ISMM

IASTED

Proceedings of the IASTED International Conference

MODELLING AND SIMULATION

Pittsburgh, PA, U.S.A.
May 10-12, 1993

Editor: M. H. Hamza

A Publication of
The International Association of Science and Technology for Development - IASTED

ISBN: 0-88986-173-0
ISSN: 1021-8181

IASTED
ANAHEIM • CALGARY- ZURICH

R. S. Sandige, B. M. Wilamowski "Logic Hazard Free Synthesis Tool", proceedings of the IASTED International Conference on Modeling and Simulation, Pittsburgh, Pennsylvania, pp. 171-174, May 10-12, 1993.

LOGIC HAZARD FREE SYNTHESIS TOOL

Richard Sandige Bogdan Wilamowski

Department of Electrical Engineering
University of Wyoming
Laramie, WY 82071

KEY WORDS

Logic Hazard, Glitch,
Asynchronous, Synthesis Tool

ABSTRACT

This paper presents a new software tool for the synthesis of logic hazard free circuits. Spurious output signals, called glitches, in combinational logic circuits are due to different delays in signal paths and changes in input signals. Such a condition, which can lead to a glitch, is called a hazard. Logic hazards are due to a single input change [1]. These hazards can be eliminated by product terms called logic hazard covers (LHCs).

The software synthesis tool presented in this paper, when provided with a minimized SOP (sum of products) Boolean function, provides the appropriate logic hazard covers for the function. Elimination of logic hazards in combinational logic functions results in logic hazard free circuits. Logic hazard free circuits are essential in the design of high speed fundamental mode (level mode) asynchronous logic circuits. The software synthesis tool being presented provides a convenient technique for identifying required logic hazard covers ([2], [3]) in such asynchronous designs.

INTRODUCTION

The realization of a logic hazard free function results in the removal of spurious outputs or glitches in the resulting circuit. This synthesis tool allows up to fifty logical variables to be used for a minimized SOP Boolean function. Reapplying the synthesis tool with all logic hazard covers included in the function, verifies the removal of all logic hazards. This tool provides designers with a software method of generating logic hazard free functions for very large as well as smaller logic circuits.

GENERATION OF A LOGIC HAZARD FREE FUNCTION VIA A KARNAUGH MAP

To illustrate the process of obtaining a logic hazard free function, consider the following Boolean specification represented by Equation 1.

$$F(A,B,C,D) = \Sigma\, m(0,1,2,5,7,10,14,15) \qquad \text{Eq 1}$$

Plotting this specification on a four variable Karnaugh map results in the plotted map illustrated in Figure 1.

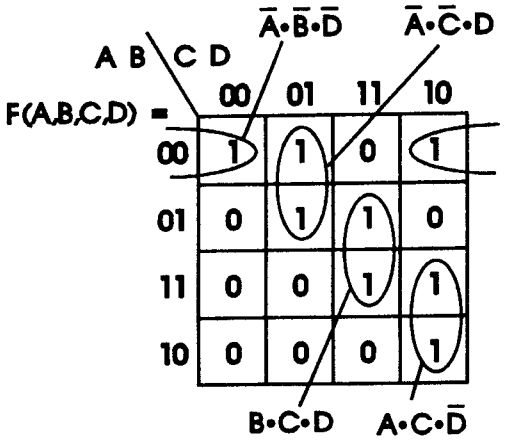


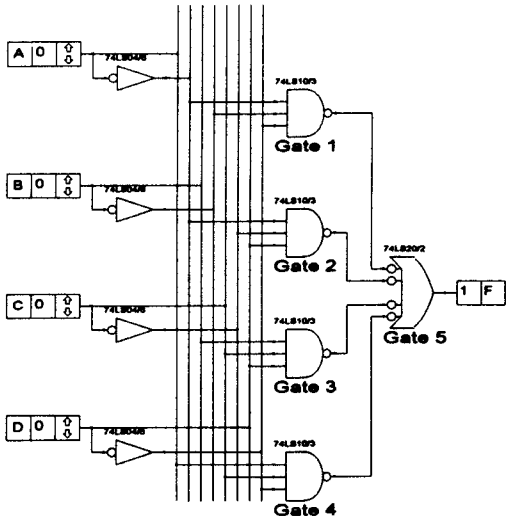
Figure 1.
Karnaugh map for Boolean specification represented by Equation 1.

