Clock Power Optimizations in VLSI design at Advanced Technology Nodes

Manjunath Rao B. M.
Dept of Electronics and Communication
RVCE, Bengaluru, Karnataka, India

H. V. Ravish Aradhya, PhD
Prof. Dept of Electronics and Communication
RVCE, Bengaluru, Karnataka, India

ABSTRACT
Power reduction in VLSI designs is one of the key design constraints along with others, namely timing, area, quality constraints, noise, etc. Even though there has been a steady growth of devices that are able to be placed in a given area of a chip as per Moore's law, the same cannot be said for battery technologies as they have never been able to catch up. Since the advent of the deep sub-micron era, speed and higher frequency of operation have become the prime goals of any design as the hunger for faster and better optimized systems are never ending. But as a consequence of faster operating speeds which basically means higher clock frequencies, power becomes one of the main constraints to be considered as the most important component of power dissipation, namely the dynamic power dissipation has a proportional relationship with the clock frequency. Hence clock power optimization is taken up as the prime objective of this paper for technologies below 14 nm as at these technologies, other secondary power dissipation components start to become more prominent. Various design techniques have been discussed and applied at both the circuit design and the RTL levels of abstraction in order to provide a complete review of most of the low power design techniques which can be used to reduce power at both these levels of abstraction. An improvement of 25% and 15.7% of power reduction are observed in clock power and overall power respectively. There is also a power reduction of 2-5% for each of the RTL level optimization techniques.

General Terms
Clock power optimization, Moore's law, Deep sub-micron era, Levels of abstraction.

Keywords
Activity factor, Clock tree, Clock gating efficiency, Data aware clock gating efficiency, low activity non enabled register, Regional clock buffer, Local clock buffer.

1. INTRODUCTION
Most of today's technologies and devices wouldn't have been possible without Microelectronic VLSI circuits as almost all modern devices, be it the digital television, mobile phones, microprocessors, microcontrollers, etc are byproducts of it. It is clear that for these devices to be possible to realize, it is essential to either have a good power source as most of these devices are handheld/mobile devices or decrease their power consumption. Hence low power designs with low power dissipation is something that is required in all kinds of markets to make it possible to realize such devices. The basic and most common expressions of power dissipation is as follows

$$\mathbf{P = I^2 * R = V * I}$$

Where I is the current flowing through the device, R is its resistance and V is the voltage across its terminals. It must be clear from the above equation that decreasing any of the three values such as resistance, current or voltage should have a positive impact on power but doing so without the thorough knowledge of VLSI design would lead to a design disaster as these parameters also have a significant impact on other parameters, one of the crucial ones being timing, because even if power consumption is a tad bit more than that intended, it may be alright but if there is a timing failure, then the entire design will be compromised as its functionality becomes uncertain. Hence a good designer must have a thorough knowledge of these parameters before he decides to play around with some of them to reduce power. Though power reduction and speed are the primary objectives in any VLSI design, reaching an optimum performance is the main design goal. Power reduction always comes at a cost of decreased performance as well as some quality issues too but it again depends on the designer's skill to ensure that there is no significant trade-off [1] [2] and it is as low as possible. The best design methodologies can be considered as ones in which the various negative tradeoffs are weighed with the positive advantages before proceeding with their implementation. To create a good balance between them, two or more metrics can be combined to form another one which shows a combined significance of both the metrics. Examples are the power-delay product and the area-delay product which are also considered as standard metrics in today's designs.

In VLSI designs, low power techniques can be classified into the levels at which they can be applied, namely the architectural, logical and circuit levels. Architectural level optimization refers to those techniques that are decided while coding the system, in other words they are the RTL level techniques. Logic level techniques are those which are applied at the logic level by optimization of the logic components either by improving their implementation logically of by removing redundant logic [3]. Circuit level techniques are those applied at the circuit and transistor levels.

There are various types of power dissipation in VLSI systems which can be broadly classified into four categories – dynamic power dissipation, static power dissipation, Short circuit power dissipation and leakage power dissipation. Dynamic power dissipation is taken up as the topic of interest of this paper as this constitutes more than 50 pc of overall power dissipated in most of the designs.

