Theorem-proving Verification of Multi-clock Synchronous Circuits on Multimodal Logic

Shunji Nishimura Motoki Amagasaki Toshinori Sueyoshi
Graduate School of Science and Technology
Kumamoto University
2-39-1 Kurokami, Chuo-ku Kumamoto 860-8555, Japan
nishimura@arch.cs.kumamoto-u.ac.jp {amagasaki,sueyoshi}@cs.kumamoto-u.ac.jp

Abstract— Formal verification methods for synchronous circuits are widely used, but almost all of the methods are limited to single-clock synchronous circuits. In this paper, we propose a formal verification method for multi-clock synchronous circuits. The proposed verification method is in theorem-proving manner and based on multimodal logic. We also show an example of verification of a clock switching circuit by using the method.

I. Introduction

Formal methods for hardware verification are being more widely adopted because of demands of comprehensive verification. Formal verification methods can be classified into equivalence-checking methods and property-checking methods; in this article, we focus on the latter type. Many tools for property checking have been released by major electronic design automation (EDA) companies, although nearly all of such tools are limited to working with only single-clock synchronous circuits.

Clarke et al. have proposed a verification method for multi-clock synchronous circuits that employs model checking tool in [1]. However, that method requires specifying the relations between clocks; it cannot be applied to sets of unrelated clocks.

In this article, we adopt a theorem-proving verification. Theorem-proving has been applied to circuit verifications since the 1980s [2, 3], but there has been no applications for multi-clock synchronous circuits. We propose a deduction system for multi-clock synchronous circuits, which is suitable even when the clocks are unrelated. The verification system is based on multimodal logic [4], and we also a verification example on that.

II. Verification system based on multimodal logic

A. Circuits definition

A multi-clock synchronous circuit is expressed as shown in Fig. 1. We assume hereinafter that the behavior of the circuit is observed under totally ordered discrete times (e.g. natural numbers). A temporal logic is built as a Kripke semantics characterized by a set of times T and a binary relation R on T. Temporal logic has two unary operators that we use here: the Globally operator $\Box$, and the Next operator $\bigcirc$. For a given test pattern pt, we can define a new operator $\textcircled{i}$ that represents the concept of 'next' for FF_i along with pt. We let the operator $\textcircled{i}P$ mean that P holds at each time point which passes one edge-of-clock for FF_i. We use $R_{\bigcirc}$ and $R_{\textcircled{i}}$ to denote binary relations of $\bigcirc$ and $\textcircled{i}$, respectively.

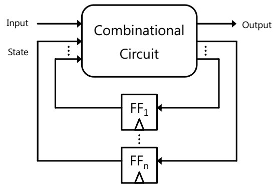


Fig. 1.: Multi-clock synchronous circuit

A multi-clock synchronous circuit can be described by using these new operators in the following equation system:

$$\begin{cases} out &= f^*(a,(s_0,\cdots,s_n)) \\ \textcircled{0}s_0 &= f_{*0}(a,(s_0,\cdots,s_n)) \\ &\vdots \\ \textcircled{n}s_n &= f_{*n}(a,(s_0,\cdots,s_n)) \,. \end{cases} \tag{1}$$

Where, a is input, out is output, s_i is the state of FF_i, f^* is the output function of the combinational part, and f_{*i} is the state function for FF_i.

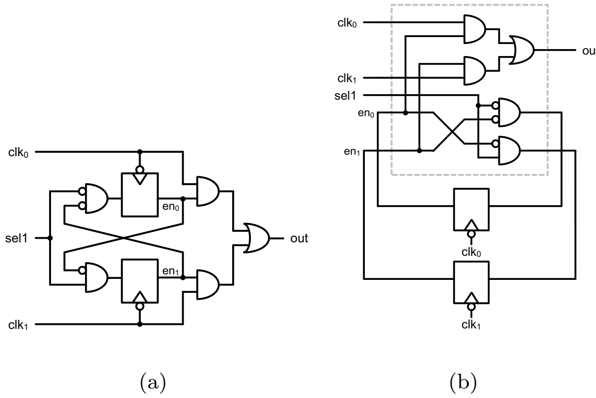
B. Definition of deduction system

The axioms for our deduction system are defined as the following properties, which consist of the semantics of the previous subsection. (Where, $\Diamond$ is defined as $\Diamond P := \neg\Box\neg P$ in the conventional way.)

Definition 1. *(Axioms of the verification system)*

$$\begin{array}{lcl}
\textit{Axiom T} & : & \Box P \;\Rightarrow\; P \\
\textit{Axiom 4} & : & \Box P \;\Rightarrow\; \Box\Box P \\
\textit{Axiom V1} & : & \Box P \;\Rightarrow\; \bigcirc P \\
\textit{Axiom V2} & : & P \;\Rightarrow\; \Diamond P \\
\textit{Axiom V3} & : & \textit{When}\;\; \exists t' .\, t\, R_{\bigcirc}\, t',\;\; t \models \bigcirc\Diamond P \;\Rightarrow\; \Diamond P \\
\textit{Axiom V4} & : & \textit{When}\;\; \exists t' .\, t\, R_{\bigcirc}\, t',\;\; t \models \bigcirc s_i \;=\; s_i \;\textit{or}\; \textcircled{i} s_i \\
 & & \textit{(where } s_i \textit{ is the state of } FF_i\textit{)} \\
\textit{Axiom VN1} & : & (0 \models P \;\textit{and}\; P \Rightarrow \bigcirc P) \;\Rightarrow\; \Box P \\
\textit{Axiom VN2} & : & \textit{when}\;\; \exists t' .\, t\, R_{\textcircled{i}}\, t',\;\; \exists n \,.\, (\, t \models \textcircled{i} = \bigcirc^n \,)
\end{array}$$

Where, $t \models P$ means that P holds at time t. Axioms T and 4 are equivalent to axioms of general multimodal logic



(a) (b)

Fig. 2.: Clock selector CLKSEL

theory [4]; they represent the reflexivity and transitivity, respectively, of relations on Kripke semantics. From our assumption of a totally ordered discrete time space, it follows immediately that those two axioms hold. Axiom *V1* and the axioms below it are novel. Axioms *V1* to *V4* show the relations between $\Box$, $\Diamond$, $\bigcirc$, and $\textcircled{i}$. Axiom *VN1* describes the principle of mathematical induction. The $\bigcirc$ operator is regarded as the unit time of behavior evaluation; this lets Axiom *VN2* follow from the system.

