# Clock Domain Crossing Aware Sequential Clock Gating

Jianfeng Liu, Mi-Suk Hong, Kyungtae Do,
Jung Yun Choi and Jaehong Park
S. LSI, Samsung Electronics Co. Ltd.
Hwasong-Si Korea
{jfliu,ms05.hong,kyungtae.do,jungyun74.choi,jaehongp}@samsung.com

Mohit Kumar, Manish Kumar, Nikhil Tripathi and
Abhishek Ranjan
Calypto Design Systems Inc.
Noida, India
{mohitk,mkumar,ntripathi,aranjan}@calypto.com

***Abstract* — Power has become the overriding concern for most modern electronic applications today. To reduce clock power, which is a significant portion of the dynamic power consumed by a design, sequential clock gating is increasingly getting used over and above combinational clock gating. With the shrinking device sizes and increasingly complex designs, data is frequently transferred from one clock domain to the other. The sequential clock gating optimizations can use signals from across sequential boundaries and thus, can introduce new clock domain crossing (CDC) violations which can cause catastrophic functional issues in the fabricated chip. Hence, it has become very important that sequential clock gating optimizations be CDC aware.**

**In this paper, we present an algorithm to handle CDC violations as part of the objective function for sequential clock gating optimizations. With the proposed algorithm, we have obtained an average of 22% sequential power savings — this is within 3% of the power savings obtained by the CDC unaware sequential clock gating. In comparison, the state-of-the-art two-pass solution is leading to an almost complete loss of power savings.**

***Keywords— Clock Domain Crossing, Sequential Clock Gating, Sequential Analysis, Sequential Optimization, Observability, Stability, Power Analysis, Power Optimization.***

## I. INTRODUCTION

Reducing power consumption in a semiconductor device is becoming one of the most important design criteria. It has been suggested that power will be the limiting factor when determining the maximum number of applications that can simultaneously be active [1] and not just the amount of functionality that can be packed in a single die as governed by Moore's law [2]. Clock and register power is one of the most power consuming components in our designs today. To reduce clock power, clock gating is used to gate the clocks when writing into the register is redundant [3]. Sequential clock gating, where design behavior is analyzed across multiple cycles to identify redundant writes into a register, has emerged as a very powerful technique to identify new clock gating conditions in the design [4][5][6]. To reduce manual effort, there are solutions [7], which can automatically identify and modify the RTL to insert new clock gating conditions based on sequential clock gating analysis.

Today's SOC systems have a multitude of components with multiple interfaces. These components are working with multiple asynchronous clock domains running at varying speeds. Major sub-blocks of the SoCs are designed to run on independent clocks to ease the problems of clock skew across large chips. The clock domains are originated from different clock sources or derivatives of those. As a result timing of the asynchronous clock domain crossing paths cannot be accurately verified since the order of the clock edges cannot be guaranteed. These asynchronous clock domain crossings (henceforth referred to as CDC) are termed as CDC violations [8][9][10].

As sequential clock gating techniques analyze the design across multiple cycles, new CDC violations may be introduced in the design. Hence, it is required that the sequential clock gating techniques should be CDC aware and should not introduce any new CDC violations. It is also essential that the power saving provided by sequential clock gating optimization is not compromised when handling CDC violations.

In this paper, we first define the state-of-the-art solution to remove CDC violations introduced by the sequential clock gating optimizations in the designs. This is a two-pass process which constitutes performing sequential clock gating optimizations and then removing those clock gating optimizations which are causing new CDC violations in the design [7]. This process has the potential of causing severe loss in power savings. To overcome the limitation of two-pass flow, we propose a method to handle CDC natively in the sequential clock gating optimization algorithm. This requires the CDC to be modeled as an objective function for the clock gating optimizations.