2. DYNAMIC POWER DISSIPATION
Dynamic power of a digital circuit is given by the equation

$$\boldsymbol{P_{dyn} = \alpha * C * (V^2) * f}$$

Where P_{dyn} is the dynamic power of the circuit, α is the activity factor which varies from 0 to 1, 0 being no activity on the net and 1 being full activity on the net wrt clock, C is the total capacitance, V is the supply voltage and f is the frequency of operation. The activity of any net is never the

same throughout unless it is an ungated clock signal as data toggles can happen at any time in a system. Hence in such cases, activity factor or AF is defined as the average switching activity of a system.

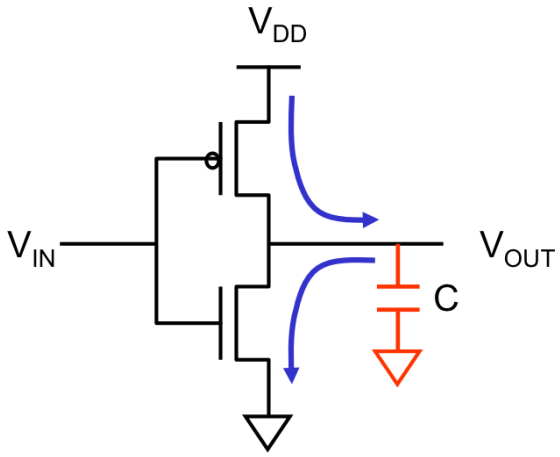

Fig 1: Switching of a CMOS inverter

When node capacitance C is discharged and charged by the power supply when input is low, as shown in Fig 1, the frequency of which being f and supply voltage V, then the charge change per cycle is given by C*V and the charge changed or moved per unit time ie per sec is given by C*V*F. As the charge is supplied by the supply at a voltage V, energy dissipation per cycle, or the dynamic power, is C*V*V*F. The dynamic power in case of a synchronous flip-flop, that can change its state once in a cycle, the dynamic power is given by

$$P_{dyn} = \frac{\alpha * C * (V^2) * f}{2}$$

Here the equation is multiplied by activity factor that was earlier explained. In case there is a gating, its value will be even lower as the activity will be smaller.

As we go to the deep submicron region, the leakage power dissipation due to subthreshold currents start to become more prominent but as we are restricting ourselves to the major power contributors, this subject will not be covered here

3. CLOCK POWER OPTIMIZATION AT CIRCUIT DESIGN LEVEL

As it is clear from the equation of dynamic power dissipation, dynamic power is a linear function of clock frequency. Although dynamic power is a quadratic function of the supply voltage, trying to reduce the supply would have an impact on timing as the device slews would degrade and trying to improve them by upsizing the devices would in turn lead to an increase in power. So we focus on reduction of the clock power here without changing the clock frequency in any way. In general there are a wide no of clock buffers that are placed in a design to distribute the clock as well as provide a degree of clock tuning (in this context, clock tuning refers to either delaying the clock or making it arrive at the sequential faster) for fixing the timing issues. By just manipulating these buffers, routing, their sizing, placement etc, a considerable amount of power can be saved.

The various circuit design level optimization techniques are as follows.

3.1 Dual and quad latch/flop conversion

There may be cases where in two or more flops or latches are driven by the same clock signal (or similar clock signal such as two different clocks with the same master clock and gating). They can be merged to form dual or quad latches to save power as the number of clock buffers, clock nets and clock routing will be reduced. These sequentials may be either in the same hierarchy or in different hierarchies, where in, in the latter case, the sequentials need to be pulled out of their hierarchies in order to merge them. There are also a few challenges of this technique, one of which is that the functional equivalence may begin to fail after the conversion. This is because when functionality of design is verified with that of the RTL, the design and the RTL are segmented into pieces called 'cones' which must have a one to one mapping for functional equivalence. But as these conversions are implemented, the verification nodes of a cone may change in name due to which the one to one mapping will happen incorrectly and the functional verification fails. The other and perhaps most critical issue with this technique is that if the sequentials exist in a highly congested and localized area, it may be very difficult to accommodate the new sequential in the given area as it will approximately be twice as large as the original sequential which would surely require the movement of other cells to accommodate it and alter the timing of all the signals through the new sequential. If it is approximated that this type of conversion would only provide a small power gain, it is not preferred due to its negative impacts on timing and area.

