IMPLEMENTATION OF ARBITER CIRCUIT FOR ASYNCHRONOUS DESIGN

R. T. Vaz da Silva, C. A. Sampaio, M. C. B. Osório, A. I. Reis, R. P. Ribas
Instituto de Informática – UFRGS
CxP.15064, CEP 91501-970 Porto Alegre – RS (BR)
{rsilva, csampaio, osorio, andreis, rpribas}@inf.ufrgs.br

ABSTRACT

The synthesis of logic event blocks represents a difficult task to automatically generate the layout of asynchronous circuits. It occurs because the logic function of these blocks is generally defined by two equations: one gives the zero logic values of the output signal and other the one logic values. When neither zero nor one voltage levels are defined the output signal is kept invariable. A general structure composed of a P-network connecting the output node and the power, an N-network connecting the output to the ground, and an output latch. Simulation results of the arbiter circuit implemented in the proposed structure are presented and compared to the implementation with standard logic gates.

1. INTRODUCTION

The popular synchronous digital design is running into serious problems like electromagnetic radiation, power and clock distribution.

Asynchronous design is the natural alternative for solving these problems. It is a very promising technique for low power circuits. In addition, the data path runs at the maximum speed allowed by the technology and system architecture. Electromagnetic noise is rarely generated by this kind of circuit.

However, asynchronous circuit design is generally more complex than the synchronous version. The absence of a global clock demands the use of local handshakes to communicate the processing blocks. In order to implement such communication protocol, some specific control circuits (logic event modules) are required.

The automatic synthesis of asynchronous has been widely investigated in order to simplify the design task. The synthesis of logic event circuits is a not obvious task because their functionality cannot be generally represented by a single logic function.

The static CMOS latched gate, could be considered to build the logic event circuits. This structure can be synthesized by using CAD tools that automatically generate non-complementary P- and N-networks to connect the output node to power and ground sources, respectively. An output latch is used to maintain the output signal when there is no logic path through the transistor networks, avoiding so the high impedance output state.

In this paper, the logic event arbiter circuit has been implemented as static CMOS logic gates and electrical simulations were carried out in order to verify the functionality and to compare such approach to the conventional CMOS logic gate version. The output latch is studied too.

2. STATIC CMOS LATCHED GATE

The handshake communication in self-timed circuits can be implemented through two- or four-phase transition signaling. In general, there are a greater variety of logic event functions in the two-phase protocol. Some of these functions are: C-Element or Muller Cell, Toggle, Select, Call, Mutex, Arbiter. The arbiter is used to guarantee mutually exclusive action of two requesters. It accepts requests on R1 and R2 and grants one at a time. After receiving the acknowledge signal that the shared resource is given up, the Arbiter grant access to the next requester. The descriptions of the others block can be found in [2].

One way to implement these blocks is using standard CMOS logic gates. Fig. 1a shows this kind of implementation for the arbiter.

Another way to implement such logic event cells trough static CMOS latched gates. Observing the truth table of these cells, it can observed four output states: the zero and one logic levels, the “don’t care” condition and the memorization or holding of output value.

Two logic equations must be extracted from the truth table: the equation to the P-network that defines the zero logic at the output node, while the equation to the N-network that defines the one logic level. When there is no path from power or ground to output node, the output latch holds the output signal unchanged. See Table I and Table II.

The Arbiter was successfully implemented by considering the static CMOS latched gate structure. It is shown in Fig. 1b. It simplifies significantly the automatic synthesis of this specific logic circuit.

Table I – Equations to Arbiter (output G1)

P – network	[(R2 . D2) + (!R2 . !D2)] . R1 . !D1
N – network	[(R2 . D2) + (!R2 . !D2)] . R1 . !D1

Table II – Equations to Call (output True)

