Asynchronous Design Methodology for an Efficient Implementation of Low power ALU

P.Manikandan, B.D.Liu, L. Y. Chiou
Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan
mani@spic.ee.ncku.edu.tw,bdliu@spic.ee.ncku.edu.tw,
lihyih@mail.ncku.edu.tw

G.Sundar, C. R. Mandal
Department of Computer Science & Engineering
Indian Institute of Technology
Kharagpur, India
sundarg@cse.iitkgp.ernet.in, chitta@cse.iitkgp.ernet.in

***Abstract*— We present a design technique for implementing asynchronous ALUs with CMOS domino logic and delay insensitive dual rail four-phase logic. It ensures economy in silicon area and potentially for low power consumption. The design has been described and implemented to achieve high performance in comparison with the synchronous and available asynchronous design. The experimental result shows significant reduction in the number of transistors as well as delay.**

I. Introduction

The power consumption problem has been raised considerable attention nowadays in design technologies. The power consumption can be reduced by decreasing the supply voltage, load capacitance and frequency. Some propose by using gated clocks to reduce the switching activity of logic in redundant cycles [1]. Xi et. a1 provides an optimization algorithm for buffer and device sizing under process variations[2]. Although it minimizes the skew, this methodology is limited when operating frequency is very high. On the other hand, Globally Asynchronous and Locally Synchronous (GALS) technique aims to eliminate the global clock, by partitioning the system into several synchronous blocks and communicating asynchronously among blocks [3]. However, the global signaling protocol increases the total area-power penalty and affects performance of the system.

Several researchers propose asynchronous approaches to cope with performance and timing issue. Tang et. al. designed a 16-bit asynchronous ALU with an asynchronous pipeline architecture [4]. In this approach, simple handshake cells embedded in pipeline stages make the ALU run fast. However, large power has consumed by this design while waiting for the incoming data. In contrast, by using Galois Field arithmetic logic and reduced switching activity in the latches, paper [5] achieved low power in their asynchronous ALU design. Several researchers propose asynchronous approaches to cope with performance and timing issue. Tang et. al. designed a 16-bit asynchronous ALU with an asynchronous pipeline architecture [4]. In this approach,

simple handshake cells embedded in pipeline stages make the ALU run fast. However, large power has consumed by this design while waiting for the incoming data. In contrast, by using Galois Field arithmetic logic and reduced switching activity in the latches, paper [5] achieved low power in their asynchronous ALU design. Since the inappropriate rotate-wire concept of data buses, the time required for each multiplication operation becomes larger. In reducing the time dependency of an asynchronous design of Quantum dot Cellular Automata (QCA), Paper [6] used GALS with delay-insensitive data encoding scheme. Here each gate has locally synchronized by corresponding clocking zone(s). Appropriate data forwarding and synchronization guaranteed at the gate-level that reduces the number of clocking zones and increases the circuit speed. Nevertheless, the overall timing of the circuit depends upon the layout. An asynchronous bundled-data pipeline for the matrix–vector multiplication core of discrete cosine transforms achieved 30% higher average throughput [7] based on a bit-partitioned carry-save multiplier. Due to its bulky control overhead, the controller drops its speed gain. These methods intend to improve the power consumption/performance of IC's using asynchronous methodology. We focus in this asynchronous design to reduce the transistors count, power consumption and delay significantly by using delay insensitive dual rail logic and bundled data bounded delay model.

The rest of the paper has organized as follows. Section II explains briefly about the background techniques used in our architecture such as 4-phase dual rail protocol for handshaking, Muller C element for completion detection with pipeline. Section III describes the design of 1-bit ALU using our approach. Section IV shows the analysis and comparison of the proposed design with existing designs.