In the subsequent sections, we will discuss types of sequential clock gating optimizations, types of CDC violations and the need for CDC awareness in sequential clock gating optimizations. We will then discuss the two-pass process to remove CDC violations and its associated limitations. Then, we propose a solution for modeling CDC as part of power optimization objective. In the results section, we will show that the proposed methodology not only provides significantly higher power savings compared to the two-pass flow but also produces a CDC clean RTL. Finally, we will conclude by summarizing our findings and provide directions for further work.

978-3-9815370-4-8/DATE15/©2015 EDAA

## II. Prior Work

Clock Gating is one of the most frequently used techniques in RTL to reduce dynamic power consumption without affecting the functionality of the design [3]. It involves inserting gating conditions in the RTL, which the synthesis tool [13][14] translates to clock gating cells in the clock-path of a register bank. This helps to reduce the switching activity on the clock network thereby reducing dynamic power consumption in the design. Since the translation is purely combinational, it is also referred to as Combinational Clock Gating.

Sequential clock gating on the other hand uses multi-cycle analysis of the design to identify writes that are either unobservable down-stream or the same value is written in consecutive cycles. The first type of redundant writes are called Observability Based Clock Gating and the second type of redundant writes are called Stability Based Clock Gating [5]. Fig. 1 shows an example of writes to the register which, under some conditions are never going to be observed at the design output.

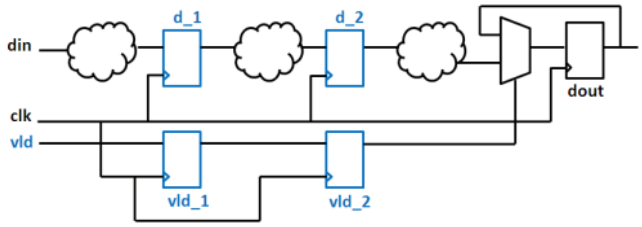


Fig. 1. Unobservable Writes

If the signal *vld_2* is low in a particular cycle, the register *dout* retains its older value. This means that the writes that would have happened in register *d_2* one cycle back and *d_1* two cycles back are redundant. Observability based clock gating would identify this redundant write and add a suitable gating condition for the register *d_1* based on the signal *vld* (and similarly based on the signal *vld_1* for the register *d_2*).

Fig. 2 shows an example where the same value gets written to registers across consecutive clock cycles.

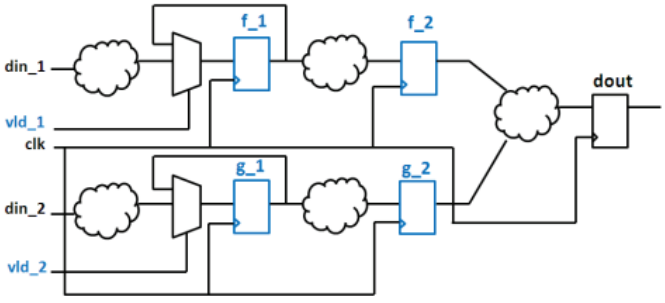


Fig. 2. Stable Writes

When the signals *vld_1* and *vld_2* are low, the registers *f_1* and *g_1* retain their previously held values. Consequently, values written to registers *f_2* and *g_2* in the next clock cycle are identical to what was written to them in the previous cycle. Stability based clock gating would identify this redundant write and add a suitable gating condition using one-cycle delayed version of *vld_1* for *f_2* (and similarly one-cycle delayed version of *vld_2* for *g_2*).

A clock domain crossing [9] occurs whenever data is transferred from a register driven by one clock domain (launch) to a register driven by another clock domain (capture). This is illustrated in Fig. 3, where the signal *B* is launched by a register in the clock domain *CLK1* and captured by a register in the clock domain *CLK2*. Asynchronous clock domain crossings are those where the launch and the capture clock domains have no constant phase and time relationship. Transferring signals between asynchronous clock domains may lead to setup and hold time violations of registers.

