

A Review on Strategies and Methodologies of Dynamic Power Reduction on Low Power System Design

Kamal K Mehta

Institute of Technology, NIRMA University, Ahmedabad. INDIA
{kamal.mehta,rasendu.mishra}@nirmauni.ac.in

Abstract: As the scale of integration increases proportionally the power requirement of computer system also increases. Since more than a decade it has been observed that energy budget and its analysis is paying a vital role on systems architecture. Many schemes were proposed to reduce power requirement. For a thin film CMOS technology, it has been noticed that, dynamic power component is one of the major factor having of power consumption. Dynamic power report to give maximum contribution towards power budget. Power consumption due to communication on system-level buses contributes a lot of power. This paper aims to provide details literature review covering dynamic power dissipation aspect and related technical issue. Details elaboration of various factors contributing toward power budget is discussed. The survey report discuss around sixteen various techniques to reduce power dissipation. It has been observed that the researchers are working on operational as well as design parameters for dynamic power reductions. The paper discuss about residue effect on systems design after dynamic power problem is incorporated. The result of the study show that, the tradeoff between adding new CMOS nodes to handle power dissipation in turn results more power for operation. It is further observed that an optimum point need to set before the design of processor is complete.

Keywords: Low Power Design, Power Dissipation, Dynamic Power.

1: Introduction

The dynamic power consumption caused by the charging and discharging of the capacitive load on each gate's output [1]. It is proportional to the frequency of the system's operation, the activity of the gates in the system, the total capacitance seen by the gate's outputs and the square of the supply voltage [9,13]. As the size of processors reduces and the power density increases. The need to reduce power consumption has become important design parameter. Process can be applied to a cell phone and its derivatives, where low power consumption and lower size are the key. Systems designers have developed several techniques to reduce dynamic power consumption at the various level like, logic, architecture, and operating systems. This paper gives detail summary on various techniques applied for reducing the power consumption.

2: Components of Power Dissipation in CMOS circuits

Power dissipation in CMOS circuits comprises two major components, namely the transient component and the static component. Transient power dissipation arises when the transistors are performing switching actions [5]. On the contrary, static power dissipation occurs even when there is no switching activity as long as the transistors are powered [15]. The component of transient power dissipation, in turn comprised of the P(DYN) subcomponent, representing dynamic power dissipation. The major cause of dynamic power dissipation is due to the switching of transistors, and the PSC subcomponent, which is due to the short circuit or "crowbar" current that flows through the PMOS-NMOS stack during a transition [9]. Static power dissipation can also be analyzed into two subcomponents, namely P(LEAK), which is the power dissipated due to leakage currents, and P(STATIC) which is the power dissipated due to currents that continuously flow from the power supply (V_{dd}) to the ground because of weakly ON transistors. Hence, the overall average power dissipation in CMOS circuits (PAVG) can be analyzed into four components, and is described by the following expression

$$PAVG = PDYN + PSC + PLEAK + PSTATIC$$

Although the peak power consumption should always be considered for reliability and correct circuit operation purposes, average power is more crucial because it is correlated with the circuit's overall expected behavior. Moreover, minimizing P(AVG) relieves peak power as well and increases reliability.

2.1: Dynamic Power Dissipation:

Dynamic power dissipation results from the charging and discharging of capacitances in a circuit during logic transitions [1,7]. It is proportional to the frequency of the system's operation, the activity of the gates in the system, the total capacitance seen by the gate's outputs and the square of the supply voltage. Thus the formula for dynamic power dissipation is given as:

$$PDYN = \alpha * VDD^2 * F * CL$$

Dynamic power dissipation has been the dominant factor of power dissipation, compared to the other components, in digital CMOS circuits [20,21]. Design technologies down to 0.35 μ m, dynamic power accounts for about 80% of a circuit's total power dissipation. For DSM technologies dynamic power increases in absolute numbers because of increasing functionality, complexity and clock rates with this increase partially compensated by the significant lowering of supply voltages (Vdd). In comparison though to the other power dissipation components, dynamic power isn't considered to be dominant in post-60nm technologies with leakage power having a rapidly increasing trend and accounting for even more than 50% of the total power dissipation.