Definition 2. *(The verification system) The proposing verification system consists of a pair of the axioms and the modal description of circuit* (1).

III. Verification example: clock selector

We adopt the previously defined system and use it to verify the clock selector circuit $CLKSEL$ depicted in Fig. 2. Panel (b) is a synchronous depiction of the same circuit as shown in panel (a). $CLKSEL$ outputs clk_0 when $sel1 = 0$, and clk_1 when $sel1 = 1$. Glitch noises are avoided by waiting for the low value during the exchanging phase. (For our purposes, we take "glitch noise" as a short pulse whose length is less than the both of low and high periods of the two clocks.) Figure 3 shows the timing chart of the following behavior: at first, clk_0 is activated by $sel1 = 0$, and then clk_1 is activated by $sel1 = 1$. The circuit enters the "exchanging" phase at the first edge of clk_0 after $sel = 1$. Neither clk_0 nor clk_1 is derived during the exchanging phase. After, en_1 becomes 1 at the first edge of clk_1 and $CLKSEL$ begins clk_1 output. Let $\textcircled{0}$ be a modal operator for the FF that yields en_0 (which is synchronized by clk_0), and let $\textcircled{1}$ be a modal operator for the FF that yields en_1 (which is synchronized by clk_1.) Then, $CLKSEL$ can be represented as follows:

$$\begin{cases} out = en_0 \ \&\ clk_0 \mid en_1 \ \&\ clk_1 \\ \textcircled{0} en_0 = \overline{sel1} \ \&\ \overline{en_1} \\ \textcircled{1} en_1 = \ sel1 \ \&\ \overline{en_0}. \end{cases} \tag{2}$$

We assume that following requirements must be met:

Spec1 **:** The selected clock must be output; and

Spec2 **:** glitch noises must be avoided.

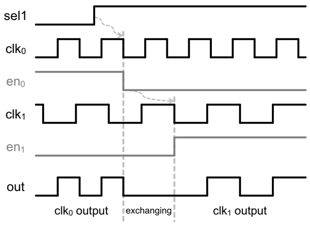


Fig. 3.: Example behavior of CLKSEL

Even though the proposed system can be used to verify that both of these specifications are met, we focus on *Spec*1, which suggests an advantage of the proposed method. To proceed, $CLKSEL$ needs an exchanging phase, as shown in Figure 3, which means that the output clock is not immediately decided by $sel1$. Specification *Spec*1 can be restated as the following rules:

Spec1-0 : When clk_0 has been selected, then
clk_0 will be derived at some point,
Spec1-1 : When clk_1 has been selected, then
clk_1 will be derived at some point.

(Here, we assume that clk_0 and clk_1 do not ever stop.) We can express these specifications in modal logic as:

$$\Box\,(sel1 = 0) \quad \Rightarrow \quad \Diamond\,\Box\,(out = clk_0)\ , \tag{3}$$

$$\Box\,(sel1 = 1) \quad \Rightarrow \quad \Diamond\,\Box\,(out = clk_1)\ . \tag{4}$$

Finally, our verification is achieved by deriving specification (3) and (4) from the $CLKSEL$ expression (2) and the axioms. The final of the deduction is as follows:

$$\begin{array}{lr} \forall n \in \mathbb{N}\,.\,\textcircled{0}\textcircled{1}\bigcirc^n\,(en_0 = 0 \ \&\ en_1 = 1) & \\ \Rightarrow\ \textcircled{0}\textcircled{1}(\forall n \in \mathbb{N}\,.\,\bigcirc^n\,(out = clk_1)) & \text{(by the top of (2))} \\ \Rightarrow\ \textcircled{0}\textcircled{1}\Box(out = clk_1) & \text{(by } Axiom\,VN1) \\ \Rightarrow\ \bigcirc^m\Box(out = clk_1) & \text{(by } Axiom\,VN2) \\ \Rightarrow\ \bigcirc^m\Diamond\Box(out = clk_1) & \text{(by } Axiom\,V2) \\ \Rightarrow\ \Diamond\Box(out = clk_1) & \text{(by } Axiom\,V3.) \end{array}$$

IV. Summary

We proposed a deduction system for theorem-proving verification of multi-clock synchronous circuits, which does not require any constraint on relations between clocks. In the future, we hope to implement our method on a theorem proving language. On which, proofs are rigorously verified, and the entire method will become completely formal.

References

[1] E. M. Clarke, D. Kroening, and K. Yorav, "Specifying and verifying systems with multiple clocks," in *2012 IEEE 30th International Conference on Computer Design (ICCD)*. IEEE Computer Society, 2003, pp. 48–48.

[2] J. Joyce and G. Birtwistle, "Proving a computer correct in higher order logic," 1985.

[3] T. F. Melham, *Higher order logic and hardware verification.* Cambridge University Press, 2009, vol. 31.

[4] P. Blackburn, M. De Rijke, and Y. Venema, *Modal logic.* Cambridge University Press, 2002.