II. BACKGROUND

Asynchronous circuits are fundamentally different from the synchronous counterpart and use handshaking among components to perform the necessary synchronization, communication and sequencing of operations. The handshaking implementation may follow any one of these protocols, 4-phase bundled data, 2 –phase bundled data or 4-phase dual rail. In all protocols, Muller pipeline is used. The 4-phase bundled data and 2–phase bundled data are pipelined designs in which matching delay elements needed to be inserted between latches to maintain correct behavior in the request signal path. On the other hand, 4-phase dual rail has designed to combine encoding of data and request. We apply it in our circuit, because 4- phase dual rail protocol provides reliable synchronization, lower power consumption with simple and faster signal transition than 2-phase model [9].

This work was supported in part by the National Science Council under Grant Number NSC 95-2220-E-006-007 and the Ministry of Economic Affairs under the Grant Number 95-EC-17-A-01-S1-031.

1-4244-0387-1/06/$20.00 ©2006 IEEE

To achieve higher performance and low power operation, the circuit has designed with CMOS domino logic, which also reduces the transistor count, parasitic capacitances and ensures the circuit a glitch-free one. The Muller C-element and Four-Phase Dual Rail Protocol are used for the completion detection. This design uses conventional CMOS domino logic since its implementation supports the glitch free circuit and the capacitance of its output node is separating by interval and load capacitance [2]. Also it ensures the lower power consumption by reducing the parasitic capacitances and transistor count.

A. *The 4-phase dual-rail protocol concept*

This protocol makes the reliable communication between blocks of designed architecture regardless of delays in the wires connecting two blocks and also which is delay-insensitive [8] [9]. It uses a single wire for each data bit and one extra control line for each data word. It provides the reliable communication between blocks by combined encoding of data and request with an acknowledge signal after completion detection. This logic uses two request wires per bit of information d:; one wire d.t is used for signaling a logic 1(or true), and another wire d.f is used for signaling logic 0(or false). In a single bit channel with 4 phase dual rail logic, the request signal can be either of d.t or d.f for handshaking purpose. Viewed together the {x.f, x.t}={1,0} and {x.f, x.t}={0,1} represent "valid data" (logic 0and logic 1 respectively) and {x.f, x.t}={0,0} represents "no data" (or "empty value" or "E"). The codeword {x.f, x.t}={1,1} is not used, and a transition from one valid codeword to another valid codeword is not allowed, as illustrated in Fig.2(a).

B. *Muller C-element and indication concept*

The concept of indication or acknowledgement plays an important role in the design of asynchronous circuits for synchronization. Muller C element is a state-holding element much like an asynchronous set-reset latch [10]. When both inputs are 0 the output is set to 0, and when both inputs are 1 the output is set to 1. For other input combinations {(0, 1) or (1, 0)} the output does not change. Consequently, one can see the output changes from 0 to 1 can conclude that both inputs are now at 1, similarly one can see the output changes from 1 to 0 may conclude that both inputs are now 0. In this circuit design, the absence of a clock means that, in many circumstances, signals are required to be "valid data: {x.f, x.t}={1,0} and {x.f, x.t}={0,1}" all the time that every signal transition has a meaning and, consequently, that hazards and races must be avoided. Signal transitions are not indicated (acknowledged) for the other signal transitions such as {(0,1),(1,0)} and that are used to avoid the source of hazards. A circuit accomplish this requirement with Muller C-element is as shown in Fig. 2(b). The Muller C-element is indeed a fundamental component that is extensively used in asynchronous circuit design [10].

C. *Muller pipeline*

A 4-phase dual-rail pipeline is based on the Muller pipeline that relays handshakes in Fig. 2 (c). In the Muller pipeline, there is a 1-bit wide and 3-stage deep pipeline that uses a common acknowledge signal per stage to synchronize. Here the pipeline stage can store empty codeword {d.t, d.f}={0,0}, causing the acknowledge signal out of that stage to be logic 0 or one of the two valid code words {0,1} and {1,0}, causing the acknowledge signal out of that stage to be logic 1. Initially all of the C-elements have been initialized to 0 and during the operation, according to the successor value, the current C element transfers its predecessor's value for handshaking [11]. To understand what happens let's consider the i'th C- element, C [i]: It will propagate a 1 from its predecessor C [i-1], only if its successor C [i+1] is 0. In a similar way it will propagate a 0 from its predecessor if its successor is 1.

