NARESH NAGULA , S. V. DEVIKA , SK. KHAMURUDDEEN/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
www.ijera.com
Vol. 1, Issue 4, pp.1922-1927

Implementation of Asynchronous Topology using SAPTL

NARESH NAGULA *, S. V. DEVIKA **, SK. KHAMURUDDEEN ***

*(senior software Engineer & Technical Lead, Xilinx India)
** (Associate Professor, Department of ECE, HITAM ,Hyderabad ,India)
*** (Assistant Professor, Department of ECE, HITAM ,Hyderabad ,India)

ABSTRACT

By using low-leakage pass transistor networks at low supply voltages SAPTL(sense amplifier-based pass transistor logic) structure can realize very low energy computation. With the introduction of asynchronous operation in SAPTL further improves energy-delay performance without a significant increase in hardware complexity. So in this paper we are going to propose and design the implementation of low-energy asynchronous logic topology using SAPTL. In this paper we present two different self timed approaches like Bundled Data and Dual rail handshake protocol. The proposed self-timed SAPTL architectures provide robust and efficient asynchronous computation using a glitch-free protocol to avoid possible dynamic timing hazards. Results obtained with this paper show that the self-timed SAPTL with dual-rail protocol exhibits energy-delay characteristics better than synchronous and bundled data self-timed approaches in 90-nm CMOS.

Keyword's— SAPTL, Bundled Data, Dual rail handshake Protocol, logical topology, Low-leakage circuits.

I.INTRODUCTION

The ability to design and build logic and computational elements that operate at extremely low energy levels is seen as a very important enabler for systems in various application domains such as mobile devices, wireless sensor networks and bio-medical systems. Devices operating at these low energy levels can also take advantage of alternative energy storage and scavenging methods that can lead to almost indefinite operational lifetimes, as well as new computing and system paradigms. Technology scaling and supply voltage reduction have been responsible for the continued energy reduction and performance improvement in complementary static CMOS circuits, the most popular logic topology in use today. However, the increased leakage energy brought about by scaling and VDD reduction is starting to limit the minimum energy that static CMOS circuits can achieve. One low-energy alternative to complementary static COMS circuits is the sense amplifier-based pass transistor logic (SAPTL) topology. The sense amplifier-based pass transistor logic (SAPTL) is a novel circuit topology that breaks this tradeoff in order to achieve very low energy without sacrificing speed. The initial SAPTL circuits were designed to operate synchronously but with the intent of being able to operate asynchronously with some minor modifications.

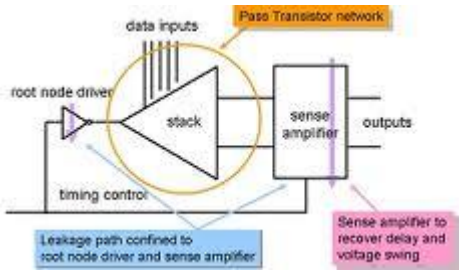

Fig 1: The sense amplifier-based pass transistor logic (SAPTL)

As the effects of process variations continue to increase dramatically with technology scaling, it is becoming harder to design variation-tolerant timing schemes using the traditional synchronous methodologies. To meet a certain timing requirement, the synchronous approach must use a very conservative "worst case" design that is slow enough for the needs of the statistically slowest circuit elements and, thus, will fail to exercise the whole capacity of statistically faster parts of the circuit. The asynchronous approach, on the other hand, can exploit local timing information to achieve "average-case" performance. An asynchronous design can get the best performance out of all components independent of statistical variations in local speed while guaranteeing correct circuit operation.

Asynchronous operation is also attractive to the low-power designer. The absence of a clock distribution network can significantly reduce the power overhead needed to generate timing information. Furthermore, an idle asynchronous system avoids consuming any active power.

Despite the advantages of asynchronous operation, the circuit complexity and performance overhead required to implement the needed handshaking protocol may not be trivial. The overhead cost might offset all benefits and make the asynchronous approach impractical. The SAPTL, however, offers a relatively easy way to realize asynchronous operation. Because of the differential signaling used, it is easy to determine when a logical operation completes. Therefore, the self-timed SAPTL topology is a promising candidate for reducing power consumption and improving speed in extremely low energy applications.

II.PROTOCOL DESIGN

Fig (2) & (3) shows two approaches to realize RTA8, which we call the early reset and the late reset protocol, respectively. In the early reset protocol, we introduce another event Reqin*↑ between the original Ackout↓ and DIN↑ events, as shown in Fig. 2. We can use the Reqin* signal, instead of Reqin , as a triggering signal to start the data reset operation earlier and avoid generating a glitch.