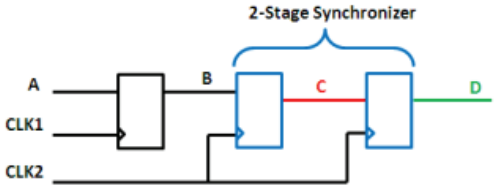


Fig. 3. Clock Domain Crossing

These violations can cause meta-stability in the design [12]. The signal *C* is termed as a meta-stable signal. These kinds of signals are unstable signals which take long to reach a valid logic value. The meta-stability issues can be mitigated by using synchronizers [11] in the capture clock domains. A commonly used synchronizer is the 2-stage synchronizer shown in Fig. 3. The signal *D* is termed as a *synchronized signal* which is synchronized with respect to the clock edge of the capture clock domain and can be safely used in the logic downstream.

The other important categories of CDC violations are divergence of meta-stable signals and re-convergence of the synchronized signals [8][9][10]. In Fig. 3, had the signal *C* been feeding other logic as well, it would have been flagged as a *divergence of meta-stable signal* type violation.

Fig. 4 shows an example of *re-convergence of synchronized signals* violation. The synchronized signals *D3* and *C3* are converging and could cause functional issues in the chip. We are not mentioning all types of CDC violations to maintain the brevity of the paper.

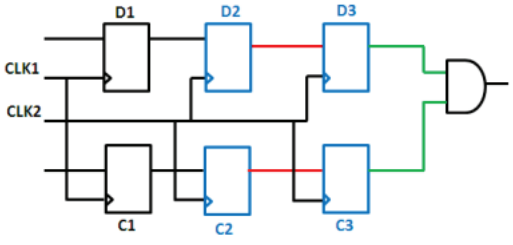


Fig. 4. Re-convergence of synchronized signals

## III. Issues With CDC Unaware Sequential Clock Gating

Sequential clock gating optimizations require reasoning about the behavior of the design across multiple cycles. This

often involves tracing signals across multiple design hierarchies. In the complex designs with multiple clock domains, it is possible that new gating condition uses signals from different clock domains. This could introduce new CDC violations in the design.

In Fig. 1, if the signal *vld* was in an asynchronous clock domain to the clock domain *clk*, a CDC unaware observability based clock gating would still add a gating condition for the register *d_1* based on *vld* signal. This would introduce a new asynchronous CDC violation on the register *d_1*. Similarly, in the Fig. 2, if the signal *vld_1* was in an asynchronous clock domain to *clk*, a CDC unaware stability based clock gating optimization would still add a gating condition using a one cycle delayed version of *vld_1* for the register *f_2*. Again, this is a new CDC violation on *f_2*.

Fig. 5 shows an example where sequential clock gating optimization can introduce re-convergence of the synchronized signals. The signals *vld_1* and *vld_2* are synchronized control signals in the clock domain *CLK2*. A CDC unaware stability based clock gating optimization would add a gating condition for the register *dout* using a one cycle delayed version of the signals *vld_1* and *vld_2*. This would introduce a re-convergence of the synchronized signals on the register *dout*.

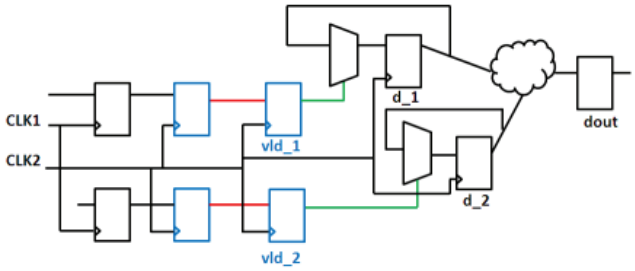


Fig. 5. CDC Unaware Stability Optimization

In Fig. 6, the signal *sync_meta* is a meta-stable signal. CDC unaware observability based gating optimization would make use of this signal to generate a gating condition for the register *d_2*. This clock gating condition would introduce a divergence of the meta-stable signal which is a CDC violation.

