FSM based Checker for Sequential Circuits

Ilya Levin and Vladimir Ostrovsky
Tel Aviv University
ilia1@post.tau.ac.il

Abstract

This paper presents a method for logic synthesis of Totally Self-checking Checkers for sequential circuits in a form of a Finite State Machine. This approach allows sufficient reduction of both the number of redundant bits of the sequential circuit, and the number of inputs of the checker.

1. Introduction

Major difficulties in designing of self-checking devices are related to complexity of decoding (i.e. verification that a given output vector is a codeword). Output vectors of devices are usually encoded by a code detecting unidirectional errors [1]. Implementing of checkers as combinational circuits is an acceptable practice since any error has to be detected immediately when occurs, while the presence of a memory seems to increase a fault latency. In this paper, we try to revise this stereotype. We develop a method for synthesis of a checker for sequential circuits (SCs) in a form of circuit with a memory without any increasing of the fault latency but with a sufficient reduction of both the number of redundant bits of the SCs and the number of checker's inputs.

Our approach is based on a fact that SC being in a particular state is able to produce a limited subset of code words, and not a whole set as it is assumed in the traditional checking scheme. This subset consists of output vectors that can be produced on all transitions from the present particular state. It allows implementing the checker in a form of Finite State Machine (FSM), states of which are in a one-to-one correspondence with states of the SC. Such a constructed checker enables checking exactly those output vectors that can be produced on transitions from the present state of the SC.

Usually, only small part of output variables (sufficient variables) can take both "one" and "zero" values on transitions from a certain state. Most of the output variables (insufficient variables) are equal to zero on each of transitions from this state.

Based on the above-mentioned properties, we propose: 1) separate coding each of a subset of SC output vectors and thus reducing of the number of redundant bits; 2) utilizing the same checker's inputs for introducing of different outputs of the SC thus reducing the number of the checker's inputs.

2. Self-checking scheme

The proposed scheme of the self-checking SC consists of: 1) an initial SC, 2) an *output compressor* that transforms SC's outputs into the checker's inputs and 3) an FSM based checker

We will illustrate the proposed approach by an example of designing a checker for SC defined by its State Transition Graph (STG) shown in Table 1. In this table: a_m and a_s are a present and a next state correspondingly, $X(a_m,a_s)$ - transition function, i.e. a Boolean function which is equal to one when SC makes the transition from state a_m to state a_s. $Y(a_m,a_s)$ – output vector generated on the transition of the SC from a_m to a_s.

2. 1. Outputs compressor

Define the outputs compressor for the initial SC. Let $Y(a_m)=\{y_{m1},\ldots, y_{mF_m}\}$ is a set of output variables on all transitions from state a_m. In our example: $Y(a_1)=\{y_2,y_3, y_4\}$;

$Y(a_2)=\{y_1,y_5\}$; $Y(a_3)=\{y_1,y_5,y_6\}$;
$Y(a_4)=\{y_6,y_7\}$; $Y(a_5)=\{y_4,y_7\}$.

$y_n \in Y$ is replaced with the $z_g \in Z$ so that if the SC is in state a_m then $z_g = y_n$. For our example output compression $Y \to Z$ is defined as follows: $z_1 = \phi(y_1,y_2)$; $z_2 = \phi(y_3,y_5,y_7)$; $z_3 = \phi(y_4,y_6)$, where ϕ - is a function $1-out-of-n$.

Tab.1.STG of theSC

a_m	a_s	$X(a_m,a_s)$	$Y(a_m,a_s)$
a_1	a_2	$x_1\bar{x}_2$	0110000
	a_4	$x_1\bar{x}_2x_3$	0001000
	a_1	$x_1\bar{x}_2\bar{x}_3$	0010000
	a_3	$\bar{x}_1$	0100000
a_2	a_4	1	1000100
a_3	a_1	$x_4\bar{x}_1$	1000010
	a_4	$\bar{x}_1$	1000100
	a_4	$\bar{x}_4$	1000100
a_4	a_5	$\bar{x}_2$	0000011
	a_1	$\bar{x}_2$	0000000
a_5	a_1	1	0001001

2. 2. FSM based checker

Define the FSM based checker.

1. Set B of states of the checker is in the one-to-one correspondence with the set of states of the initial SC. In our example: $B=\{b_1, b_2, b_3, b_4, b_5\}$.

2. Set I of input variables of the checker consists of: 1) set Z; 2) a set of the unordered coding [2] variables R and 3) an additional variable z_0 that takes the value "one" when all insufficient output variables of a present state are equal to zero. $I = Z \cup R \cup z_0$. In our example: $I=\{z_1, z_2, z_3, r, z_0\}$. z_0 is implemented by the output compressor as follows: $z_0 = B_1 z_0^1 + \ldots + B_M z_0^M$, where B_m is a product of state variables corresponding to state b_m; $z_0^m = 1$, if all insufficient output variables of a state a_m take a value "zero". For example: $z_0^1 = \overline{y_7 + y_5 + y_6 + y_7}$.

3. A set of output variables of the FSM checker forms an error vector signal $E=\{e_1, e_2\}$.

4. Transition of the checker from state b_m to state b_s occurs according to input vector $I(b_m,b_s)$, with producing an output vector $E(b_m, b_s)$.

The STG of the checker is shown in Table 2.

Tab. 2. STG of the FSM based checker

b_m	b_s	$Z(b_m, b_s)$	$R(b_m,b_s)$	$E(b_m,b_s)$
b_1	b_2	$\bar{z}_1\bar{z}_2z_3z_0$	$\bar{r}$	01
	b_4	$\bar{z}_1z_2\bar{z}_3z_0$	$\bar{r}$	01
	b_1	$\bar{z}_1z_2\bar{z}_3\bar{z}_0$	r	10
	b_3	$\bar{z}_1z_2\bar{z}_3\bar{z}_0$	r	10
b_2	b_4	$z_1\bar{z}_2z_0$	$\bar{r}$	01
b_3	b_1	$z_1\bar{z}_2\bar{z}_3z_0$	$\bar{r}$	01
	b_4	$z_1\bar{z}_2z_3z_0$	$\bar{r}$	10
b_4	b_5	$\bar{z}_4\bar{z}_2z_0$	$\bar{r}$	01
	b_1	$\bar{z}_4\bar{z}_2z_0$	r	10
b_5	b_1	$z_4z_2z_0$	$\bar{r}$	01

We assume that on transitions that are not mentioned in the table, output vector $E(b_m,b_s)$ is equal to 00.

In our example the total number of inputs for the proposed FSM based checker is equal to 5 while the traditional checker has 10 inputs.

3. Conclusion

We have proposed a novel technique for synthesis of self-checking checkers for sequential circuits. The proposed technique is based on an architecture comprising the outputs compressor and the FSM based checker. Unordered coding in this case requires a small number of coding bits. A specific way of output's compression allows reducing the number of checker's inputs.

4. References

1. P. Lala. Self-checking and Fault-Tolerant Digital Design. Morgan Kaufmann Publishers, 2000.

2. J. Smith, "On Separable Unordered Codes", IEEE Transaction on Computers, Vol. C-33, No. 8, August 1984.