Fig 2 shows conversion of two flops into a dual flop

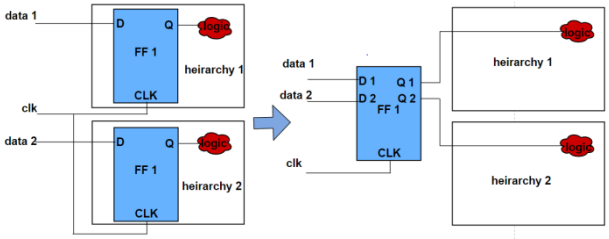

Fig 2: Dual flop conversion

3.2 Oversized clock drivers

Every device can be characterized by its device width (denoted by Z) and the output capacitance seen by the device (denoted by C). If the C/Z ratios of devices in the clock paths are too small, ie the clock drivers are larger than what is required to drive their loads, a significant amount of power is consumed as well as they are bad from the timing perspective as the input devices of these buffers would not be able to drive these oversized buffers and also would have to be sufficiently large to drive the large buffers which degrades power further [4]. Hence these buffers with large C/Z are reduced in size. Care must be taken though as this changes the arrival times of clock signals at the sequentials. So the extent of timing change must be kept in mind while resizing these buffers.

3.3 Inefficient clock multiplication and stacked buffer

This technique deals with optimizing the clock tree (here clock tree refers to the clock network in a design which is similar to the branching seen in a tree). Fig3 shows two buffers that are placed immediately after a branching node. If these two buffers driving a similar load and have similar driving strengths, they can be combined to form a larger

buffer placed before the branch node or completely eliminate a buffer stage.

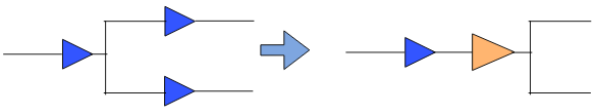

Fig 3: Clock multiplication optimization

If a net has a number of stacked buffers as in Fig 4 and provided that they do not cause any timing issue, they can be combined to form a single larger buffer. This also greatly reduces the clock delay on that net which may impact setup time as the data would have to arrive that much earlier than before. This not only reduces the dynamic power but reduces the static power dissipation as well as the effective length of the device channel is reduced.

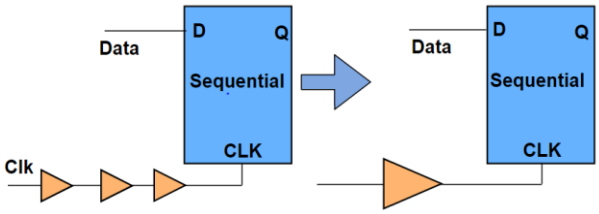

Fig 4: Stacked buffer optimization

3.4 Sequential movement

This deals with the movement of sequentials either before its preceding logic or moving it after its succeeding logic. This is best explained by considering the case of a multi-input, multi-output logic scenario. For example consider a 16:4 encoder. If sequentials are implemented at all the input nets, they would account to a total of 16 sequentials but in case they are moved after the encoder, only 4 sequentials would be required which reduces the no of required sequentials by four times, also saving a significant amount of power. This can have a timing impact because depending upon whether it is a decoder or an encoder, an extra decoder/encoder delay now shows up after/before the sequential respectively. Similar technique can be applied to muxes as well. Fig 5 shows latch movements in decoder and encoder circuits.