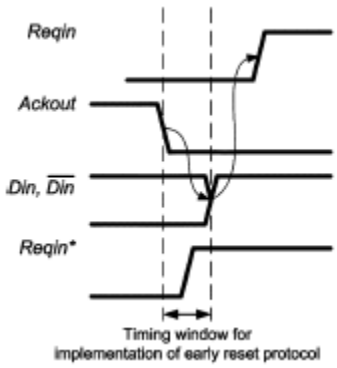

Fig 2: Early reset.

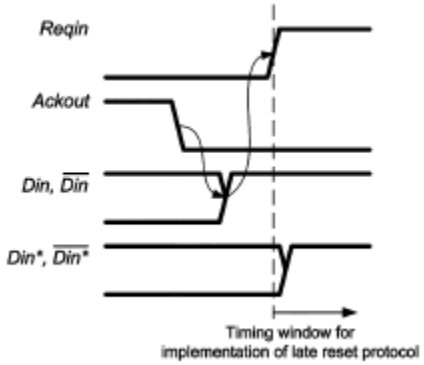

Fig 3: late reset glitch-free handshaking protocol.

The other way to implement glitch-free operation, which is the late reset protocol, is shown in Fig. 2. The stack employs signals Din* and $\overline{Din*}$, which are the replica delayed versions of Din and $\overline{Din}$, as reset input signals in the data reset cycle. The only requirement for and is that both signals be triggered later than. From Fig (2) & (3), we can observe that the timing slack needed to implement the early reset protocol is smaller than for the late reset protocol or, in other words, the implementation of the early reset protocol requires stricter RTAs. However, employing the early reset protocol will not affect the original latency of the data reset operation.

Therefore, the early reset protocol can achieve higher speed performance than the late reset protocol. Moreover, the early reset protocol can also minimize the leakage energy consumed per handshaking operation of the SAPTL by keeping the root of the *stack* at logic 0 longer. This advantage of low-leakage operation makes the early reset protocol the preferred option.

III.CIRCUIT IMPLEMENTATION

The circuit implementation of a self-timed SAPTL module with the early reset glitch-free handshaking protocol is shown in Fig. 4, and the corresponding timing diagram for two successive evaluation and reset cycles is shown in Fig. 5. The additional OR gate and extra input Reqin to the C-element do not change the functional behavior of self-timed SAPTL in the data evaluation cycle. In a data reset cycle, however, Reqin* will be pulled up to a logical 1 and will start resetting the internal nodes of the stack immediately after the SAPTL module de-asserts the acknowledge signal Ackout. No glitch will be

generated during the data reset cycle if Reqin* is charged to a logical 1 and drives the stack root input to a logical 0 before Din*and $\overline{Din}$ go high, which can be described by the following relative timing constraint for glitch-free operation.

$$T_{Driver} + T_{OR} < T_{C-elemen} + T_{SA,reset} \quad \text{(RTA9)}$$

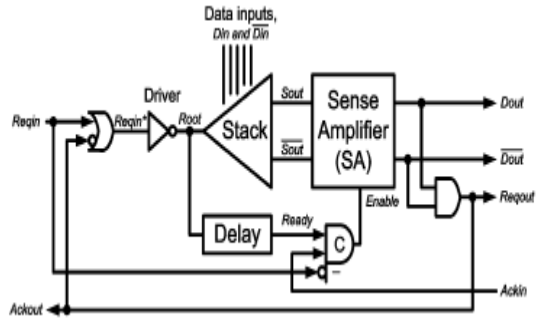

Fig 4: Self-timed SAPTL structure with early reset glitch-free protocol.

Because the OR gate and driver may be merged into one NOR gate, the total delay in the left-hand side of RTA9 is really small and RTA9 can thus easily be met. The extra input Reqin to the

C-element is essential to maintain the reset state of the current SAPTL stage until the previous SAPTL raises Reqin for the next data evaluation cycle, thus the use of a three-input C-element. Note that the third C-element input signal Reqin is necessary only in the data reset cycle, but not in the data evaluation cycle. Therefore, we are able to use an asymmetric C-element circuit in this glitch-free self-timed SAPTL to minimize delay and energy consumption.

