International Journal of Electronics and Electrical Engineering

Volume 3 | Issue 4 Article 2

April 2015

POWER REDUCTION BY GUARDED EVALUATION CONSIDERING LOGIC ARCHITECTURE AND USING CLOCK GATING

G. SURESH
VLSI Design, Sir C.R.REDDY College of Engineering, Eluru, suresh@gmail.com

A.RAM KUMAR
Dept. of E.C.E, Sir C.R.REDDY College of Engineering, Eluru, aramkumar@gmail.com

Recommended Citation

SURESH, G. and KUMAR, A.RAM (2015) "POWER REDUCTION BY GUARDED EVALUATION CONSIDERING LOGIC ARCHITECTURE AND USING CLOCK GATING," *International Journal of Electronics and Electrical Engineering*: Vol. 3 : Iss. 4 , Article 2.
DOI: 10.47893/IJEEE.2015.1161
Available at: https://www.interscience.in/ijeee/vol3/iss4/2

POWER REDUCTION BY GUARDED EVALUATION CONSIDERING LOGIC ARCHITECTURE AND USING CLOCK GATING

G. SURESH[1], A.RAM KUMAR[2]

[1]M. Tech VLSI Design, [2]M.Tech, Assistant Professor, Dept. of E.C.E, Sir C.R.REDDY College of Engineering, Eluru

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- *Flip-flops, Logic Synthesis, Low power design, Cad tools.*

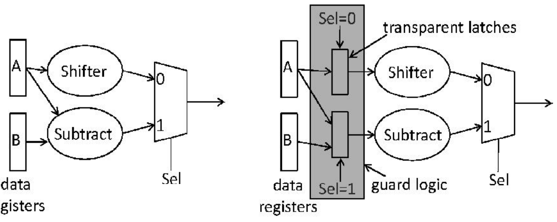
I. INTRODUCTION

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 circuit that 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 electronics systems 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 output 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, such as adder, muxes, memory, and flip-flops. Standard cells put together are called technology library. Normally the technology library is known by the transistor size. A circuit description is written in Hardware Description Language (HDL) such as Verilog. The designer should first understand the architectural description. Then he should consider design constraints such as timing, area, testability, and power. High level design is less prone to human error because designs are described at a higher level of abstraction. High-level design is done without significant concern about design constraints. Conversion from high-level design to gates is done by synthesis tools.

II. BACKGROUND

A. GUARDED EVALUATION



(a) (b)

Fig1(a). Before guarded evaluation. Fig 1(b). After 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. We used binary decision diagrams to discover logical implications that permit certain sub circuits 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.

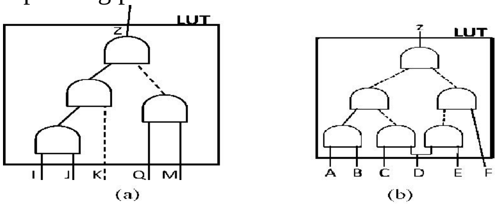


Fig 2 (a). Identifying gating inputs on LUTs using non inverting paths.
(b). Identifying trimming inputs on LUTs using partial non inverting paths.

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 area-efficiency. 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.

$$\text{Power}=\text{Capacitance} * (\text{Voltage})^{2} * (\text{Frequency})$$

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. HOW TO IMPLEMENT CLOCK GATING

When there is no activity at a register “data” input, there is no need to clock the register and hence the “clock” can be gated to switch it off. If the clock feeds a bank of registers, an “enable” signal can be used to gate the clock, which is called the “clock gating enable”.

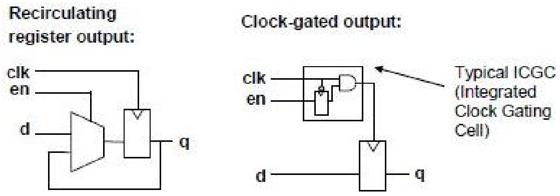

Recirculating register output:
clk
en
d
q
Clock-gated output:
clk
en
d
q
Typical ICGC (Integrated Clock Gating Cell)

Fig 3.(a). Feedback mux Fig 3.(b). Integrated clock gating This clock gating logic is generally in the form of "Integrated clock gating" (ICG) cells. However, note that the clock gating logic will change the clock tree structure, since the clock gating logic will sit in the clock tree[6]. As shown in Figure 3.(a), when an “explicit” clock enable exists in the RTL code, synthesis tools may choose between two possible implementations. The implementation as shown in Figure 1a, is a “re-circulating register” implementation, where the enable is used to either select a new data value or re-circulate the previous data value.