Equation 2 shows one of the two possible minimized SOP equations for the given Boolean specification.

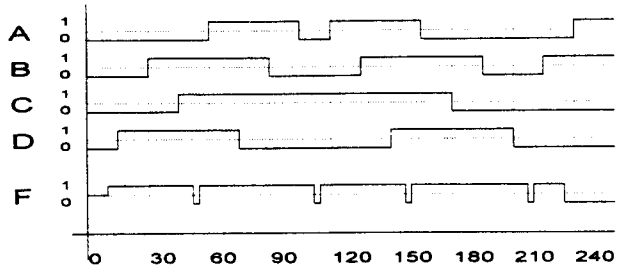
$$F = \bar{A}\cdot\bar{B}\cdot\bar{D} + \bar{A}\cdot\bar{C}\cdot D + B\cdot C\cdot D + A\cdot C\cdot\bar{D} \qquad \text{Eq 2}$$

A static 1 logic hazard [4] can be detected by observing cells in the map where only one variable changes from 0 to 1 or from 1 to 0 as in the case of cells 5 and 7. For example, the function in the Karnaugh map is represented by inputs ABCD and as these inputs change from 0101 to 0111 or from 0111 to 0101 (variable C changes from 0 to 1 or from 1 to 0) a logic hazard exists in the circuit, that is, output F can change from 1 to 0 then back to 1. By covering (linking) the ones in cells 5 and 7 to form the redundant prime implicant or hazard cover $\bar{A}\cdot B\cdot D$ and then logically adding this product term to Equation 2, the static 1 logic hazard that exists between cells 5 and 7 can be effectively eliminated.

Figure 2a shows Equation 2 in schematic form using the schematic capture software package B^2 Logic [5]. The simulation timing diagram for the circuit is shown below the schematic in Figure 2b.



(a)



(b)

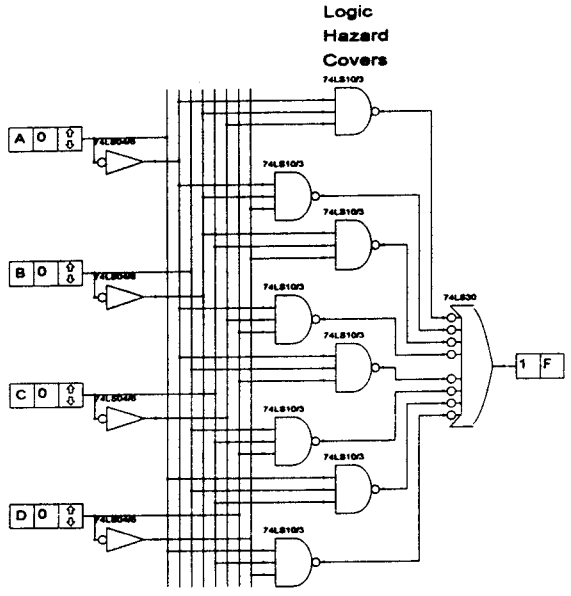
Figure 2.
(a) Schematic for Equation 2 using the software package B^2 Logic. (b) Simulation timing diagram showing logic 0 glitches.

By inspecting the timing diagram one can see that a logic 0 glitch occurs when variable C changes from 0 to 1 (the first glitch in the timing diagram) if the propagation delay $\Delta t2$ is less than $\Delta t3$. Changing the propagation delay $\Delta t2$ such that it is greater than $\Delta t3$ causes a logic 0 glitch to occur when variable C changes from 1 to 0. The delays $\Delta t2$ and $\Delta t3$ represent the propagation delays that exist from the input signal line C (in this case) to the outputs of gate 2 and gate 3 respectively.

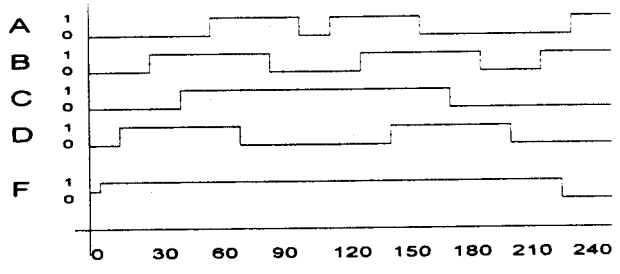
Logic hazards also exist between cells 0 and 1, cells 14 and 15, and cells 10 and 2 in Figure 1. Logic hazard covers for these logic hazards are product terms $\overline{A}\cdot\overline{B}\cdot\overline{C}$, $A\cdot B\cdot C$, and $\overline{B}\cdot C\cdot\overline{D}$ respectively. Equation 3 shown below has the same functionality as Equation 2, only the logic hazard covers have been added to make Equation 3 a logic hazard free (LHF) function.

$$F_{LHF} = \overline{A}\cdot\overline{B}\cdot\overline{D} + \overline{A}\cdot\overline{C}\cdot D + B\cdot C\cdot D + A\cdot C\cdot\overline{D} + \overline{A}\cdot B\cdot D + \overline{A}\cdot\overline{B}\cdot\overline{C} + A\cdot B\cdot C + \overline{B}\cdot C\cdot\overline{D} \qquad \text{Eq 3}$$

The logic hazard covers in Equation 3 insure that the output F_{LHF} will stay at a value of 1 when a single input variable changes from 0 to 1 or from 1 to 0 or stay at a value of 0 when a single input variable changes from 0 to 1 or from 1 to 0 ([4], [6]). Figure 3a shows the circuit for the function F_{LHF} in Equation 3.



(a)



(b)

Figure 3.
(a) Schematic for Equation 3 using the software package B^2 Logic. (b) Simulation timing diagram showing removal of all glitches.

For the same propagation delays represented in the timing diagram in Figure 2a, the simulation timing diagram in Figure 3b shows that the logic 0 glitch between cells 5 and 7 has been removed. Further testing also shows that for function F_{LHF} all logic 0 and logic 1 glitches resulting from static 1 and static 0 logic hazards have been removed by the logic hazard covers in Equation 3.

GENERATION OF A LOGIC HAZARD FREE FUNCTION VIA THE LOGIC HAZARD COVER ALGORITHM

For large or small functions, a systematic way of generating logic hazard covers for minimized SOP functions is represented by the logic hazard cover algorithm illustrated in Figure 4. The logic hazard cover algorithm allows one to identify all logic hazard covers without using a Karnaugh map. In general, Karnaugh maps are useful for perhaps up to six variables and then they become very time consuming to draw as well as difficult to use. The advantage of using the logic hazard cover algorithm is the fact that it can be used for any number of variables either by hand calculation or by computer synthesis.

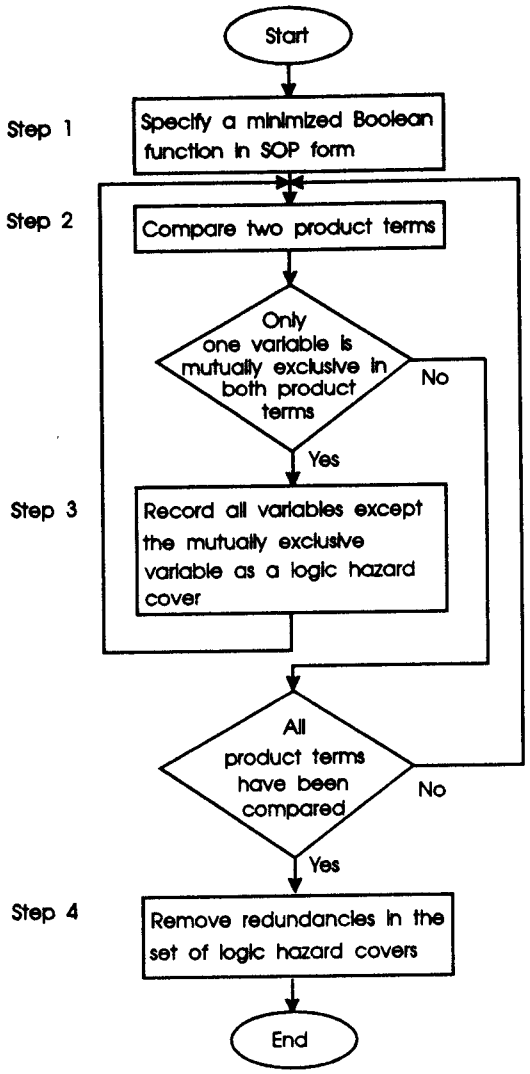


Figure 4.
Logic Hazard Cover Algorithm

Figure 5a shows the hand calculation technique for computing the logic hazard covers for function F (Equation 2) while Figures 5b&c show the results of using the logic hazard synthesis tool (computer synthesis) to calculate the logic hazard covers for function F. The logic hazard synthesis tool is much faster and easily confirms both the hand calculations and syntheses using Karnaugh maps. To provide additional conformation, the logic hazard synthesis tool can be rerun with logic hazard covers appended to the minimum SOP part of the function. The logic hazard synthesis tool will indicate any logic hazard covers that are missing.

$$F = \bar{A}\cdot\bar{B}\cdot\bar{D} + \bar{A}\cdot\bar{C}\cdot D + B\cdot C\cdot D + A\cdot C\cdot\bar{D}$$

$$LHC = \bar{A}\cdot\bar{B}\cdot\bar{C}$$

none

$$LHC = \bar{B}\cdot C\cdot\bar{D}$$

$$LHC = \bar{A}\cdot B\cdot D$$

none

$$LHC = A\cdot B\cdot C$$

(a)

```
/A/B/D
/A/CD
BCD
AC/D
```

(b)

```
*   input data:
/A/B/D
/A/CD
BCD
AC/D
*  logic hazard covers:
/A/B/C
/BC/D
/ABD
ABC
```

(c)

Figure 5.
(a) Hand calculation technique for computing logic hazard covers for function F (Equation 2). (b) File generated using a text editor with the extension .LIN (Logic INput) for function F. (c) Result of using the logic hazard synthesis tool to calculate the logic hazard covers for function F.

CONCLUSIONS

A special software synthesis tool has been written and simulations have been run using B^2Logic to verify the removal of logic hazards using logic hazard covers. The software synthesis tool verifies that logic hazards exist in hazardous circuits (circuits incorporating logic hazardous equations) by the generation of logic hazard covers. The synthesis tool may then be rerun using the logic hazard covers to verify that logic hazards do not exist in the final function with the logic hazard covers logically added to the function. A schematic capture tool that incorporates a timing simulator can also be used to verify that all the hazards have been removed from the logic hazard free function. The software logic hazard free synthesis tool presented in this paper (for PC and PC compatibles) is available from the authors at no charge.

REFERENCES

[1] R. S. Sandige, *Modern Digital Design, McGraw-Hill*, New York, 1990.

[2] R. F. Tinder, *Digital Engineering Design A Modern Approach*, Prentice Hall, New Jersey, 1991.

[3] C. Innes, "Avoiding Programmable Logic Hazards", National Anthem, National Semiconductor, January/February 1988, p. 5.

[4] S. H. Unger, *The Essence of Logic Circuits*, Prentice Hall, New Jersey, 1989.

[5] J. A. Engelbert, *User Manual for B^2 Logic v2.2*, Beige Bag (B^2) Software of Ann Arbor, Michigan, 1990.

[6] E. J. McCluskey, *Logic Design Principles with Emphasis of Testable Semicustom Circuits*, Prentice Hall, New Jersey, 1986.