LOW POWER VLSI DESIGN WITH RESISTIVE FEEDBACK LOGIC

[1]C.Ashok Kumar, [2]Dr.B.K.Madhavi, [3]Dr.K.Lal Kishore

[1]. Prof.& Head, ECE dept, TRR Engg. College, Hyderabad, AP., cakk_kumar@yahoo.co.in

[2]. Prof., ECE Dept, SNIST, Ghatkesar, Hyderabad, AP-India.

[3]. Rector, JNTU college of Engg. Kukatpally, Hyderabad-India.

ABSTRACT

These papers focus on the development of low power VLSI design methodology on system level modeling and circuit level modeling for power optimization. The developed transition optimization approach further merged with circuit level power optimization using Glitch minimization technique. A resistive feed back method is developed for the elimination of glitches in the CMOS circuitry, which result in power consumption and reducing performance of VLSI design. The optimized sequence is then processed through a 8-bit register bank modeled in CMOS level for data transfer to observe the glitch effect. Tanner EDA tool is used for the designing of the CMOS circuitry with resistive feedback mechanism for power optimization.

Keywords

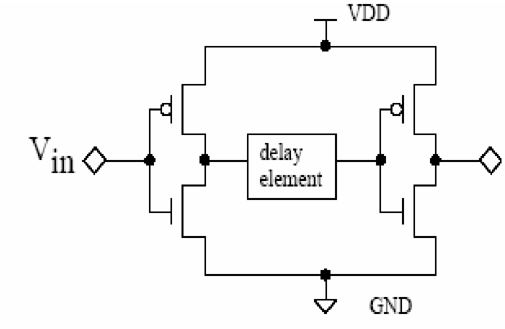
Low power VLSI, glitch free modeling, Resistive feedback logic, stray capacitance.

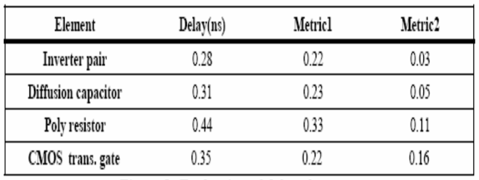
1. INTRODUCTION

Dynamic power is a major component in the overall power dissipation of a CMOS circuit. It can be reduced by minimizing the number of transitions of signals. Besides the logic transitions, glitches (or hazards) also consume power. Reducing these can save up to 30 to 70% of the total power dissipation. A general solution to glitch elimination involves gate delay manipulation to balance paths and to filter hazards as discussed in recent papers by Agrawal and Raja et al. [5],[6],[7],[8]. These methods are not applicable to standard-cell ASICs, where delays of library cells cannot be arbitrarily changed. We design a resistive feed through cell whose delay can be customized. Using this, we accomplish path delay balancing and hazard filtering for ASICs. For details the reader may refer to [11]. Most of the reported work on standard cell ASICs, i.e., by Hashimoto and Onodera [1], Masgonty et al [4], Scott and Keutzer [9], Sulistyo and Ha [10], and Zhang et al. [13] ignores the glitch suppression aspect of low-power design.

2. DESIGN OF A DELAY CELL

To implement a delay element as standard cell that can fit into a 0.25µ CMOS cell library used in this work, we analyzed several delay elements. These were an inverter pair, an n-diffusion capacitor, a poly silicon resistor and a CMOS transmission gate. Among these, the inverter pair and the transmission gate have been used before by Mahapatra et al. [3] and Raja et al [8]. The delays of these elements are controlled by changing the W/L ratio of the transistors. The formulation might become non-linear, since changing the W/L ratio of a MOSFET changes the channel resistance as well as the associated parasitic capacitances. This motivated us to look at delay elements that are entirely capacitive or entirely resistive so that the RC time constant of the element can be controlled linearly. Thus, we tried an n-diffusion capacitor and a resistor made from poly silicon wire with a silicide blocking mask over it. The silicide blocking mask increases the resistivity of the poly silicon resistor. To simulate a realistic situation, each delay element is driven by an inverter gate and is also loaded with an inverter. The circuit set up is shown in Figure 1. The connection for the n-diffusion capacitor is different from that depicted in the figure. The capacitor is connected such that it increases the load capacitance of the driving inverter. The circuit simulator used to measure the delay and power consumption is the SpectreTM circuit-level simulator from Cadence (2003). The results of this experiment are presented in Figure 1. The inverter and the transmission gate cells are made of transistors with minimum sizes. The resistor cell was designed for a resistance of 15.4kΩ and the capacitor cell was designed for a capacitance of 2.7fF.





Element	Delay(ns)	Metric1	Metric2
Inverter pair	0.28	0.22	0.03
Diffusion capacitor	0.31	0.23	0.05
Poly resistor	0.44	0.33	0.11
CMOS trans. gate	0.35	0.22	0.16

Figure 1: Evaluation of delay elements.

The average delay in the table of Figure 1 is the average of the rise and fall delays of the delay element alone. This value is calculated as the difference between the delay of the entire circuit shown in Figure 1 and the delay of the same circuit without the delay element. The delay is expressed in nanoseconds. The delay element should also be optimized for size and power consumption. We use two metrics for the circuit of Figure 1, delay/power (metric 1) and delay/area (metric 2). The delay values used to calculate both metrics are given in column 2 of the table in Figure 1. The power consumption values used to calculate metric 1 are expressed in µW. Since the delay elements are implemented as standard cells with fixed height, the area is measured in terms of the number of grid units along the width. We observe that the poly silicon resistor introduced the maximum amount of average delay. Since the poly silicon resistor also had the highest metric 1 value, we can conclude that this element introduces maximum delay per unit power consumption of the circuit in Figure 1. Similarly, metric 2 indicates the amount of delay introduced per unit area overhead. Though the transmission gate had the highest metric 2 value, since saving power is our primary objective, we chose the poly silicon resistor in our effort to reduce the glitch power. We call this delay element a resistive feed through cell. At logic level this element was modeled as a fictitious buffer. This cell was designed as a parameterized cell, i.e., the physical design of the cell is generated on-the-fly according to the required delay. Resistors are relatively easy to integrate and have been used extensively in the past to build analog circuits as discussed by Hastings (2001). The resistance of a rectangular slab of length L, width W and thickness t can be calculated in terms of a material specific constant called resistivity ρ, as R=ρL/(Wt). Integrated resistors consist of diffusions or depositions that can be modeled as films of constant thickness. It is therefore customary to combine resistivity and thickness into a single term called the sheet resistance Rs. The formula can be written as: R = Rs L/W, where Rs=ρ/t. The L/W ratio is given the units of squares and Rs of a material is given in units of Ω per square. Layout designers generally maintain uniform width and control the resistance by varying the length. This is done to avoid two main factors that affect the integrated resistor -- width bias and non uniform current flow. The details of these effects can be found in a book by Hastings (2001). This practice has an indirect advantage that the delay across an integrated resistor can be formulated as a linear function of the length. Resistors are often folded in a serpentine manner employing rectangular turns. The rectangular turns are easy to draw and the spacing between the turns is easily adjusted. A rectangular corner adds a resistance of approximately 1/2 square. In a CMOS process, a resistor can be implemented using metal layers, the poly silicon layer, diffusion layers or the n-well layer. Among these materials the poly silicon forms the best resistor. The narrower poly silicon pitch will result in a smaller layout and they are less susceptible to parasitics. However, the poly silicon used for constructing MOS gates is heavily doped to improve conductivity resulting in smaller sheet resistance. Most CMOS processes provide a silicide blocking mask to block the doping and increase the sheet resistance. For a 0.25μ CMOS process, we found that the silicide poly has a sheet resistance of 3.6Ω/square but with a silicide blocking mask the sheet resistance is 173.6Ω/square. We implemented the feed through cell as a serpentine path of poly with a silicide blocking mask over it. The width of the path is kept to a minimum. This cell is designed as a parameterized cell so that delay can be varied continuously. The delay is controlled by varying the length of the poly wire. Since the delay depends only on the length of the poly wire, we get a linear formulation. Also, we get very high resolution since the length can be varied in steps of the minimum feature size (λ) of a particular technology. For the 0.25μ CMOS technology with λ=0.15μ, we found that the resistance of the feed through cell could be varied in steps of 95Ω. The layout of a resistive feed through cell to give a resistance of 20kΩ is shown in Figure 2. Since the poly silicon resisitive cell has a regular geometry, the automation of the physical design is fairly simple and these cells can be easily implemented as parameterized cells with the length of the poly silicon wire as the parameter. Compared to a library where all logic gates are implemented as parameterized cells, this approach is economical in design effort.