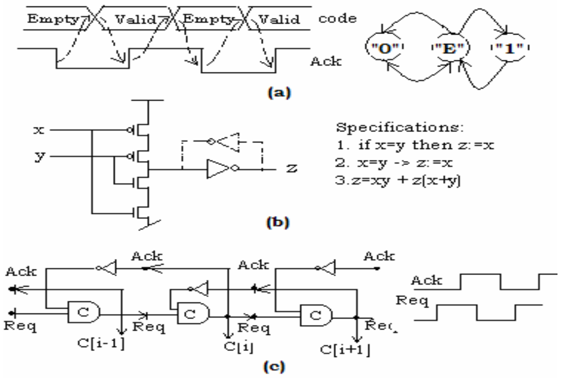

Fig. 2. (a) The 4-phase dual rail logic (b) Muller C-element and indication concept and (c) Muller pipeline

III. DESIGN AND IMPLEMENTATION

Using the logics and principles outlined in Section II, an ALU has designed at the transistor level for single bit operation as shown in the Fig. 3 to demonstrate our design concept. A single bit-slice ALU uses only 53 transistors and is capability of operations in Table 1. Since we emphasize on the design of asynchronous component, there is no hardware implementation for 4-phase dual rail with Muller C. However, the proposed circuit assumes the signaling from such logic blocks. For example, C0out and C1out act as two wires of 4-phase logic, which makes reliable operation between its predecessor and successor blocks.

Architecture design & implementation

We designed a 32-bit ALU, which requires 1696 transistors. The basic principle of Bundled data – Bounded delay model of Sutherland's micro pipelines is used here [12]. The timing characteristics of all data busses of this architecture are bundled together. Here the statuses of the data busses are indicated by 4 phase-dual rail hand shake signals. The clock power reduction at the architectural level is mainly due to pipeline technique. The dynamic logic of completion detection unit ensures precise internal operation, because of its 4-phase dual logic. It is also carrying the timing information because it uses common timing characteristics.

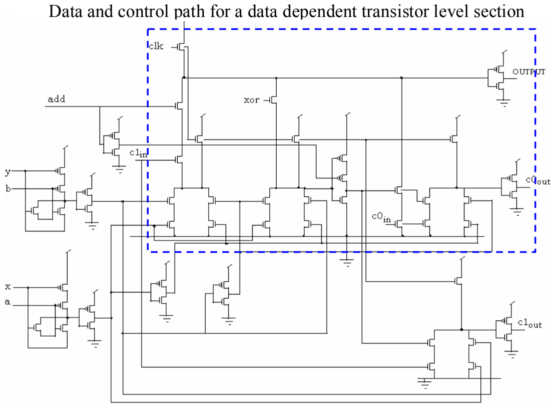
Data and control path for a data dependent transistor level section
clk
add
xor
OUTPUT
c1in
c0out
y
b
c0in
x
a
c1out

Fig. 3. CMOS level Asynchronous ALU circuit for 1-Bit operation

TABLE I. FUNCTIONS AVAILABLE FOR THE PROPOSED ALU

Logic Funcion	Basic Operation	a-input	b-input
and	AND	true	true
add	AND	true	true
add with carry	AND	true	true
subtract	AND	true	complement
reverse subtract	AND	complement	true
subtract with carry	AND	true	complement
rev. subtract with carry	AND	complement	true
test bits	AND	true	true
compare	AND	true	complement
compare negative	AND	true	true
bit clear	AND	true	complement
xor	XOR	true	true
test equal	XOR	true	true
or	OR	true	true
move	OR	zero	true
move NOT	OR	zero	complement

IV. ANALYSIS AND COMPARISON

Addition is one of fundamental functions of the ALU. We start by analyzing the number of transistors used in the addition. About 80% of the operations require some form of addition [13]. If we improve the processing time of addition operation, the performance of complete ALU can also be improved. The latency required by our design is depended upon the operation, the input data at that incident and the carry flow across the whole word length, i.e. it needs to propagate carry until it has predicted by the completion detection stage. The average length of the mean carry propagation distance is varying according to input data. In this 32-bit operation, a sum of 140 transistors has used for precharging (domino logic) and buffer purposes to meet the specifications at the layout. Our logic so far discussed ensures very simple circuitry than existing designs and provides three benefits at a time such as reduction in size, less power consumption and improvement in performance.

