INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 4, Issue 2 JULY

FPGA Power Reduction by Guarded Evaluation Considering Logic Architecture

CHALAM TIRUNAGARI, M.TECH, **VLSI SD**
KONDRA GOPI, ASST.PROF, E.C.E
KSHATRIYA COLLEGE OF ENGINEERING

Abstract:

In this paper, guarded evaluation is a dynamic power reduction technique by identifying sub circuits inputs and kept constant at specific times during circuit operation. In certain condition, some signals within the digital design are not observable at output. So make such signals as guarded (constant). There by reducing the dynamic power. Here we apply this technique for all digital circuits. The problem here is to find conditions under which a sub circuit input can be held constant with disturbing the main circuit functionally (correctness). Here we propose a solution for discovering the gating inputs based on inverting and non-inverting methods. By including "clock gating" we still reduce the dynamic power and leakage power especially for sequential circuits and also used to some small combinational circuits.

Keywords—AGFF, LACG, Flip Flops, Clock Gating.

I. INTRODUCTION

Flip-Flops (FFs) are the basic storage elements used extensively in all kinds of digital designs. The digital designs nowadays often adopt intensive pipelining techniques and employ many FF rich modules and also estimated that the power consumption of clock system. In electronics, a flip-flop or latch is a circuitthat has two stable states and can be used to store state information. A flip-flop is a bitable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are a fundamental building block of digital electronicssystems used in computers, communications, and many other types of systems. Flip-flops and latches are used as data storage elements. Such data storage can be used for storage of state, and such a circuit is described as sequential logic. When used in a finite state machine, the utput and next state depend not only on its current input, but also on its current state (and hence, previous inputs). It can also be used for counting of pulses, and for synchronizing variably timed input signals to some reference timing signal.Flip-flops can be either simple (transparent or opaque) or clocked (synchronous or edge-triggered),the simple ones are commonly called latches. The word latch is mainly used for storage elements, while clocked devices are described as flip-flops. A latch is level-sensitive, whereas a flip-flop is edge-sensitive. That is, when a latch is enabled it becomes transparent, while a flip flop's output only changes on a single type (positive going or negative going) of clock edge.

Logic synthesis is the process of converting a high level description of design into an optimized gate level representation[1]. Logic synthesis uses a standard cell library which have simple cells, such as basic logic gates like and, or, and nor, or macro cells, Logic synthesis is the process of converting a high level description of design into an optimized gate level representation[1]. Logic synthesis uses a standard cell library which have simple cells, such as basic logic gates like and, or, and nor, or macro cells,

II. BACKGROUND

A. GUARDED EVALUATION

We first described important techniques for guarded evaluation in ASICs. The key idea is shown in Fig. 1. In Fig. 1(a), a multiplexer is shown receiving its inputs from a shifter and a subtraction unit, depending on the value of select signal Sel[2]. Fig. 1(b) shows the circuit after guarded

evaluation. Guard logic, comprised of transparent latches, is inserted before the functional units. The latches are transparent only when the output of the corresponding functional unit is selected by the multiplexer, i.e., depending on signal Sel. When the output of a functional unit is not needed, the latches hold its input constant, eliminating toggles within the unit. Here, one can view Sel as the "guarding signal."We applied this concept to gate-level networks, where the difficulty was in determining which signals could be used as guarding signals for particular sub circuits.

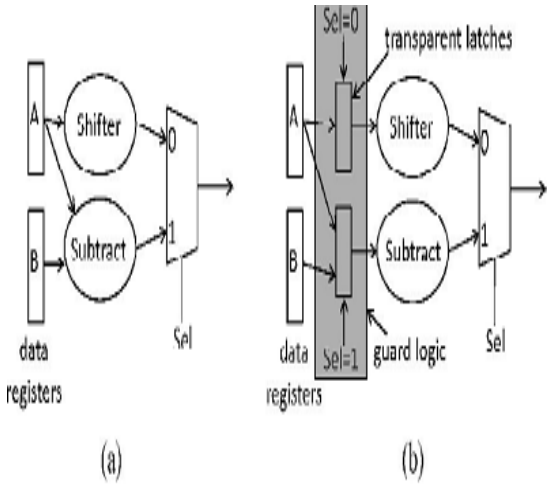

Fig1(a). Before guarded evaluation. Fig 1(b). After guarded evaluation.