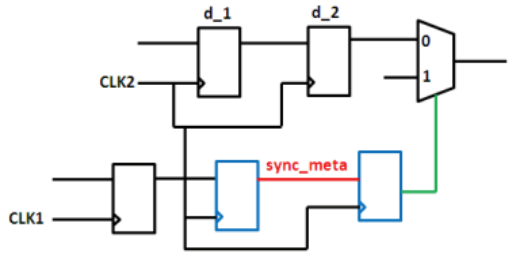


Fig. 6. CDC Unaware Observability Optimization

Clearly, sequential clock gating should not be creating structures which will lead to CDC violations in the design. However, care has to be taken that power savings achieved by sequential clock gating is not adversely impacted.

## IV. State-of-the-Art Solution for Removing CDC Violations

While determining the clock gating condition, signals from multiple clock domains can end up participating in the clock gating condition. This would create CDC violations in the power optimized RTL. In this section, we describe the state-of-the-art solution for getting a CDC clean power optimized RTL— this is a two-pass process.

In this two-pass process, the CDC violations are removed from the RTL design as a post-process of the sequential clock gating. Fig. 7 shows a traditional RTL design flow which makes use of the violations reported by CDC checking tools [15][16][17] to remove the clock gating conditions which are introducing new CDC violations.

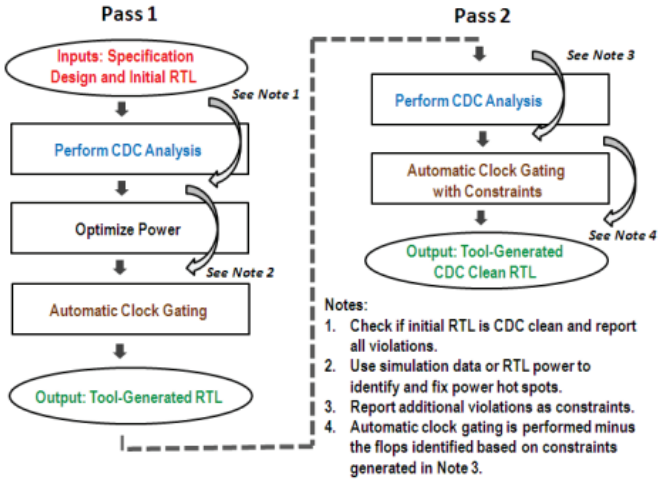


Fig. 7. Two-Pass Flow

In the RTL design flow, the input design specifications and the initial RTL is taken through a CDC checking tool. Based on the information of the clock domains, these CDC tools do the structural and functional analysis of the design. These tools report all the clock domain crossing signals. In addition to these signals, these tools also report CDC violations like re-convergence of the synchronized signals and divergence of the meta-stable signals to name a few [8][9][10].

Sequential clock gating optimizations can be performed using tools like [7]. These tools provide an automated way to implement sequential clock gating in the design. Once the clock gating logic has been inserted into the design, RTL is again taken through the CDC checking tool. Additional CDC violations are reported which are due to the newly inserted clock gating logic. The clock gating logic responsible for the additional CDC violations can be removed by providing constraints to the automated clock gating optimization tool [7].

The two-step flow is an iterative way of removing clock gating conditions. A major limitation of this method is that it could end up removing several clock gating conditions, thereby bringing down power savings drastically. Additionally, it is also runtime intensive; the process requires running the sequential clock gating tool and the CDC tool multiple times to get a CDC clean design.

Fig. 8 illustrates the first limitation by using observability based clock gating as an example. The clock domain of the registers *d_1* and *d_2* is *CLK1*. The complete observability based clock gating condition for the register *d_1* constitutes one cycle earlier values of the registers *vld_1* and *cntl_1*. The one cycle earlier value of the register *cntl_1* is the signal *cntl* which is in a different clock domain, *CLK2*. The usage of the signal *cntl* in the clock gating condition for the register *d_1* would create a new CDC violation from the signal *cntl* to the register *d_1*.

The two-step process would remove the entire clock gating condition to eliminate the CDC violation. Interestingly, a smaller clock gating condition with the signal *vld* is still possible. It might provide lesser power savings as compared to the complete clock gating condition but is CDC clean.