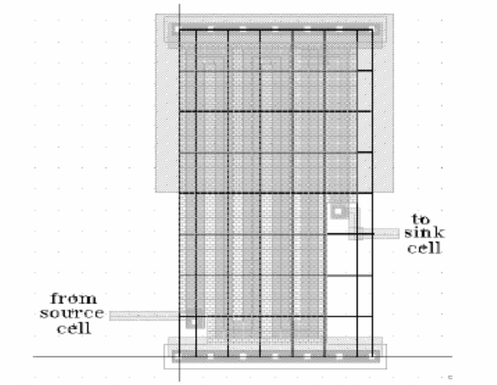
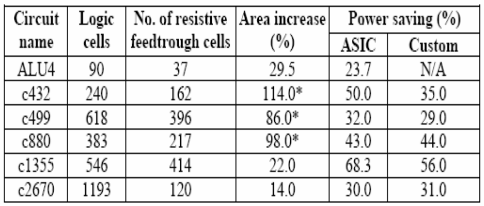


Figure 2: Layout of a resistive feedthrough cell.

3. GLITCH-FREE PHYSICAL DESIGN

The resistances found from the lookup table are automatically designed as standard cells according to the procedure described above. These feed through cells are inserted into the original circuit by modifying a Verilog netlist of the circuit. The place-and-route layout of the modified netlist is then done using a commercial tool such as Silicon EnsembleTM (2003). The circuit is extracted from layout and simulated to check for any violations in the timing behavior. If necessary, we trim the resistances of the feed through cells to correct the routing delays. The trimming can be done by reducing the silicide blocking mask over the poly silicon resistors to reduce the resistance value. If an increase in resistance is required, then, instead of introducing a new feed through cell that could change the entire physical design of the circuit we can change the metal interconnects into poly silicon routes. We conducted experiments to verify that insertion of resistive feed through cells saves power by reducing the glitches. The Verilog netlist of a circuit and the timing information of the target standard cell library in ASCII format are given as inputs to our experimental setup to generate the layout of the minimum transient energy design. A C-program parses the circuit netlist and the cell delay file to output the LP in a language called AMPL. These files are used by the linear program solver called CPLEX (see Fourier et al. (1993)) to find the necessary wire delays. The wire delays are then used by a PERL script to generate the physical design of resistive feed through cells and to modify the initial circuit netlist by inserting the new delay elements. This netlist is then used for a final place-and-route by any commercial tool. In our experiments we used Silicon Ensemble TM (2003). The target cell library used for the physical design of the modified netlist also includes the various delay elements generated.



Circuit name	Logic cells	No. of resistive feedtrough cells	Area increase (%)	Power saving (%)	
				ASIC	Custom
ALU4	90	37	29.5	23.7	N/A
c432	240	162	114.0*	50.0	35.0
c499	618	396	86.0*	32.0	29.0
c880	383	217	98.0*	43.0	44.0
c1355	546	414	22.0	68.3	56.0
c2670	1193	120	14.0	30.0	31.0

Table 2: ISCAS85 benchmarks in standard-cell.

The power dissipation was estimated in two ways. At logic level we used an event driven logic simulator to estimate the dynamic power consumption as discussed by Raja [6]. This program uses the cell delays and wire delays obtained from LP and the node capacitances extracted from layout to estimate the power dissipation before and after optimization. At post layout level, the SpectreTM circuit simulator from Cadence (2003) was used to simulate the extracted circuit. The average power for various input vectors is then calculated by integrating the instantaneous current waveform. The results obtained by logic simulation of various circuits are presented in Table 2. The dynamic power saving is the decrease in the average power due to random vectors as a percentage of the average power of the un optimized circuit for the same vectors. These values are compared with a glitch removal technique for custom circuits that uses variable input delay gates described by Raja [6]. Table 2 shows that the average power saving is comparable to that achieved in the custom design of Raja et al [8]. For circuit's c432 and c1355 the power saved by our method is significantly greater. This is because the custom design had to insert 61 and 64 delay buffers in c432 and c1355, respectively. In our case the range of the resistance values that can be introduced on a signal line is large. Table 2 shows large area increases for three circuits, marked with asterisk (*). We should point out that the added resistances could have been reduced by selecting alternative larger delay logic cells available in the library. Then the area increase will be similar to other circuits. All optimized circuits had no increase in the input to output overall delay. We have assumed that the glitch removal technique has little or no impact on the short circuit and leakage power. To verify the validity of this assumption we performed the post layout circuit level simulations for the 4-bit ALU circuit. The pre-optimized and post optimized circuits were simulated for 1,000 random vectors with a cycle period of 10ns. Simulation gave a 22% power saving.

4. IMPLEMENTATION

Buses have been used as an efficient communication link among functional modules in very large scale integration (VLSI) systems. Whereas the size of functional modules decreases with the development of semiconductor technology, the size of VLSI chips increases, and so does the number of functional modules on a chip. Increasing communication requirements among the modules demand more complicated and more efficient buses. Currently, internal bus design plays important roles in the performance of a chip. Since too many modules rely on the buses for their communication, the buses are usually heavily loaded so that they dissipate quite an amount of power in operation. Activation of external buses consumes significant power as well, because many input–output (I/O) pins and large I/O drivers are attached to the buses. Typically, 50% of the total power is consumed at the I/Os for well-designed low-power chips by R.Wilson. Thus, reducing the power dissipated by buses becomes one of the most important concerns in low-power VLSI design. The dynamic power dissipated in a bus is expressed as the following N.Weste and K.Eshraghian [15]:

$$P_{\mathrm{BUS}} = \sum_{\mathrm{line}} C_{\mathrm{load}} V_{\mathrm{DD}}^{2} N_{\mathrm{trans}}$$

Where Cload is the total load capacitance attached to a bus line, VDD is the voltage swing at operation, and Ntrans is the number of transitions per second. There are two approaches to reduce the dynamic power of buses. One is to save the dynamic power per activation by reducing either Cload or VDD. Reduced swing bus is an example of this approach Y.nakagone through H.zhang. The other is to reduce the number of bus activations by coding. Bus-invert (BI) coding by M.R.Stan and W.P.Burleson, Gray code by C.L.Su, C.Y.tsui and A.M.Despain, and the beach solution by L.Benini, g.De micheli are some examples of this approach.Before the advent of internet and multimedia systems, most of I/O data used inVLSI chips are granulated data which are requested aperiodically on demand and consist of few bytes of discrete information.The previous coding schemes were devised for the applications with this kind of I/O patterns. However, a new pattern of data transmission has become a great concern with the widespread use of internet and multimedia systems. Data are transferred like a stream when used in applications such as MP3 players and video players. Once an operation is started, it requires to transmit large amounts of data from a few kilobytes to hundreds of megabytes. For example, web-surfing or downloading files from the internet involves transmission of a few kilobytes of data, while playing music or movies needs streaming data transmission up to a few gigabytes. As streaming becomes one of the major data transfer patterns, we have one more degree of freedom, i.e., the sequence of data that we can exploit to reduce the number of bus transitions during data transmission. A new coding scheme called sequence-switch coding (SSC) is proposed in this paper. It is different from previous transition-reduction coding schemes in that it is aimed at applications with the stream-type data transfer pattern. SSC reduces the number of bus transitions by rearranging the transmission sequence of data. An algorithm called lagger algorithm is presented to show the feasibility of SSC. This algorithm reduces around 10% of bus transitions in transmission of the benchmark files. Performance of SSC algorithms is evaluated by simulations. For the brevity of description, let us define some terms and notations first. In the rest of this paper, the term "word" is used as the unit of data transmission. Hence, the number of bits in a word is the same as the number of lines in a bus. A (transmission) cycle is defined as the period of time to send a word through a bus, typically one clock cycle. A switched sequence is defined as the sequence of words transmitted according to an SSC algorithm. Let us express the hamming distance between a word, W and a bus, B, as H (W; B).

4.1 Unbounded Lagger Algorithm

In this algorithm, only two most recent words are considered in switch operation. The encoder has a register called lagger register to store a word temporarily. Any word that stayed in this register is tagged as a lagger. At every cycle, two words, a lagger and a new incoming word compete for transmission so that the winner is sent and the looser is held as the lagger for next competition. The coding information is transmitted simultaneously with the word by an auxiliary line called S-line that is added in parallel to the bus. The S-line is set to 1 when a latter word is selected while it is reset to 0 whenever a lagger is transmitted. If a switched sequence is compared with the original sequence, all words displaced from the original positions are laggers.

Furthermore, the original sequence can be recovered from a switched sequence by relocating every lagger to the position of its preceding lagger.

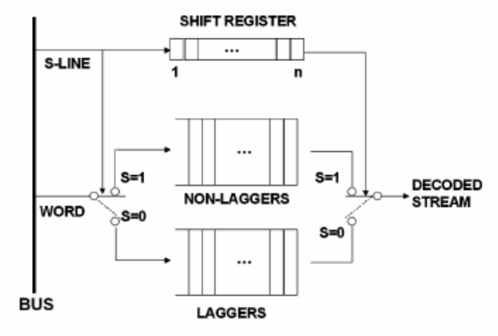


Fig. 3. An implementation of the decoder for lagger algorithm.

The first lagger is placed at the beginning of the sequence. Using this property, decoding can be achieved with two first in–first out (FIFO) buffers and a shift register (SR) as in Fig. 3. The one buffer is reserved for laggers and the other is for non laggers. The SR consists of n bits that are initially reset to 0. Assume that the decoder sends a decoded word at the first-half cycle while it receives a word from the bus at the second-half cycle. For every cycle, the decoder receives a word with an S-value from the bus, and stores the word in either of the buffers according to the S-value. SR is shifted from SR1 to SRn, and the S-value goes to SR1. For the first n - 1 cycle, i.e., until the first S-value reaches to SRn-1, the decoder only stores the received words. From nth cycle, the decoder starts to send a word to the next stage (client). According to the value of SRn, the decoder fetches a word from either of the buffers and sends it to the client. The latency of the decoder becomes n - 1 cycles. Let us define lagging distance of a lagger as the number of its losses in the competitions, and the maximum lagging distance (lmax) of a switched sequence as the biggest lagging distance of all laggers in the sequence. Then, n should be greater than or equal to lmax +2, and the FIFOs should have at least lmax + 1 slots. The shortcoming of UL is that lmax is not predictable in advance. Because it can be a very large number, the decoder should prepare a huge number of slots.

5. RESULT OBSERVATIONS

For the implementation of the suggested method a simulation modeling of the switch logic and a CMOS modeling of the feedback logic is developed. The obtained simulation results and the implementation are as presented below, The CMOS modeling developed is developed on the tanner EDA tool.

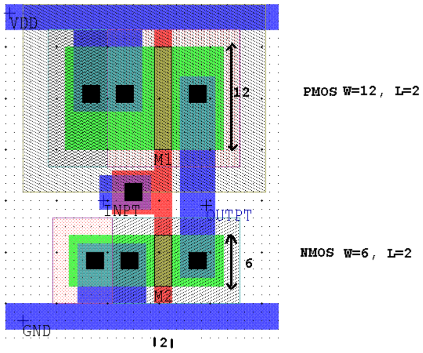
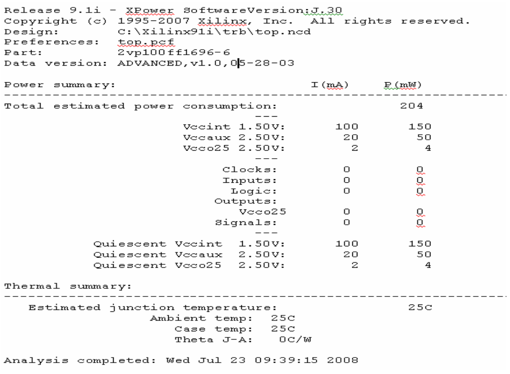


Figure 4: CMOS logic developed with a inverter operation



```
Release 9.1i - XPower SoftwareVersion:J.30
Copyright (c) 1995-2007 Xilinx, Inc.  All rights reserved.
Design:        C:\Xilinx91i\trb\top.ncd
Preferences:   top.pcf
Part:          2vp100ff1696-6
Data version: ADVANCED,v1.0,05-28-03

Power summary:                            I(mA)    P(mW)
----------------------------------------------------------------
Total estimated power consumption:                   204
                                  ---
                      Vccint 1.50V:        100       150
                      Vccaux 2.50V:         20        50
                      Vcco25 2.50V:          2         4
                                  ---
                          Clocks:           0         0
                          Inputs:           0         0
                           Logic:           0         0
                         Outputs:
                           Vcco25           0         0
                         Signals:           0         0
                                  ---
           Quiescent Vccint  1.50V:        100       150
           Quiescent Vccaux  2.50V:         20        50
           Quiescent Vcco25  2.50V:          2         4

Thermal summary:
----------------------------------------------------------------
   Estimated junction temperature:                    25C
                    Ambient temp:  25C
                       Case temp:  25C
                       Theta J-A:   0C/W

Analysis completed: Wed Jul 23 09:39:15 2008
```

Figure 5: Obtained power report for the implemented logic

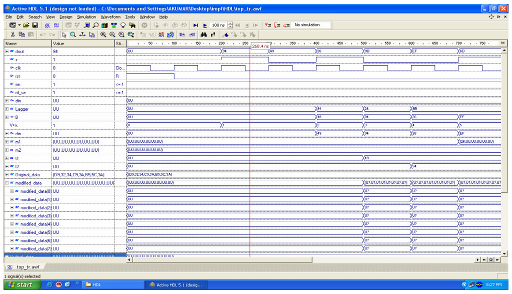


Figure 6: Obtained timing observation of the developed switching logic

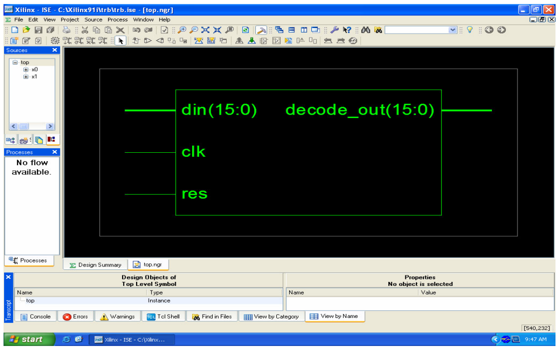


Figure 7(a) RTL View of Implemented Design

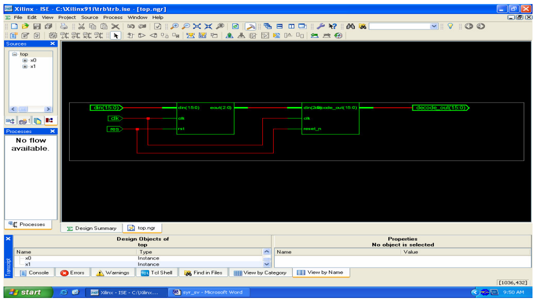


Figure 7(b) RTL View of Implemented Design

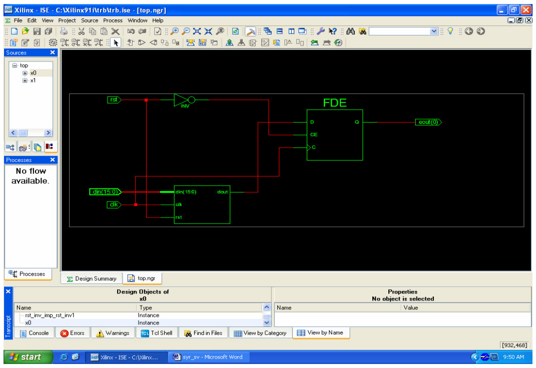


Figure 7(c) RTL View of Implemented Design

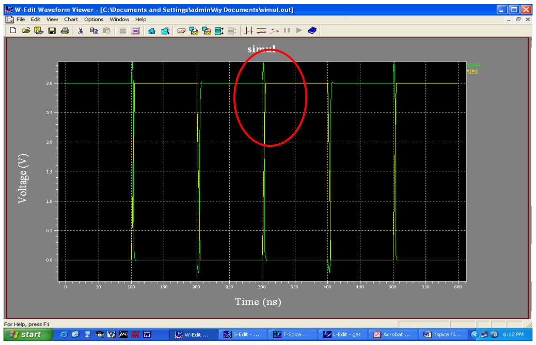
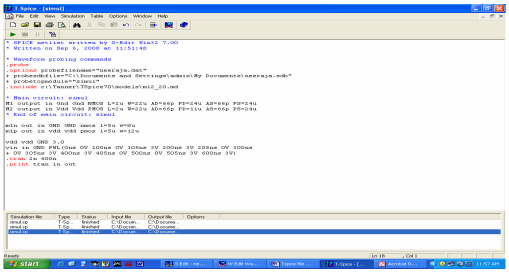


Figure8: Obtained glitch observation for the developed CMOS model



```
* SPICE netlist written by S-Edit Win32 7.00
* Written on Sep 6, 2008 at 11:51:48

* Waveform probing commands
.probe
.options probefilename="neeraja.dat"
+ probesdbfile="C:\Documents and Settings\admin\My Documents\neeraja.sdb"
+ probetopmodule="simul"
.include c:\Tanner\TSpice70\models\ml2_20.md

* Main circuit: simul
M1 output in Gnd Gnd NMOS L=2u W=22u AD=66p PD=24u AS=66p PS=24u
M2 output in Vdd Vdd PMOS L=2u W=22u AD=66p PD=11u AS=66p PS=24u
* End of main circuit: simul

m1n out in GND GND nmos l=5u w=8u
m1p out in vdd vdd pmos l=5u w=12u

vdd vdd GND 3.0
vin in GND PWL(0ns 0V 100ns 0V 105ns 3V 200ns 3V 205ns 0V 300ns
+ 0V 305ns 3V 400ns 3V 405ns 0V 500ns 0V 505ns 3V 600ns 3V)
.tran 2n 600n
.print tran in out
```

Simulation file	Type	Status	Input file	Output file	Options
simul.sp	T-Sp...	finished	C:\Docum...	C:\Docume...	
simul.sp	T-Sp...	finished	C:\Docum...	C:\Docume...	
simul.sp	T-Sp...	finished	C:\Docum...	C:\Docume...	

Ready

Figure 9: Offered resistive load on the drain path for the charge discharge at the junction

