DESIGN OF ASYNCHRONOUS AUTOMATA

Mihai Timis[1], Alexandru Valachi[2]
[1] "Gh.Asachi" Technical University of Iasi,
Automatic Control and Computer Engineering Faculty
Computer Engineering department, mtimis@cs.tuiasi.ro
[2] "Gh.Asachi" Technical University of Iasi,
Automatic Control and Computer Engineering Faculty
Computer Engineering department, avalachi@cs.tuiasi.ro

Keywords: Asynchronous Digital Systems Design, Delay Insensitive Gates, Self Timed Circuits, Handshake Protocol.

Abstract: The use of asynchronous sequential circuits has brought many advantages to system development, given the following examples:

- Signal interface protocols (for example SCSI)
- An asynchronous sequential system can be built from a number of modules, by interfacing them.
- For asynchronous systems, output signal are generated instantly, without waiting the clock signal for synchronization.
- The absence of the clock signal reduces the EMI(Electro Magnetic Interferences).

1.INTRODUCTION

The design of asynchronous circuit is more difficult than the design of synchronous ones considering the propagation delay of signal trough elementary circuits. To simplify the analysis and the design of asynchronous systems, their operation is considered to be in fundamental mode. Operating in fundamental mode means that the input signals can change their state only if the circuit is in a stable state, meaning that no state variable is modifying. In fundamental mode all inputs are considered to be levels. Sometimes there is considered the impulse mode, where the inputs can provide impulses (level variations).

An asynchronous sequential system does not contain a clock signal for synchronization. Every part of the system synchronizes locally with neighboring circuits. This deters clock slew periods.

As it was mentioned before, in order to reduce power consumption, CMOS circuits are used, because the power requirements for them are lower in idle mode.

Some recent implementations show the potential of systems implemented unsing asynchronous circuits:

- 80C51 is a full asynchronous microcontroller implemented by Philips Research and Philips Semiconductor [1]; the system consumes only a fourth of the power consumed by a corresponding synchronous system. This microcontroller can be found in some pagers produced by Philips.
- A asynchronous instruction decoder, produced by Intel, whose performance is boosted 3,4 times that the synchronous counterpart [2].
- Amulet2e circuit, a microprocessor produced by the Manchester University [3]. This includes an asynchronous ARM processor with level 1 cache. It has better performance than in synchronous variant, the dissipated power being 1 microwatt when the processor is in idle state.
- A digital filter, part of a synchronous sequential system used for hearing aid, was implemented totally asynchronous by the Dutch Technical University and Oticon, Inc. [4]. The asynchronous system dissipates five times less power than the synchronous system.
- RISC processor, based on MIPS-X architecture [5].

Other examples include: a asynchronous sequential system used for solving equations [1], asynchronous processor [4], Reed-Solomon error detector used in digital audio systems [5], high performance divisor circuit[16], infrared communications processor [3], parallel processing router processor [2].

Although synchronous sequential systems are widely spread, there is a series of applications that deter their use:

- The circuit has input variables that can change at any given time and cannot be synchronized with the clock signal
- If a system in complex enough and the time response of the logic elements is low, it can be hard to assure that the clock signal propagates simultaneous to all the flip-flops
- The applications has to be fast enough and the system cannot tolerate loss of time due to the clock wait states

In such cases, the use of asynchronous sequential system is required. They are not synchronized by a general clock signal; when an input modifies, the state of the circuit may change almost instantaneously.

- All impulse inputs must have a minimum time period, in order to command flip-flops
- The time interval between two successive impulses must be long enough for the system to respond accordingly.

2. ASYNCHRONOUS SEQUENTIAL SYSTEMS CLASSIFICATION

A delay insensitive type of circuit functions correctly regardless of the propagation delay trough circuits and wires. This concept was introduced by Clark and Molnar [1]. A speed independent type of circuit functions correctly regardless of the propagation delay trough circuits. The wires have specific propagation time, this concept was introduced by David Muller [3].

A self timed type of circuit consists of a collection of elements internally temporized, where the propagation delay trough wires are known or negligible. Usually, they are built as speed independent circuits, using time informations [2]. The above circuits function in input/output mode.

The general category is represented by the asynchronous circuits [4]. These do not have a global clock signal, functioning using an internal timing. This category includes latches and flip-flops [5], that have setup and hold periods.

3. BASIC CONCEPTS USED IN ASYNCHRONOUS SEQUENTIAL SYSTEMS SYNTHESIS DATA COMMUNICATION

When two systems communicate, the receiver must know the moment when the data is available. In a synchronous system, the transmitter puts out the data synchronously with a clock signal. The data must not violate the constraints imposed by the setup and hold time. In asynchronous systems, there is no clock signal, that is why the receiver must be signaled upon start and stop of data transmission. This can be realized by introducing an extra signal named Ack. In asynchronous systems, between two blocks, always one is active while the other one is passive and vice versa. The active block initiates the communication, while the passive one responds to the actions started by the active one.