2.2: Principles for Power Reduction:

The idea behind coping with nowadays increased power consumption is to attack the underlying mechanisms of the dominant power components and subcomponents presented [19]. As denoted, these are P(DYN) and P(LEAK) with the latter being dominated by sub threshold leakage currents. Therefore, the vast majority of low power techniques are concerned about reducing these two components. This work as well, focuses on presenting such techniques [24].

P(DYN) is analogous to the switching activity, load capacitance CL, the square of VDD and the operating frequency. Apparently, power reduction can be achieved by reducing these factors, independently or in a combined fashion. The two most efficient and most followed ways for diminishing PDYN, concern the reduction of the supply voltage and the effective capacitance. Supply voltage VDD reduction is the most aggressive technique because of the quadratic relation to PDYN. All other things being constant, halving VDD reduces PDYN to a quarter of its initial value [22]. However, this increases propagation delays thus degrading the overall circuit's performance. More sophisticated techniques, based on the same concept, propose applying multiple simultaneous operating voltages instead of a single lowered one. The idea behind this practice is the identification of the time critical and non-critical logic paths in order to introduce cells with an improved power profile, e.g. operating on a lower VDD, thus sparing power consumption without affecting performance. In addition to these techniques, voltage can be varied in a temporal fashion as well. This is called dynamic voltage scaling, DVS, and is a workload based technique that requires software-hardware cooperation. If the operating frequency is varied as well then the technique is enhanced into dynamic voltage and frequency scaling, DVFS.

The next large group of techniques dedicated at attacking P(DYN), concerns techniques that aim at reducing the effective capacitance. This is accomplished through resizing of transistors, improving inter connects overhead and/or by trying to minimize the essential switching activity. The latter can be achieved by introducing novel techniques at the architectural level like, architectural transformations, use of low power operands, glitch suppression techniques, data and FSM state encoding, signal gating e.g. clock gating, arithmetic representations etc.

The combined effect of some of these techniques can furthermore cut down PDYN consumption, but not as aggressively as the aforementioned VDD reduction based techniques. Using this methodology though doesn't require any other new technology and allows the existing circuit to be redesigned for low power.

3: Techniques for Reduction of lowering logic transitions by selectively pre Dynamic Power Consumption

Techniques for reducing the transient component of power dissipation are briefly presented in this section.

3.1: Operator Selection for Low Power:

When it comes to select an arithmetic operator, certain aspects have to be considered. Profoundly, effective capacitance should be minimized which in turn is a function of the operator's architecture, area, number of logic

levels, fan out distribution and possible signal encoding. Glitches can also add up to an operator's power dissipation. Usually a high fan out along with a large number of logic levels implies higher glitch propagation.

A common set of adders is the one including the ripple carry (RPL), forward carry look ahead (CLF), carry look ahead (CLA) and the Brent-Kung (BK) adders. Ripple carry adders may be attractive for their small size but the carry signal propagates through all the stages thus consuming power and time. This occurs to a lesser extent in CLA and CLF adders, but at the cost of higher number of logic levels. The best option for area, speed and power tradeoff is the Brent-Kung (BK) adder. The BK adder is a parallel prefix adder that was originally proposed as a simple and area efficient adder. It offers a good architecture for minimizing the number of logic levels, wiring tracks, fan out, and gate count. As for multipliers, Wallace advantages over carry-save multiplier due to its uniform switching propagation, less logic levels and lower average fan out. More power savings can be accomplished by utilizing special kinds of signal encoding like the Hybrid Encoding, which is a compromise between the minimal input dependency offered by the binary encoding and the low switching characteristics of the Gray encoding.

Operators are usually inferred from HDL code and their actual implementation is selected by synthesis tools, according to which one best meets the design constraints. It is therefore important to know the properties of each operator's possible architectures, thus being able to instruct the tools which one to select out of the list when it comes to low power design.