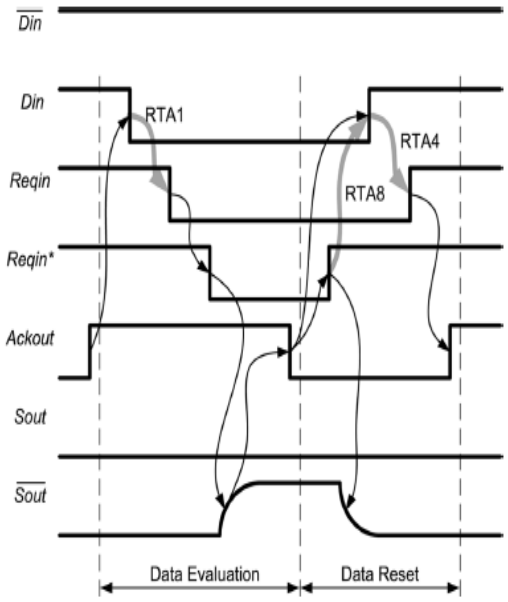

Fig 5: Timing diagram of glitch-free self-timed SAPTL.

By employing an additional NOR gate and a higher fan-in asymmetric C-element, the self-timed SAPTL architecture can perform the following: 1) achieve more robust handshaking; 2) consume lower energy; and 3) avoid the glitch problem. In addition, there are two main advantages to resetting the stack immediately after the SAPTL module sends the acknowledge signal. First, Sout and $\overline{Sout}$ will stay above logical 0 for only the short period required by the sense amplifier to latch the stack data. Once Dout and $\overline{Dout}$ has reached full swing, every internal node of the stack, as well as Sout and$\overline{Sout}$, will be reset to logical 0. Therefore, during the remainder of the cycle, the stack will stay in the data reset mode and consume minimum leakage energy.

Second, because the handshaking events with the previous SAPTL stage and the data reset events within the current SAPTL *stack* operate in parallel, the SAPTL will have a lower latency data reset cycle and thus achieve better performance. Thus, the delay line now becomes the major performance limiter in the self-timed SAPTL design using the bundled data protocol.

