ will be observable at the primary outputs $\boldsymbol{po^t}$ for the first time and for how long an assignment at the primary inputs maximally affects the circuit.

Theorem 1. *SEC-based latency analysis calculates μ and λ.*

Proof: The unrolling of the circuit is satisfiable by applying values corresponding to a path in the circuit. Now consider μ at first. If there exists no assignment for t such that there are differences at the primary outputs, then there exist no two paths as required by Definition 5. Since the circuit is unrolled incrementally, the smallest t is determined where two paths exist as required by Definition 5.

This is similar for λ: If there exists an assignment for the circuit such that there are differences in the state variables, this assignment and the corresponding paths will be found for the same reason. ■

2) 3-Valued Approach: The 3-valued method uses an input alphabet $I = \mathbb{A}^n$, states $S = \mathbb{A}^k$, and an output alphabet $O = \mathbb{A}^m$, where $\mathbb{A} = \{0, 1, X\}$. We encode the well-known semantics of 3-valued simulation into the SAT-instance. However, 3-valued simulation is known to be conservative in the propagation of an X, such that an X may be propagated instead of an actual value.

Example 2. *In Fig. 3 we show an example for the conservative propagation of X. For the assignment of $a = 1$ and $b = X$ we will get an X at output f. However, the circuitry driven by b is redundant. Consequently, independently of the assignment of b, f is functionally equivalent to a.*

The principle of the 3-valued approach is shown in Fig. 4 and described below. Initially, the unknown value X is assigned to at least one of the primary inputs $pi_i^0 \in I$ such that

$$\iota_{3V}^0 = \bigvee_{i=0}^{n} pi_i^0 \equiv X.$$

At the same time it has to be ensured that no X has been assigned to the initial state variables s_i^0 because otherwise an X may be recognized at the outputs po_i^t that did not originate at the inputs pi_i^0:

$$\sigma_{3V}^0 = \bigwedge_{i=0}^{k} s_i^0 \neq X$$

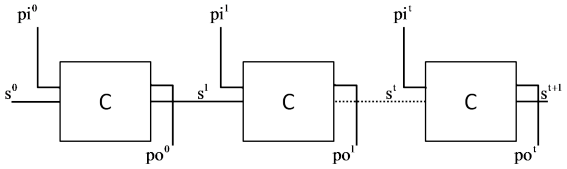


Fig. 4. Principle of 3-valued latency analysis

Subsequently, the circuit is unrolled incrementally analogous to the SEC approach and the primary outputs po_i^t are checked whether an X is assigned to one of them in time step t. This is expressed by the following formula:

$$\phi_{3V}^t = \rho_C^t \wedge \iota_{3V}^0 \wedge \sigma_{3V}^0 \wedge \bigvee_{i=0}^{m} po_i^t \equiv X. \quad (1)$$

With satisfying Formula (1) in time cycle t we have an under-approximation $\mu_{3V} = t$ for the minimal latency.

Similarly, the maximal latency λ is determined by checking for how long the X is propagated through the state elements s_i^t. Thus, we have an over-approximation $\lambda_{3V} = t$ for the maximal latency, if the following formula is unsatisfiable at time cycle t:

$$\sigma_{3V}^t = \rho_C^t \wedge \iota_{3V}^0 \wedge \sigma_{3V}^0 \wedge \bigvee_{i=0}^{k} s_i^t \equiv X.$$

Theorem 2. *Let μ_{3V} be calculated by the 3-valued approach and μ be the exact latency. Then $\mu_{3V} \leq \mu$.*

Proof Idea: Given an exact value μ there exist two paths π and π'. Then there exists an assignment π_{3V} to σ_{3V}^{μ} derived from π and π', such that $\sigma_{3V}^{\mu}(\pi_{3V}) = 1$.