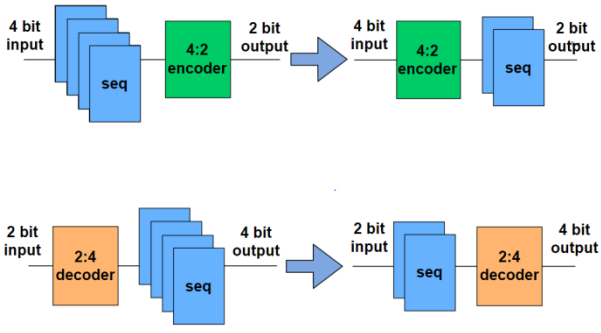

Fig 5: Sequential movement in encoders and decoders

3.5 Clock logic optimization

As it is already firmly stated, clock power is the the largest contributor to the overall power of any design. So the primary goal of power optimization must be to reduce the overall clock power. Fig 6 shows an example of how simple logic optimizations can lead to a good amount of power saving by reducing clock nets as well as clock devices.

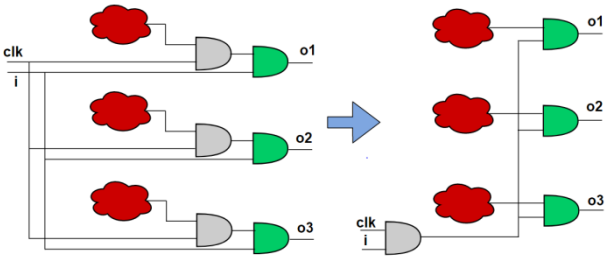

Fig 6: Clock logic optimization

3.6 High AF nets

The primary and most primitive method of fixing the setup violations is to reduce the delay in the path by upsizing the device. But this can have serious implications on power if the device is present on a path with a high AF (activity factor) as upsize of any device on high AF nets would lead to more power dissipation cycles of a larger device and as it has been upsized, the preceding device may have to drive and switch a larger capacitance which may require its proper sizing as well and would lead to a power disaster. Hence before upsizing any device, it is extremely important to look at the AF of that net especially if it's in a clock path.

4. CLOCK POWER OPTIMIZATION AT THE RTL LEVEL

The techniques discussed so far are implemented during the circuit design stage whereas the optimization techniques that will be discussed, require a modification in the RTL for its implementation [5]. These are mainly based on the various clock gating techniques that can be effectively implemented.

These are very reliable and efficient techniques if implemented correctly, hence reducing power in major power hog areas of the design. These techniques primarily employ or change the controls of groups of sequentials with a common enable.

VLSI designs basically consist of many stages of clock buffers, the higher stage buffers ones called the RCBs (Regional clock buffers) and lower stage buffers called the LCBs (local clock buffers). The RCBs are generally connected to one or more LCBs depending upon their AF.

A few terms can be defined that can be used to quantify the gating effectiveness are as follows.

1. Clock gating percentage – It is the percentage of registers/sequentials which are clock gated.
2. Clock gating efficiency (CGE) – It is the percentage of time for which the master clock is gated. This gives an accurate estimation for goodness of gating [6].
3. Data Aware Clock Gating Efficiency (DACGE) – This is similar to CGE but considers clock toggles with respect to data toggles or in other words, it gives the percentage of time in which data also toggles when the clock toggles [7].

A register using a typical latch based clock gating is shown in Fig 7.

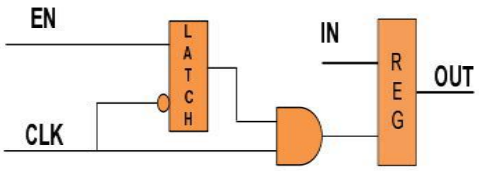

Fig: 7: Latch based clock gating for register

The clock input of the register is gated by the enable signal to ensure that the clock toggles only when the enable signal is high. The enable latch is used to maintain a constant voltage level for the entire half cycle, absence of which would create irregular clock waveforms with at the register input. It must be noted that the same enable can be used by other latches as well having similar activity.

The various types of gating techniques to improve DACGE, CGE and to efficiently combine or split the RCBs and LCBs are as follows.

4.1 Low activity non enabled register

In this type of register, shown in Fig 8, clock toggles only when there is a change in data, hence called data aware clock gating. The XOR gate output would change only when there is a change in the input data hence the data controls when the clock is applied to the sequential. This is used to improve the DACGE.