A push channel represents a data channel where data is sent by the active block. The active block sends a confirmation signal to the passive block, indicating the fact that data can be read.

A pull channel is a data channel where the active block receives data from the passive one.

The concept used in the design of asynchronous sequential systems is named handshake and is represented in figure 1.2.
The data transfer is started by the source by activating the data signals named Data and Strobe, while the recognition is realized by the recipient by generating a Data Request. If the transmitter wishes the receiver to process data, it sends data on the Data Bus and activates the Strobe signal. When the recipient receives the data it activates the Data Request signal.

The Data Request and Strobe signals are represented by transitions or events. If a parasite signal appears, it can be noticed as an event, and the circuit can enter in a erroneous functional state. The data transfer can be initated by the source or the destination.
In the case of a transfer initiated by the destination, figure 1.2.

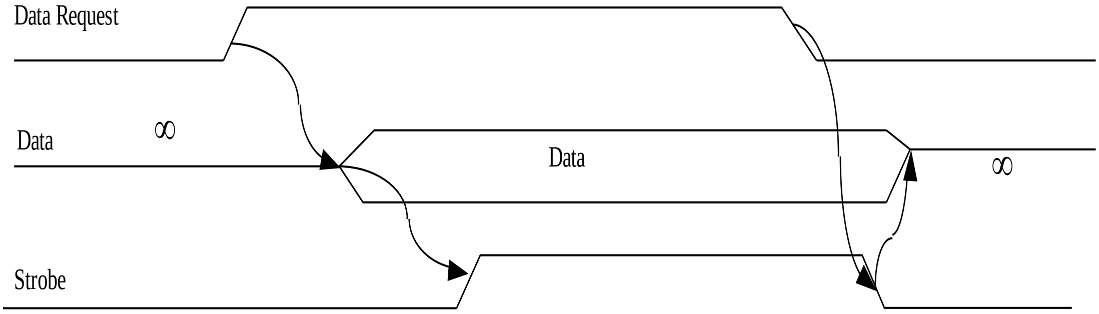


Figure 1.2. Data request and strobe

3.1. CIRCUIT MODELS

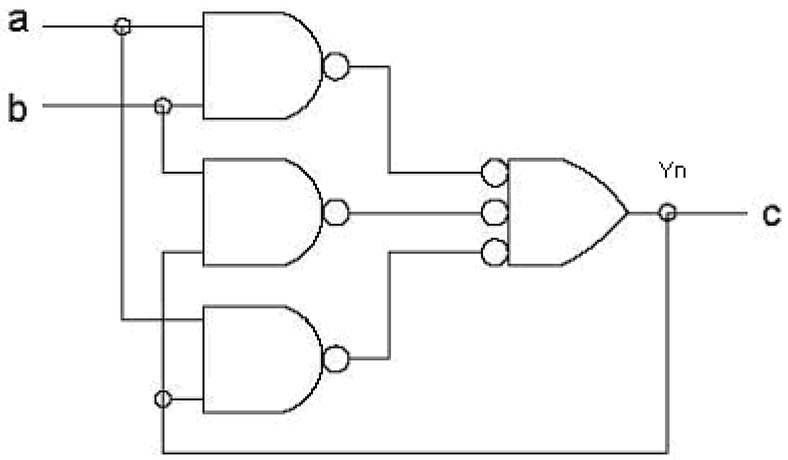
A complete circuit represents a closed system composed of two parts: the circuit and the environment around it. The model describes the way delays are associated with the logic gates. The environment model describes how it interacts with the circuit.

3.2. DELAY MODEL

The delay is an important factor concerning sequential systems [2]. There are two types of delays: stray delay – it inherits the physical properties of every circuit; delay elements – they are added by the designer. The stray delays appear at logic gates as a result of their physical properties, impedance, capacitance, propagation delays in gates and wires.
The way a circuit interacts with the environment can be classified in two different categories. The first is known as input-output mode, at it assumes that time constraints are not imposed. In the opposed case, if time constraints are imposed, the circuit functions with timed circuits. The circuit models describe the working mode imposed by the two delay types: stray delay and delay elements. Also we have handle delays and bounded delays. In the case of unbounded delays, any delay can have an arbitrary value for the necessary propagation time. For bounded delays, the values are chosen from an interval. Circuits that use inputs and outputs for environment interaction use the concept of unbounded delays. Circuits that use a specific propagation time when interacting with the environment, use the concept of bounded delays.
Common to all circuits that function in input-output mode is the fact that we must know the time at which a command is started and stopped. For example, to be sure all receiving circuits have got a message from the transmitter, the receivers must generate an ACK signal to the transmitter. This signal shows that the sent signals were received and

executed. Circuits that work in input-output mode can be divided into 4 main types of asynchronous circuits.