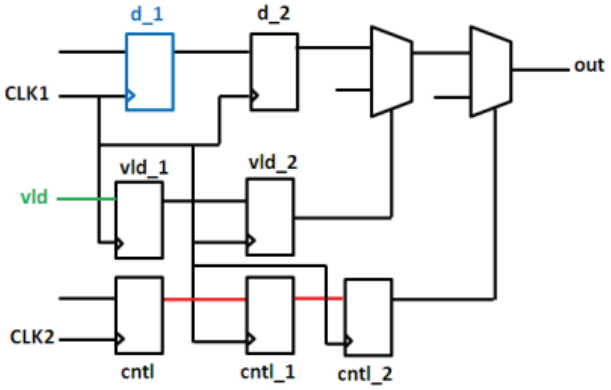


Fig. 8. Complete Loss of Gating Condition in the Two-Pass Flow

In the next section, we propose a new solution to mitigate the limitations of the two-pass flow by modeling the CDC as part of the objective function for the power optimizations.

## V. Proposed Solution

We propose an algorithm to model CDC natively in the cost/benefit analysis of the sequential clock gating optimization algorithms to ensure that the inserted clock gating condition is CDC clean. Out of multiple signals which could lie in different clock domains, gating condition is created to obtain maximum power savings without introducing new CDC violations.

This algorithm would be much faster than the two-step process as it would produce CDC clean clock gating condition in single pass only.

Before delving into the algorithm, it is essential to understand the key requirements to model CDC natively:

- Fast and incremental CDC analysis
- Handling of all types of CDC violations
- Incorporating CDC in the cost function of Sequential Clock Gating

### A. Fast and Incremental CDC Analysis

Given that sequential clock gating can introduce gating logic for large number of registers [6], fast and incremental CDC analysis is of utmost importance. Based on the clock domain specification of the primary input signals, the clock domain of a signal *s* i.e., *CD(s)* can be computed as follows: *The clock domain of a signal driven by a register is inferred as the clock domain of the register. The clock domain of a signal driven by a combinational logic gate is the union of the clock domains of all the inputs of the combinational logic gate.*

Once all design signals are annotated with the appropriate clock domains, it is easy to apply the same procedure to compute the clock domain of the signals for the new clock gating logic generated. This is equivalent to clock domain computation on the design except that it is on a smaller logic with the available clock domains on the support signals. Here, because analysis is performed on the additional clock gating logic, CDC analysis is incremental and fast.

### B. Handling All Types of CDC Violations

Apart from the new asynchronous clock domain crossings, all types of CDC violations should be handled *i.e.,* divergence of meta-stable signals, re-convergence of synchronized signals [8][9][10] etc.

To handle all CDC violations, already existing meta-stable and synchronized signals should be identified. This can be achieved by identifying all types of synchronizer patterns in the design. Most commonly used synchronizers are 2-stage synchronizers, multi-stage synchronizers and MUX-synchronizers [15][16][17]. In all of these synchronizers, specific signals are attributed as meta-stable and synchronized signals. This knowledge can be leveraged to generate a gating condition free of CDC violations.

### C. Incorporating CDC in the Cost Function of Sequential Clock Gating

A typical cost function of sequential clock gating optimization comprises of the area and power impact analysis of the clock gating condition [6]. The area estimation of the gating condition is straight forward. To compute the gate area, gating condition is converted to the form of an AND-OR-INVERT balanced tree and the area of these cells is then computed from the target technology library.

The power estimation is done based on the signal transition densities and probabilities of the support signals of the gating condition [6]. Statistical methods are used to propagate the signal transition densities and probabilities across all the logic gates in the gating condition [18][19]. Using the transition density on each signal of the gating condition, power can be computed easily.