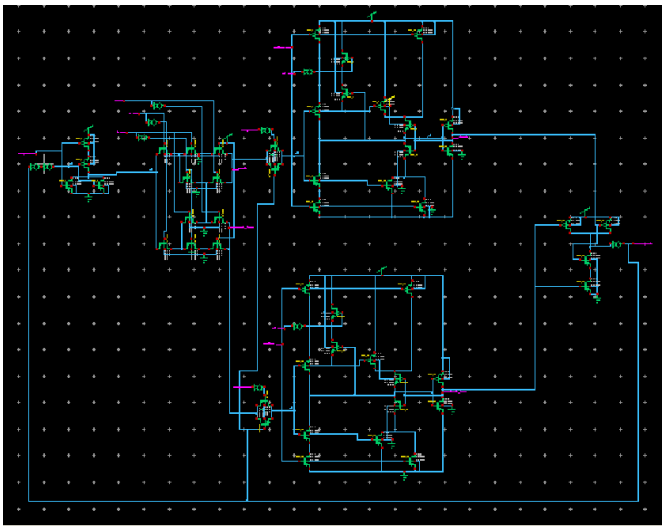

Fig 6: Glitch free handshaking protocol schematic using tanner tool

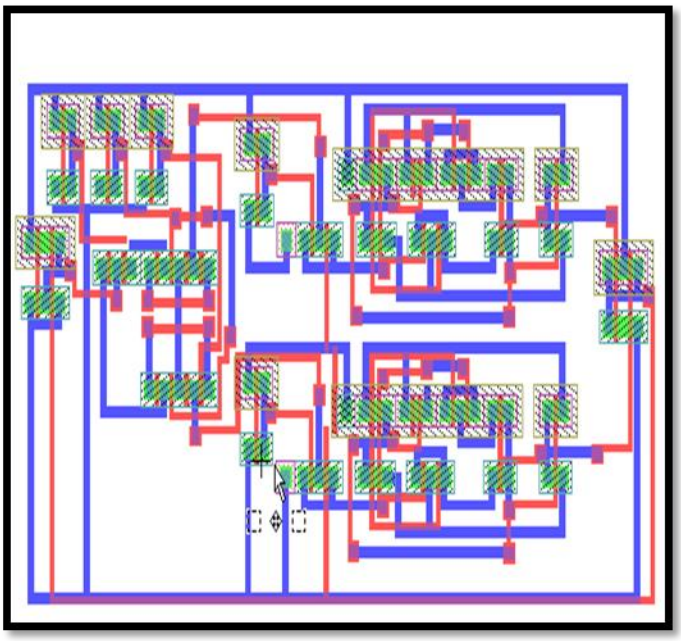

Fig 7: Corresponding layout for glitch free handshaking protocol

IV.RESULTS

LOGICS	NO. OF TRANSISTORS USED			POWER FROM TIME 0 TO 4e-008 s		
	P-MOS	N-MOS	TOTAL	AVG POWER	MAX POWER	MIN POWE
RATIOED	2	6	8	9.932225e-005 watts	5.073397e-003	1.130351e-00
DOMINO	2	8	10	8.176752e-006 watts	1.756024e-002	3.771018e-00
DCVSL	2	12	14	9.176304e-005 watts	1.565550e-002	3.849001e-00
CMOS LOGIC	6	6	12	5.321445e-006 watts	1.367566e-002	1.873325e-00
NMOS PASS TRANSISTOR LOGIC	0	8	8	7.519514e-007 watts	8.810182e-003	0.000000e+000

Table 1: Comparison of Various Logics with nMos Pass Transistor Logic

MODULES	POWER CALCULATED FROM TIME 0 TO 4e-008 s		
	AVG POWER CONSUMED	MAXIMUN POWER	MINIMUN POWER
Synchronous SAPTL	2.928049e-005 watts	2.637207e-002 watts	5.059357e-009 watts
Bundled Data Self – timed SAPTL	2.870605e-005 watts	1.920554e-002 watts	1.781765e-008 watts
Glitch free self timed SAPTLmodule with dual rail protocol	5.857818e-004 watts	5.857818e-004 watts	5.857818e-004 watts

