A NEW METHOD OF ANALYSIS AND SYNTHESIS OF ASYNCHRONOUS SEQUENTIAL CIRCUITS

Mihai TIMIŞ

Technical University "Gh. Asachi", Iaşi, România
Automatic Control and Computer Science Faculty
Computer Science department
mtimis@cs.tuiasi.ro

INTRODUCTION

In present paper, I propose a new method for analysis and synthesis of an asynchronous sequential system. The main idea consist on building a CK drive in signal, using the transition system function graph. It contains a clock signal who will drive on the states of the sequential system. To control sequential system, it is used a clock signal who contain a set of Master – Slave D latches circuits, logic gates. It was built from the sequential system transition matrix. All the system components are asynchronous, it has no clock signal.

For system transitions control actions, it use a built in local clock signal who will drive in the digital system. Using the system fluence graph and fluence table will be build an asynchronous clock signal. It is implemented using logic gates like AND, OR, XOR.

Using this method, the asynchronous sequential system can be easy implemented, and it works free of logic hazards.

METHOD DESCRIPTION

A such control system using clock drive in signal on asynchronous sequential system is presented in figure 1. It contains a combinational logic system, dates storage elements – D latches circuits, input signals – input1, input2……inputN, output signals – output1, output2……outputM, states variables – s1,s2….sk. The combinational logic system used by the drive in clock signal can be named like synchronisation unit, it is used for control states mode of the digital asynchronous system. Unlike sequential synchronous systems, the states of an asynchronous sequential systems can be changed only when a new set of input signals was generated. So, the clock drive in signal depends only about the states and inputs signals of asynchornous sequential digital system.

The clock signal is inversed, 0 or 1, initally it isn't stored on to some logic device. It controls the latch memory circuits, latch1 and latch2. Every state variable contain one Master-Slave latch1 – latch2. When the clock signal is high, the states variables are stored on latches.

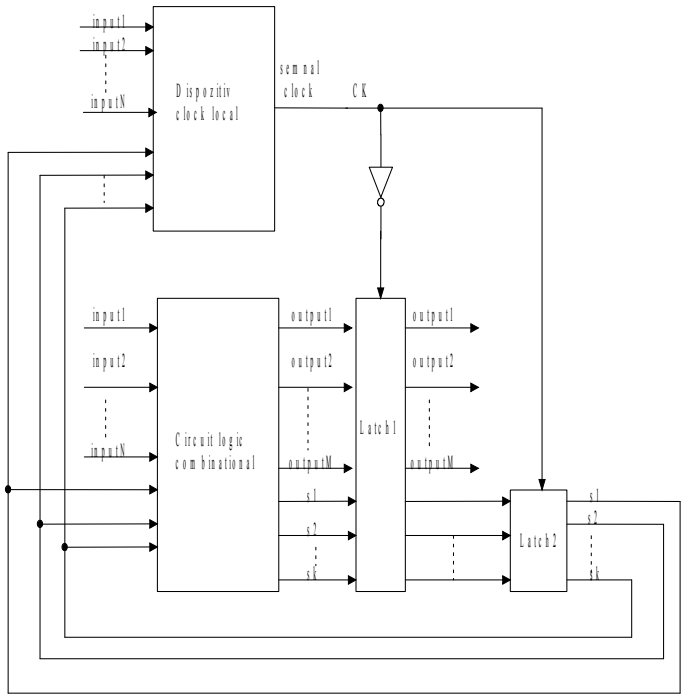

Figure 1. System Description

When the clock signal is low, the latch1 circuits are comanded to hold the output signals values and the states signals of the asynchronous system. When the CK signal become High, the Master latch1 circuits is disconnected, and the Slave latch2 circuit is available. It take the system states values, memorize them and send on feedback wire to clock combinational logic circuit. Every system states will contain output values, the system process the input signals from the transition system matrix table. A Master – Slave D latch2 circuit is show on figure 2.

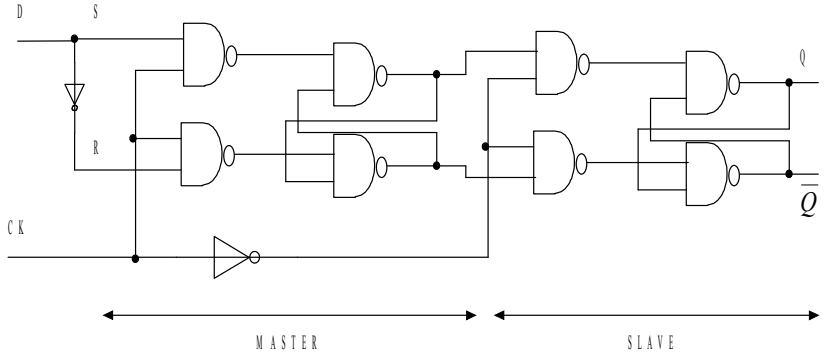

Figure 2. Master Slave D Latch

It needs a way to modify the output signals only on transition of the clock CK signal.

- Edge activation, the inputs signals can be modified on the others time periods, the output signals are not affected by this.
- On positive edge activation, the transition command is generated by 0 -> 1 logic transition.

The asynchronous digital system runs very good when a time setup period is consider and the input signals can't be changed. Also the time hold period is consider, the input signals can't be changed. These time periods are on when the CK signal is high.