Similarly, CDC analysis can be incorporated as part of the cost function of the sequential clock gating. The meta-stable support signals of the gating condition would need to be eliminated from the gating condition. The support signals pertaining to the asynchronous clock domain as against the target register would either need to be eliminated from the gating condition or can be synchronized to the target clock domain using a synchronizer. The additional area and power of the synchronization logic would then have to be considered in the cost function. In a gating condition, only one synchronized signal can participate; otherwise, it would lead to violation caused by re-convergence of synchronized signals. Of the

synchronized signals, the signal providing maximum power savings is kept as part of the gating condition while others are discarded. The algorithm to model CDC as part of sequential clock gating cost/benefit analysis is described below — along with the inputs required and the output generated.

*Inputs*

- A gating condition as a Boolean function $G(v_1, \ldots, v_n)$;
- Probability [18][19] of each support variable $v_i$ stays true, i.e., $Pr(v_i=1)$, i = 1, .., n;
- Transition Density [18][19] of each support variable $v_i$ i.e. $TD(v_i)$, i = 1, …, n;
- The register in the design which G is targeted for;
- Target technology library;

*Output*

- A boolean function $G'(v_1, \ldots, v_k)$ satisfying:
  - G' => G, i.e., G covers G';
  - Among all candidate functions that are covered by G, G' optimally reduces power consumption related to the register including the gating logic itself;
  - G' would not introduce any new CDC violation;

*Algorithm*

```
Procedure CDC_AWARE_COST_FUNCTION(R, G, TD, Pr, CD)
    // R: Target Register
    // G: Complete Gating Condition with v1,  vn supports
    // TD: Transition Densities of R inputs and G supports
    // Pr: Probabilities of R inputs and G supports
    // CD: Clock Domains of R inputs and G supports
begin
    V := {v1, …, vn}; // Support set of G;
    C := CD(R);      // Clock domain of register R;

    V' = {}; // Support set of G without meta-stable signals;
    PRE_SYNCED = {}; // set of already synchronized signals;
    NEW_SYNCED = {}; // set of new synchronized signals;
    for each support v in V {
        if v is a meta-stable signal {
            Eliminate v from G; // Universal quantification of v
            V'= V – v;
        }
        if v is already synchronized {
            PRE_SYNCED += v;
        }
         if (CD(v) != C) {
            Add synchronizer on signal v;
            NEW_SYNCED += v;
         }
    }

     // Remove all synchronized signals from G to form G';
    G' = G – {PRE_SYNCED + NEW_SYNCED};
    P := ComputeTotalPower(R + G');
    for each support v in V' {
          if v Є {PRE_SYNCED + NEW_SYNCED} {
              Eliminate all synchronized signals other than v from G
              to form G'';
              if v Є NEW_SYNCED {
                  P' = ComputeTotalPower(R + G'' +
                                         synchronizer power);
              } else {
                  P' = ComputeTotalPower(R + G'');
              }
              if  P' < P {
                  P := P';
                  G':= G'';
              }
          }
      }
      return G';
end
```

Fig. 9 illustrates how the above algorithm works to identify synchronized signals that would produce maximum power savings. If the signal *sync1* is low in a particular cycle, a write on the register *d_1* in that cycle would not be observable at the design output; likewise for signal *sync2*.

Assume that the probabilities of signals sync1 and *sync2* being high are 0.2 and 0.8 respectively. The complete observability based gating condition for the register *d_1* would constitute both the signals *sync1* and *sync2* [4][5]. However, only one of these signals can participate in the final gating condition as both these signals are synchronized signals. Based on the probabilities of these signals, it is evident that the signal sync1 (probability 0.2) remains low for more duration as compared to the signal sync2 (probability 0.8). Since the total power of the register *d_1* with the gating condition comprising only of signal *sync1* would be lesser than the gating condition comprising of only signal *sync2*, signal *sync1* would be selected by procedure CDC_AWARE_COST_FUNCTION to participate in the final gating condition.