In addition, because the value X is propagated conservatively through the circuit, as shown in Example 2, an X can be observed at time cycle $t < \mu$, such that $\mu_{3V} \leq \mu$. ■

Theorem 3. *Let λ_{3V} be calculated by the 3-valued approach and λ be the exact latency. Then $\lambda_{3V} \geq \lambda$.*

Proof: Analogously to $\mu_{3V} \leq \mu$, $\lambda_{3V} \geq \lambda$ is proven. ■

3) Simulation: In the simulation-based approach certain values are randomly simulated at the input signals pi_i^t of a circuit C in each time cycle t. An approximation λ_{Sim} for the maximal latency λ and an approximation μ_{Sim} for the minimal latency μ are determined corresponding to the same constraints described for the SEC-based approach. To compare simulation to the other procedures the initial state signals are assigned with random values such that also unreachable states are created.

Theorem 4. *With simulation latency values are obtained that over-approximate the minimal latency $\mu_{Sim} \geq \mu$ and under-approximate the maximal latency $\lambda_{Sim} \leq \lambda$.*

Proof: Since only random values are simulated, as a consequence assignments may be missed, leading to a smaller value for μ or to a larger value for λ. ■

C. Comparison of the Approaches

The proposed procedures may deliver different values for the latency of the same circuit. The SEC-approach yields exact values for μ and λ that are in between the results of the simulation-based approach and the 3-valued approach:

$$\begin{aligned} \mu_{3V} &\leq \mu_{SEC} = \mu &&\leq \mu_{Sim} \\ \lambda_{3V} &\geq \lambda_{SEC} = \lambda &&\geq \lambda_{Sim} \end{aligned}$$

The simulation-based approach tends to over-approximate μ and under-approximate λ while the 3-valued approach applies conservative X-propagation that under-approximates μ and over-approximates λ.

D. Extensions

The methodologies proposed above can be extended in several ways. Typically, the functionality of certain primary inputs is known. For example, consider a counter with a reset input. Resetting one copy of the counter while not resetting a second copy will cause the two circuits to remain in different states for an unbounded number of future time steps. In such a case restricting the latency analysis to the remaining inputs is useful.

In general, a set of inputs may be excluded from the analysis as follows. Given a set $\Gamma \subseteq \{1, \ldots, n\}$ the latency analysis is performed by replacing the constraints ι^0_{SEC} on the primary inputs for the SEC-based approach by

$$\iota'^0_{SEC} = \bigvee_{i \in \Gamma} pi_i \oplus pi'_i \wedge \bigvee_{i \notin \Gamma} pi_i \equiv pi_i$$

and the constraints ι^0_{3V} on the primary inputs for the 3-valued approach by

$$\iota'^0_{3V} = \bigvee_{i \in \Gamma} pi_i \equiv X \wedge \bigwedge_{i \notin \Gamma} pi_i \oplus X.$$

Otherwise the approaches are working as described above. Analogously, the analysis may be restricted to a set of state variables, a set of output variables, or to certain internal variables, respectively.

Having loops in a circuit, often the primary input values can be chosen such that the difference in the states can be propagated over all time cycles. To avoid an infinite run of the approaches a possible extension is *loop detection*. With loop detection we check whether a current state where latency is still observable represents a repeated state of the covered path. Loop detection has been implemented for SEC and is only carried out if already a minimal latency μ has been observed. Given two paths $\pi = (\boldsymbol{\nu^{\mu}}, \boldsymbol{\gamma^{\mu}}, \boldsymbol{\nu^{\mu+1}}, \boldsymbol{\gamma^{\mu+1}}, \ldots, \boldsymbol{\gamma^{t-1}}, \boldsymbol{\nu^{t}})$ and $\pi' = (\boldsymbol{\nu'^{\mu}}, \boldsymbol{\gamma^{\mu}}, \boldsymbol{\nu'^{\mu+1}}, \boldsymbol{\gamma^{\mu+1}}, \ldots, \boldsymbol{\gamma^{t-1}}, \boldsymbol{\nu'^{t}})$ with $t \leq \lambda$ loop detection LD is defined by

$$LD = \bigvee_{i=\mu}^{t-1} \boldsymbol{\nu^{i}} \equiv \boldsymbol{\nu^{t}} \wedge \boldsymbol{\nu'^{i}} \equiv \boldsymbol{\nu'^{t}}.$$

With this formula loops in both paths π and π' are detected in the same time cycle t. Note that there may be different loops in the two copies of the circuit with different lengths k and l. These loops are detected in time cycle $t = k \cdot l$.