On this paper, I propose a system design using a Master-Slave circuit. First, the Master circuit capture the input signals on to the positive edge of the CK signal, second, the Slave circuit uses the output of the Master latch on to the negative edge of the CK signal. The output signal of the latch circuit will be modified, on the negative edge of the CK input signal.

DIGITAL SYSTEM EXAMPLE

Let's consider the sequential system describe by the fluence graph from figure 3. Using the method explained above, the asynchronous sequential system can be implemented like on figure 4. Every output signal is feed on a D latch circuit, the latch1 circuit store the values of the output signals of asynchronous sequential digital system, while the latch2 circuits are blocked. Every state variable is stored on a Master-Slave D latches circuits. It is drive by the CK input signal only to a new state transition.

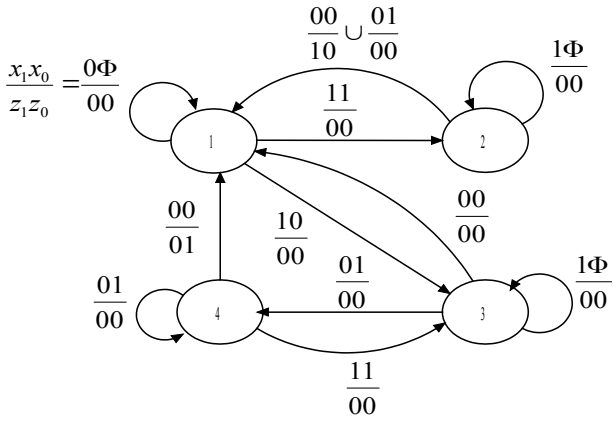

Figure 3. System Fluence Graph

The CK input signal is implemented with logic gates, it's unlached. When the latch1 is drive on, the CK signal is low, when the inputs signals show a new state transition, the CK input signal goes high, then low, like on system fluence table. If the system hold it's state, the CK input signal remain low.

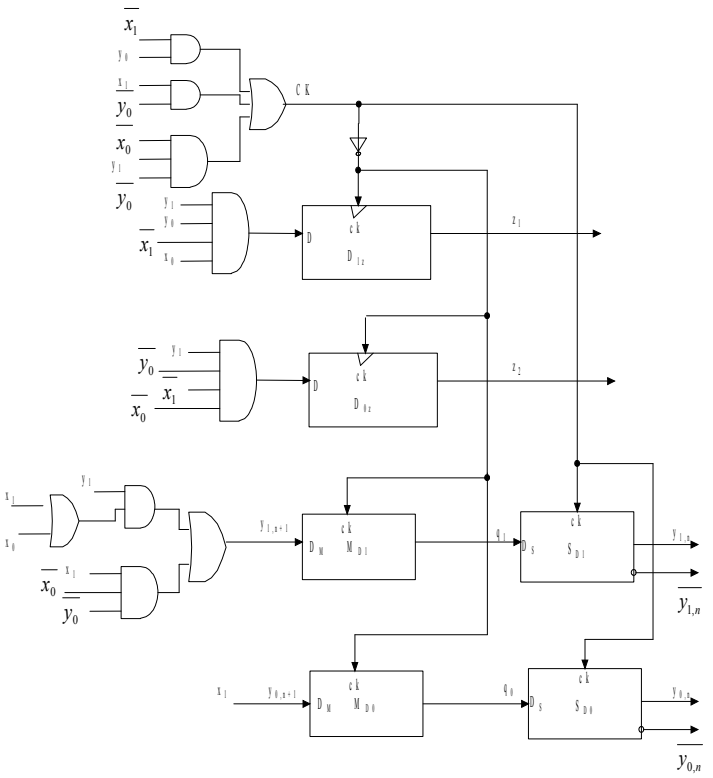

Figure 4. System Implementation

Initially, the CK input signal is low, input signals, state signals and output signals are on a low logic level. The latch1 circuits are transparent, the data flow through them. It needs to avoid the appearance of the hazard situations on output signals. When the x_1x_0 became high, the z_1z_0 outputs remains low until all the input signals are fully generated.

From the fluence graph it's observed that the output signals remains on a low logic level, the system transit on the 2 state. When the $x_1x_0 = 00$, on 2 state, the output signals remains low untill all the input signals are generated, when $z_1z_0=10$, the system back to the 1 state. The $z_0=1$ when the system is on the 4 state, the input signals are $x_1x_0 = 00$. The figure 4 show the block design of the asynchronous sequential digital system.

CONCLUSION

When the system remains on a state, hold that state, the CK input signal remain on a low logic level. Also, the CK input signal is generated only when a new transition from a state to another is started, like on fluence table, if no transition is started, the CK signal remain low.

The output signals are generated when the CK input signal is low. The logic implementation of the CK input signal is implemented with gates.

Using the local clock CK the system run asynchronous, but on the same time his transitions are controlled by it.

REFERENCES:

[1] Alexandru Valachi, Radu Silion, Mihai Timis. "Improvement of FSM Synthesis Using MSI and LSI Circuits", 8th International Symposium on Automatic Control and Computer Science - SACCS, Iasi, Octomber 2004.

[2] Calin Monor, Mihai Timis. "Synthesis Of Reconfigurable Sequential Digital Devices", 7th International Symposium on Automatic Control and Computer Science - SACCS, Iasi, Octomber 2001.

[3] Mihai Timiş. "Analiza şi Sinteza Dispozitivelor Numerice. Aplicații", Performantica, Iaşi, 2003.