*IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*
*Volume 4, Issue 4, Ver. I (Jul-Aug. 2014), PP 64-69*
*e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197*
*www.iosrjournals.org*

# A Survey on Static Power Reduction Techniques in Asynchronous Circuits

V. Madhurima[1], Dr. Kesari Padma priya[2]
[1](Assistant professor, ECE department, SV college of engineering, Tirupati, AP, INDIA)
[2](Associate professor, ECE department, JNTUK University college of engineering, Vizianagaram, AP, INDIA)

***Abstract:*** *With the technology down scaling the area of each device in a chip reduces. Lesser area increases the power consumption. In current technologies leakage current is the major part in power consumption. Power gating is a technique which has been used to reduce leakage power by shutting off the power when no activity done by the logic. These helps to reduce the power consumption, delay and switching times of the logic. This survey paper mentions the different techniques to reduce leakage power in asynchronous logic.*
***Keywords:*** *Asynchronous logic, power gating, sleep mode etc.*

## I. Introduction

Reducing power consumption has become very important in recent years due to increase in transistor density and clock frequency as well as consumer trends in high-performance, portable and embedded applications. There is lot of methods available to reduce the power in chip. Total power is divided into static and dynamic. Power loss at the time of transistor working at weak inversion region is called static power. Dynamic power is because of capacitor charging and discharging. Dynamic power losses are important, but can be minimized by using the technique called clock gating, which reduces the power consumption of idle sections of synchronous circuits [1].Asynchronous circuit's implements the equivalent of a fine-grained clock gating network. However, while dynamic power loss has been dominant problem in the past, static power loss has become a considerable contributor to power consumption in nano-scale technologies [4, 3] due to leakage currents. So if the technology is down scaling the area of each device in a chip reduces lesser area increases the power this serious cause of static power loss are leakage currents. There are many techniques are designed to leakage currents [2, 6, 5].The most effective technique is power gating circuits.

Power gating technique is a technique used to reduce the power when the logics are in idle state i.e. Essentially cutting off the pull-up network and pull-down network from one or both power rails during idle or "sleep" periods. During active period the circuit is reconnected to the power rails in a process known as 'wake up" or power up. While power gating has been adopted for use in asynchronous circuits [7, 8] most of these efforts involve direct application of synchronous techniques to asynchronous systems. As such, the unique capabilities of non synchronous circuits have not been leveraged in the context of power gating. Many asynchronous circuit families are robust to a wide range of supply voltages, ambient temperatures and process variations. This robustness in the context of power gating to enable a zero-delay wakeup scheme for pipelined computation: the first token traveling through a pipeline turns on downstream pipeline stages, hiding the latency cost of wakeup in the computation time of upstream pipeline stages.

Synchronous circuits cannot take full advantage of such forceful power gating control schemes, as local supply voltage must reach nominal values to prevent the synchronous circuit from violating its time requirements, e.g. setup/hold constraints on state-holding elements. Therefore, inputs can only be applied to a pipeline stage once the supply voltage has reached an acceptable threshold. By leveraging the supply voltage operating range of asynchronous circuits, which can avoid this requirement and begin useful computation before the supply voltage has stabilized, reducing the forward latency seen by the first input token.

In general, power gating techniques increase the effective resistance of leakage paths by inserting sleep transistors (power gating transistors) between power supply rails and transistor stacks. Common implementation methods of power-gating techniques include multi-threshold CMOS, boosted-gate CMOS, super cut-off CMOS, variable threshold CMOS, and zigzag super cut-off CMOS. The power consumption of power gating circuitry is consumed by the sleep controller, the sleep signal distribution network, and the sleep transistors. The

fundamental challenge for any power gating technique is to ensure that the saved standby power outweighs the power overhead of the power gating. Power gating techniques are classified into two types: coarse-grain power gating and fine-grain power gating. Coarse grain system of fewer larger components has fine grain system. The coarse grain description of a system regards large sub components while fine grained description regards smaller components of which the larger ones are composed. In coarse-grain power gating, a large number of lookup tables (LUTs) share a single sleep controller so the area and power overheads of the sleep controller are relatively small. However, if any LUT within a coarse-grain power-gated domain is active, none of the LUTs which share the same sleep transistor can be set to the sleep mode. Asynchronous circuit with coarse-grain power gating also causes a large dynamic power and area overhead in the sleep signal distribution network since it is distributed to many LUTs through programmable interconnection resources. On the other hand, in fine-grain power gating, each LUT has its own sleep transistor and related sleep controller, so when any LUTs are inactive, they can be set to the sleep mode immediately. These results in much lower standby power compared to coarse-grain power gating.

