Evaluation of 4 bit Self-timed Adder Through Asynchronous Applications

CARLOS A. SAMPAIO, MÁRIO OSÓRIO, GIOVANI H. SARTORI, ANDRÉ I. REIS, RENATO P. RIBAS

Instituto de Informática – UFRGS Av. Bento Gonçalves, 9500 – CxP 15064 CEP 91501-970 Porto Alegre – RS, Brasil
{csampaio, osorio, sartori, andreis, rpribas}@inf.ufrgs.br

Abstract

This paper presents a procedure to evaluate different topologies of Adder circuits. This goal is achieved through asynchronous application circuits that are used in the process of characterization and evaluation. Comparison between them is presented through electrical simulation.

1. Introduction

In this work we propose to study different topologies of sum circuits (DIMS, ECDL, DCVS and Martin approach) by electrical simulation. One of the great problems in validate this kind of circuits is the question about the test vectors and the definition of the worst case latency path that will be used to define the maximum frequency that they had to be supplied

With the construction of the application circuits, the problem of vector testing supplying is avoided, but still persist the problem with critical path and consequently the maximum frequency that the circuit will work. Then, to suppress this problem, this application circuits were constructed using asynchronous techniques.

The use of asynchronous techniques suppresses these problems because of the "absence" of a clock signal that is replaced by a handshake protocol. When initialized the circuit will start to work at the maximum frequency supported, these maximum frequencies only rely on the delays of the circuits that compose the application and in the delays of the handshake protocol and components as will be explained beneath, in Section 3.

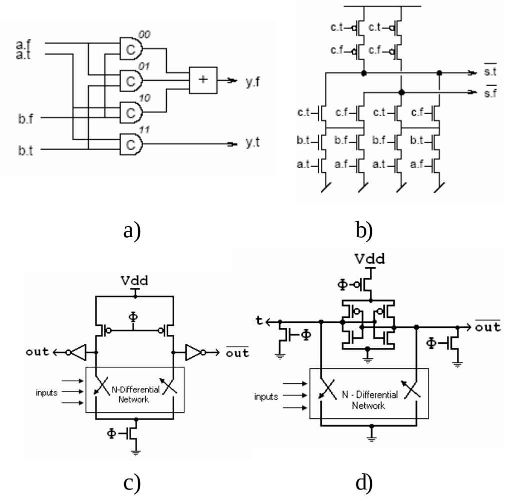
2. Studied Topologies

The topologies presented beneath are all dual rail approaches considered in this work.

This protocol characteristic requires that any combinatorial block must supply the reset logic necessary to the "empty" value generation.

In the DIMS approach [5] the adders are built in a sum of products like structure shown in Fig1 (a) changing the AND gate by one Muller C cell [5]. The circuit resulting of this process is a static implementation where the logic reset is given by the C cell. In Martin [5] design style, the gates are constructed in a complementary way (Fig1 (b)), but also taking attention to some early evaluation considerations, as described in his paper in an algorithmic form. The logic reset is given by the "p" network.

When designing DCVS circuits [1], an "n" network that corresponds to the logical function implementation and his complement that are connected to a pre-charge circuit (Fig1 (c)) is projected. In this kind of approach, an evaluation state that happens when the control signal is in high value occurs. The ECDL circuit [2] is constructed in a similar way (Fig1 (d)), with the "n" network evaluating the function, but, in this case, a pre-discharge scheme where the evaluation period is set when the control signal is low. Both technologies (ECDL, DCVS) have the logical reset supplied by the pre-discharge or pre-charge circuit, respectively and are multiple outputs [4]. That means to say that when constructing recursively functions they could use the same "n" tree for sharing diverse intermediary nodes as outputs.



a) b)

c) d)

Fig1 – Adders structure examples; a) DIMS; b) Martin; c) DCVS; d) ECDL.

3. Application Circuits

The applications circuits constructed, as said before, are asynchronous circuits. All application circuits designed share a common basic structure, with some differences in the data logic manipulation.

The basic structure (Fig2) that is part of all applications circuits presented is composed by three latches, the control circuit and the logic part. The tree latches serve to achieve the correct protocol function as explained in

[5] in the section about Muller pipelines. The control circuit is basically composed by C Cells that are responsible by the handshake between blocks, as described in the same book section as presented above, and by some decisions that depends of the application running, like computation end. The logic circuit is really the part that will differentiate one application from another, these part will be explained more meticulously in the next five paragraphs, because of the changing in his structure depending on the application desired.