The implementation as shown in Figure 3.(b) is a “gated clock” implementation[8]. When the enable is off, the clock is disabled. The output of the two implementations will always be identical, but the timing and power behavior will be different.

A. CLOCK GATING USING ASIC DESIGN

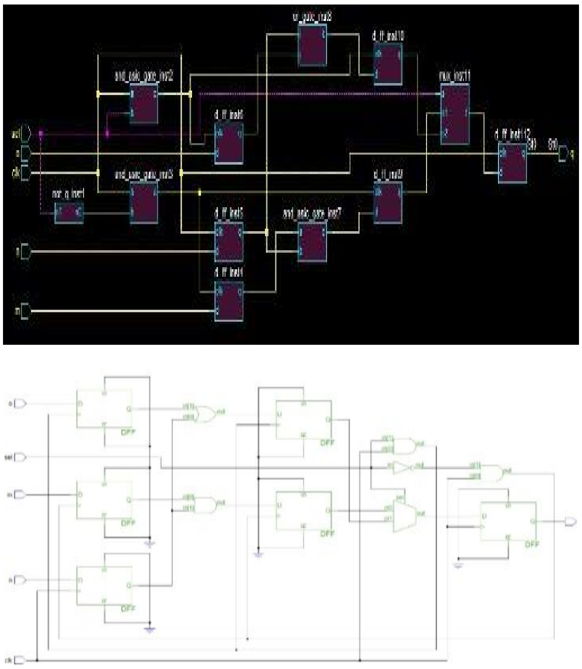


Fig 4. Clock gating in Asic Design

Fig. 4 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.

IV. RTL POWER ESTIMATION FLOW

Early power estimation at RTL can help the designer to quickly explore different architectures like replacing large memories with smaller memories or register files and find power bugs early in the design before it is found too late at the gate-level where synthesis and place and route steps will have to be iterated to meet the required power budget for the design.

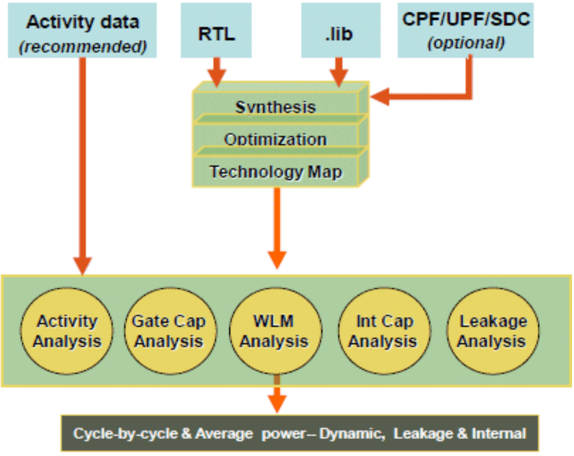


Fig 5. RTL Power Estimation Flow

Fig. 5 illustrates the details of the required and optional inputs to the RTL power estimation flow with Atrenta’s SpyGlass-Power solution: Synthesizable RTL Design – In order to understand the gate count and power characteristics of a design, it must be synthesizable[8]. Portions of the design which are not synthesizable or not finished yet can be represented as black boxes and power data can be provided for these as part of the power library data. Power Library Data for the Process – The liberty format has a representation for power data which most library providers use[9]. v Activity Data – In order to estimate power accurately, waveforms for an RTL simulation of the design should be provided in VCD, FSDB, or SAIF format. Power Intent – UPF/CPF can be used to define the power intent for estimating the power at RTL[10]. Timing Constraints (optional) – There are several Synopsys Design Constraints (SDC) timing constraints which may be useful for power estimation, such as set_case_analysis or set_output_load. An SDC file may be optionally supplied.