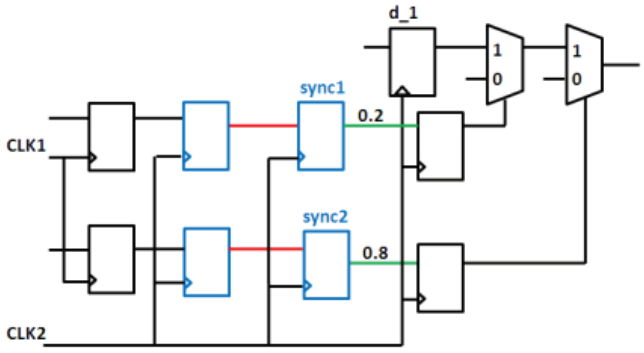


Fig. 9. CDC and Power Aware Sequential Clock Gating

## VI. RESULTS

Ideas discussed in previous sections have been tried on six large design blocks having millions of gates (MG) from a smart phone application. Details of the designs are presented in the first three columns of Table 1. All these designs have three asynchronous clock-domains.

The numbers shown in the "CDC Unaware Flow'' section of the table show the performance (total number of gated registers in KG (Kilo Gates) and their percentage with respect to total design registers, sequential power savings and total runtime) of Sequential Clock Gating tool [7] when it is working in the CDC unaware mode. These numbers provide the baseline for performance of a CDC aware solution.

TABLE I. RESULTS COMPARING PERFORMANCE OF TWO-PASS AND CDC AWARE FLOWS

| | | | CDC Unaware Flow | | | Two-Pass Flow | | | CDC Aware Flow | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Design** | **Size (MG)** | **Total Registers (KG)** | **Gated Registers (KG)** | **Sequential Power Savings** | **Run time (s)** | **Gated Registers (KG)** | **Sequential Power Savings** | **Run time (s)** | **Gated Registers (KG)** | **Sequential Power Savings** | **Run time (s)** |
| DESIGN1 | 8.1 | 204.6 | 48.1(23.5%) | 18.7% | 25059 | 3.0(1.5%) | 1.2% | 52600 | 44.8(21.9%) | 16.8% | 26992 |
| DESIGN2 | 8.4 | 298.7 | 123.7(41.4%) | 26.8% | 59459 | 3.7(1.2%) | 1.1% | 90918 | 119.2(39.9%) | 26.4% | 59954 |
| DESIGN3 | 3.2 | 83.3 | 21.8(26.1%) | 24.8% | 7296 | 2.0(2.4%) | 1.9% | 15218 | 19.1(22.9%) | 22.5% | 7603 |
| DESIGN4 | 2.3 | 54.8 | 17.1(31.2%) | 33.5% | 2737 | 2.7(5.0%) | 3.9% | 5774 | 15.8(28.8%) | 32.8% | 2965 |
| DESIGN5 | 4.8 | 121.5 | 26.4(21.7%) | 17.1% | 5669 | 1.4(1.2%) | 1.8% | 11938 | 24.2(19.9%) | 7.3% | 6186 |
| DESIGN6 | 1.4 | 35.4 | 4.7(13.2%) | 27.2% | 2982 | 4.5(12.7%) | 26.5% | 6264 | 4.5(12.7%) | 26.6% | 2837 |

Power savings are computed by running an industrial Power Analysis tool [7] on the original and sequentially-clock-gated designs. The "Two-Pass Flow" section of the table shows the performance of two-pass solution. The results of the two-pass flow show a *drastic reduction in number of gated registers (almost 10x) as well as sequential power savings*. At the same time, runtime of two-pass flow is about 2x that of CDC unaware flow.

The last section shows the performance of CDC aware flow proposed in Section V. It is clear from the results that modeling CDC as part of sequential clock-gating optimization objective provides much superior performance compared to a two-pass flow. In fact, percentage of gated registers and sequential power savings (22% on average) of CDC aware solution is almost similar to that of CDC unaware flow's sequential power savings (25% on average). In five out of six designs, the reduction in percentage of gated registers and power savings is less than 3%. Results show the clear superiority of our proposed algorithm versus state-of-the-art two-pass flow.