P – network	[(D . D2) + (!D . !D2)] . R1
N – network	[(!D . D2) + (D . !D2] . R1

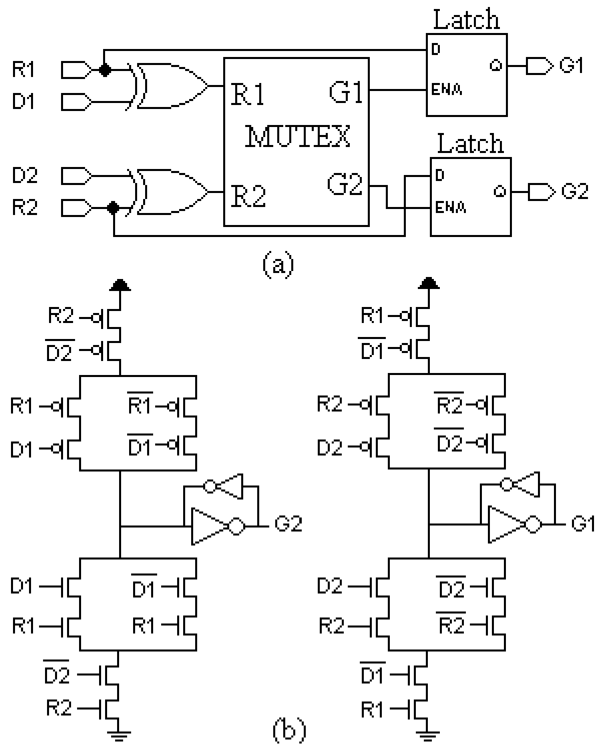

Figure 1. Arbiter: (a) logic gates and (b) latched gate

3. OUTPUT LATCH

Three different storage elements were evaluated for the output latch. See Fig 2.

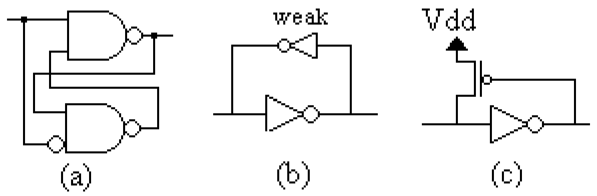

Figure 2. (a) SR latch, (b) SRAM, (c) Inverter-transistor

Table III shows the electrical simulation results obtained for these three latches.

The third latch version presents the best results. However, the positive feedback of this structure is somewhat unstable. The SR latch, in turn, uses more transistors than the others and on the average it does not have better delay results than the SRAM latch. Consequently, the SRAM latch was chosen because of its little layout area with only four CMOS transistors and the delays are similar than SR latch.

Table III - Comparison of evaluated latches.

	Delay (ps)				Transistors
	tdlh	tdhl	tr	tf	number
SR latch	91	296	177	211	10
SRAM	151	127	290	192	4
Inverter-transistor	139	30	227	56	3

4. DESIGN AND PERFORMANCE ANALYSIS

The arbiter function was implemented using the static CMOS latched gate structure. The electrical simulations of these circuits have been carried out in the Cadence environment, taking into account the AMS 0.35µm CMOS process parameters.

Table IV presents the results for the two implementation of the arbiter cell. Timing characteristics, power dissipation and number of transistors have been compared.

The static CMOS latched gate implementation use less transistors and present lower power dissipation. However, the circuit is slower than the standard logic version. The results for the others logic events modules are similar.

Table IV – Results for the two implementations (Arbiter)

		Static CMOS logic gate	Static CMOS latched gate
Delay (ns)	tdlh	1.01	1.25
	tdhl	1.08	2.57
	tr	0.17	0.57
	tf	0.11	1.09
Transistor number		56	40
Power dissipation (uW)		204	192

5. CONCLUSIONS

The logic event circuits cannot be generally represented by a single logic equation, becoming the synthesis of these blocks a not obvious task. A good strategy to overcome this drawback is the implementation of such a kind of circuit through static CMOS latched gates. In this paper, the arbiter has been evaluated taking into account this strategy. The results will improve a CAD tool for asynchronous circuit synthesis, in development by our team.

6. ACKNOWLEDGMENTS

The authors R. T. Vaz da Silva, C. A. Sampaio, M. C. B. Osório are supported by CNPq/Milenio, FAPERGS and CNPq scholarships, respectively. The authors A. I. Reis and R. P. Ribas have a research support from CNPq agency.

7. REFERENCES

[1] Ivan E.Sutherland. *Micropipelines. Communications of the ACM* , Vol. 32, No. 6, Jun. 1989, pp.720-738.

[2] Jens Sparso, Steve Furber. *Principles of asynchronous circuit design – A system Perspective*. Kluwer Academic Publishers. September 2001

[3] Steve Furber. *Computing without Clocks: Micropipelinig the ARM Processor*. Asynchronous Digital Circuit Design, Birtwistle and Davis editores, Springer-Verlag, pp. 211-262, 1995.