V. 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.

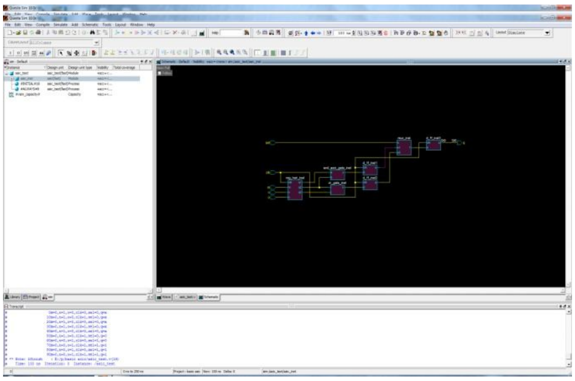


Fig 6(a). Before Asic Design Schematic in Questasim Tool

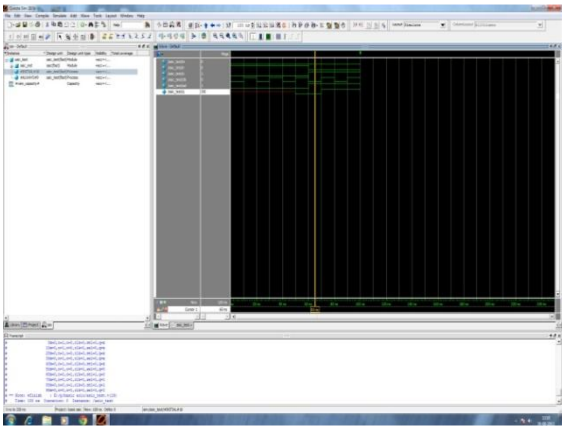


Fig 6(b). Before Asic Design Wave Form in Questasim Tool

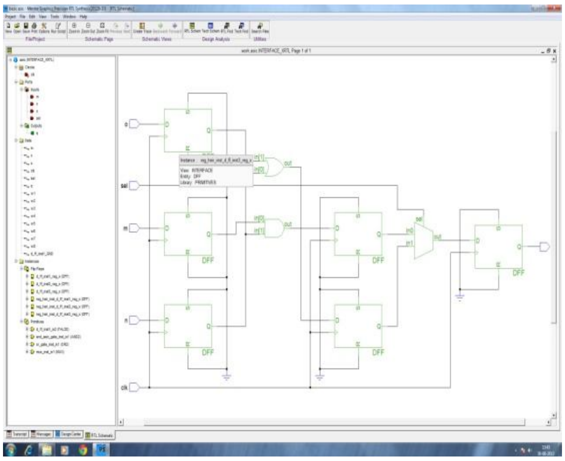
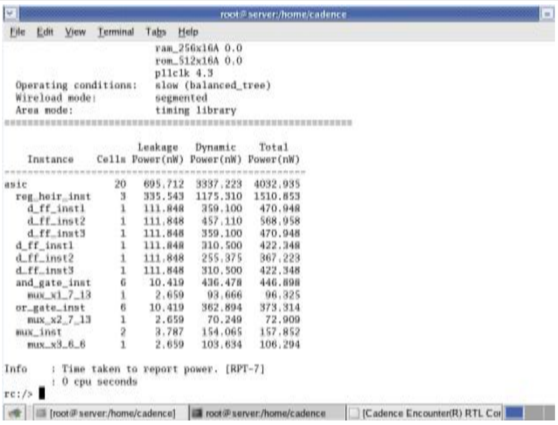


Fig 6(c). Before Asic Design Schematic in Precision Tool



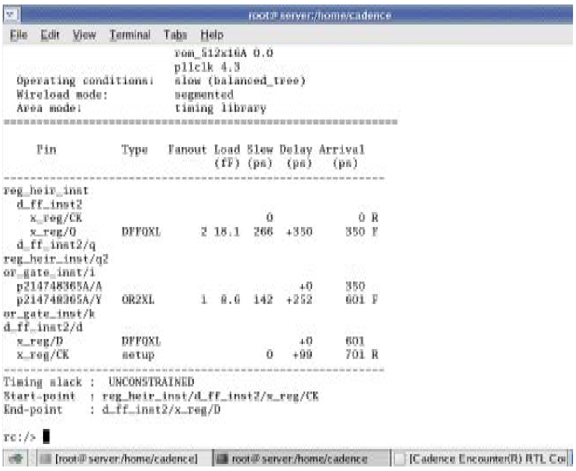
```
                         ram_256x16A 0.0
                         rom_512x16A 0.0
                         pllclk 4.3
  Operating conditions:  slow (balanced_tree)
  Wireload mode:         segmented
  Area mode:             timing library
================================================================

                     Leakage   Dynamic    Total
   Instance   Cells Power(nW) Power(nW) Power(nW)
-------------------------------------------------------
asic             20   695.712  3337.223  4032.935
  reg_heir_inst   3   335.543  1175.310  1510.853
    d_ff_inst1    1   111.848   359.100   470.948
    d_ff_inst2    1   111.848   457.110   568.958
    d_ff_inst3    1   111.848   359.100   470.948
  d_ff_inst1      1   111.848   310.500   422.348
  d_ff_inst2      1   111.848   255.375   367.223
  d_ff_inst3      1   111.848   310.500   422.348
  and_gate_inst   6    10.419   436.478   446.898
    mux_x1_7_13   1     2.659    93.666    96.325
  or_gate_inst    6    10.419   362.894   373.314
    mux_x2_7_13   1     2.659    70.249    72.909
  mux_inst        2     3.787   154.065   157.852
    mux_x3_6_6    1     2.659   103.634   106.294

Info    : Time taken to report power. [RPT-7]
        : 0 cpu seconds
rc:/> 
```