Asynchronous circuits employ local handshaking for transferring data between neighboring modules, so they are data-driven and active only when performing useful work. That is, asynchronous circuits do not switch when inactive and inherently have the advantage of offering the equivalent of fine-grain clock gating. Although asynchronous circuits in inactive mode have no dynamic dissipation, they still suffer leakage dissipation. Recently, several techniques have been proposed for employing power gating techniques to reduce the static power of asynchronous circuits at different levels of granularity.

## II. The Conventional Asynchronous Logic Pipeline

In each combinational block in the conventional asynchronous four-phase bundled-data pipeline [9] is provided by both a header and a footer sleep transistor. When the latch controller in a pipeline stage detects valid input data, it absorbs the data in the data latch and turns on the sleep transistors. So, that the combinational block can wake up and process the input data to generate the output data. When the output data are received by the next pipeline stage, an acknowledge signal is sent back to this stage, and the latch controller can turn off the sleep transistors of the associated combinational block to reduce leakage dissipation.

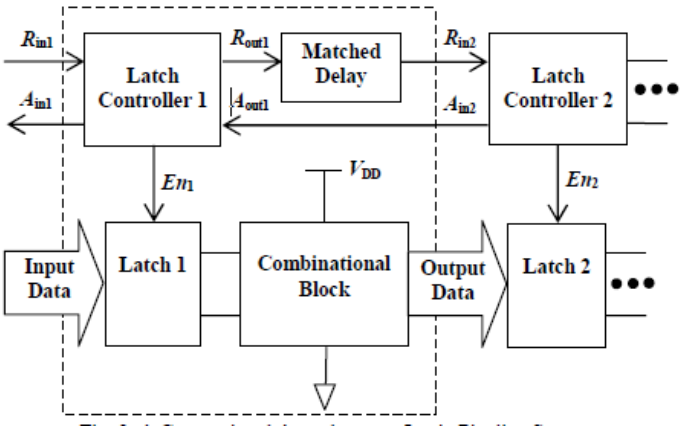


**Figure 1:** Conventional Asynchronous-Logic Pipeline Stage

When the input data is ready, a request signal, Rin1, will be asserted. Seeing that, Latch Controller 1 will enable Latch 1 to capture the input data through En1 and raise the output request signal Rout1 as well as the acknowledge signal Ain1. It will then wait for Rin1 to be de-asserted, and respond by de-asserting Ain1. While the latched data of Latch 1 is being processed by the Combinational Block, the output request signal Rout1 is also passing down a Matched Delay element. The delay of the Matched Delay element is matched to at least the worst-case delay of the Combinational Block. This is necessary to ensure the computed data values and the output request signal arrives at the input of the next stage (Latch Controller 2 and Latch 2) around the same time, and trigger a new round of handshaking events.

## III. Automatic Power Regulation Based On Asynchronous Activity Detection

This paper presents an innovative solution to detect incoming asynchronous activity, which associated to an automatic power regulation [10], efficiently reduces the supply voltage and thus the leakage power. In order to control the leakage of an asynchronous unit shown in Figure 2, a voltage regulator is used in order to power down the asynchronous logic unit when in standby mode. Activity detection on the incoming and outgoing channels is performed using channel monitors. When no more input and output activity is detected, the

voltage regulator powers down the asynchronous logic unit in standby mode for reducing the leakage power. When new incoming activity is detected, the voltage regulator powers up the asynchronous logic in normal mode, without any additional software control and at minimal latency cost.

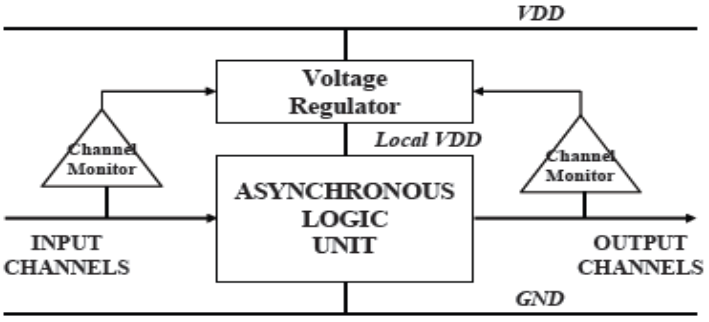


**Figure 2:** Asynchronous Activity detection scheme

Due to their robustness to operating conditions, asynchronous circuits can be easily supplied at low voltage for power reduction. Nevertheless, many issues need to be addressed in this simple proposal. The proper protocol must be chosen in order to detect traffic with a fast and reliable detection. The second constraint is regarding the voltage regulator and the definition of the standby mode.