V. Experimental Results

The latency analysis approaches described in the previous section have been evaluated on a set of ITC'99 benchmark circuits. These include among other functionality FSMs (`b01`, `b02`), an arbiter (`b03`), simple algorithms and games (`b04`, `b12`), and parts of processors (`b14`, `b15`). The benchmarks consist of up to $243k$ gates (`b19`). More information about the properties of the benchmarks is available in [18].

The experiments have been carried out on an *Dual-Core AMD Opteron(tm) Processor 2222SE* with 64 GB of memory. The latency values μ and λ as well as the run times in CPU seconds have been measured. We restricted the latency analysis for all approaches to a time limit of 60 minutes per instance. In addition, the simulation-based procedure has been limited to 2000 simulation traces and 2000 time cycles per simulation trace at the maximum for the same instance. Since the introduced measures are new, there exist no other approaches for the latency analysis of circuits. Thus, a comparison to other procedures is not possible so far.

In Table I at first the three proposed procedures are compared to each other. In addition, in column $\texttt{SEC}_{ld}$ SEC is carried out with loop detection. A time out is denoted as `t.o`. Time outs are especially expected for circuits with loops for the computation of the maximal latency using SEC or the 3-valued approach. In the case of a time out the maximal latency λ represents the time cycle t, where the analysis procedure was canceled. Consequently, $\lambda \geq t$ can be assumed. The run times are relatively small for calculating the minimal latency, where only a few time steps have to be considered. Only for calculating the maximal latency, run time becomes an issue. Interpolation techniques could be used to speed up the calculation as shown for reachability analysis before [19].

The minimal latency μ which is quite small is retrieved for almost all circuits by the SEC-based exact approach. Also the two heuristic approaches retrieve the exact values in most cases. Simulation coarsely over-approximates μ only in case of *b09* where a value of 10 is retrieved instead of the exact value 1. The 3-valued approach is even more accurate, where μ is underestimated by one for *b11*, while the exact result is retrieved in all other cases. For the large benchmarks `b18` and `b19` the two heuristic procedures provide an approximation for the latencies. Both approaches return the same value for the minimal latency. Thus, the value is exact and the computational more expensive approach based on SEC is not required.

The computation of the maximal latency λ with SEC exceeds the time limitation in most cases. This also happens for the 3-valued over-approximation. The reason for the long run times of the approaches is explained by the SEC approach with loop detection. In all cases where the analysis runs out of time, a loop is detected which leads to infinite maximal latency values. Because of the conservative X-propagation the 3-valued approach may also run into a loop in circuits that actually do not have a loop, see, for example, *b01* and *b02*.

Due to the large search space, the simulation-based approach is quite inaccurate when the maximal latency is considered. This behavior is expected as in many cases a circuit may be influenced for an unbounded number of time steps by certain events – remember the example of a counter with a reset signal in Section IV-D.

The simulation runs additionally yield statistical data about the circuit. For example, μ and λ are equal for `b02` for all random simulation runs. This means if an error is recognized at the primary outputs it will most likely disappear in the next time cycle. For benchmark `b07` a difference at the primary inputs does not affect the primary outputs for 90% of the simulation runs. Similarly, for `b11` just in 4% of all runs different input stimuli affect the primary outputs.