We used binary decision diagrams to discover logical implications that permit certain subcircuits to be disabled at certain times. We proposed using guarded evaluation in ASICs to attack both leakage and dynamic power [2]. The guarding signals were used to drive the gate terminals of NMOS sleep transistors incorporated into CMOS gate pull-down networks, putting sub circuits into low-leakage states when their outputs were not needed. Their approach produced encouraging power reduction results by exploiting select signals on steering elements (multiplexers) to serve as guarding signals and is therefore limited to specific types of circuits.

A. Gating Inverting And Non Inverting

Fig. 2(a) gives an example of a LUT and the corresponding portion of a covered AIG.

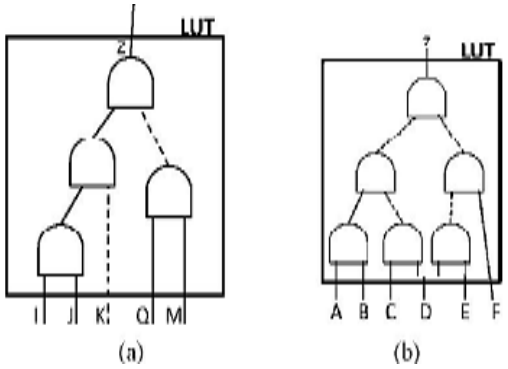

Fig 2 (a). Identifying gating inputs on LUTs using non inverting paths.(b). Identifying trimming inputs on LUTs using partial non inverting paths.

The logic function implemented by the LUT is Z = I · J ·K ·Q ·M. Examine the AIG path from the input I to the root gate of the AIG, Z. The path comprises a sequence of AND gates with none of the path edges being complemented[3]. Recall that the output of an AND gate is logic- 0 when either of its inputs is logic-0. For the path from I to Z, when I is logic-0, the output of each AND gate along the path will be logic-0, ultimately producing logic-0 on the LUT output. We therefore conclude that I is a gating input to the LUT. The LUT in Fig. 2(a), in fact, has three gating inputs, I, J, and K. Input J is the same form as input I in that there exists a path of AND gates from J to root gate Z and none of the edges along the path
are inverted. Observe, however, that the situation is slightly different for input K. For input K, the "frontier" edge crossing into the LUT is inverted; however, aside from this frontier edge, the remaining edges along the path from K to the root node Z are "true" edges. This means that when K is logic-1, the output of the AND gate it drives will be logic-0, eventually making the LUT's output signal Z logic-0. K is indeed a gating input, though it is K's logic-1 state1 (rather than)its logic-0 state) that causes the LUT output to be logic-0. Non inverting paths are therefore chains of AND gates without edge inversions. Gating inputs to LUTs can be easily discovered through a traversal of the

underlying AIG.In [3], the notions of gating inputs and non inverting paths were applied to map circuits into a new logic block architecture that delivers improved areaefficiency. Here, we apply the ideas for power reduction through guarded evaluation.

III. CLOCK GATING

In today's semiconductor designs, lower power consumption is mandatory for mobile and handheld applications for longer battery life and even networking or storage devices for low carbon footprint requirements. Clock power consumes 60-70 percent of total chip power and is expected to significantly increase in the next generation of designs at 45nm and below. This is due to the fact that power is directly proportional to voltage and the frequency of the clock as shown in the following equation.
Power=Capacitance * (Voltage)2* (Frequency)

Clock gating is a popular technique used in many synchronous circuits for reducing dynamic power dissipation[4]. Clock gating saves power by adding more logic to a circuit to prune the clock tree. Pruning the clock disables portions of the circuitry so that the flip-flops in them do not have to switch states. Switching states consumes power. When not being switched, the switching power consumption goes to zero, and only leakage currents are incurred load to node X causes speed and power performance degradation[7]. Clock gating works by taking the enable conditions attached to registers, and uses them to gate the clocks. Therefore it is imperative that a design must contain these enable conditions in order to use and benefit from clock gating[5]. This clock gating process can also save significant die area as well as power, since it removes large numbers of muxes and replaces them with clock gating logic.