Table 2: Power Results For Various SAPTL Architectures

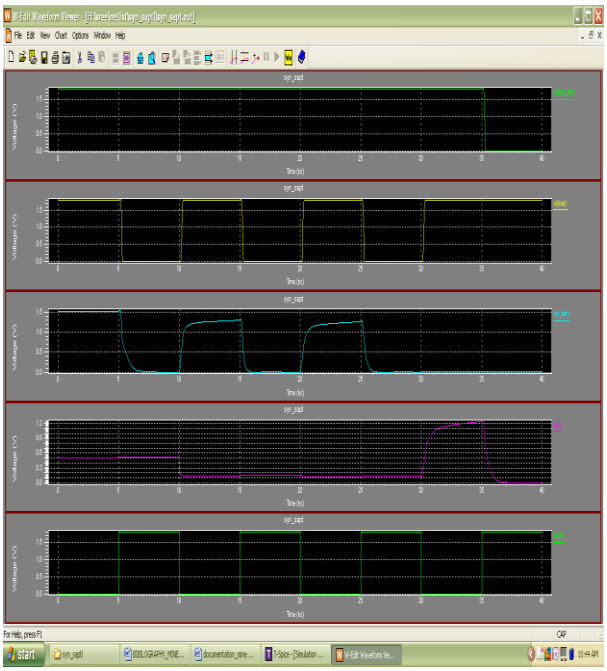

Fig 8: Output Waveforms of Synchronous SAPTL

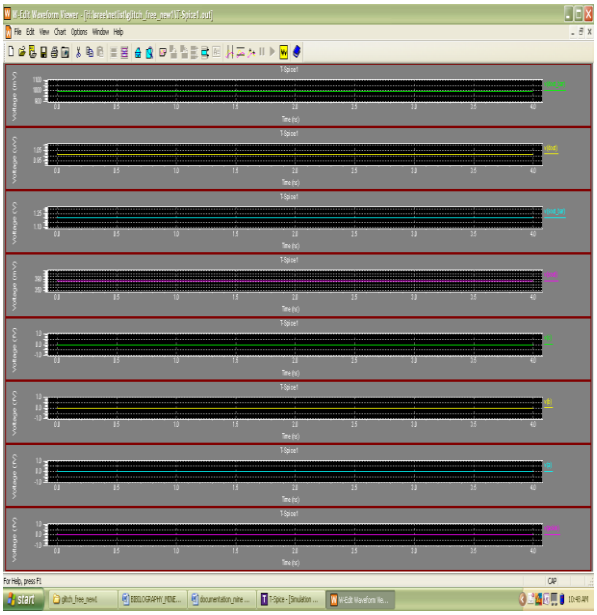

Fig 10: Output Waveforms Of Glitch-Free Handshaking Protocol

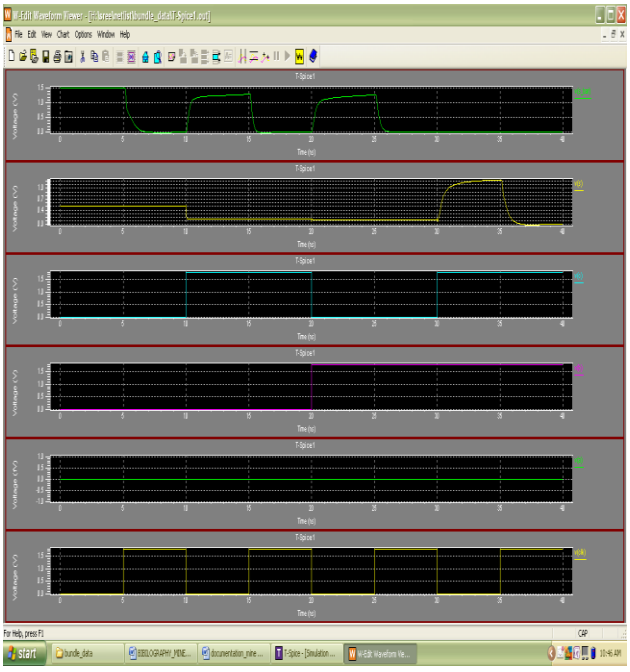

Fig 9: Output Waveforms For Bundled Data

V.CONCLUSION

The sense amplifier-based pass transistor logic (SAPTL) is a logic topology that utilizes an inverted pass transistor tree, or the stack, as a passive logic implementation network. Unlike conventional pass transistor networks, the stack only has feed forward paths from a single root node to two low-swing pseudo-differential outputs. Gain is then added in the form of the root driver and sense amplifier. Note that both the driver and sense amplifier do not contribute anything to the Boolean functionality of the logic block. This effective decoupling of functionality and gain is a key SAPTL characteristic.

The asynchronous operation of the SAPTL provides robustness in the presence of variability as well as performance advantages over synchronous operation. While the self-timed SAPTL using the bundled data protocol can potentially achieve higher speed performance by overlapping the data evaluation and reset cycle, the self-timed design based on the dual-rail protocol has less rigid relative timing constraints, which leads to better energy and speed performance in technologies with increased process variations.

The early reset operation of self-timed SAPTL not only prevents dynamic timing hazards from glitches but also improves both energy and speed Performance. The low implementation cost of the asynchronous operation makes the self-timed SAPTL family a very promising candidate to realize robust and low-energy computations.

REFERENCES

[1] T. Sakurai, "Perspectives on power-aware electronics," in *ISSCC Dig. Tech. Papers*, 2003, vol. 1, pp. 26–29.

[2] L. Alarcón, T.-T. Liu, M. Pierson, and J. Rabaey, "Exploring very lowenergy logic: A case study," *J. Low Power Electron.*, vol. 3, no. 3, pp. 223–233, Dec. 2007.

[3] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design*. Norwell, MA: Kluwer, 2001.

[4] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Englewood Cliffs, NJ: Prentice- Hall, 2003.

[5] H. Li, S. Bhunia, Y. Chen, K. Roy, and T. Vijaykumar, "DCG: deterministic clock-gating for low-power microprocessor design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 3, pp. 245–254, Mar. 2004.

[6] N. Banerjee, K. Roy, H. Mahmoodi, and S. Bhunia, "Low power synthesis of dynamic logic circuits using fine-grained clock gating," in *Proc. DATE*, Mar. 2006, vol. 1, pp. 1–6.

[7] T.-T. Liu, L. Alarcón, M. Pierson, and J. Rabaey, "Asynchronous computing in sense amplifier-based pass transistor logic," in *Proc. 14th IEEE Int. Symp. ASYNC*, Apr. 2008, pp. 105–115.

[8] T. Williams, "Performance of iterative computation in self-timed rings," *J. VLSI Signal Process.*, vol. 7, no. 1/2, pp. 17–31, Feb. 1994.

[9] K. Stevens, R. Ginosar, and S. Rotem, "Relative timing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 1, pp. 129–140, Feb. 2003.

[10] I. Sutherland, "Micropipelines," *Commun. ACM*, vol. 32, no. 6, pp. 720–738, Jun. 1989.

[11] S. Narendra, "Scaling of stack effect and its application for leakage reduction," in *Proc. ISLPED*, Aug. 2001, pp. 195–200.

AUTHORS:

Mr .Naresh Nagula Working as senior software Engineer & Technical Lead, Xilinx India, His area of interest of are Embedded systems & VLSI. He can be reached at naresh.nagula@ xilinx.com .

Mrs. S. V. Devika Working as Associate Professor in Hyderabad Institute of Technology & Management, Her area of interest are communications & VLSI. She can be reached at chin.devika@gmail.com .

Mr Sk. Khamuruddeen Working as Assistant Professor in Hyderabad Institute of Technology & Management, His area of interest is VLSI. He can be reached at khd485@gmail.com.