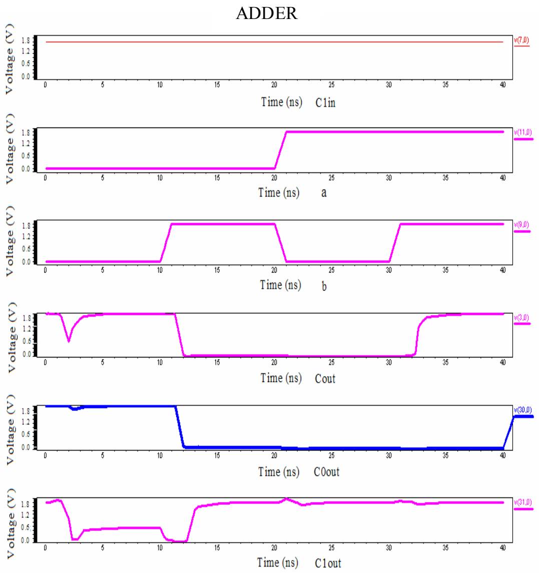

Fig. 4. Simulated waveforms for the 1 bit addition operation

The simulated output waveform for the addition operation performed by this ALU presented in Fig. 4. It is performed with V_{DD}=1.8V, input sequence C1in=1111, C0in=0000, A=0011,B=0101 and the simulated output sequence is output=1001, C0out=1000, C1out=0111 which coincides the expected specification. This simulation has done by HSPICE with 10ns local clock period at room temperature.

The simulation results for the power consumption of typical addition operation with different supply voltages are shown in the Fig. 5. The simulation results of HSPICE-0.18um technology shows, the average power consumption for typical addition operation is 1.02e-4w under 1.8V supply with 1000 sample inputs at room temperature and average time delay is 2.5ns.

A comparison of the simulated time performance and transistor count of this design with other published alternatives is shown in Table II and Table III. They clearly indicate a significant reduction in transistors count. Our design has much reduction in silicon area. In addition, this architecture enables to have reduced switching capacitance because of missing master clock in the transistors of the ALU circuit design. It gives reduced switching actions for every arithmetic operation and reduction in silicon area. In summary, our proposed design gives a better throughput with minimum number of transistors.

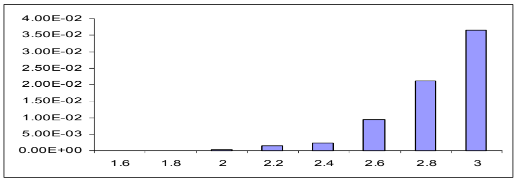

Fig. 5. Simulation Results for power consumption at different $V_{DD.}$

TABLE II. SIMULATIVE RESULTS FOR TIME PERFORMANCE

Time(ns)	***Best case***	***Worst case***	***Average***
Ref [14]	2.5	7.5	5
Ref [15]	3	6	4.5
Ref [16]	3	6	4.5
Our design	3	4	3.5
Synch	-	-	4

TABLE III. COMPARISON OF REPORTED DESIGN

Compariso n	**Existing Designs and Discussed Design**		
	Synchronous ARM ALU [14]	***Asynchronous ARM ALU [14]***	***Our Design***
Technology	1.2um CMOS	1.2um CMOS	0.18um CMOS
Supply voltage	~ 5V (CO/SS)	~ 5V (CO/SS)	~ 1.8V (CO/MS)
Self Time Unit	3000 (# transistors)	2300 (# transistors)	1696 (# transistors)
Timing Purpose	--	140 (# transistors)	140 (# transistors)
Data width	32 bit	32 bit	32 bit