## IV. Zero- Delay Ripple Turn On Power Gating (ZDRTO)

Many asynchronous circuit families are robust to a wide range of supply voltages, ambient-temperatures, and process variations. To exploit this robustness in the context of power gating the ZDRTO power gating technique [11] is used. In this technique for a zero-delay wakeup scheme in pipelined computation: the first token traveling through a pipeline turns on downstream pipeline stages, hiding the latency cost of wake up in the computation time of upstream pipeline stages. Asynchronous N pipeline stages are grouped into clusters, each with its own local $g_{vssv}$ and $g_{vddv}$ power nets and associated sleep transistors, allowing us to power gate each cluster individually, as shown in Figure 3.

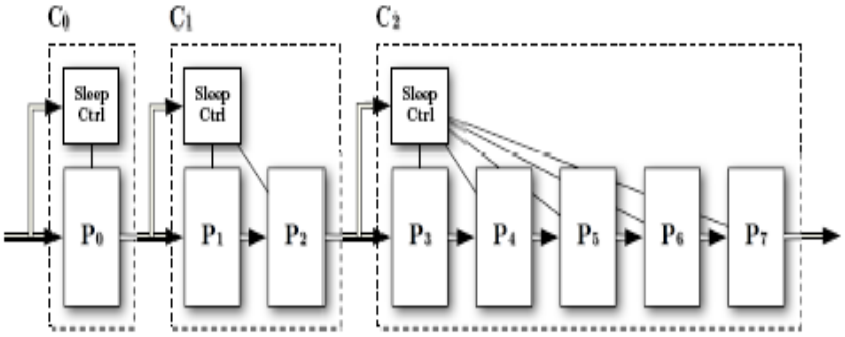


**Figure 3:** Block diagram of our Zero-Delay Ripple Turn on (ZDRTO) power gating control scheme. A sample pipeline of 8-stages is divided into three unequal clusters: C0, C1, and C2. Each cluster controls the power gating of the next inline cluster. With respect to Eq. 1, j = i + 1.

The ripple turn on effect occurs upon arrival of an input token to program P. At this time, we wake up the first cluster, which wakes up the second cluster, and so on. This continues as the token travels through the pipeline with cluster i waking up cluster j, until the last cluster is active. Note that i and j do not have to be consecutive clusters—a token arriving at cluster i could potentially wake up the next few clusters.

In order to achieve the "zero-delay" effect, the cluster grouping should be chosen so that the forward propagation delay, $t_{fp}(i, j)$, from cluster i to j hides the latency, $t_w(j)$, of waking up cluster j, as seen in Eq. 1.

$$t_w(j) \leq t_{fp}(i, j) \quad \forall \{ i, j \mid i < j \} \qquad \text{---------- 1}$$

Achieving this requirement is not difficult in modern processes, especially for low duty cycle pipelines. Note that the value of $t_w$ is variable, as asynchronous circuits have a wide operating voltage range. Furthermore, by selecting different power gating techniques the value of $t_w$ is coarsely tunable. A conservative

choice of $t_w$ such that $g_{vssv}$ and $g_{vddv}$ are equal to GND and VDD, respectively, for any particular cluster by the time the first token arrives—with the exception of the first cluster—ensures each cluster is ready to perform useful computation the moment data arrives. This is the origin of the "zero-delay" latency hiding effect. A more aggressive choice of $t_w$ such that $g_{vssv}$ > GND and $g_{vddv}$ < VDD results in additional power savings at the cost of a longer forward propagation delay of the first tokens for that cluster—and a longer pipeline latency overall. Correctness and stability are conserved, so long as $g_{vssv}$ and $g_{vddv}$ have reached safe values when $t_w$ has elapsed.

As discussed in section, to implement our Zero-Delay Ripple Turn on (ZDRTO) power gating control scheme, it must organize the pipeline stages into clusters and the clusters are simply the different operations of the AES round computation described earlier, each of which is a pipelined computation. BS and SR are transformations on individual bytes, by slicing the data path in 8-bit chunks, and could swap their ordering with no effect on correctness and swap them now because the BS operation has a higher transistor count, as seen in Table I, and thus takes a longer time to wake up. Furthermore, reordering the BS and SR stages also allows for hardware reuse between encryption and decryption. The final pipeline stage clustering is as follows: AK, SR, BS, and MC.

INTERLEAVED COUNTER OVERHEAD

| | Transistor Count | Static Power (nW) |
|---|---|---|
| Additional Bit | 400 | 19 |
| Constant Overhead | 1900 | 95 |

To fully implement power gating in a pipeline, it needs empty pipeline detection in the form of interleaved empty pipeline detection counter. The total depth of our AES round pipeline is 10 half-stages, so we use a 4-bit interleaved counter. The overheads added by the counter are summarized in above Table-I for our 90nm process, broken up by the overhead of adding additional bits and the constant overhead of the counter arbitration and control circuitry. The average operating frequency is relatively low-50MHz in 90nm. Given these characteristics, interleaved counter is suitable for deep low energy pipelines.

## V. Asynchronous Adiabatic Logic

The basic idea of asynchronous adiabatic logic (AAL) [12] is illustrated following. For the initial testing, 2N-2N2P logic [5] has been used as the logical block and many different kinds of simple and complex C&R structures has been designed and tested to get the best power efficiency out of the AAL system.

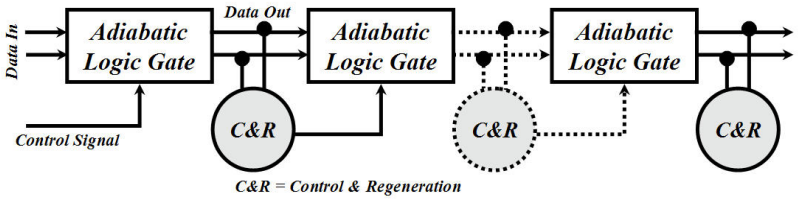


**Figure 4:** Block diagram of AAL

A simple implementation of the AAL is depicted in Figure 4. It is a chain of inverters, with the logical part designed using 2N-2N2P logic, where as the control part of the C&R block is made of pass gate logic and regeneration part is made of conventional CMOS logic. Each stage in an AAL circuit consists of an adiabatic gate, which implements the logic function of this stage, and a control and regeneration (C&R) block, whose output supplies power to the associated adiabatic logic gate. When the C&R block detects that the input to the adiabatic gate becomes valid, the output of the C&R block transits to HIGH, and the adiabatic gate can acquire power to evaluate its output; when the C&R block detects that the input to the adiabatic gate becomes empty, the output of the C&R block transits to LOW, and the adiabatic gate is not powered and becomes idle. Test chip consists of AAL logic circuits has been designed and fabricated in CMOSP18 0.18um technology as a proof of concept. It consists of closed loop chain of different adiabatic NAND gates as the logical block and conventional CMOS OR as C&R block.

Optimizations of the logic delays and more precise control over the slope of the control signal can make the AAL a more power efficient logic. Evolution of devices with technologies such as SOI or MEMS based switches to limit the leakage currents at lower speeds will be very helpful for the power efficient operation of our proposed design and the efficient low frequency operation of adiabatic logic circuits.

## VI. Asynchronous Fine-Grained Power Gated Logic With PCR

This paper has proposed the AFPL. In the AFPL circuit, the logic blocks become active only when performing useful computations, and the idle logic blocks were not powered and have negligible leakage power dissipation. With fine-grain power gating, the AFPL approach has more opportunities to reduce leakage at run-time than other coarse-grain power gating techniques. The AFPL [13] circuit employs ECRL logics to construct its logic blocks to avoid the occurrence of the short-circuit current from VDD to the ground, and to eliminate the requirement for additional standalone pipeline latches. Several logic families, including ECRL, 2N-2N2P, IPGL, PFAL, and DTGAL, can be used to construct the function blocks in the AFPL circuit. The PCR mechanism can be incorporated in the AFPL circuit to form the AFPL-PCR circuit. With the PCR mechanism, part of the charge on the output nodes of a discharging ECRL logic gate can be reused to charge another ECRL logic gate about to evaluate, reducing energy dissipation required to complete the evaluation of an ECRL logic gate. The AFPL-PCR pipeline uses the enhanced C∗-element in its handshake controllers such that an ECRL logic gate in the AFPL-PCR pipeline can enter the sleep mode early to reduce leakage dissipation once its output has been received by the downstream pipeline stage. Two techniques of circuit simplification have been developed to mitigate the hardware overhead of the AFPL circuit.