A. Clock Gating Using Asic Design

Fig. 3 Any RTL modifications to improve clock gating will result in functional changes to the design (since the registers will now hold different values) which need to be verified. Sequential clock gating is the process of extracting/propagating the enable conditions to the upstream/downstream sequential elements, so that additional registers can be clock gated. Although asynchronous circuits by definition do not have a "clock", the term perfect clock gating is used to illustrate how various clock gating techniques are simply approximations of the data-dependent behavior exhibited by asynchronous circuitry[8]. As the granularity on which you gate the clock of a synchronous circuit approaches zero, the power consumption of that circuit approaches that of an asynchronous circuit: the circuit only generates logic transitions when it is actively computing.

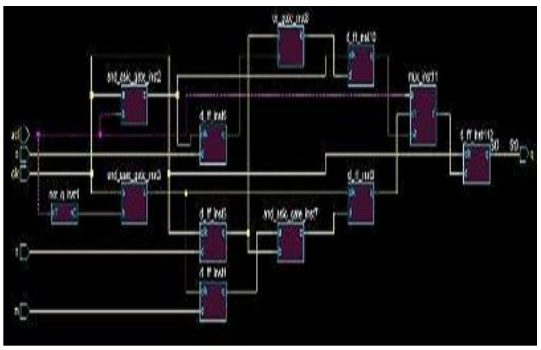

Fig 3. Clock gating in Asic Design

IV.SIMULATION RESULTS

The simulation results are same the guarded asic designs ans clock gating asic designs. Since delay is small difference and the dynamic power reduction and the leakage power is reduced. The simulation results are schematic circuits designed in Questasim (10.2a) Tool and the power calculations are carried out the CADENCE RTL COMPLIER Tool is used as shown in Fig 6(a).Before Asic Design Schematic in Questasim Tool, Fig 6(b).Before Asic Design Wave Form in Questasim Tool, Fig 6(c). Before Asic Design Schematic in Precision Tool, Fig 6(d). Before Asic Design RTL Power in, Fig 6(e). Before Asic Design Delay in Cadence Tool, Fig 7(a). Clock gating Asic Design Schematic in Questasim Tool, Fig 7(b).Clock gating Asic Design Wave Form in Questasim Tool, Fig 7(c). Clock gating Asic Design Schematic in Precision Tool, Fig 7(d). Clock gating Asic Design RTL Power in

Cadence Tool, Fig 7(e). Clock gating Asic Design Delay in Cadence Tool.