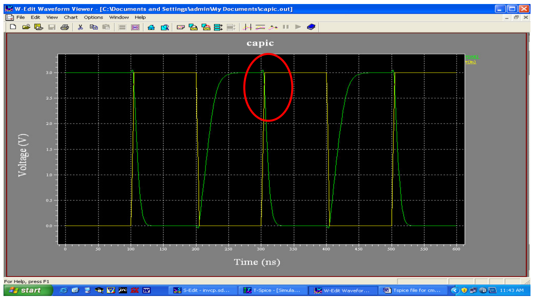


Figure 10: Obtained glitch reduction with offered feedback load

6. CONCLUSION

A Resistive feed back method was thus developed for the elimination of the glitches in CMOS circuitry, which results in power consumption and reducing performance of VLSI design. The glitch power elimination objective is successfully incorporated into the standard cell library based design flow. This was done without re-design of the library. The new design flow is effective in designing minimum transient energy standard cell based digital CMOS circuits. Due to the high level of automation the flow permits, the performance of a standard cell based design is improved without sacrificing the fast design cycle time feature. Results have shown circuit design examples with, on average, 40% reduction in average power for no speed degradation. The increase in area, on average, was 58%. The switch transition minimization approach is observed to be an additional approach for the reduction of power consumption in case of logical transitions. The integration of logical power consumption based on transition minimization and logical feedback resistive path can hence reduce the overall power consumption in case of digital systems for low power VLSI designing.

7. REFERENCES

[1] Hashimoto, M. and Onodera, H. (2001), Post-Layout Transistor Sizing for Power Reduction in Cell-Based Design, in Proc. Asia South Pacific Design Automation Conf. (ASPDAC), pp. 359-365.
[2] Hastings, A. (2001), the Art of Analog Layout, Upper Saddle River, New Jersey: Prentice Hall.
[3]Mahapatra, N.R., Garimella, S.V., and Tareen, A. (2000), An Empirical and Analytical Comparison of Delay Elements and a New Delay Element Design, in IEEE Workshop on VLSI, Apr., pp. 81-86.
[4] Masgonty, J.M., Cserveny, S., Arm, C., and Pfister, P.D. (2001), Low-Power Low-Voltage Standard Cell Libraries with a Limited Number of Cells, in Proc. Int. Workshop Power and Timing Models, Optimization and Simulation, Oct.
[5] Raja, T. (2002), A Reduced Constraint Set Linear Program for Low-Power Design of Digital Circuits, Master's Thesis, Rutgers University, Dept. of ECE, New Brunswick, NJ, Mar.
[6] Raja, T. (2004), Minimum Dynamic Power CMOS Design with Variable Input Delay Logic, PhD Thesis, Rutgers University, Dept. of ECE, New Brunswick, NJ, May.
[7] Raja, T., Agrawal, V.D. and Bushnell, M.L. (2003), Minimum Dynamic Power CMOS Circuit Design by a Reduced Constraint Set Linear Program, in Proc. Int. Conf. VLSI Design, Jan., pp. 527-532.
[8] Raja, T., Agrawal, V.D. and Bushnell, M.L. (2004), CMOS Circuit Design for Minimum Dynamic Power and Highest Speed, in Proc. Int. Conf. VLSI Design, Jan., pp. 1035-1040.
[9] Scott, K. and Keutzer, K. (1994), Improving Cell Libraries for Synthesis, in Proc. Custom Integrated Circuits Conf., May, pp. 128-131.
[10] Sulistyo, J.B. and Ha, D.S. (2002), A New Characterization Method for Delay and Power Dissipation of Standard Library Cells, in Proc. Int. Conf. VLSI Design, Jan., pp. 667-678.
[11] Uppalapati, S. (2004), Low Power Design of Standard Cell Digital VLSI Circuits, Master's Thesis, Rutgers University, Dept. of ECE, New Brunswick, NJ, Oct.
[12] Wroblewski, A., Schimpfle, C.V., and Nassek, J.A. (2000), Automated Transistor Sizing Algorithm for Minimizing Spurious Switching Activities in CMOS Circuits, Proc. Int. Symp. Circuits and Systems.
[13] Zhang, Y., Hu, X., and Chen, D.Z. (2001), Cell Selection from Technology Libraries for Minimizing Power, Proc. of the Design Automation Conf., Jun., pp. 609-614.
[14] R.Wilson, "Low power and paradox," Electron. Eng. Times, p. 38, Nov.1993.
[15] N. Weste and K. Eshraghian, Principles of CMOS VLSI Design, a Systems Perspective. Reading, MA: Addison-Wesley, 1988.