Next we consider different results for the minimal latency of single input variables in Table II. Only those circuits are shown where the minimal latency varies for different inputs. In the table, we give the range of μ for individual inputs of a benchmark. As can be seen the variation is quite small for most circuits – an exception is *b08*. Assigning different input values to single primary inputs, a minimal latency between 1 and 17 is observed. A further enhancement of the resolution is possible by separately considering each output or sets of outputs that are important for debugging, e.g., where an error has been observed.

In Table III we analyzed the latency of single primary inputs of the voting system `b10` with the SEC approach. We achieve results for the minimal latency μ that vary between 1 and 4. However, the maximal latency always is $\lambda = 9$. This means that an error at input 0 at the maximum will stay within the circuit for 5 time cycles while an error at input 2 may influence the results for 8 time cycles only. Moreover, the analysis of single signals always leads to a maximal latency λ while the

TABLE I
RESULTS FOR MINIMAL LATENCY μ AND MAXIMAL LATENCY λ

benchmark				SEC				3V				Simulation				SEC_{ld}			
	#pi	#po	#ff	μ_{SEC}	λ_{SEC}	time μ_{SEC}	time λ_{SEC}	μ_{3V}	λ_{3V}	time μ_{3V}	time λ_{3V}	μ_{Sim}	λ_{Sim}	time μ_{Sim}	time λ_{Sim}	$\mu_{SEC_{ld}}$	#cycles ($\lambda_{SEC_{ld}}$)	time $\mu_{SEC_{ld}}$	time $\lambda_{SEC_{ld}}$
b01	2	2	5	**1**	**6**	< 1	< 1	**1**	$\geq$ **2037**	< 1	t.o.	**1**	**5**	< 1	< 1	**1**	**6**	< 1	< 1
b02	1	1	4	**2**	**5**	< 1	< 1	**2**	$\geq$ **2955**	< 1	t.o.	**2**	**4**	< 1	< 1	**2**	**5**	< 1	< 1
b03	4	4	30	**4**	$\geq$ **418**	< 1	t.o.	**4**	$\geq$ **1277**	< 1	t.o.	**4**	**44**	< 1	< 1	**4**	**6**(∞)	< 1	< 1
b04	11	8	66	**1**	$\geq$ **120**	< 1	t.o.	**1**	$\geq$ **340**	< 1	t.o.	**1**	**481**	< 1	< 1	**1**	**2**(∞)	< 1	< 1
b05	1	36	34	**1**	$\geq$ **476**	< 1	t.o.	**1**	$\geq$ **257**	< 1	t.o.	**1**	$\geq$ **2000**	< 1	< 1	**1**	**3**(∞)	< 1	< 1
b06	2	6	9	**1**	$\geq$ **325**	< 1	t.o.	**1**	$\geq$ **2187**	< 1	t.o.	**1**	**43**	< 1	< 1	**1**	**2**(∞)	< 1	< 1
b07	1	8	49	**1**	$\geq$ **491**	< 1	t.o.	**1**	$\geq$ **644**	< 1	t.o.	**1**	$\geq$ **2000**	< 1	< 1	**1**	**42**(∞)	< 1	2856
b08	9	4	21	**1**	$\geq$ **385**	< 1	t.o.	**1**	$\geq$ **1103**	< 1	t.o.	**1**	**901**	< 1	< 1	**1**	**17**(∞)	< 1	13
b09	1	1	28	**2**	$\geq$ **458**	< 1	t.o.	**2**	$\geq$ **1252**	< 1	t.o.	**10**	**75**	< 1	< 1	**2**	**11**(∞)	< 1	< 1
b10	11	6	17	**1**	$\geq$ **320**	< 1	t.o.	**1**	$\geq$ **785**	< 1	t.o.	**1**	$\geq$ **2000**	< 1	< 1	**1**	**2**(∞)	< 1	< 1
b11	7	6	31	**3**	$\geq$ **200**	< 1	t.o.	**2**	$\geq$ **260**	< 1	t.o.	**3**	$\geq$ **2000**	< 1	< 1	**3**	**4**(∞)	< 1	< 1
b12	5	6	121	**1**	$\geq$ **214**	< 1	t.o.	**1**	$\geq$ **271**	< 1	t.o.	**1**	$\geq$ **2000**	< 1	8.7	**1**	**5**(∞)	< 1	1.4
b13	10	10	53	**1**	$\geq$ **649**	< 1	t.o.	**1**	$\geq$ **759**	< 1	t.o.	**1**	$\geq$ **2000**	< 1	1.1	**1**	**2**(∞)	< 1	< 1
b14	32	54	245	**2**	$\geq$ **66**	4.2	t.o.	**2**	$\geq$ **72**	4.0	t.o	**2**	**740**	< 1	3.7	**2**	**4**(∞)	3.8	6.0
b15	36	70	449	**2**	$\geq$ **128**	3.6	t.o.	**2**	$\geq$ **113**	3.9	t.o	**2**	$\geq$ **2000**	< 1	31.8	**2**	**3**(∞)	5.1	2.2
b17	37	97	1415	**2**	$\geq$ **41**	22.8	t.o.	**2**	$\geq$ **39**	17.7	t.o.	**2**	$\geq$ **2000**	< 1	327	**2**	**3**(∞)	18.9	9.5
b18	37	22	3320	**-**	**-**	t.o.	-	**2**	$\geq$ **13**	146	t.o.	**2**	$\geq$ **2000**	2.2	823	**-**	**-**	t.o.	-
b19	24	27	6642	**-**	**-**	t.o.	-	**2**	$\geq$ **6**	2246	t.o.	**2**	$\geq$ **173**	4.6	t.o.	**-**	**-**	t.o.	-
b20	32	22	490	**2**	$\geq$ **38**	833	t.o.	**2**	$\geq$ **43**	12.2	t.o.	**2**	$\geq$ **2000**	< 1	25.0	**2**	**4**(∞)	2704	17.8
b21	32	22	490	**2**	$\geq$ **39**	1369	t.o.	**2**	$\geq$ **44**	8.9	t.o.	**2**	$\geq$ **2000**	< 1	46.4	**2**	**4**(∞)	512	42.7
b22	32	22	735	$\geq$ **1**	**-**	t.o.	-	**2**	$\geq$ **33**	19.4	t.o.	**2**	$\geq$ **2000**	< 1	14.6	$\geq$ **1**	**-**	t.o.	-