[root@server:/home/cadence]
root@server:/home/cadence
[Cadence Encounter(R) RTL Coi

Fig 6(d). Before Asic Design RTL Power in Cadence Tool



```
                          rom_512x16A 0.0
                          pllclk 4.3
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          segmented
  Area mode:              timing library
==============================================================

    Pin          Type    Fanout Load Slew Delay Arrival
                                (fF) (ps)  (ps)   (ps)
------------------------------------------------------------
reg_heir_inst
  d_ff_inst2
    x_reg/CK                              0          0 R
    x_reg/Q      DFFQXL       2 18.1  266  +350    350 F
  d_ff_inst2/q
reg_heir_inst/q2
or_gate_inst/i
  p2147483651A/A                           +0     350
  p2147483651A/Y OR2XL        1  8.6  142  +252    601 F
or_gate_inst/k
d_ff_inst2/d
  x_reg/D        DFFQXL                    +0     601
  x_reg/CK       setup                0    +99     701 R
------------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : reg_heir_inst/d_ff_inst2/x_reg/CK
End-point    : d_ff_inst2/x_reg/D

rc:/> 
```

[root@server:/home/cadence]
root@server:/home/cadence
[Cadence Encounter(R) RTL Coi

Fig 6(e). Before Asic Design Delay in Cadence Tool

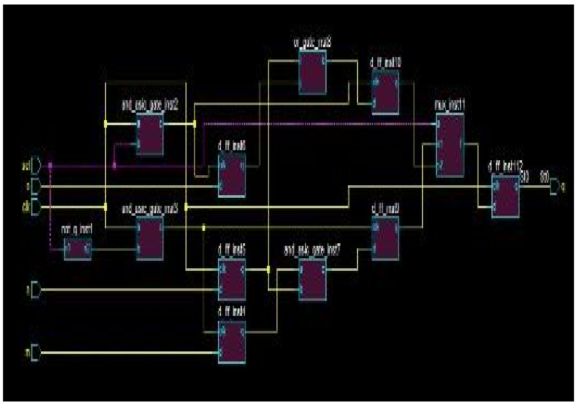


Fig 7(a). Clock gating Asic Design Schematic in Questasim Tool

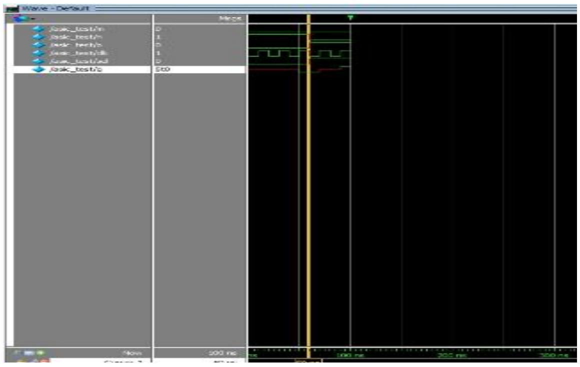


Fig 7(b). Clock gating Asic Design Wave Form in Questasim Tool

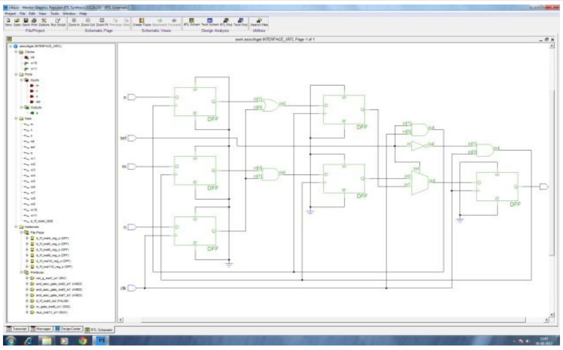
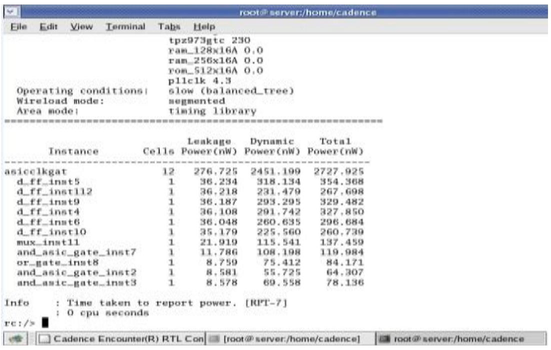


Fig 7(c). Clock gating Asic Design Schematic in Precision Tool

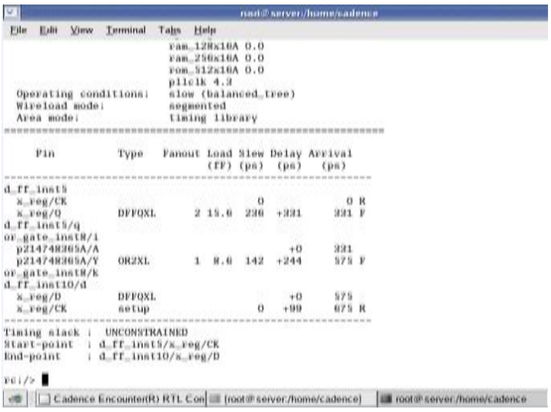


```
                         tpz973gtc 230
                         ram_128x16A 0.0
                         ram_256x16A 0.0
                         rom_512x16A 0.0
                         pllclk 4.3
  Operating conditions:  slow (balanced_tree)
  Wireload mode:         segmented
  Area mode:             timing library
============================================================