3.2: Pre Computation: Pre computation is a logic optimization technique that aims at evaluating the output values of a logic function one clock cycle before they are required, and then using the pre evaluated values to reduce internal switching activity in the succeeding clock cycle[23]. The issue of not being able to pre compute any input combination, due to the constrained pre computation input subset of the previous architecture, is alleviated by the introduction of the complete input disabling pre computation architecture proposed. In that architecture pre computation logic can be a function of all input variables, allowing the pre computation of any input combination. Experimental results showed significant power reductions over both original circuits as well as those synthesized under the latter architecture [25]. A pre computation architecture for multi output circuits is also possible at the expense of being significantly more complex and with oversized predictor functions.

3.3: Guarded Evaluation: There are various levels for optimizations, guarded evaluation is a gate level power optimization technique that relies on blocking the inputs of complex data path circuits for transition reduction, if these inputs do not contribute to output generation for a given sensitization vector. In other words, if under certain conditions a fan out is not observed, that is, if it has observable don't care

(ODC) conditions, then transparent latches or floating gates can be inserted at the corresponding fan-in. Such parts of the data path are denoted as guarded. Usually multiplexers generate ODC conditions. Whenever an input vector belonging to the guarded logic ODC set appears, the guarding logic does not allow any signal transitions pass through the guarded logic.

3.4: Operand Isolation: In a data path intensive design, the complex combinational blocks may contribute to the majority of power consumption of the design. Operand isolation is based on the same principle as guarded evaluation but with this technique designs described at the Register Transfer Language (RTL) are processed. By taking advantage of ODC conditions an RTL exploration can be done for identifying circuit portions which perform redundant computations, thus suppressing switching activity.

3.5: Operator Reduction: Operator reduction is a technique based on transformations of operations into computationally equivalent implementations at the algorithmic level. Its objective is to optimize the number and type of computational modules, their interconnection and their sequencing of operation, while input/output behavior is preserved. This is an apparent approach for reducing the effective capacitance in a circuit. There are cases that a reduced number of operators can be achieved at the cost of a longer critical path. This implies a higher supply voltage if we want a realization that retains the initial throughput. Thus, this technique can have side effects as well making the associated power minimization task a tradeoff problem. Operator reduction is one of the techniques belonging to the High-Level Synthesis Transformations set, along with operation substitution, word length reduction, control step reduction etc.

3.6: Data Representation: The switching activity from a data path element is directly proportional to the number of

bits switched between successive data accesses [4,6]. Therefore, an optimized style for data representation could result in lower switching activity. The proposed method is based on the observation that the distribution of data value transitions is usually highly skewed, and exploits this observation in choosing a data representation, so consecutive data value pairs that appear frequently are encoded to have a smaller Hamming Distance. This implies that information regarding data value transitions for a target application has to be computed in advance. In order to maintain compatibility within the system between parts that use different data representations, converters are used in the data path to change the low power representation to normal and vice versa. Authors claim a 22% reduction in switching activity for MPEG-1 decoders [17].

Using Sign-Magnitude instead of two's complement data representation can also yield power savings under certain conditions. Although for most signal processing applications two's complement is chosen for performing arithmetic operations easy, that can have a negative impact on switching activity. Sign-Extension causes the MSB sign bits to switch when a signal changes values around zero. Thus, if the signals being processed frequently switch from positive to negative values without utilizing the whole bit-width, then switching activity can be significantly increased. In such cases sign-magnitude can be a good alternative for power minimization, since only one bit is allocated for the sign for this representation. In general, these techniques are quite application specific and have to be judiciously utilized. It has been experienced that, suitable tools for data representation is not available to experiment on simulated results. A customized software has been developed and used for experimentation.

3.7: Pipelining and Parallelism: There is a dependency between the operating frequency of a circuit and its supply voltage given by

$$f \sim (VDD - VTH)^2 / VDD \Rightarrow f \sim VDD$$

Assuming $VTH \ll VDD$.

