Asynchronous Computing in Low Power Based Sense Amplifier Pass Transistor Logic

JOHN PRAGASAM.D. #1, D.S.SHYLU SAM#2
Department of Electronics and communication
Karunya University
Coimbatore
INDIA
1d.johnpragasam@gmail.com, 2shylusam@karunya.edu

Abstract: - This paper presents the design and implementation of a low-energy asynchronous logic topology using sense amplifier based pass transistor logic (SAPTL). The SAPTL structure can realize very low energy computation by using low leakage pass transistor networks at low supply voltages. So the self-timed SAPTL with bundled data protocol power consumption is better than the synchronous SAPTL.

Key-Words: - Low-leakage circuits, low-voltage circuits, pass transistor, self-timing, sense amplifier based pass transistor logic(SAPTL).

1. Introduction
Everyday CMOS technology continues to scale, both supply voltage and device threshold voltage must scale down together to achieve the low power. Lowering the supply voltage effectively reduces dynamic energy consumption but is a accompanied by a dramatic increase in leakage energy due to the lower device threshold voltage needed to maintain the performance[1].

The sense amplifier-based pass transistor logic (SAPTL) [2] 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 [2] but with the intent of being able to operate asynchronously with some minor modifications.

In 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 in formation. The SAPTL is 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.

2. SAPTL Architecture
The basic architecture of the SAPTL [2] circuit is shown in Fig. 1. It is composed of a pass transistor stack, a driver, and a sense amplifier.

![Fig1. Architecture of SAPTL module with synchronous timing control.](image)
The total energy consumed by the SAPTL is composed of the following: 1) The energy used by the driver to energize the stack, 2) The energy used by the sense amplifier to resolve the correct logical levels and drive the inputs of the fan-out stacks, (3) The energy needed to generate the appropriate timing information, either globally, such as clock distribution networks, or locally, as in handshaking circuits.

2.1 Stack and Driver

The stack consists of an nMOS-only pass transistor tree with full-swing inputs and low-swing pseudo differential outputs to perform the required logic function, as shown in Fig.2. The stack can implement any Boolean expression by connecting the minterm branches of the tree to one output and the maxterm branches to the other as illustrated by the programming switches in the diagram. In our current implementation, the logic function of an SAPTL stack is determined and permanently fixed at fabrication by replacing the programming switches with hardwired connections. Because the stack has no supply rail connections.

2.2 Sense Amplifier

The sense amplifier circuit shown in Fig.3 consists of two stages. The first stage acts as a preamplifier to reduce the impact of mismatch in the actual technology environment, and the second stage acts as a cross-coupled latch which retains the processed data even after the stack is reset.

The sense amplifier is designed to detect input voltages that are less than \((V_{th} - V_{dd})\), thus reducing the performance degradation due to the low stack voltage swings and the absence of gain in the pass transistor network.

3. Self-timed SAPTL Protocol

The communication between two self-timed SAPTL modules based on a request–acknowledge handshaking protocol is shown in Fig. 4. The operation of the self timed SAPTL involves two parts: 1) Data evaluation, 2) Data reset. The reset cycle, both data inputs must be reset to a logical 1 level rather than the commonly used logical 0 [3]. Various relative timing assumptions (RTAs) [6] are presented to ensure that sufficient voltage levels are present at the stack outputs in order to guarantee correct SAPTL operation.

4. Bundled data self-timed SAPTL design

The circuit implementation of the self-timed SAPTL module using the bundled data protocol is shown in Fig 5. The data path which is composed of a driver and stack evaluates data or resets after receiving the request signal Req and data input signals Din and Din from the previous SAPTL stage [5]. The control path, which consists of a delay line and a C-element, produces the local clock signal Enable to trigger the sense amplifier. The delay line mimics the delay of the stack to generate the control signal ready indicating that the stack has finished an operation.

The C-element produces Enable by collecting Ready and the acknowledge signal Ack in from the next SAPTL stage.
When triggered by Enable, the sense amplifier latches the stack output data or resets depending on the logical state of Enable. The full-swing data output signals $D_{out}$ and $\overline{D}_{out}$ are made available at the outputs of the sense amplifier. The AND gate serves as a completion detection circuit, generating the handshake signals Ackout and Reqout that indicate the completion of the current operation.

The relationship between the input and output signals of the SAPTL stage as follows,

$$D_{out}, \overline{D}_{out} = f(Enable_i, S_{out}, \overline{S}_{out})$$  \hspace{1cm} (1)

$$Ackout_i = Reqout_i$$  \hspace{1cm} (2)

$$Enable_i = h(Regin_i, Ackin_i)$$  \hspace{1cm} (3)

Where for the subsequent $(i+1)$ th SAPTL stage

$$Din_{i+1} = D_{out}$$
$$\overline{Din}_{i+1} = \overline{D}_{out}$$

4.1 Data Evaluation Cycle

When the self-timed SAPTL stage finishes a data reset cycle, it raises the acknowledge signal Ackout and waits for Din, $\overline{Din}$ and a falling before performing a data evaluation cycle. Before the driver applies the evaluation current, which is controlled by $Regin$, the data input signals Din and $\overline{Din}$ must be set to the correct logical levels for proper operation. The Din indicates voltage difference have been developed between the two data input signals Din and $\overline{Din}$, which means that one of the two data input signals has been pulled down to logical 0, while the other stays at logical 1. The $S_{out}$ represents the development of a sufficient voltage difference between the two stack output signals $S_{out}$ and $\overline{S}_{out}$, meaning that one of the output signals has been charged high enough, while other stays at logical 0.

4.2 Data Reset Cycle

In contrast to the micropipeline two-phase handshaking protocol which is able to exploit both phases for data transmission in order to achieve maximum throughput, the self-timed SAPTL architecture can transmit data only during alternate phases because the stack must be reset after every data evaluation cycle. Thus, one handshake phase in the self-timed protocol is always dedicated to data reset.

5. Results

The Synchronous SAPTL circuit has drawn using Tanner tool and the output waveform in fig 6.

![Fig6. Synchronous output waveform](image6.png)

The output power is calculated by using Tanner tool and its max power output is $4\mu$Watts.

The Asynchronous bundled data SAPTL circuit also has drawn using Tanner tool and the output power waveform in fig7.

![Fig7. Asynchronous bundled data SAPTL output waveform](image7.png)
The output power is calculated by using Tanner tool and its power output is less 3.5µWatts.

6. Conclusion

The synchronous and asynchronous SAPTL architecture described in the paper is analyzed. The synchronous and asynchronous bundled data protocol power is calculated for the single SAPTL architecture. Finally, the self-timed SAPTL with bundled data protocol power consumption is better than the synchronous SAPTL.

7. Future work

SAPTL architecture consists of five blocks SAPTL, with feedback from each block is connected to Reqin and Ackin Comparison of area of synchronous and asynchronous SAPTL will be found. The Leakage current and Energy delay of SAPTL architecture of both as synchronous and asynchronous will be calculated as whole.

References: