EFFECTIVE CONTROLLER IN OPTIMIZED ASYNCHRONOUS LOGIC

P.Sudha¹, P.Kavitha²
¹Faculty, department of ECE, M.A.M School of Engineering, Siruganur, Tamilnadu, India,
²Faculty, department of ECE, M.A.M School of Engineering, Siruganur, Tamilnadu, India,

Abstract
Asynchronous Fine-grain power gated Logic (AFL) which includes Modified Efficient Charge Recovery Logic (M-ECRL) gates to implement the logic function of the stage with a handshake controller which comprises of C-element to handle the control signals with the neighboring stages and provides power to M-ECRL gate. AFL adopts an partial charge reuse (PCR) mechanism, part of the charge on the output nodes of M-ECRL gate which entering the discharge phase can be used to charge the output nodes of another M-ECRL gate which is more enough to complete evaluate phase, thus reducing the power consumption. To design the efficient asynchronous styles with effective controller from their available CMOS topology. Moreover, study is to scrutinize the use of different controller implementations in a single design in order to generate hybrid and optimized designs.

Index terms - Asynchronous circuits, logic gates, low-power electronics, power gating.

I. INTRODUCTION
The interest in non-synchronous circuits is increasing. The International Technology Roadmap for Semiconductors (ITRS) describes a clear need for asynchronous communication protocols for control and synchronization in integrated circuits (ICs) for the next decades. The power dissipation has become an important concern in nano scale CMOS VLSI design as the feature size continues to shrink and the corresponding transistor density increases. In CMOS circuits, the power dissipation can be categorized into dynamic dissipation and static dissipation. Dynamic power is the power dissipated when the device is active, and static power is the power dissipated when the device is powered up but no signals are changing their values [9]. Dynamic power consists of the switching power, caused by charging and discharging of load capacitance, and the internal power, caused by short-circuit current and the currents needed to charge the internal capacitance of the cell. Static dissipation results from leakage currents and the main source of leakage include sub-threshold leakage, gate leakage, gate - induced drain leakage, and junction leakage. As threshold voltage, channel length, and gate oxide thickness continue to shrink, leakage dissipation is becoming a significant contributor to the total power dissipation. In a nanometer CMOS circuit, leakage power can constitute as much as a third of total power.

Common implementation methodologies of power-gating techniques include multi threshold CMOS, boosted-gate CMOS, super cut-off CMOS, variable threshold CMOS, and zigzag super cut-off CMOS. For asynchronous circuits, power gating can be implemented in the fine-grain or coarse-grain manner. The fine-grain power gating approach has more opportunities to reduce leakage at run-time than the coarse-grain power gating approach. Common implementation methodologies of power-gating techniques include multi-threshold CMOS [6], boosted-gate CMOS [1], super cut-off CMOS [3], variable threshold CMOS, and zigzag super cut-off CMOS. First, fine-grain power-gating needs significant buffering and routing resources to distribute the sleep control signal to all the cells in the synchronous system [8]. Second, at the wakeup, the local supply voltage of a cell must reach a nominal value before inputs can be applied to the cell in order to prevent the synchronous system from violating the timing requirements [8], in order to supply the worst case current required by the cell, the power switch (sleep transistor) can be up to three times as large as the rest of the cell.

Therefore, most of today’s power-gated synchronous designs use coarse grain power gating. Nevertheless, a coarse-grain power-gated synchronous system has some disadvantages. Although asynchronous circuits in inactive mode have no dynamic dissipation, they still suffer leakage dissipation. A novel low power logic family called asynchronous fine-grain power gated logic (AFL) can achieve fine-grain power gating to mitigate static power dissipation without superfluous hardware overhead.

II. MODIFIED ECRL LOGIC GATES
The functional block of proposed APL circuit is replaced to an Modified Efficiency Charge Recovery Logic (M-ECRL) gate. Since ECRL has some drawbacks. The ECRL circuit uses four phase power clocks. The problem of multi-phase clocking charge recovery circuits include clock skew, complicated

www.ijsret.org
power clock tree and multi-phase clock generators that increasing energy dissipation.

Moreover, in order to maintain pipeline operation, some buffers must be inserted in ECRL circuits which result in extra area overhead and increase the complexity of layout place and route. From this discussion the Improved Efficiency Charge Recovery Logic (M-ECRL) gate is proposed here. The structure of Improved M-ECRL circuits shown in Fig 1. In the M-ECRL circuits, combinational logic blocks are supplied by a single-phase power clock. The schematic of the M-ECRL circuit is the same as ECRL. All combinational logic blocks of the M-ECRL circuits are driven by a single phase power clock while is unlike as the basic ECRL ones. The operation of the M-ECRL can be separated into two processes: Evaluation (t1) and recovery (t2). In the evaluate phase, the voltage of Vpi ramps up from 0 V to VDD, and gate Gi can draw current from Vpi and begin to evaluate its outputs.

In the recovery phase, the voltage of Vp ramps down from VDD to 0V, the charge on the output nodes is transferred back to power node Vp, and the output eventually become empty (i.e., out.t and out.f both become LOW).

III. C-ELEMENTS

A. Overview

To enable most non-synchronous styles, the C-element is a fundamental device that has to be available as logic primitive. A recently proposed design flow improved a standard cell library, adding to it a set of typical asynchronous cells. To achieve this, it adds a new degree of freedom to cell design. The new standard cell set encompasses over 500 different C-element implementations. Most of the asynchronous design techniques proposed to date require devices other than the ordinary logic gates and flip-flops available in current standard cell sets. These include e.g. metastability filters, event fork, join and merge devices, C-elements, etc.
This work presents a comparison between the three C-element mentioned implementations. The comparison considered layout effects on a DSM technology, together with the systemic effect of using these C-elements in building asynchronous cores. The results obtained show that previous findings for the electrical behavior of C-elements must be reevaluated.

IV. PCR MECHANISM AND HANDSHAKE CONTROLLER.

PCR mechanism and the C-element are presented in AFL-PCR pipeline with three stages. There are two main differences between AFL-PCR and AFL without PCR Fig.5 (a) and (b). First, AFL-PCR employs the PCR unit PCR_{i+1} to control charge reuse between pipeline stages S_i and S_{i+2}. Second, the handshake controller HC_i in AFL-PCR employs a C-element to control the power node Vp_i of the associated M-ECRL gates. The C-element offers the advantage that an M-ECRL gate can achieve early discharging if its outputs are no longer required, without waiting for the next empty token to arrive at this stage.

![Diagram of PCR mechanism and handshake controller](image)

The C-element in HC_i has three inputs, R_i, A_{ini}, and A_{ini}, the latter two of which are complementary. R_i is the request signal from the CD in HC_i. A_{ini} and A_{ini} are the acknowledge signals from HC_{i+2}. After reset, R_i = 0, A_{ini} = 0, and A_{ini} = 1. The transitions of R_i and A_{ini} involve the following four events.

1) Event Req↑: R_i transits from LOW to HIGH. This event occurs when a valid token arrives at stage S_i.
2) Event Ack↑: A_{ini} transits from LOW to HIGH and A_{ini} transits from HIGH to LOW. This event occurs when the valid output of stage S_i has been received by stage S_{i+2}.
3) Event Req↓: R_i transits from HIGH to LOW. This event occurs when an empty token arrives at stage S_i.
4) Event Ack↓: A_{ini} transits from HIGH to LOW and A_{ini} transits from LOW to HIGH.

Thus, the M-ECRL logic gates in S_i can enter the discharge phase to achieve early discharging as soon as the valid output of stage S_i has been received by stage S_{i+2}. In the AFL-PCR pipeline, the arrival of a valid token at stage S_{i+1} forces stage S_i to discharge and turns on the switch in PCR_{i+1}. Part of the charge on the output nodes of gate G_i is reused to charge the output nodes of gate G_{i+2} to reduce energy dissipation. The use of the C-element makes it possible to synchronize the discharging of gate G_i with the evaluating of gate G_{i+2}, and to have gate G_i enter the sleep mode early to further reduce static power dissipation.

V. RESULTS AND DISCUSSIONS

Figure 6 shows details the power consumption of the C-elements from the total power consumed by the placed and routed netlists. The total power consumed by the Sutherland, the weak feedback and the van Berkel C-elements, in their respective netlists, was 54%, 73% and 54% respectively. These results show that, in a realistic application, the reason for the Sutherland C-element to consume less power than the van Berkel is because the internal power consumed by the latter is more significant. One aspect that worsens internal power consumption is the amount of transistors in short circuit when switching the van Berkel C-element output logical value.

![Power consumption chart](image)

Van Berkel C-element appears as the lowest static power consuming implementation, it has been shown to consume more dynamic power than the Sutherland C-element for bigger input slopes. Moreover, the dynamic power consumption represents a bigger portion of the
total power than the static power consumption. In this way, the Sutherland C-element appears as the most indicated for low power designs, regardless of input slope variations. The weak feedback presented the worst propagation delay, regardless output load or input slope variations. Moreover, the van Berkel and the Sutherland C-elements proved to be equally robust to output load variations. Therefore, the van Berkel C-element appears as the most speed efficient implementation. The results on required area for each C-element showed that the van Berkel implementation is the most silicon area consuming, while the weak feedback is the most area efficient. Hence, the latter is the most suitable for high density designs. Finally, the work described here shows that the choice of C-element type in a DSM asynchronous design is a triple (speed/area/power) tradeoff.

In this section simulation is performed with microwind tool. The simulation results are carried out for AFL without PCR and AFL-PCR pipeline with three stages are discussed here. In this fig.7, the Vp gets high value when both the input of C-element are high. The empty token has not arrived at the input stage and output of the logic gates are no longer required i.e. ack bar equals 1, it maintains the previous Vp value. The logic gate performs the function when the value of Vp goes to high and produces output. The simulation results of AFL without PCR pipeline is shown in fig.7. This figure describes three AFL without PCR are cascaded and creating a pipeline chain. If the ack bar and Ri is high, the C-element produces high value which activates the logic gate and produces high output.

The fig.8 shows the simulation result of AFL-PCR. It is similar to AFL-PCR, even though one difference is however empty token is not received at the input stage and the output stages are no longer required i.e. ack bar equals 1, the output of Vp goes to discharge state. This is also one of the advantages of AFL-PCR.

The simulation results of APL-PCR pipeline is shown here. If the valid token reaches the stage 3 the PCR unit is activated and output of Vp rises to high value, which activates the logic gate. Then the logic produces output value either as high or low corresponding to the input token.

VI. CONCLUSION

In designing of AFL 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 AFL approach has more opportunities to reduce leakage at run-time than other coarse-grain power gating techniques. The AFL circuit employs M-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. The proposed C-element is implemented in AFL circuit and its power consumption is compared with the existing C-element in AFL circuit. Although the AFL implementation has the advantage of lower power dissipation, it suffers the problem of a lower maximum sustainable throughput rate. The proposed AFL circuit with C-element, which is more efficient compared to other existing method. Power consumption is reduced by using the C-element in AFL circuit compared to the existing method.
REFERENCES