This implies that we can use performance speed-up transformations, and tradeoff performance gains for power through voltage scaling. A common method for reducing a circuit's critical path is pipelining. Logically deep internal nets are typically more affected by primary input switching, therefore they are more susceptible to glitches. Pipelining can attain glitch reduction as well, since it shortens the depth of combinatorial logic by register insertion. On the other hand, it increases clock tree power but overall power consumption in a design is lowered. We can achieve significant power savings by using parallelism as well, which is a method based on the same concept as pipelining. The main idea is to maintain throughput at reduced supply voltages through hardware duplication. By using parallel, identical units, the speed requirements on each unit are reduced, allowing for a reduction in the circuit's voltage.

During the study it was felt that, increasing instances of hardware to implement parallelism and pipelining would increase overhead due to added components. It has been further observed that the publications does not include significant research work to calculate overhead findings. Thus a new branch is open for research to find overhead calculation and give optimum design in parallel architecture.

3.8: Register Retiming: Register retiming is a sequential optimization technique that moves registers through the combinational logic gates of a design to optimize timing and area. This proves particularly useful when some stages of a design exceed the timing goal while others fall short.

Retiming can be used for minimizing dynamic power in a twofold fashion. One way is to reduce switching activity of "busy" computational elements, especially of those who drive large capacitive loads, by moving registers to their outputs. This way outputs will change only once per clock cycle, thus masking power costly glitches. The heuristic algorithm for the selection of the candidate gates to be registered, is based on the amount of glitching that occurs at the output of each gate and the probability that this glitching can further propagate.

Study revealed second approach is similar to pipelining and parallelism, retiming provides an efficient and straightforward way to reduce critical paths while preserving throughput and the number of operations. Performance gains can then be exchanged for lower supply voltage. In a hybrid retiming and supply voltage scaling technique, we not only scale down the voltage of computational elements that are out of the critical paths but also, through retiming, try to move registers around in order to maximize the number of elements off the critical paths, thus incurring further power savings.

3.9: Clock Gating: Clock gating is the primary means of dynamic power management in synchronous circuits. It is a very efficient technique that provides a way to selectively stop the clock, whenever the computation that is to be carried out at the next clock cycle is redundant. In other words, the clock signal is disabled according to the idle conditions of the logic network, thus evading a considerable power consumption by combinatorial logic, flip-flops and by the clock buffer tree in the design. Clock gating actually works by identifying groups of flip-flops, which share a common enable term that determines when a new value is to be clocked into the flip-flops. The implementation of clock gating mechanism can be as simple as an AND or an OR gate, depending on the edge on which flip-flops are triggered. Due to the fact that such a simple implementation is susceptible to hazards/glitches of the enable term that could corrupt the clock signal to the registers, a transparent latch is usually inserted between the glitch enable term and the AND/OR gate. Some drawbacks of clock gating are that it may hinder timing closure and can make design for testing and verification more complex. Overall, the clock gating technique is very advantageous for circuits that are often idle for long periods and get activated by request.

3.10: Gated Clock FSM: As with arithmetic operators, finite-state machines (FSM) are also very common parts of digital systems used to generate signal sequences, to check a signal sequences or to control large data path parts. It is therefore imperative to try to minimize their power consumption. The basic idea of gated-clock FSM is to avoid any redundant switching activity in the next state logic block and in the state register [26], if the FSM present state is the same as the next one. Put differently, this technique discusses the application of clock gating in finite state machines.

3.11: Finite State Machine State Encoding:

Efficient encoding of the states of a FSM can help reducing the power consumption in the next state and output combinatorial blocks, by suppressing logic transitions. The main idea is to use an encoding scheme that minimizes switching from one state to another, if such a transition is very likely to happen. If we identify the most probable cycles in a FSM and encode the states on these cycles with minimum Hamming Distance Codes[27], then power dissipation can also be minimized. In general this is not an undemanding task since special tools that propagate transition probabilities on the FSM inputs and calculate the probability of each transition are required [28]. There are cases though where this technique could be more easily applied.