CO-Correct Operation, SS-Slow Speed, MS-Medium Speed

V. CONCLUSIONS

The proposed design can reduce silicon area and improve the performance on average. This is mainly due to the use of the asynchronous design concepts with CMOS dynamic domino logic. They result in reduction in the transistor count & parasitic capacitances. Subsequently, power consumption can be reduced. Furthermore, this architecture provides glitch free operation, which is an important key factor to better reliability and performance. Effective reduction in area, power consumption and reasonable performance improvement are the special features of our designed architecture.

REFERENCES

[1] L. Benini, and G. De Micheli," Transformations and synthesis of FSM's for low power gated clock implementation", *IEEE Trans. Computer-Aided Design*, vol. 15, June 1996.

[2] J. M. Rabaey, and M. Pedram, *Low Power Design Methodologies*, Kluwer Academic Publishers, 1996, ISBNO-7923-9630-8.

[3] A. Hemani, T. Meincke , S. Kumar, A. Postula, P. Nilsson, J. Oberg , P. Ellervee, and D. Lundqvist, "Lowering power consumption in clock by using globally asynchronous locally synchronous design style," in *Proc. IEEE Design Automat. Conf.*, June 1999, pp. 873-878.

[4] T. Y. Tang, C. S. Choy, J. Butas, and C. F. Chan, "An ALU design using a novel asynchronous pipeline architecture," in *Proc. IEEE ISCAS*, May 2000, pp. vol. 5, 361-364.

[5] P. K. Leung, C. S. Choy. C. F. Chan and K. P. Pun, "A low power asynchronous GF(2173) ALU for elliptic curve crypto-processor," in *Proc. IEEE ISCAS*, May 2003, vol. 4, pp. 337-340.

[6] M. Choi, and N. Park, "Locally synchronous, globally asynchronous design for quantum-dot cellular automata (LSGA QCA)," in *Proc. IEEE NANO*, July 2005, pp. 121-124.

[7] S. Tugsinavisut, Y. Hong, D. Kim, K. Kim, and P. A. Beerel, "Asynchronous bundled-data pipelines for DCT matrix–vector multiplication", *IEEE Trans. VLSI Systems*, vol. 13, pp. 448-461, Apr. 2005.

[8] S. Hauck, "Asynchronous design methodologies: an overview," Proc. *IEEE*, vol. 83, pp. 63-69 ,Jan 1995.

[9] J. Sparso, and S. Furber, *Principles of Asynchronous Circuit Design: A Systems Perspective*, Kluwer Academic Publishers, 2001.

[10] I. E. Sutherland, and J. Ebergen, *Computers without Clock*, Scientific American, Inc. Aug. 2002.

[11] S. B. Furber, P. Day, J. D. Garside, N. C. Paver, and J. V. Woods, "A micropipelined ARM," *Proc. IEEE* Compcon, Mar. 1994, pp. 476-475.

[12] I. E. Sutherland, "Micropipelines", Communications of the ACM, pp 720-738, vol. 32, No. 6, June 1989.

[13] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel and F. Baez, "Reducing power in high performance microprocessors," in *Proc. IEEE Design Automat. Conf.*, June 1998, pp. 600-610.

[14] J. D. Garside, "A CMOS VLSI implementation of an asynchronous ALU", in *Proc. IFIP working conference on Asynchronous Design Methodologies*, Manchester, M139PL, U.K.(1993).

[15] A. D. Gloria, and M. Olivieri "Statistical carry lookahead adders," *IEEE Trans. on Comput.*, vol. 45, pp. 340-347, Mar. 1996.

[16] D. Kearney, and N. W. Bergmann, "Bounded data asynchronous multipliers with data dependent computation times," in *Proc. IEEE Advanced research in asynchronous ciruits and systems*, Apr. 1997, pp. 186-197.