A delay insensitive circuit can have arbitrary inputs corresponding to logic gates or wires. A delay insensitive circuit functions correctly regardless of the propagation delay of the wires. A delay insensitive type circuit is composed of modules and the interconnecting wires. Every module must be designed as the delay variation must not influence the stability of the circuit. Due to acknowledgement signal confirmation for every circuit, in the case of a complex system, the bounded logic can be very complex [3]. So, delay insensitive circuits can be made only by using Muller elements and active low logic gates; the truth table is presented in figure 1.3.



a b	c
0 0	0
0 1	C_{n-1}
1 0	C_{n-1}
1 1	1

a b y_n	y_{n+1}
0 0 -	0
0 1 -	y_n
1 0 -	y_n
1 1 -	1

Figure 1.3. C Muller truth table

The equation for the C Muller element is: $c = \bar{a} \cdot b \cdot c_{n-1} + a \cdot \bar{b} \cdot c_{n-1} + a \cdot b$

If we consider c=y_{n+1}, then the equation becomes: $y_{n+1} = \bar{a} \cdot b \cdot y_n + a \cdot \bar{b} \cdot y_n + a \cdot b$

A speed independent circuit functions correctly regardless of the propagation delay of the logic gates.

In the bounded delay model, the delays in the logic gates and in the wires must be less that a maximum considered value. This is the most common used delay type, although

functional problems may appear when considering time constraints.
A self timed element can be implemented as an speed independent one using more types of logic gates, by analyzing their response time. The quasi delay insensitive circuits have arbitrary delays in gates and wires as well [3]. A logic gate is a component that calculates instantaneously the value of a logic function, then after a certain delay assign that value at its outputs. For example a AND gate calculates the logic multiplication of the two input values after a certain delay time. A logic gate is stable if the value of the associated logic function matches the output. Also there is are stable wires, which means it has a stable value for a certain amount of time. a logic circuit is stable when each gate and each wire is stable. It can be said that a circuit functions in fundamental mode if its inputs can be modified only after the circuit is in a stable state. A simple way for classifying combinational circuits is regarding the propagation delay. Every circuit class has a different type of delay. For example a circuit model whose logic gates assume arbitrary delays, is not suited for implementing a asynchronous sequential system. Opposite to asynchronous systems, synchronous systems have no reaction speed dependence. The reliability of a circuit also depends on propagation delays and latencies.

4. CONCLUSIONS

The design of asynchronous circuit is more difficult than the design of synchronous ones considering the propagation delay of signal trough elementary circuits. To simplify the analysis and the design of asynchronous systems, their operation is considered to be in fundamental mode. Operating in fundamental mode means that the input signals can change their state only if the circuit is in a stable state, meaning that no state variable is modifying. In fundamental mode all inputs are considered to be levels. Sometimes there is considered the impulse mode, where the inputs can provide impulses (level variations).

References:

1. Unger, S.H. Sequential Switching Circuits. Wiley Interscience, 1969, ISBN 0-89874-565-9.
2. David L. Dill. Trace Theory for Automatic Hierarchical Verification of Speed Independent Circuits, MIT Press, Cambridge, MA, 1989.
3. Jo Ebergen. A formal approach to designing delay/insensitive circuits.Distributed Computing, 5(3):107-119, 1991.
4. J.E.Hopcroft and J.D. Ullman. Introduction to Automata Theory,Languages and Computation. Addison-Wesley, Reading, MA, 1979.
5. Mihai TIMIS, Alexandru Valachi. SYNTHESIS OF THE ASYNCHRONOUS DIGITAL SYSTEMS USING THE IMPROOVED LOCALLY CLOCK METHOD. Technical University „Gh.Asachi" Iasi, Automatic Control and ComputerScience Faculty, Computer Science departament, 9th International Symposium on Automatic Control and Computer Science Iasi, Romania, 2007.
6. Alexandru Valachi, Mihai Timis, Mirela Danubianu. Some Contributions to Synthesis and Implementation of Multifunctional Registers, 11th WSEAS Int.Conf. on AUTOMATIC CONTROL, MODELLING & SIMULATION (ACMOS'09), Proceedings of the 2nd WSEAS International Conference on Multivariate Analysis and its Application in Science and Engineering Istanbul, Turkey, May 30 - June 1, 2009, ISSN: 1790 5117, ISBN: 978-960-474-
083-3, WSEAS Press, www.wseas.org, page.146-149, Indexed by Thomson ISI.
7. Roth, Charles H. (1999). Fundamentals of Logic Design, West Publishing Company.
8. Al. Valachi, R.Silion, M. Timis – Improvement of FSM Synthesis using MSI and LSI Circuits. Advances in Electrical and Computer Engineering. Academy of Techincal Sciences of Romania, "Stefan cel Mare" University of Suceava, vol 5(12), nr.1/23, 2005.
9. Al.Valachi,M.Chis-Ster – States Coding Program for Wired Sequencers Designed with Shift Registers, 4th International Symposium on Automatic Control and Computer Science, SACCS'93, 29-30 Oct. 1993, Iasi.
10. Viorel Onofrei, Al. Valachi – Systematic Synthesis of Wired Sequencers Using Counters and Registers – MICROCAD'1999 – Miskolc, Hungary.