3.12: FSM partitioning: The basic idea of FSM partitioning is to decompose a large finite state machine into two or more simpler machines that jointly produce the equivalent input-output behavior, as the original machine, for low power purposes. The new sub-FSM will be composed of smaller state registers and combinatorial logic blocks.

The technique is meaningful if the original, bulky FSM is partitioned by searching for a small subset of states with high probability of transitions among them and a low probability of transitions to and from other states. The high transition activity subset of states will constitute a small sub-FSM that is consequently active most of the time. Therefore, such a partition enables the application of clock gating on the larger, and most of the time inactive, sub-FSM thus saving dynamic power.

3.13: Bus Encoding: Modern SoC designs are characterized by wide and long buses, which interconnect various internal blocks, or internal blocks with the external environment. These buses constitute a major source of dynamic power dissipation mainly due to their large capacitance and significant switching activity. Therefore, it is imperative to apply techniques that can reduce bus activity [10], thus yielding significant power savings overall (Especially when driving off-chip modules)[11]. The main idea is to reduce power consumption by properly coding the data and/or address bus values so as to minimize the number of transitions that occur on the bus [18].

One common bus encoding method is the Bus Invert Coding (BIC). In BIC, before sending data, the emitter compares its current value with the previous one and decides whether to send it or to send its inverted value along with a polarity signal.

This decision depends on the Hamming distance between the present bus value $b(t)$ and the previous one $b(t-1)$:

$$(B(t), \text{Inv}(t)) = (b(t), 0) \text{ if } H \leq N/2$$

Otherwise $(b'(t), 1)$,

Where

N=Number of bus lines and

H=Hamming distance of two consecutive words.

A bank of XOR gates at both ends implements inversions when needed. Thus, switching activity on highly capacitive buses can be reduced at the expense of additional switching in the decoder/encoder and the polarity line. BIC is effective when the data to be transmitted are randomly distributed in time, but it is not as efficient for data that exhibit sequentially and locality, e.g. sequential addresses.

3.14: Multi-Supply voltage Design (Vdd):

Supply voltage to a CMOS node is drain to drain voltage. Because dynamic power is proportional to VDD^2 , even a small reduction in supply voltage would reduce dynamic power exponentially. For example, the CPU and RAM blocks might need to be faster than a simple peripheral block. This implies a design can be partitioned into multiple voltage domains (also referred to as “Voltage Islands” in layout), based on timing criticality. For logic blocks that can operate at low clock speeds and are not timing critical, the supply voltage can be reduced to a level that just maintains reliable operation of the block. Of course level converters are needed in this technique as well, for signals travelling through different voltage domains. Generally, this is a non-trivial technique to apply on a design because of the difficulties in placing the level converters and reducing their overhead, acquiring cell libraries characterized for every operational voltage and efficiently partitioning the design for maximum power savings. A detailed knowledge of the functionality of the design is usually required. Complicated board-level design, high production costs and large energy costs outside the chip are also a big concern. It should also be noted that leakage power is also reduced, to a lesser degree, by voltage scaling due to its linear variation with VDD causes a quadratic decrease in power consumption. On the other hand, by decreasing the supply voltage the circuit’s delay is influenced negatively. Multi supply voltage design introduces the idea of preserving performance, while also reducing power consumption. This can be achieved by assigning the high VDD (VDDH) to the gates that belong to the critical path, while the lower set of supply voltages is assigned to off- critical path remaining gates according to their timing slack. If only two different supply voltages are provided, VDDH and VDDL, the technique is simplified into dual-VDD design. To avoid excessive static power dissipation, due to the inability of VDDL gates completely cutting off driven VDDH gates, the use of level converters placed between the VDDL and VDDH supplied gates is necessitated, which imposes area and power overheads. Layout is an important issue when dealing with multi VDD designs due to different n-well voltages of different supply voltage cells. In brief, two algorithms have been proposed for optimal assignment of cells to the layout, clustered voltage scaling (CVS) and extended CVS (ECVS). CVS allows only one voltage transition along a path and level conversions only at flip-flops. On the other hand, ECVS allows multiple voltage transitions along a path and placement of level converters even between logic cells. In general, power savings arising from the adoption of multiple supply voltages technique may be insignificant, due to the use of additional level converters, therefore dual- or triple-VDD techniques are mostly used.