For the AFPL-PCR implementation of an eight-bit five-stage pipelined Kogge–Stone adder, the handshake controllers and PCR units account for 14% of the total transistor count. Compared with the static CMOS counterpart, the AFPLPCR implementation of the Kogge–Stone adder can reduce power dissipation by 30.6%–55.3% for an input data rate ranging from 30MHz to 900MHz. Moreover, the AFPL-PCR implementation of the Kogge–Stone adder can reduce static power dissipation by 85.5% while in idle mode. Compared with the asynchronous PS0 pipeline counterpart, the AFPL-PCR implementation of the Kogge–Stone adder can reduce static power dissipation by 90.3% when the adder has no valid inputs, and reduce power dissipation by 82.6%–93.0% when the input data rate ranges from 30 to 900 MHz Although the AFPL-PCR implementation has the advantage of lower power dissipation, it suffers the problem of a lower maximum sustainable throughput rate. Compared with the PS0 counterpart, the AFPL-PCR implementation of the Kogge–Stone adder has a performance loss of 25%. Simulation results have shown that the AFPL-PCR circuit is robust to process, supply voltage, and temperature variations.

## VII. Comparative Study

This paper presents a comparative study of different techniques to reduce the leakage current. In conventional asynchronous four phases bundled data logic pipeline is a simple technique to reduce power dissipation; however the hardware overhead is large and still suffers with leakage dissipation. In AAL circuit consists of an adiabatic gate which implements the logic function and control and regeneration block supplies output to the associated adiabatic gate. In this the synchronization between neighboring stages is accomplished via a unidirectional control signal rather than bidirectional handshake signal so it suffers with diverse propagation delay. When voltage regulator in an ANOC node detects to reduce the leakage power. However specific power domains must be chosen in order to detect traffic with a fast and reliable detection. The second constraint is regarding the voltage regulator and the definition of the standby mode. In AFPL-PCR, PCR mechanism is combined with AFPL to reduce the energy dissipation required to complete the evaluation of logic block. Compared to the other methods mentioned the AFPL-PCR is the technique which reduces the power consumption in ideal/active logic, however it suffers with lower maximum sustainable throughput rate.

## VIII. Conclusion

As the technology is down scaling, the power dissipation in integrated circuit is dominant. Our aim is to reduce the maximum power with the decrement of area penalty and delay. In recent sub-micron and nano technologies, the static power is more than the dynamic power. We are reducing the leakage power using power gating techniques. This survey paper compares the different techniques used for reducing static power in asynchronous circuits.

## References

[1]. G Tellez, A Farrahi, and M Sarrafzadeh. Activity-driven clock design for low power circuits. ICCAD, pages 62 – 65, Nov 1995.
[2]. Fatih Hamzaoglu and Mircea Stan. Circuit-level techniques to control gate leakage for sub-100nm cmos. ACM ISLPED, Aug 2002.
[3]. K Roy, S Mukhopadhyay, and H Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits. Proc. IEEE, 91(2):305 – 327, Feb 2003.
[4]. A Keshavarzi, K Roy, and C Hawkins. Intrinsic leakage in low power deep submicron cmos ics. IEEE ITC,pages 146 – 155, Nov 1997.
[5]. K Shi and D Howard. Sleep transistor design and implementation – simple concepts yet challenges to be optimum. IEEE VLSI-DAT, pages 1 – 4, Apr 2006.
[6]. [6] Youngsoo Shin and Hyung-Ock Kim;. Cell-based semicustom design of zigzag power gating circuits. IEEE ISQED, pages 527 – 532, Mar 2007.

[7]. M Imai, K Takada, and T Nanya. Fine-grain leakage power reduction method for m-out-of-n encoded circuits using multi-threshold-voltage transistors. IEEE ASYNC, pages 209 – 216, May 2009.
[8]. Tong Lin, Kwen-Siong Chong, Bah-Hwee Gwee, and J Chang. Finegrained power gating for leakage and short-circuit power reduction by using asynchronous-logic. IEEE ISCAS, pages 3162 – 3165, May 2009.
[9]. Tong Lin, Kwen-Siong Chong, Bah-Hwee Gwee, and J Chang. Finegrained power gating for leakage and short-circuit power reduction by using asynchronous-logic. IEEE ISCAS, pages 3162 – 3165, May 2009.
[10]. Y Thonnart, E Beigne, A Valentian, and P Vivet. Automatic power regulation based on an asynchronous activity detection and its application to anoc node leakage reduction. IEEE ASYNC, pages 48–57, 2008.
[11]. C. Ortega, J. Tse, and R. Manohar, "Static power reduction techniques for asynchronous circuits," in Proc. IEEE Symp. Asynchronous Circuits Syst., May 2010, pp. 52–61.
[12]. M. Arsalan and M. Shams, "Asynchronous adiabatic logic," in Proc. IEEE Int. Symp. Circuits Syst., May 2007, pp. 3720–3723.
[13]. Meng-chouChang,Wei-HsiangChang "Asynchronous Fine-GrainPower-Gated Logic"proc.IEEE.VOL.21.no.6,JUNE2013.