TABLE II
RANGE OF μ FOR SINGLE INPUTS

benchmark	#pi	#po	#ff	μ
b08	9	4	21	1-17
b10	11	6	17	1-4
b12	5	6	121	1-2
b13	10	10	53	1-3
b14	32	54	245	2-3
b15	36	70	449	2-3
b17	37	97	1415	2-3

TABLE III
μ AND λ OF B10 FOR SINGLE INPUTS

b10		
input	μ	λ
0	4	9
1	4	9
2	1	9
3	1	9
4	4	9
5	2	9
6	2	9
7	2	9
8	2	9
9	2	9
10	2	9

consideration of all inputs always contains loops such that an infinite maximal latency has been determined. The analysis of single signals or a subset of signals especially becomes interesting if parts of circuits have to be analyzed.

In summary, all of the approaches are applicable for the circuits considered. Run time is not an issue for most of the circuits considered here. For the minimal latency also the two approximate approaches may be used as they yield almost exact values. The maximal latency is typically infinite for the benchmark circuits considered here if a design contains loops. But in practice, a range of designs will exhibit a different behavior. For example, most pipelined designs or data sampling designs only depend on a limited number of past time steps.

VI. CONCLUSION

We introduced a formal definition for the latency of sequential circuits and evaluated three methods for computing minimal and maximal latency. The two approximate approaches are orthogonal in the sense that one is optimistic while the other one is pessimistic. Together they yield bounds on the latency values and the difference between these bounds shows a potential inaccuracy. The described methods have been effectively applied on ITC'99 benchmark circuits.