3.15: Dynamic Voltage and Frequency Scaling: Dynamic Voltage and Frequency Scaling (DVFS), is an adaptive “version” of the previously discussed static Multi-VDD design technique. It allows devices to change their components voltage and corresponding frequency during runtime to adapt to the workload demand. DVFS can be applied to different levels of granularity, on large modules or even on individual logic blocks on the critical path at the expense of complexity overheads. This is an advanced technique that requires proper software and hardware cooperation and is usually correlated with processors, since these components are extensively used within embedded systems and are power hungry. Obviously, DVFS results in processors with variable performance. The technique is based on the key observation that a processor’s peak performance isn’t always required (large positive slacks), therefore the processor (or any other relevant hardware component) can be safely slowed down to save power, on the fly. Variable performance can be made unnoticeable to the end-user by accurately predicting the upcoming workload requirements and then adjusting the processor’s voltage and frequency accordingly. If we consider a task with a deadline of 25 ms running on a processor with a 50MHz clock speed and 5.0 V supply voltage then if the task requires 5×10^5 cycles for its execution, the processor executes the task in 10 ms and becomes idle for the remaining 15 ms. However, if the clock speed and the supply voltage are lowered to 20 MHz and 2.0 V, it finishes the task at its deadline, 25ms, resulting in ~84% energy reduction.

3.16: Power Optimization Results:

This section presents the estimated power results of the optimized Asterix, after applying the optimization flow. These results are the output of a pre-layout analysis flow. The impact of each technique on Asterix's power dissipation, active area and performance is extracted and evaluated independently. Moreover, we focus on the power-driven feature of clock gating, comparing the results with and without its usage.

4.1: Impact on Power Dissipation: A stacked column chart was utilized so as to provide information on how each power component is affected. As with previous analysis, each column is analyzed into cell leakage power, net switching power and cell internal power. The first column identifies the un optimized Asterix block. The second, third and fourth columns identify the optimized Asterix after gate-level power optimizations, clock gating and operand isolation respectively. Column five shows the design without ripple-carry operators while the final column identifies the combined impact of all the techniques on Asterix. The vertical axis corresponds to the estimated absolute power values in mW. Moreover, the value of each column's power component is designated to allow for direct and exact comparisons. The use of non-ripple-carry operators marginally improves dynamic power, however it is good to verify that such an option is on the positive side. Unfortunately, the operand isolation technique could not be evaluated, because the tool couldn't find an opportunity to apply the technique on Asterix. Finally, it can be seen that by combining all of the optimization techniques, a superposition of the impacts is achieved. Additionally, the negative side effects of leakage optimization seem to be partially offset. Quantitatively, in terms of percentages, our optimization flow manages to improve Asterix's overall power consumption by 30%, while for leakage power alone the improvement is much larger reaching a 52.2%.

4.2: Impact on Active Area: This section explains how the design's active area is affected by the applied optimization techniques. Figure 6-5 illustrates the impact of each of the applied power optimization techniques on Asterix's area. Each column identifies an optimization technique and the vertical axis corresponds to the absolute values of active area, in μ -meter. Because of the relatively small differences in area from column to column, a chart with broken Y-axis has been utilized so as to show up these differences. As discussed in the previous section, apart from the reduction in switching activity clock gating also reduces area.

Above concept is an alternative implementation to load enable registers, unnecessary activity is saved by replacing the multiplexers and feedback loops with clock gating logic. Apparently the latter requires less area, thus meriting the technique. The rest of the techniques do not affect Asterix's area. It is also seen that by combining all of the optimization techniques, a superposition of the impacts is achieved in terms of area. Summing up all together, there is an impact of power optimization on area, which is not significant but notable, around 4%, owing to the aforementioned "Positive Side Effects" of clock gating.