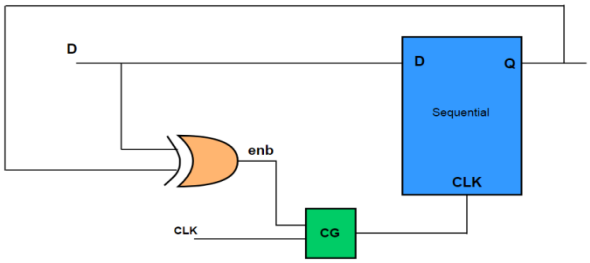

Fig 8: Low activity non enabled register

4.2 Backward assertion of enables

Consider a typical mux based enable register as shown in Fig 9. If 'EN' is true, only then will the register change to the new value, else it retains its old value.

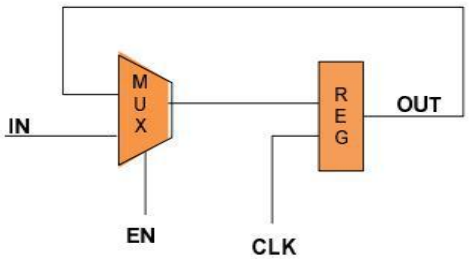

Fig 9 : Mux based data gating

Fig 10 depicts a chain of register, each of them having single fanouts. If 'en' in the last stage is zero, irrespective of previous Register's outputs. d_out remains unchanged [8]. This clock gating technique can be used especially in situations where in the clock AF of the last register in a chain of registers is very low in which case, even though there are significant number of toggles in previous registers (R1 and R2), the expected output of final register would have low AF, hence it would be impractical to allow the inputs of R1 and R2 to always toggle. So this enable of the final register can be pulled forward to ensure R1 and R2 toggle only when R3 toggles

In other words, the extra toggles at outputs of R1 and R2 are redundant and can be removed.

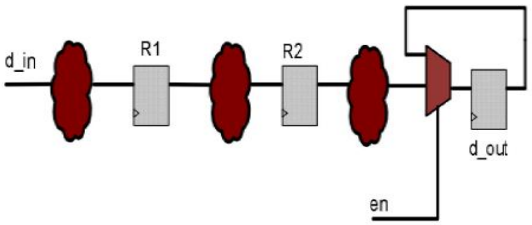

Fig 10: Series of registers with the possibility of backward assertion of enables

Fig 11 shows how the enable can be backwards asserted. The 'en' signal of R3 is given to R1 and R2 as well.

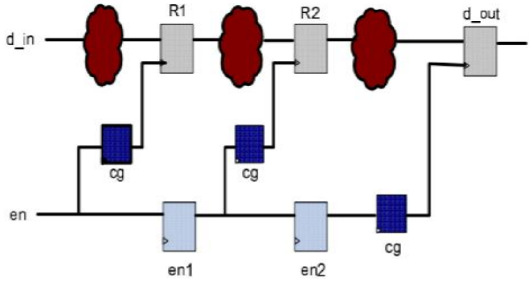

Fig 11: Series of registers with backward asserted enable

4.3 Forward assertion of enables

Consider in Fig 12 that the input to the final latch is a function of the outputs of R2 and R4

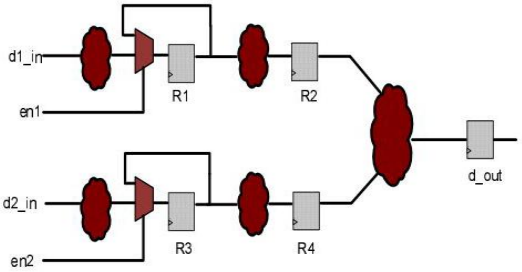

Fig 12: Series of registers with possibility of forward assertion of enables

The enables of R1 and R3 can be asserted forward in case the AF of R1 and R3 are considerably smaller than those of the succeeding registers, ie R2 and R4 as shown in Fig 13

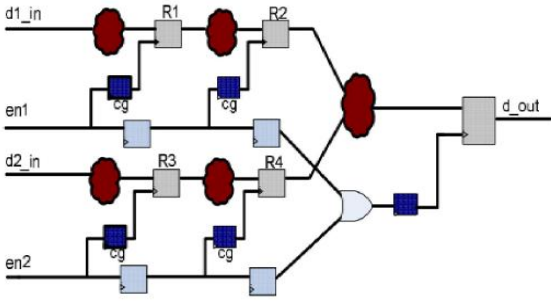

Fig 13: Series of registers with forward asserted enable

In Fig 13, the clock input of the final register is a function of the enables of R1 and R3, an Or function basically as the data should be latched on to the register if there is a change in either en1 or en2.