In future work we will utilize the new measure within debugging algorithms and the analysis of fault tolerant circuits.

REFERENCES

[1] G. Fey, S. Staber, R. Bloem, and R. Drechsler, "Automatic fault localization for property checking," *IEEE Trans. on CAD*, vol. 27, no. 6, pp. 1138–1149, 2008.

[2] M. Khalil, Y. L. Traon, and C. Robach, "Towards an automatic diagnosis for high-level design validation," in *International Test Conference*, 1998, pp. 1010–1018.

[3] M. Renieres and S. P. Reiss, "Fault localization with nearest neighbor queries," in *18th IEEE International Conference on Automated Software Engineering*, 2003, pp. 30–39.

[4] R. Leveugle, "A new approach for early dependability evaluation based on formal property checking and controlled mutations," in *IEEE International On-Line Testing Symposium*, 2005, pp. 260–265.

[5] U. Krautz, M. Pflanz, C. Jacobi, H. W. Tast, K. Weber, and H. T. Vierhaus, "Evaluating coverage of error detection logic for soft errors using formal methods," in *Design, Automation and Test in Europe*, 2006, pp. 176–181.

[6] M. Bozzano, A. Cimatti, and F. Tapparo, "Symbolic fault tree analysis for reactive systems," in *Automated Technology for Verification and Analysis*, ser. LNCS, vol. 4762, 2007, pp. 162–176.

[7] J. Hayes, I. Polian, and B. Becker, "An analysis framework for transient-error tolerance," in *VLSI Test Symposium*, 2007, pp. 249–255.

[8] G. Fey, A. Sülflow, and R. Drechsler, "Computing bounds for fault tolerance using formal techniques," in *Design Automation Conference*, 2009, pp. 190–195.

[9] M. Mneimneh and K. Sakallah, "SAT-based sequential depth computation," in *ASP Design Automation Conference*, 2003, pp. 87–92.

[10] N. Drechsler and R. Drechsler, *Exploration of Sequential Depth by Evolutionary Algorithms*, ser. IFIP. Springer Boston, 2006, vol. 200, pp. 73–83.

[11] M. Awedh and F. Somenzi, "Increasing the robustness of bounded model checking by computing lower bounds on the reachable states," in *International Conference on Formal Methods in CAD*, ser. LNCS, vol. 3312. Springer Verlag, 2004, pp. 230–244.

[12] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu, "Symbolic model checking without BDDs," in *Tools and Algorithms for the Construction and Analysis of Systems*, ser. LNCS, vol. 1579. Springer Verlag, 1999, pp. 193–207.

[13] A. Biere, A. Cimatti, E. Clarke, M. Fujita, and Y. Zhu, "Symbolic model checking using SAT procedures instead of BDDs," in *Design Automation Conference*, 1999, pp. 317–320.

[14] D. Kroening and O. Strichman, "Efficient computation of recurrence diameters," in *International Conference on Verification, Model Checking, and Abstract Interpretation*, ser. LNCS, vol. 2575. Springer Verlag, 2003, pp. 298–309.

[15] S. Prabhakar and M. Hsiao, "Using non-trivial logic implications for trace buffer-based silicon debug," in *Asian Test Symposium*, 2009, pp. 131–136.

[16] J.-K. Zhao, J. Newquist, and J. Patel, "A graph traversal based framework for sequential logic implication with an application to c-cycle redundancy identification," in *VLSI Design Conference*, 2001, pp. 163–169.

[17] J. Whittemore, J. Kim, and K. Sakallah, "SATIRE: A new incremental satisfiability engine," in *Design Automation Conference*, 2001, pp. 542–545.

[18] ITC99 Benchmark Home Page. (1999) http://www.cerc.utexas.edu/itc99-benchmarks/bench.html.

[19] K. L. McMillan, "Interpolation and SAT-based model checking," in *Computer Aided Verification*, ser. LNCS, 2003, pp. 1–13.