4.3: Impact on Performance (Overhead): This section examines the impact of power optimization on the design's performance. Performance is evaluated in terms of maximum attainable frequency. The chart's vertical axis corresponds to the absolute values of clock frequency, in MHz. For this work Asterix was synthesized with a maximum frequency constraint of 250 MHz, though the design is able to operate at much higher frequencies. This translates into a plethora of non-critical paths. It is also seen that performance is only affected by the gate-level power optimization techniques, in particular by leakage optimization. Thus, the second column, which identifies this technique, illustrates in the best way the tradeoff between positive slacks and leakage power reduction. Apparently, over constraining a design in terms of performance reduces the number of non-critical paths, thus shrinking the available optimization margins.

Conclusion

The chief motive behind this paper has been the addressing of the emanating problem of high power dissipation in CMOS circuits, which mainly stems from the downscaling of process technologies and low power market trends, in both terms of a theoretical and methodological approach. This translated into a set of goals such as investigating the mechanisms behind power dissipation, exploring the state of the art power saving techniques and implementation/recommending efficient workflows and tool usage for power analysis and optimization. This section summarizes the main results of this work, as listed below:

Extraction from literature and efficient organization of information on CMOS power dissipation components, their underlying mechanisms and their correlation to supply voltage, temperature and feature size. Extraction from literature, organization, efficient classification and presentation of Fifteen power reduction techniques that cover both dynamic and leakage power. Generic principles for low power design flows and power regression test methodology are laid down. The importance of coping with power early in these flows is also justified and highlighted. Formulation of tables that serve as knowledge tools for estimating the implementation cost and the expected power impact of various techniques, thus facilitating proper decision making about which power saving techniques to apply on a certain design. More than a ten of cited literature references to power saving techniques, design methodologies, tool usage etc., thus providing a compressive paper for researchers to refer.

References:

- 1: Srivastava N, Tripathi G. S. "Modelling of Parasitic Capacitances for Single Gate, Double Gate and Independent Double Gate MOSFET", International Journal of Computer Applications (0975 8887),35(9),December 2011.
- 2: Ahmed Abdelmotalib, Zhibo Wu "Power Consumption in Smartphones (Hardware Behaviorisms)", School of Computer Science and Technology ,Harbin Institute of Technology ,Harbin - China IJCSI ,International Journal of Computer Science 3(9),No 3, May 2012 ISSN ,(Online): 1694-0814.
- 3: Benini L, Bogliolo, A and De Michli "A Survey of Design Techniques for Systems Level Dynamic Power Management" IEEE Transaction on Very Large Scale Integration. 8(3), pp 299-366 June 2000.
- 4: A. R. Brahmabhatt, J Zhang Qinru Qiu and Q Wu , "Adaptive Low Power Bus Encoding Based on Weighted Code Mapping" Proc. Of IEEE Int. Symposium on Circuits and Systems, May 2006.
- 5: Werner W Bachmann and Sorin A Huss, "Efficient Algorithm for Multilevel Power Estimation of VLSI Circuits." IEEE Transaction on VLSI,13(2), feb 2005.
- 6: M. Olivieri, F Pappalardo, and G. Visalli. "Bus Switch Coding for Reducing Power Dissipation in off Chip Busses". IEEE Transaction on Very Large Scale Integration Systems. 12:1374-1377, Dec 2004.
- 7: M.Madhu, V. Srinivasa Murty and V. Kamakoti "Dynamic Coding Technique for Low Power Data Bus", IEEE Computer Society annual symposium on VLSI 2003.
- 8: Dinesh C Suresh, Jun Yang, Chuanjun Zhang, Banit Agrawal, Walid Najjar "FV-MSB: A Scheme for Reducing Transition Activity on Data Busses", Int. Conf on high performance computing Dec. 2003 Hyderabad India.
- 9: Sotiriadis, P.P. "Interconnect Modeling and Optimization in Deep Sub-Micron Technology", Thesis (Massachusetts Institute of Technology), May 2002.
- 10: J. Yang and R. Gupta, "FV Encoding for Low Power Data I/O", ACM/IEEE International Symposium on Low power Electronic Design, pages 84-87 2001.
- 11: Mehta K. K, Gajbhiye Samta, Sharma H. R. "Uni-Distance Encoding Scheme for Reduction of Bus Transition Activity" Journal on Software Engineering, 1(4), 70-74, 2007 ISSN 0973-5151.
- 12: Y. Zhang, J. Yang and R. Gupta, "Frequent Value Locality and Value Centric Data Cache Design ", The ninth International Conference on Architectural Support for Programming Languages and Operating systems (ASPLOS IX) , Pages 150-159, 2000.
- 13: Mehta K. K, Choubey A. S, Kowar M, Sharma H.R. "Delay Calculation of Gray Bus Encoder as Per Micro Electronics Standard" Journal on Electrical Engineering 2009, 3(2), ISSN 2230-7576.
- 14: Sotiriadis, P.P. Chandrakasan, A. "Low Power Bus Coding Techniques Considering Inter-Wire Capacitances" IEEE Conference Proceedings on Custom Integrated Circuits CICC. 2000 page(s): 507-510.
- 15: M.K. Gowan, L.L. Biro, D.B. Jackson, "Power Considerations in the Design of the Alpha 21264 microprocessor," Proc. of Design Automation Conference, 1999.
- 16: Mehta K. K, Patel R. N, Kowar M. K, Sharma H, R. "Reduction of Power Dissipation using Gray Bus CODEC as per Micro Electronics Standard" Journal on Electronics Engineering 2010 1(1) 31-37 ISSN 2249-0760
- 17: Benini, L. De Micheli, G. Macii, E. Poncino, M. Quer, S. "Power Optimization of Core-Based Systems by Address Bus Encoding" IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Dec 1998 6(4), pp554-562.
- 18: M.R. Stan and W.P. Burleson, "Bus Invert Coding for Low Power I/O" IEEE Transaction on very Large Scale Integration (VLSI) systems, pages 49-58, Vol.3 1995.
- 19: Mehta K. K, Sharma H. R, "Software Model to Create Data Profile for Analysis of Gray Bus CODEC" Journal on Software Engineering 2010,5(2) 58-62 ISSN 0973-5151.

- 20:A.P. Chandrakasan and R.W. Broderson, "Minimizing Power Consumption in Digital CMOS Circuits," Proc. of the IEEE, vol. 83, pp. 498-523, April 1995.
- 21:D. Liu and C. Svensson, "Power Consumption Estimation in CMOS VLSI Chips," IEEE Journal of Solid-State Circuits, vol. 29, pp. 663-670, June 1994.
- 22:Najm F.N. "A Survey of Power Estimation Techniques in VLSI Circuits", IEEE Trans. 1994,VLSI-2(4), pp 446-455.
- 23:A. P. Chandrakashan, "Low Power CMOS Digital Design," IEEE J. Solid State Circuits vol. 27 no 4 pp 473-483, 1992.
- 24:Mehta K. K, Sharma H. R. "Evaluation of Uni-Distance Codec for 4-to-32 bits Information for Power Reduction Initiatives" Journal on Software Engineering, 2008, 2(3), pp 57-60, ISSN 0973-5151
- 25:Mehta K.K, Mishra P. "Modelling of Low Power Communication System with Working Capacitance & Transition" ECE 2006, RESPOGRAPH, pp 18-20, Arora Academy Hyderabad.

- 26:Latika Borker, Manisha Sharma, K K Mehta "A Report on Digital Differential Analysis for Bus CODEC" International Journal of Engineering & Advanced Technology, 1(2) pp 34-36, 2011.
- 27:Mehta K. K. "Expectation and Variance Based Analysis of Information Processed in Frequent Value Encoding Scheme" IEEE Advanced Computing Conference IACC 2009.
- 28:Mehta K K "Probabilistic Modeling of Frequent Value Bus Encoding Scheme for Low Power Computation" 2009 , Kowar M. K, Sharma H. R, Mehta K. K.