A CDC-checking tool [15] was used on the original designs and outputs of CDC aware flow as well as the two pass flow. There were no violations in any of them. Number of CDC violations in CDC unaware flow can be obtained by subtracting number of gated registers in CDC unaware flow from the number of gated registers in 2-pass flow. Use of signals to create gating expressions from controllers in asynchronous clock domains was causing the CDC violations in a large number of gating expressions in the CDC unaware flow.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we presented the need for handling CDCs as part of sequential clock gating. A new approach was presented where CDCs were natively modeled as a cost function in the power optimization objective. We showed that the approach provides significantly superior power savings compared to the two-pass flow. In fact, the power savings of this approach are comparable to the flow which is not aware of CDC restrictions. One significant aspect of the CDC aware flow is that its runtime is comparable to the CDC unaware flow and is about two times faster than the post-processing based two-pass flow.

Our approach needs to be enhanced to handle designs where there are existing CDC violations. In some cases, it is possible that new clock gating optimizations are dropped even if a CDC violation already existed on the register being gated.

## REFERENCES


[1] J. Markoff, "Progress Hits Snag: Tiny Chips Use Outsize Power," http://www.nytimes.com/2011/08/01/science/01chips.html?_r=2.

[2] Gordon Moore, "Cramming More Components onto Integrated Circuits," Electronics, Volume 38, Number 8, April 19, 1965.

[3] L. Benini and G. De Micheli, "Automatic synthesis of low power gated clock finite state machine," IEEE Trans. Computer-Aided Design, vol. 15, pp. 630--643, June 1996.

[4] J. Sukumar. et al, "Clock gating for power optimization in ASIC design cycle theory & practice," ISLPED, pp. 307--308, 2008.

[5] J. Sukumar, S. Das, A. Ranjan et al, "RTL Power Optimization in Sequential Analysis Platforms," poster presentation in DAC'2010.

[6] N. Vyagrheswarudu, S. Das, A. Ranjan, "PowerAdviser: An RTL Power Platform for Interactive Sequntial Optimizations", Design Automation and Test in Europe Conference (DATE), pp. 550-553 , 12-16 March, 2012.

[7] PowerPro CG, Calypto Design Systems Inc. (http://www.calypto.com/).

[8] S. Chaturvedi, "Static Analysis of Asynchronous Clock Domain Crossings", Design Automation and Test in Europe Conference (DATE), pp. 1122-1125 , 12-16 March, 2012.

[9] Cadence Design Systems, "Clock Domain Crossing: Closing the Loop on Clock Domain Functional Implementation Problems", technical paper. http://www.cadence.com/india/newsletters/icon_2005-05.pdf

[10] C. E. Cummings, "Clock Domain Crossing (CDC) Design and Verification Techniques using SystemVerilog," SNUG 2008.

[11] P. Parakh and S. Kommrusch, "A Smart Synchronizer: Pragmatic way to cross asynchronous clock domains", DVCON, 2011.

[12] C. L. Portmann and T. H. Y. Meng, "Metastability in CMOS Library elements in reduced supply and technology scaled applications," IEEE Journal of Solid-State Circuits, 1995.

[13] Design Compiler, Synopsys Inc. (http://www.synopsys.com/).

[14] RTL Compiler, Cadence Inc. (http://www.cadence.com/).

[15] Name witheld upon request.

[16] Questa CDC, Mentor Graphics Inc. (http://www.mentor.com)

[17] SpyGlass CDC, Atrenta Inc. (http://www.atrenta.com)

[18] F. Najm, "Low-pass filter for computing the transition density in digital circuits," IEEE Transactions on Computer-Aided Design, vol. 13, no. 9, pp. 1123--1131, September 1994.

[19] Michael G. Xakellis, Farid N. Najm, "Statistical estimation of the switching activity in digital circuits," Proceedings of the 31st annual Conference on Design Automation, pp.728--733, June 06-10, 1994