

POWER AND DENSITY AWARE PARTIAL-BUS INVERT ADAPTIVE CODING USING GDI

¹Prasanna Angel Bokinala, ²Dr. B. Nancharaiah, ³K. Babu Rao

¹M.Tech, Dept. of E.C.E, Usha Rama College of Engineering and Technology, AP, INDIA

² Professor & H.O.D, Dept. of E.C.E, Usha Rama College of Engineering and Technology,
AP, INDIA

³Assistant Professor, Dept. of E.C.E, Usha Rama College of Engineering and Technology,
AP, INDIA

ABSTRACT:

In CMOS circuits more power is dissipated during charging and discharging the load capacitance. This can be lowered by minimizing the number of transitions inside CMOS circuit. This paper includes efficient encoding technique to reduce the transition activity. An Adaptive Encoding technique is one among all available methods. To reduce the power of interchip interconnects, an adaptive encoding scheme called adaptive word reordering (AWR) is proposed, which effectively decreases the number of signal transitions, leading to a significant power reduction. A novel circuit is implemented, which exploits the time domain to represent complex bit transition computations as delays and, thus, limits the power overhead due to encoding. The effectiveness of AWR is validated in terms of decrease in both bit transitions and power consumption. As an extension of this concept, Gate Diffusion Input based architectures are introduced to design proposed encoding process in order to reduce power constraints.

Keywords: adaptive word reordering, Gate diffusion Input, signal transitions, Latency, Interconnect.

1. INTRODUCTION:

Interconnect wires account for a significant fraction (up to 50% [1]) of the energy consumed in an integrated circuit, and this fraction is only expected to grow in future. In fact, it is projected that, as technology scales to the nanometer regime, the delay and energy consumption of global interconnect structures will prove to be a major bottleneck for SoC design [2, 3]. With designers striving to improve the lifetime of battery operated, personal computing devices [4], minimizing the energy consumed in on-chip interconnects becomes crucial. However, doing so requires the use of a structured, interconnect oriented, design methodology at all layers of the hierarchy from the system-level down to physical design. Interconnect oriented design poses a whole new set of challenges for designers. The advent of deep sub-micron technology also brings several signal integrity issues to the forefront [5, 6]. Supply voltage scaling, used extensively to reduce energy consumption, has led to decreased noise margins, making interconnects less immune to the vagaries of power supply noise, inter-wire crosstalk, radiation induced soft errors, electromagnetic interference, etc. With next generation SoCs expected to have multi-billion transistors, ensuring efficient and reliable transport for data signals becomes a daunting task indeed. Existing low power system design methodologies are ill-equipped to tackle this problem because they follow an error avoidance paradigm, where the aim is to eliminate all noise related errors through fine-tuned circuit design. Clearly, this will become

very costly, if not impossible, given the complexity of future SoCs. Therefore, the paradigm has to change to one of error tolerance in which, after a moderate amount of design time optimization, the focus is on error resilience techniques to combat run time errors and ensure correct system functionality. Power optimization is important to achieve high reliability. The important requirement is to know for particular application how much power the circuit will dissipate. With the increase in speed and complexity, there is an increase in power consumption. Power consumption is proportional to switching activity. By reducing the bus switching, bus power can be reduced. The chips between different functional blocks are called on-chips and the chips between different IC's or PCB are called off chips. In micro processor related systems load capacitance of off-chip buses is orders of magnitude greater than that of internal nodes. Power dissipation on these buses mainly occurs during signal transitions. Reducing these signal transitions will reduce the power dissipation. Power dissipation is classified as static power dissipation and dynamic power dissipation. Static power dissipation is due to leakage currents and dynamic power dissipation is due to switching activity. Switching activity is the probability of transitions ref [4]. Dynamic power dissipated by CMOS circuit is

given by, $P_{dyn} = \sum_{i=1}^N C_{Li} \times V_{dd}^2 \times f_{CLK} \times \alpha_i$

where V_{dd} is the supply voltage, C_{Li} is load capacitance, f_{CLK} is chip clock frequency, α_i is activity factor. To reduce the power dissipation one of the above factors must be minimized. V_{dd} cannot be lowered because of required threshold voltages of pMOS and nMOS transistors. Clock is turned off for the portions of chip which are not used. Load capacitance is the gate capacitance of transistor it is limited during fabrication according to the technology. Thus, reducing activity factor is the best way to reduce overall power. Lowering activity factor is reducing number of transitions (at behavioral domain) or reducing glitches (unwanted transitions) at logic level.

2. LITERATURE SURVEY

Parallel buses multiplexed into a serial link enables an improvement in terms of reducing interconnect area, coupling capacitance, and crosstalk, but it increases the overall switching activity factor (AF) and energy dissipation. Therefore, an efficient coding method needed to reduce the switching AF is an important issue in serial interconnect design. Many studies attempt to reduce the AF of parallel buses. Stan and Burleson introduced a bus-invert method that transmits the original or inverted pattern to minimize the switching activity. Researchers have proposed many techniques to improve the bus-invert coding method, such as the partial bus-invert coding and weight-based bus invert coding methods. The schemes mentioned above use an extra channel to send the inversion indication signal. Kuo et al. proposed the serial coding technique to solve the extra channel problem. They append extra information bits to the back of the original data word. Although this approach resolves the area overhead problem, it increases data latency. Three level differential encoding is proposed for parallel bus to enable multiple drivers at the transmitter and to recycle the same current and reduce power consumption. Joint crosstalk avoidance code and error correction code are proposed to reduce the power in parallel bus. Huang et al. further proposed combining serializing bus with the joint crosstalk avoidance code and error correction code to reduce the power.

Serialized low-energy transmission (SILENT) is a coding method used in reducing the switching activity for serial links. This approach encodes every single bit in the parallel bus using the XOR gate, and multiplexes the encoded parallel buses into a serial link. The XOR operation sets an adjacent bit with the same value to zero. The greater the correlation is, the more zeros the encoder produces this method is designed for data with strong correlation. Bharghava et al. proposed the transition inversion coding (TIC) technique to reduce switching activity for random data and to detect errors. Their technique counts the transitions in the data word, and inverts the transition states if the number of transitions in a data word is more than half of the word length. The scheme sets the current bit in the serial stream to be the same as the previous encoded bit when there is a transition. Otherwise, it is set to the inversion of the previous encoded bit. A transition indication bit is added in every data word. This extra bit not only increases the number of transmitted bits, but also increases the transitions and latency. Serial links as communication channels on on-chip network architecture for SoC. The serial links reduce the area of communication channels by 57% compared with a non serialized approach. This approach also reduces the switch activity because the coupling capacitance of the interconnect wires. The embedded transition inversion (ETI) coding scheme is proposed to solve the extra bit indication. This scheme eliminates the need of sending an extra bit by embedding the inversion information in the phase difference between the clock and the encoded data. When there is an inversion in the data word, a phase difference is generated between the clock and data. Otherwise, the data word remains unchanged and there is no phase difference between the clock and the data. The improvement of transition reduction is 19% compared with that of the TIC. The receiver side adopts a phase detector (PD) to detect whether the received data word has been encoded or not. A Viterbi decoder for the specification of (3, 1, 3) is designed and implemented in HDL [7] with no asynchronous techniques. However, a robust Add Compare Select (ACS) Unit of the Viterbi decoder [8] is designed using asynchronous architecture based on QDI PCFB template to design the internal blocks. Along with the QDI template, Martin synthesis method of directly converting the Communicating Sequential Process (CSP) to transistor level instead of gate level is also used for reducing power dissipation of ACS unit of the decoder. New, low power memory efficient trace back scheme for high constraint length [9] Viterbi decoder is developed. The buffer based memory bank architecture, due to which the area of the overall proposed trace back is increased, is explored for path merging of SMU. Analog design of Viterbi decoding [10] for Forward Error Correction (FEC) is used in channel coding for digital communications. Analog is used to reduce the size and power consumption of channel decoders like Viterbi decoders. A differential analog Viterbi decoder architecture is implemented using 32 nm Carbon Nanotube FET (CNTFET) transistors. Increased speed is obtained in the nanotubes, as it holds hefty current and higher driving capacity. The current mode architecture using CNTFETs further reduces the number of transistors, but the analog parameters considered for the design are tedious. Analysis of different logic styles and their performances in terms of number of transistors, static power, restoring of logic, cascade ability, and robustness are given. From the comparison [11] of the static logic circuits, the Differential Cascode Voltage Switch (DCVS) logic better suits the dual rail encoding in asynchronous QDI.

3. EXISTING METHOD:

3.1 ADAPTIVE WORD REORDERING:

The proposed encoding scheme (AWR) is based on the observation of the data stream over a fixed window of N words and the dynamic reordering of these words in order to decrease the total number of transitions on the encoded bus. The proposed scheme can be applied to memory interfaces that do not use asymmetric termination such as LPDDR3 where power can be saved by reducing the number of bit transitions. Furthermore, access to LPDDR3 is burst oriented. Therefore, a block of read or write data is fully available at the beginning of the transmission, which is required by the encoder of AWR in order to not introduce additional latency.

The route followed by the NN algorithm to reduce transitions is highlighted. The pseudocode of the algorithm is shown in below Figure. N transmissions are required to transfer the N words over the interconnect. Therefore, index i is used to count the transmissions, whereas index j is used to iterate between all the words to find the one with the minimum Hamming distance for each transmission. If a word has been transmitted, then the Hamming distance is not calculated and the word is not considered for retransmission.

```

while Queue with words not empty do
  Update the  $N$  words to be reordered
  Assign a unique code of  $K$  bits to all  $N$  words
  for  $i \leftarrow 0$  to  $N - 1$  do
    for  $j \leftarrow 0$  to  $N - 1$  do
      if word[ $j$ ] not transmitted then
        Calculate the Hamming distance between
        word[ $j$ ] and previously transmitted word
        including the unique code of  $K$  bits
      end if
    end for
    Transmit the word with the minimum Hamming
    distance
  end for
end while

```

Fig1. Pseudocode of the NN algorithm amended to the requirements of AWR.

Three state-of-the-art encoding techniques are selected that do not require a priori knowledge of data statistics and a circuit implementation is provided. BI [5] is a low-power adaptive scheme that calculates the Hamming distance of consecutive data words and inverts the transmitted data word if the Hamming distance is higher than half of the word length. To indicate whether a word is inverted or not, one extra bus line is used. APBI [19] observes the data stream for a window of a fixed number of N words and forms a mask with the bus lines with a higher probability of switching. BI is then applied to these bus lines and one extra bit is used to inform the decoder about inversion. ABE selectively encodes a cluster of highly correlated bus lines. First, observe the data characteristics over a window of N words. Based on these characteristics, a cluster is formed and the line with the maximum correlated transitions with the lines of the cluster is selected as the basis line.

The lines in the cluster are finally XORed with the basis. The basis and cluster information are transmitted using a redundant bus line and an additional clock cycle at the beginning of the window, respectively. The parameters of the compared techniques APBI and ABE are specifically selected such that the techniques yield the highest reduction in switching activity according to [19] and [28]. In this way, a fair comparison is conducted that provides the highest savings for each technique. For all of the simulations, the bus width is considered to be $M = 64$ bits, whereas the observation window is $N = 32$ words for APBI and $N = 16$ words for ABE. The mask computation of the APBI technique can be executed in each window of N words (APBI1), but intervals of 16 windows (APBI16) are also explored to further reduce the power consumption of encoding and decoding and provide a fairer comparison. Therefore, both of these scenarios are included. Furthermore, two cases for ABE are considered. In the first case, ABE is applied to the entire bus (ABE1), whereas in the second case, the bus is split into four groups of $M/4$ bits and ABE is applied individually to each group (ABE4) to best exploit this technique. The decrease in switching activity of AWR is reported using both $N = 32$ (AWR32) and $N = 64$ (AWR64) reordered words. The savings increase with a higher number of reordered words; however, the power overhead of encoding and decoding increases faster. Therefore, 32 and 64 reordered words are selected as high savings in switching activity are provided, while the power overhead is restrained.

3.1.A. ENCODER: The encoding of data is implemented as follows. In each clock cycle, the Hamming distances between the previous word and the words that have not yet been transmitted are evaluated. The word with the lowest Hamming distance is then transmitted through the bus. This method requires N registers at the transmitter and the receiver to store the reordered words. Conventionally, the computation of the Hamming distance is implemented using adder trees. This approach is highly inefficient in terms of power, especially for wide buses, counteracting any energy savings produced from encoding. Therefore, a different approach is followed, where the Hamming distance is determined as a delay in the time domain, drastically reducing the overhead in power due to encoding. The proposed encoder circuit comprises three stages, as shown in Fig. 3. The first stage is the race stage, where a variable delay line is assigned to each word. In each delay line, a clock pulse is propagated and delayed according to the number of bits that switch. The delay is shorter for a lower number of transitions and, thus, the fastest signal corresponds to the word with the lowest Hamming distance. The DEL signal that arrives first at the finish line of the race stage prevents the others from propagating. This condition is implemented in the finish stage. Before any signal arrives, a “0” is stored in all of the latches (LA) and the PER signal is set to 1 using a weak pull-up resistor. The signal that arrives first sets the respective latch and resets PER. After all the DEL signals are reset, PER switches slowly back to 1 due to the weak pull-up resistor

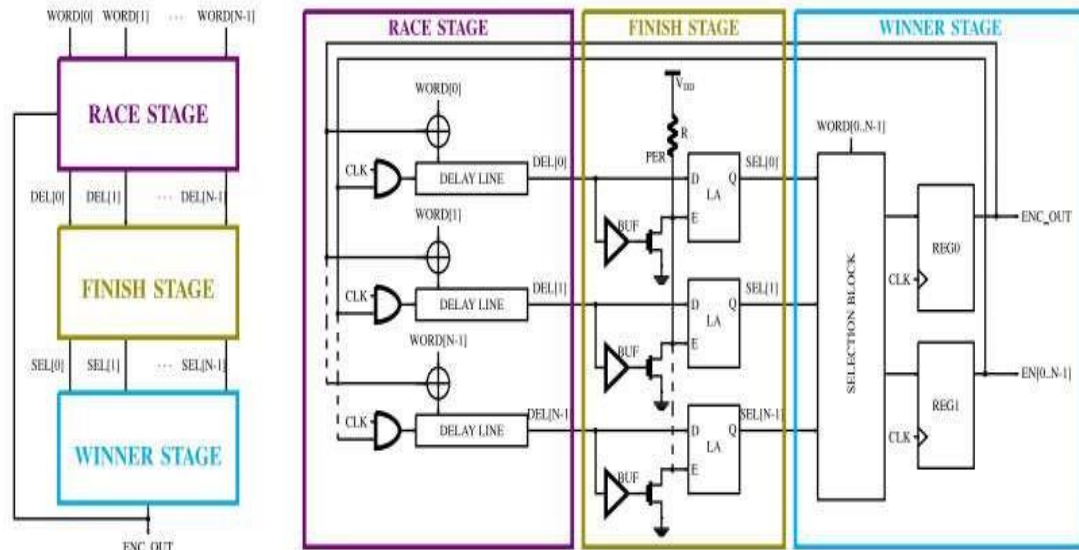


Fig. 2. Encoder circuit.

The winner stage is composed of the selection block and two registers. The selection block is a digital circuit that decides which word wins the race according to the received SEL[0 ... N] signals. In case two or more signals arrive at the same time (i.e., these words yield the same number of switching bits), such that more than one of the SEL signals is equal to 1, the word with the lowest index is chosen. Transmitting the word with the lowest index can potentially decrease the delay of decoding. For example, if WORD[0] and WORD[7] arrive at the same time, there is no benefit in transmitting WORD[7] since the receiver cannot utilize WORD[7] if all the previous words have not been read. The winning word is stored in the register REG0. To keep track of the transmitted words, a second register (REG1) is used, where the enable signals EN[0 ... N - 1] are stored. Once the word to be transmitted is selected, the respective EN signal is switched to 0 and remains low until all N words are transmitted. All EN signals switch to high whenever a new block of N words is to be transmitted. EN signals enable or disable the respective delay lines allowing the propagation of the clock only for the words that have not been transmitted. To enable and disable the delay lines, an AND gate is added in the beginning of each delay line. This mechanism not only prevents from transmitting a word twice but also saves dynamic power since the inverters in the delay lines of transmitted words do not switch. An example of signal propagation with N = 2 is shown in Fig. 4. It is assumed that in the first clock cycle, the Hamming distance of Word[1] is the lowest; therefore, DEL[1] is set to 1 faster than DEL[0]. DEL[1] causes both SEL[1] and PER signals to transition. Thus, SEL[1] is set to 1 and the latches are disabled before DEL[0] transitions to 1. PER switches slowly back to 1 after both DEL[1] and DEL[0] are reset. In the second clock cycle, the state of EN[1] changes to 0 as Word[1] was selected, whereas the clock pulse does not propagate through the delay line of Word[1]; therefore, DEL[1] remains 0. Word[0] is selected in this cycle since DEL[0] is the only signal that transitions to 1, causing SEL[0] and PER to flip. In the third clock cycle, both EN[0] and EN[1] are equal to 1 in order to enable the reordering of the next two words.

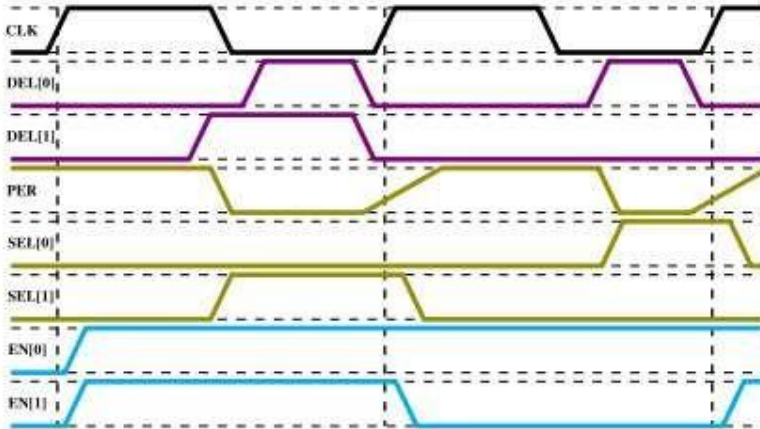


Fig. 3 Signal propagation of the encoder circuit

The delay line consists of a modified inverter chain as shown in Fig. 4, where W is the minimum width and the length is the minimum for all devices as determined by the utilized technology library. Each inverter is connected either to the ground or the supply voltage through a pair of devices. A detailed description of this delay line can be found in [5]. Briefly, when the i th bit switches, $T(i)$ switches to 1 and the i th inverter is connected to ground (i is odd) or VDD (i is even) through only one device. In case no transition takes place, the inverter is connected to either VDD or ground through two devices connected in parallel. Hence, in the former case, the delay is larger than in the latter case. The delay of only the first edge of each inverter (falling for even and rising for odd) is affected by the Hamming distance since the pair of devices is only added to either the pull-down (even inverters) or pull-up (odd inverters). The delay of the second edge of each inverter is constant. This implementation ensures that the delay of the falling edges of the DEL signals is constant. Therefore, the PER signal switches to 1 at a specific time in every clock cycle as this happens exactly when DEL signals switch to 0. Hence, the risk of a DEL signal switching to 1 while PER is still low is eliminated.

3.1.B. DECODER:

The role of the decoder at the receiver side is to place the words back in the initial order. To achieve that, the decoder

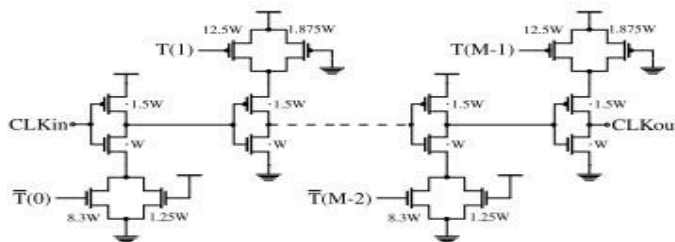


Fig. 4. Delay line circuit

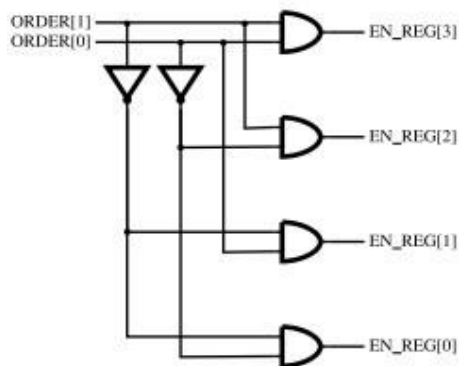


Fig. 5. Decoder circuit.

uses N registers to store the words in the right order as they arrive. The order of each word is transmitted by using low spatial redundancy. K more bits are added to each word that indicates the order. Thus, for N words, $K = \log_2 N$ additional bus lines are required. The decoder reads the K bits and enables the corresponding register to store the word, whereas the rest of the registers remain disabled. The circuit that implements this function is a basic K -to- N decoder with each output connected to the enable input of the appropriate register. A two-to-four decoder circuit is shown in Fig. 5. The K bits are also considered in the encoding process, and thus, they are included in the calculation of the Hamming distance. Consequently, it is ensured that they do not considerably increase the power overhead.

4. PROPOSED METHOD:

4.1 GATE DIFFUSION TECHNIQUE:

The GDI method is based on the use of a simple cell as shown in Fig. 1. At a first glance the basic cell resembles the standard CMOS inverter, but there are some important differences: GDI cell contains three inputs – G (the common gate input of the nMOS and pMOS transistors), P (input to the outer diffusion node of the pMOS transistor) and N (input to the outer diffusion node of the nMOS transistor). The Out node (the common diffusion of both transistors) may be used as input or output port, depending on the circuit structure.

The GDI cell is similar to a CMOS inverter structure. In a CMOS inverter the source of the PMOS is connected to VDD and the source of NMOS is grounded. But in a GDI cell this might not necessarily occur. There are some important differences between the two. The three inputs in GDI are namely-

- 1) G - common inputs to the gate of NMOS and PMOS
- 2) N - input to the source/drain of NMOS
- 3) P - input to the source/drain of PMOS

Bulks of both NMOS and PMOS are connected to N or P (respectively), that is it can be arbitrarily biased unlike in CMOS inverter. Moreover, the most important difference between CMOS and GDI is that in GDI N , P and G terminals could be given a supply 'VDD' or can be grounded or can be supplied with input signal depending upon the circuit to be designed and hence effectively minimizing the number of transistors used in case of most logic circuits (eg.

AND, OR, XOR, MUX, etc). As the allotment of supply and ground to PMOS and NMOS is not fixed in case of GDI, therefore, problem of low voltage swing arises in case of GDI which is a drawback and hence finds difficulty in case of implementation of analog circuits.

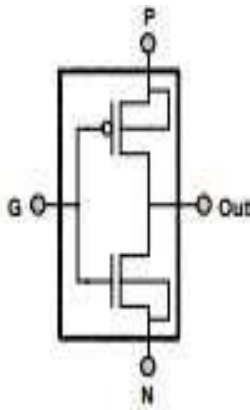


Fig 6: Basic GDI gate

Multiple-input gates can be implemented by combining several GDI cells. The buffering constrains, due to possible VT drop are described in detail in [8], as well as the technological compatibility with CMOS (and with SOI). Morgenshtein has proposed basic GDI cell shown in Fig.1 [8]. This is a new approach for designing low power digital combinational circuit. GDI technique is basically two transistor implementation of complex logic functions which provides in-cell swing restoration under certain operating condition. This approach leads to reduction in power consumption, propagation delay and area of digital circuits is obtained while having low complexity of logic design. An important feature of GDI cell is that the source of the PMOS in a GDI cell is not connected to VDD and the source of the NMOS is not connected to GND. Therefore GDI cell gives two extra input pins for use which makes the GDI design more flexible than CMOS design.

5. COMPARATIVE STUDY OF GDI AND CMOS SCHEMATICS:

Gate		
XOR		
	Transistor count- 4	Transistor count- 8 +4(for generating A' & B')

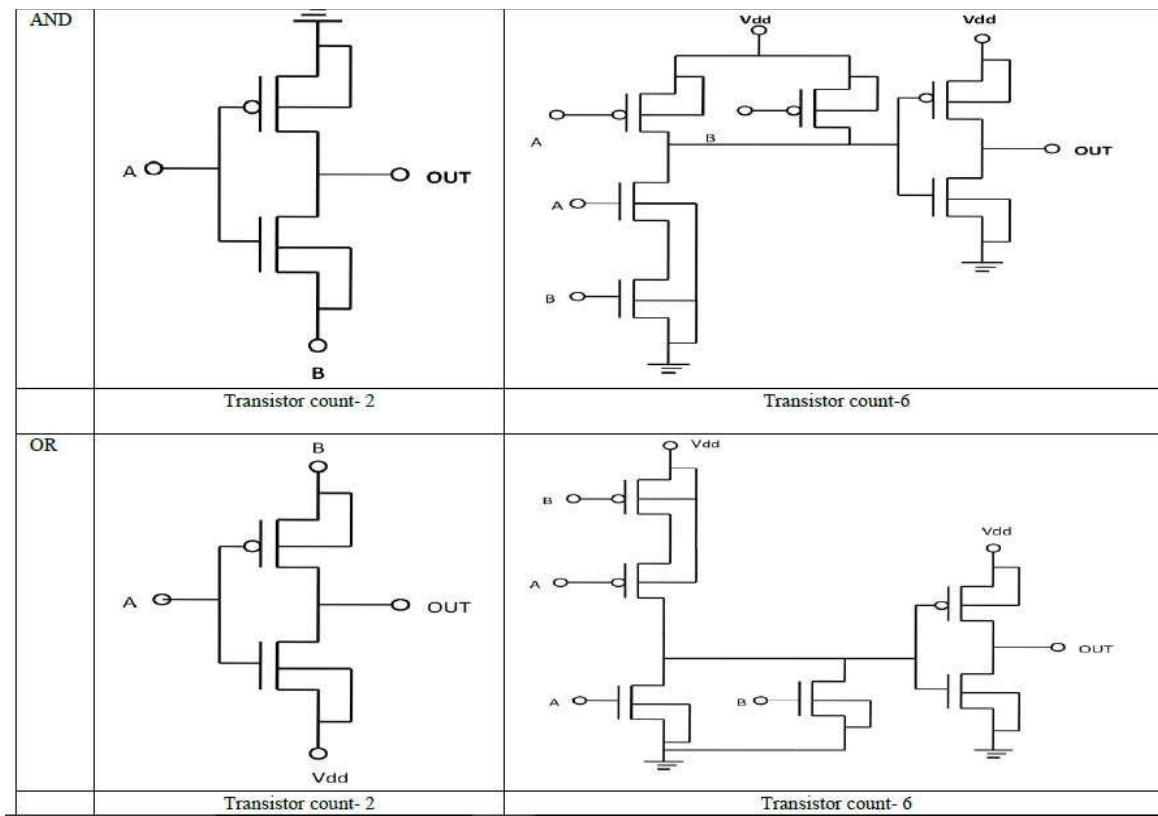


Fig 7: GDI vs CMOS gates

The GDI cell has $n + 2$ inputs when compared to CMOS; Arkadiy Morgenshtein et al. (2002) analysed the performance of GDI in terms of noise margin, body effect, fan out and delay etc. In realizing the function, $F1 = I.b$, it has been found that the behavior of GDI cell is similar to that pMOS pass transistor logic in which the output is at V_{tp} instead of 0. This is the only case where logic degradation of one V_t takes place. In order to restore the logic, the GDI based buffer is added at the output. The GDI cell can act as a buffer when $P=1$ which performs its logic evaluation and also restore the logic. This is a main advantage of GDI cell. The GDI cells have less V_t drop in some of the transitions at the output of the cell. With a proper interconnection of the cells, several GDI cells are connected in series or parallel without accumulating the voltage drop. In order to restore the logic swing, an inverter is used at the output (which can also perform its logical function). In this way, the V_t drop and noise margin reductions are localized. However, there are also potential advantages in GDI in terms of reliability. They include these:

- x The lower voltage levels have lower impact due to crosstalk on neighboring wires.
- x The fact that complex functions are built by using multiple instances of the same GDI cell contributes to reduced variability,
- x Smaller area and number of transistors in GDI mean shorter interconnects and less crosstalk and these enable more efficient place and route.

Various logic functions for different input combinations of GDI cell, which are used in this design are furnished. Most of these functions when implemented using CMOS, as well as in regard to standard pass transistor logic implementations are complex (6–12 transistors). But they are very simple (only two transistors per function) in the GDI design method.

6. **TANER EDA:**

Tanner EDA is a suite of tools for the design of integrated circuits. Tanner EDA is mainly used to analyze circuits at switch level & gate level. These are tool used to enter schematics perform SPICE simulations do physical design (i.e., chip layout) perform design rule checks (DRC) and layout versus schematic (LVS) checks.

Tanner EDA Design Tools:

- S-edit - a schematic capture tool
- T-SPICE - the SPICE simulation engine integrated with S-edit
- L-edit - physical design tool
- W-edit - waveform formatting

Improve simulation accuracy with advanced modeling features

T-Spice provides extensive support of behavioral models using Verilog-A, expression controlled sources, and table-mode simulation. Behavioral models give you the flexibility to create customized models of virtually any device. T-Spice also supports the latest industry models, including the transistor model recently selected as the next standard for simulating future CMOS transistors manufactured at 65 nanometers and below—the Penn State Philips (PSP) model. PSP will simplify the exchange of chip design information and support more accurate digital, analog, and mixed-signal circuit behavior analysis.

- Enables easy creation of syntax-correct SPICE through a command wizard.
- Highlights SPICE Syntax through a text editor.
- Provides Fast, Accurate, and Precise options to enable optimal balance of accuracy and performance.
- Enables you to link from syntax errors to the SPICE deck by double clicking.
- Supports Verilog-A for analog behavioral modeling, allowing designers to prove system level designs before doing full device level design.
- Provides “.alter” command for easy what-if simulations with netlist changes. Perform sophisticated analysis

Sophisticated Analysis

T-Spice uses superior numerical techniques to achieve convergence for circuits that are often impossible to simulate with other SPICE programs. The types of circuit analysis it performs include:

- DC analysis (DC Operating Point Analysis & DC Transfer Analysis.)
- AC analysis
- Transient analysis with Gear or trapezoidal integration.
- Noise analysis.
- Monte Carlo analysis over unlimited variables and trials.
- Virtual measurements with functions for timing, error, and statistical analysis.
- Parameter sweeping using linear, log, discrete value, or external file data sweeps.
- Transient Analysis, Power-up Mode.

7. RESULTS:

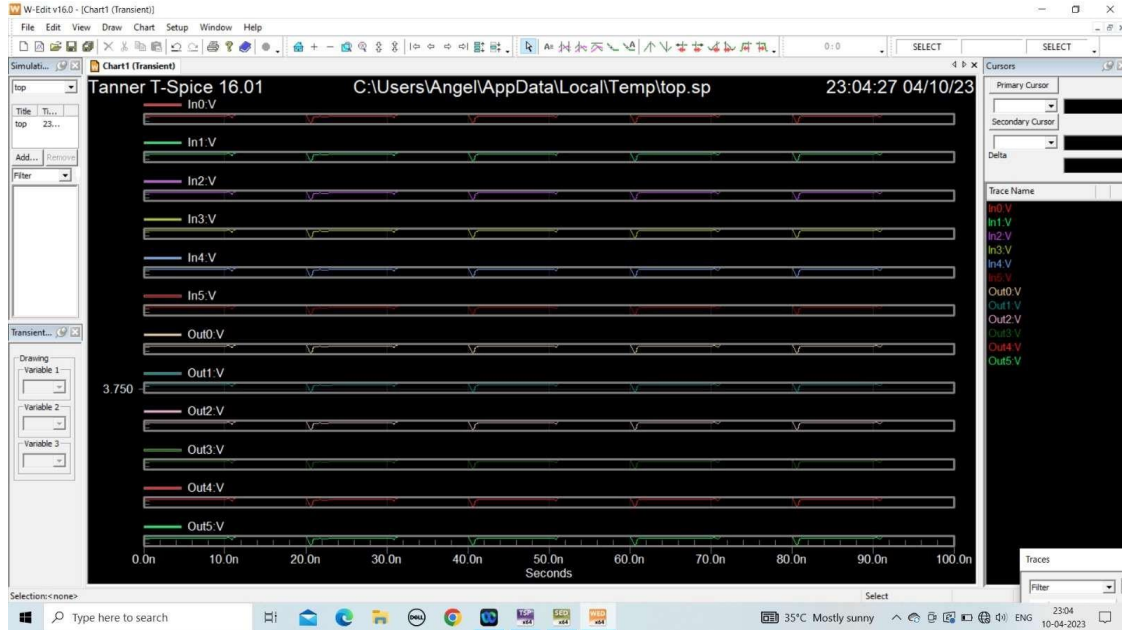


Fig 8: Proposed simulation result

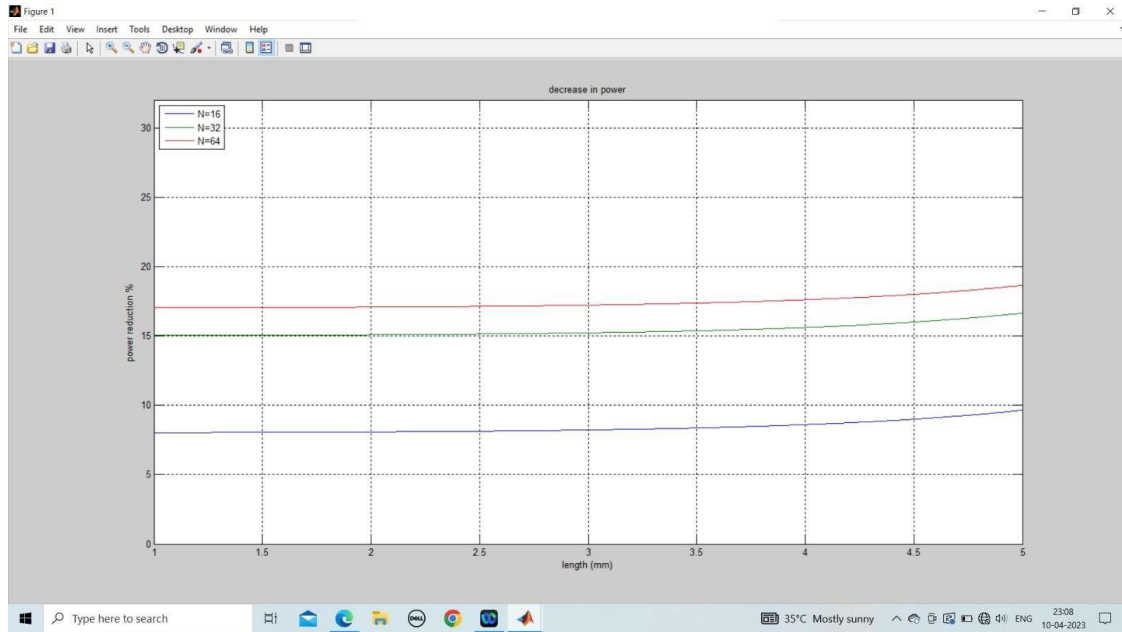


Fig 9: Proposed Power reduction analysis

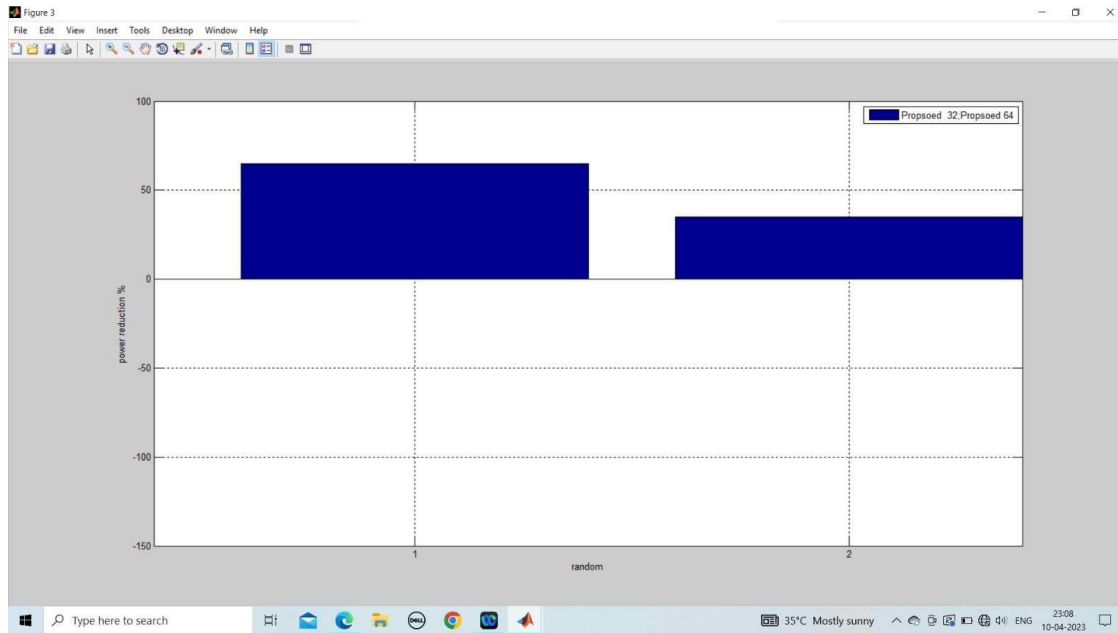


Fig 10: Proposed power reduction for random numbers

Table1: Comparison table

	EXISTING	PROPOSED
AREA (um)	23526	20556
TIME (Sec)	3	2.32
POWER Reduction (Watts)	16.4	13.0

8. CONCLUSION: In this project, an adaptive encoding scheme (AWR) was proposed for parallel off-chip interconnects. AWR decreased the interconnect power by changing the order of the transmitted words to effectively reduce switching activity. AWR was adaptive, and thus, a priori knowledge of the statistical characteristics of data was not required. Notably, the circuit power overhead of AWR was restrained by proposing a novel encoder circuit, which exploited the time domain for the computation of Hamming distances by replacing power-hungry adder trees with delay lines. AWR outperformed state-of-the-art techniques in terms of both decrease in switching activity and overall power savings for real data streams for several applications relating to high-performance computing where energy reduction is a primary objective. As an extension of this method, Gate Diffusion input architectures are implemented in designing of encoder circuit in order to reduce power constraints and gate density parameters.

9. FUTURE SCOPE:

Future work should focus on the minimization of transition state by correct techniques of the region, and the overhead efficiency. We are still seeking to use such an encoding strategy that reduces the area overhead due to encoder and decoder circuit.

References:

- [1]. Abinesh R., Bharghava R., Suresh Purini, Govindarajulu Regeti, "Transition Inversion based Low Power data coding scheme for Buffered Data Transfer", Accepted for publication in special issue of Journal of Low Power Electronics, October 2010.
- [2] Abinesh R., Bharghava R., Suresh Purini, Govindarajulu Regeti, "Transition Inversion based Low Power data coding scheme for Buffered Data Transfer", 23rd International Conference on VLSI Design, January 2010
- [3] Abinesh R., Bharghava R., M.B. Srinivas, "Transition Inversion Based Low Power Data Coding Scheme for Synchronous Serial Communication", in VLSI, pp.103-108, 2009 IEEE Computer Society Annual Symposium on VLSI, 2009
- [4].<http://www.intel.com/consumer/products/style/netbook.htm> Net book vs. Laptop and Entry Level Desktops.
- [5]. Net book design considerations by Texas Instruments. <http://focus.ti.com/docs/solution/folders/print/581.html>
- [6]. Nano-cmos scaling problems and implications. Nano-CMOS Circuit and Physical Design, Ban P. Wong, Anurag Mittal, Yu Cao, and Greg Starr, John Wiley & Sons Inc.
- [7] M. Ghoneima, Y. Ismail, M. Khellah, J. Tschanz, and V. De, "Serial-link bus: A low-power on-chip bus architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 2020–2032, Sep. 2009.
- [8] B. Razavi, "Challenges in the design of high-speed clock and data recovery circuits," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 94–101, Aug. 2002.
- [9] Eleni Maragkoudaki , Member, IEEE, and Vasilis F. Pavlidis , Senior Member, "Energy-Efficient Time-Based Adaptive Encoding for Off-Chip Communication" in IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS.
- [10]. Dhiman, G. and Rosing, T. S. 2007. Dynamic voltage frequency scaling for multitasking systems using online learning. In Proceedings of the 2007 international Symposium on Low Power Electronics and Design (Portland, OR, USA, August 27 - 29, 2007). ISLPED '07. ACM, New York, NY, 207-212.
- [11]. M. R. Stan, W. P. Burleson. Bus-Invert Coding for Low Power I/O, IEEE Transactions on Very Large Integration Systems, Vol. 3, No. 1, pp. 49-58, March 1995.
- [12] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, "Asymptotic zero-transition activity encoding for address busses in lowpower microprocessor-based systems," in Proc. Great Lakes Symp. VLSI, Mar. 1997, pp. 77–82.
- [13] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano, "Address bus encoding techniques for system-level power optimization," in Proc. Conf. Design, Automat. Test Eur., 1998, pp. 861–866.

- [14] L. Benini, G. De Micheli, E. Macii, M. Poncino, and S. Quez, "Systemlevel power optimization of special purpose applications: The beach solution," in Proc. Int. Symp. Low Power Electron. Design, 1997, pp. 24–29.
- [15] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "A coding framework for low-power address and data busses," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 7, no. 2, pp. 212–221, Jun. 1999.
- [16] R. Wille, O. Keszocze, S. Hillmich, M. Walter, and A. Garcia-Ortiz, "Synthesis of approximate coders for on-chip interconnects using reversible logic," in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Mar. 2016, pp. 1140–1143.
- [17] L. Benini, A. Macii, M. Poncino, and R. Scarsi, "Architectures and synthesis algorithms for power-efficient bus interfaces," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 19, no. 9, pp. 969–980, Sep. 2000.
- [18] Y. Shin, S.-I. Chae, and K. Choi, "Partial bus-invert coding for power optimization of system level bus," in Proc. Int. Symp. Low Power Electron. Design (ISLPED), 1998, pp. 127–129.
- [19] C. Kretzschmar, R. Siegmund, and D. Müller, "Adaptive bus encoding technique for switching activity reduced data transfer over wide system buses," in Proc. Int. Workshop Integr. Circuit Design, Power Timing Modeling, Optim. Simulation, Sep. 2000, pp. 66–75.
- [20] M. Alamgir, I. I. Basith, T. Supon, and R. Rashidzadeh, "Improved busshift coding for low-power I/O," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2015, pp. 2940–2943.
- [21] E. Musoll, T. Lang, and J. Cortadella, "Working-zone encoding for reducing the energy in microprocessor address buses," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 6, no. 4, pp. 568–572, Dec. 1998.
- [22] J. Yang and R. Gupta, "FV encoding for low-power data I/O," in Proc. Int. Symp. Low Power Electron. Design (ISLPED), Aug. 2001, pp. 84–87.
- [23] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, "An adaptive dictionary encoding scheme for SOC data buses," in Proc. Design, Automat. Test Eur. Conf. Exhib., Mar. 2002, pp. 1059–1064.
- [24] M. N. Bojnordi and E. Ipek, "DESC: Energy-efficient data exchange using synchronized counters," in Proc. IEEE/ACM Int. Symp. Microarchitecture, Dec. 2013, pp. 234–246.
- [25] P. Behnam, N. Sedaghati, and M. N. Bojnordi, "Adaptive time-based encoding for energy-efficient large cache architectures," in Proc. 5th Int. Workshop Energy Efficient Supercomput., Nov. 2017, pp. 1–8.
- [26] P. Behnam and M. N. Bojnordi, "STFL: Energy-efficient data movement with slow transition fast level signaling," in Proc. 56th Annu. Design Automat. Conf., Jun. 2019, pp. 1–6.
- [27] P. Behnam and M. N. Bojnordi, "STFL-DDR: Improving the energyefficiency of memory interface," IEEE Trans. Comput., early access, Mar. 6, 2020, doi: 10.1109/TC.2020.2978826.