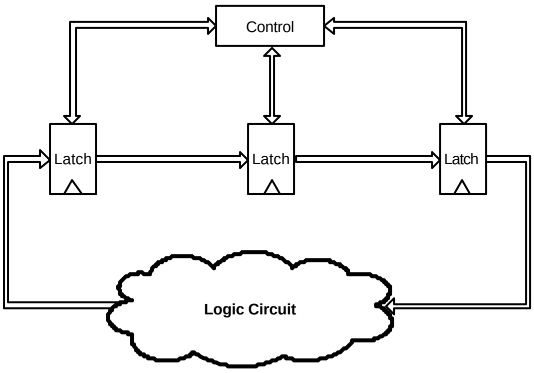


Fig2 – Basic Structure Block Diagram

- Minimum Common Divisor (MCD) – In the MCD the logical area is composed by two adders that made the generic operation “A+A0”, two muxes that are controlled by an block that realizes A<B. The algorithm realized in the logic circuit is: if “A<B” “A<=A+A0” else B<=B+B0.
- Great Common Divisor (GCD) – In the GCD the logical area is composed by two adders that made the generic operation “A-B”, two muxes that are controlled by an block that realizes A<B. The algorithm realized in the logic circuit is: if “A<B” “A<=A-B” else B<=B-A.
- Root Square (RS) – In the Root Square the logical area is composed by two adders that made the generic operation “A + Constant” and one value shifted left (SL). The algorithm realized in the logic circuit is: “A+1”; “SL(A)”; “B+SL(A)+1”.
- Entire Division (ES) – In the Entire Division the logical area is composed by one adder that made the generic operation “A - A0” and one Counter. The algorithm realized in the logic circuit is: “A-A0”; “count +1”.
- Remain Division (ReD) – In the Remain Division the logical area is composed by one adder that made the generic operation “A - B”, a mux controlled by the carry out of the adder and one value shifted left (SL). The algorithm realized in the logic circuit is: SL(A); if “A-B<0”; “A<=A0” else “A<=A-B”.

4. Preliminary results

The application circuits were simulated using the tool Spectre from Cadence, using parameters from process AMI 0.5 [3]. The parameters measured were delay, area and power consumption to a 4bit version of the applications. The results are presented in the tables below.

Table 1 - Delay Table (ns)

	DIMS	Martin	DCVS	ECDL
MCD	181,9	197,9	161,7	167,7
GCD	89,21	70	46,44	51,32
RS	38,67	32	24,27	26,2
ES	156,1	103,7	101,4	106,4
ReD	287,7	225,4	228,9	264,9

Table 2 - Area Table (transistor count)

	DIMS	Martin	DCVS	ECDL
MCD	3023	1493	1438	1578
GCD	2771	1265	1184	1394
RS	2327	1155	937	1033
ES	1747	827	771	766
ReD	2209	1289	1236	1280

Table 3 - Power Consumption Table (mW)

	DIMS	Martin	DCVS	ECDL
MCD	16,72	17,54	4,46	15,4
GCD	12,83	7,14	1,92	4,78
RS	3,69	3,58	0,93	3
ES	12,14	6,76	10,2	4,48
ReD	15,28	8,32	2,22	5,57

5. Conclusions

With this preliminary work we could conclude that our applications circuits are suitable as platforms to test and characterize the adders proposed in relation to the parameters proposed.

As future work we pretend to extend these circuits to 8, 16 and 32 bits and evaluate these circuits with more styles of adders design, like DPTL, for example. Another possibility that will be evaluate in the future is the construction of application circuits with modifications in the original architecture, like put one calculus stage between each pair of latches and verify the impact of this modifications in the parameters measured.

References

[1] G. A. Ruiz, “Compact four bit carry look-ahead CMOS adder in multi-output DCVS logic,” Electronic Letters, vol. 32, no. 17, Aug. 1996, pp. 1556-1557.
[2] S. –L. Lu, “Implementation of iterative networks with CMOS differential logic,” IEEE Journal of Solid-State Circuits, vol. 23, no. 4, Aug. 1988, pp. 1013-1017.
[3] www.mosis.org
[4] M. C. B. Osorio et al., “Estruturas CMOS Dinâmicas com Multiplas Saídas,” Iberchip 2003.
[5] Jens Sparso et. Al., “Principle of Asynchronous Circuits Design,” Kluwer Academic Publishers.