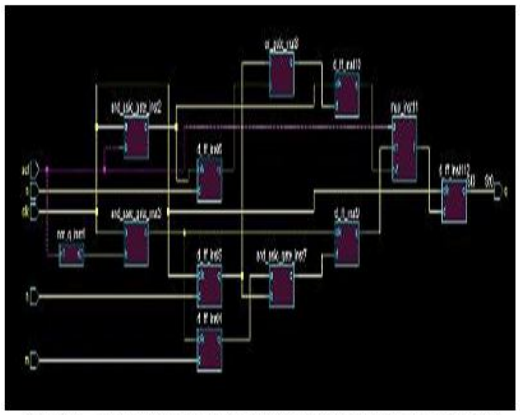

Fig 7(a). Clock gating Asic Design Schematic in Questasim Tool

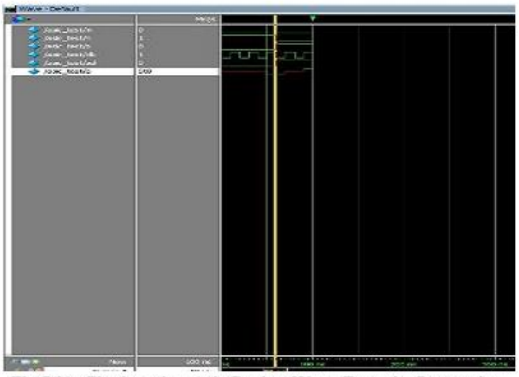

Fig 7(b). Clock gating Asic Design Wave Form in Questasim Tool

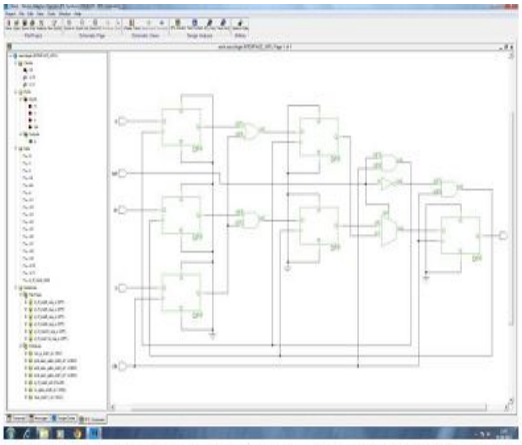

Fig 7(c). Clock gating Asic Design Schematic in Precision Tool

V. RESULT COMPARISON

The comparison of result summarizes some important performance indexes of these table1 is dynamic power and leakage power. Table2 is no of cell and delay.

Table1. Power Comparison of Guarded circuit and Clock gating circuit

CIRCUIT	LEAKAGE POWER(nW)	DYNAMIC POWER (nW)	TOTAL POWER (nW)
	GUARDED		
Asic Design	695.712	3337.223	4032.9
Lut	46.154	568.517	614.51
Shift & subtract Design	231.186	2040.321	2271.508
Alu	101.722	1196.208	1297.9
Full Adder	76.220	1350.404	1426.6
	Clock Gating		
Asic Design	276.725	2451.199	2727.9
Lut	6.033	234.240	240.27
Shift&subtract Design	207.160	1795.065	2002.2
Alu	36.267	632.743	669.01
Full Adder	94.963	723.955	818.62

Table2. Cells and Delay Comparison of Guarded circuit and Clock gating circuit

CIRCUIT	No.of.Cells	Delay (ps)
	GUARDED	
Asic Design	9	701
Lut	4	287
Shift & subtract Design	7	651
Alu	6	-
Full Adder	5	664

	Clock Gating	
Asic Design	12	654
Lut	2	168
Shift&subtract Design	9	712
Alu	5	-
Full Adder	1	271

VI. CONCLUSION

In this paper, the various digital designs and especially sequential circuits are used main aim is reducing the dynamic power and leakage power. In paper is implementing in FPGA's only the different FPGA powers are required only use cad tools can required RTL power.

REFERENCES

[1] J. Anderson and C. Ravishankar, "FPGA power reduction by guardedevaluation," in ACM/SIGDA FPGA, 2010, pp. 157–166.
[2] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs,"IEEE Trans. On CAD, vol. 26, no. 2, pp. 203–215, Feb. 2007.
[3] J. Anderson and F. Najm, "Power-aware technology mapping for LUTbased FPGAs," in IEEE Int'l Conference on FPT, 2002, pp. 211–218.
[4] S. Jang, B. Chan, K. Chung, and A. Mishchenko, "Wiremap: FPGAtechnology mapping for improved routability and enhanced LUT merging," ACM Trans. on Reconfigurable Technology and Systems, vol. 2,no. 2, pp. 1–24, 2009.
[5] J. Lamoureux and S. Wilton, "On the interaction between power-awareFPGA CAD algorithms," in IEEE/ACM ICCAD, 2003, pp. 701–708.
[6] K. Vorwerk, M. Raman, J. Dunoyer, Y.-C. Hsu, A. Kundu, and A. Kennings, "A technique for minimizing power during FPGA placement," inIEEE Int'l Conf. on FPL, 2008, pp. 233–238.
[7] J. Lamoureux, G. Lemieux, and S. Wilton, "GlitchLess: an active glitchminimization technique in FPGAs," in ACM/SIGDA FPGA, 2007, pp.156–165.
[8] "Berkeley logic synthesis and verification group, ABC– a system for sequential synthesis and verification, "http://www.eecs.berkeley.edu/~alanmi/abc/, 2009.
[9] K. Poon, A. Yan, and S. Wilton, "A flexible power model for FPGAs,"in Int'l Conf. on FPL, 2002, pp. 312–321.
[10] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deepsubmicron FPGA performance and density," in ACM/SIGDA FPGA, 2002, pp. 85–94.