4.4 LCB merge and LCB enable pull in

For RCBs with single fanouts, the opportunities to pull the second enable (en of LCB) should be evaluated so that it can be combined with the enable of the RCB to completely eliminate the LCB in order to remove the long routing of high AF net and replacing it by a low AF enable net. Fig 14 shows before and after LCB pull in.

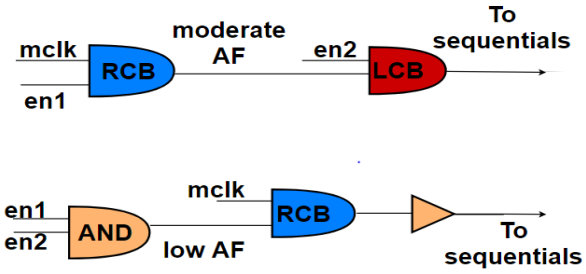

Fig 14 : Before and after LCB pullin

In case multiple LCBs have similar activity, they can be effectively replaced by a single LCB. This is called an LCB merge.

4.5 RCB split

In case an RCB in particular design is driving a large number of LCBs such that a bunch of them have high activity and the remainder have low activity, it is meaningless to drive all the LCBs with the same RCB. Hence the RCB should be split such that one RCB is connected to the high AF LCBs and the other RCB is either gated by a different enable to match with the activity of its LCBs or and LCB pull in is performed if the LCBs are observed to be having similar activity.

5. RESULTS

The power clock optimization techniques were implemented in a functional unit using a process technology lesser than 14 nm, which is clearly in the deep sub-micron region. These optimization techniques were performed keeping other constraints in mind so as to not have a significant impact on other constraints.

Fig 15 and Fig 16 show the power distribution before and after optimization at the circuit design level.

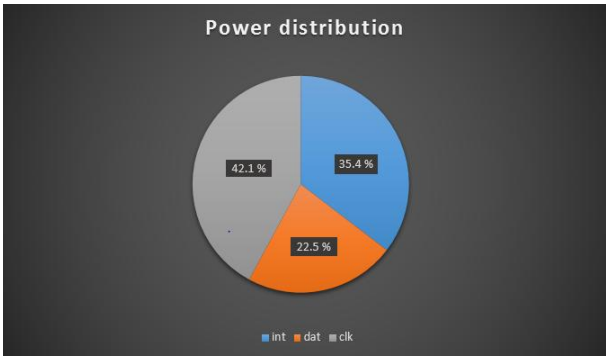

Fig 15: Power distribution before optimization

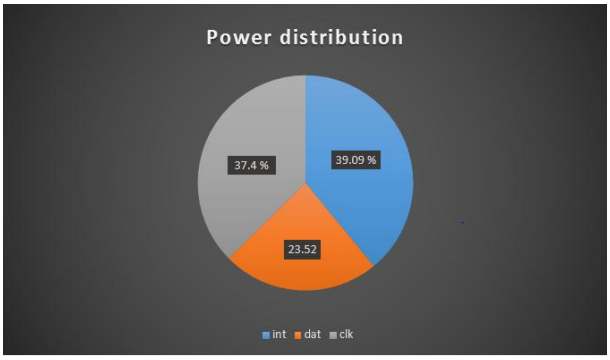

Fig 16: Power distribution after optimization

Comparing the two pie charts, it is evident that there is a clock power reduction. The exact power values of data, clock and interface power are tabulated in Table 1.

Table 1. Power reduction comparison

	Clock power (mW)	Data power (mW)	Interface power (mW)	Total power (mW)
Power before optimization	**0.110**	**0.059**	**0.092**	**0.261**
Power after optimization	**0.082**	**0.051**	**0.086**	**0.220**
Power saved diff	**0.028**	**0.008**	**0.006**	**0.041**

From Table 1, it is evident that there is a reduction of clock power by 25% and a net power reduction of 15.7%.

The RTL power optimization techniques were also applied and the results are tabulated in Table 2.

Table 2. Power reduction comparison at RTL level

Power experiments	Power gains (avg gain)
RCB merge	**4.5%**
Multibit sequential conversion	**5.2%**
LCB enable pull-in, clock path optimization and redundant buffer removal	**2-3%**
RTL clock gating improvement analysis	**Successful**
Dynamic RF to Static RF conversion	**WIP**

Power gains of 2-5% are observed for each of the RTL power optimization techniques as well. Conversion of dynamic to static ram to save more power at the cost of area is still a work in progress (WIP).

6. CONCLUSION

Low power devices and circuits are a necessity in today's scenario because as the form factor of devices reduces, there will be a proportional increase in the heat generated as more devices are accommodated in the available area. As the congestion increases, so do the problems of removing the extra heat as it becomes increasingly difficult to employ good heat sinks for all the devices given the close proximity of devices to one another.

As a solution to problems of high power consumption, various low power techniques have been successfully applied in a deep sub-micron environment with the attainment of positive results for both the categories of techniques, namely the circuit level and RTL level techniques. The primary goal was to reduce the power dissipated by the clock network and a clock power reduction of 25%, along with an overall power reduction of 15.7% was achieved. As far as the RTL level techniques were concerned, a reduction of 2-5% were observed for each of them.

In future, the work can be extended to further reduce the power of the design by converting Dynamic register files to static register files at the cost of an area penalty which would require a careful estimation and analysis of how this sort of memory array conversion would have an impact on other constraints in the design

7. REFERENCES

[1] P. Corsonello, S. Perri and G. Cororullo, "Area-time-power tradeoff in cellular arrays VLSI implementations", in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 8, no. 5, pp. 614-624, Oct. 2000. doi: 10.1109/92.894167

[2] A. Nayak, M. Haldar, P. Banerjee, Chunhong Chen and M. Sarrafzadeh, "Power optimization of delay constrained circuits", Proceedings of 13th Annual IEEE International ASIC/SOC Conference (Cat. No.00TH8541), Arlington, VA, 2000, pp. 305-309. doi: 10.1109/ASIC.2000.880754

[3] Mayank Chakraverty, Harisankar PS and Vaibhav Ruparelia," Low Power Design Practices for Power Optimization at the Logic and Architecture Levels for VLSI System Design", IEEE conference publications,International conference on energy efficient technologies for sustainability,2016.

[4] T. Enomoto, S. Nagayama and N. Kobayashi, "Low-Power High-Speed 180-nm CMOS Clock Drivers", 2007 Asia and South Pacific Design Automation Conference, Yokohama, 2007, pp. 126-127. doi: 10.1109/ASPDAC.2007.357973

[5] Kai-Shuang Chang, Chia-Chien Weng and Shi-Yu Huang, "Accurate RTL power estimation for a security processor", Conference, Emerging Information Technology 2005, pp. 3 pp.-.doi: 10.1109/EITC.2005.1544353.

[6] J. Srinivas, M. Rao, S. Jairam, H. Udayakumar and J. Rao, "Clock gating effectiveness metrics: Applications to power optimization", 2009 10th International Symposium on Quality Electronic Design, San Jose, CA, 2009, pp. 482-487. doi: 10.1109/ISQED.2009.4810342

[7] T. Na, J. H. Ko and S. Mukhopadhyay, "Clock data compensation aware clock tree synthesis in digital circuits with adaptive clock generation", Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, Switzerland, 2017, pp. 1504-1509

[8] I. Han, J. Kim, J. Yi and Y. Shin, "Register grouping for synthesis of clock gating logic", 2016 International Conference on IC Design and Technology (ICICDT), Ho Chi Minh City, 2016, pp. 1-4. doi: 10.1109/ICICDT.2016.7542070