                              Leakage   Dynamic    Total
       Instance        Cells Power(nW) Power(nW) Power(nW)
------------------------------------------------------------
asicclkgat               12   276.725  2451.199  2727.925
  d_ff_inst5              1    36.234   318.134   354.368
  d_ff_inst112            1    36.218   231.479   267.698
  d_ff_inst9              1    36.187   293.295   329.482
  d_ff_inst4              1    36.108   291.742   327.850
  d_ff_inst6              1    36.048   260.635   296.684
  d_ff_inst10             1    35.179   225.560   260.739
  mux_inst11              1    21.919   115.541   137.459
  and_asic_gate_inst7     1    11.786   108.198   119.984
  or_gate_inst8           1     8.759    75.412    84.171
  and_asic_gate_inst2     1     8.581    55.725    64.307
  and_asic_gate_inst3     1     8.578    69.558    78.136

Info    : Time taken to report power. [RPT-7]
        : 0 cpu seconds
rc:/>
```

Fig 7(d). Clock gating Asic Design RTL Power in Cadence Tool



```
                         ram_128x16A 0.0
                         ram_256x16A 0.0
                         rom_512x16A 0.0
                         pllclk 4.3
  Operating conditions:  slow (balanced_tree)
  Wireload mode:         segmented
  Area mode:             timing library
============================================================

     Pin          Type    Fanout Load Slew Delay Arrival
                                 (fF) (ps)  (ps)   (ps)
---------------------------------------------------------
d_ff_inst5
  x_reg/CK                              0          0 R
  x_reg/Q         DFFQXL       2 15.6  236  +331   331 F
d_ff_inst5/q
or_gate_inst8/i
  p214748365A/A                              +0    331
  p214748365A/Y   OR2XL        1  8.6  142  +244   575 F
or_gate_inst8/k
d_ff_inst10/d
  x_reg/D         DFFQXL                     +0    575
  x_reg/CK        setup                 0   +99    675 R
---------------------------------------------------------
Timing slack :  UNCONSTRAINED
Start-point  : d_ff_inst5/x_reg/CK
End-point    : d_ff_inst10/x_reg/D

rc:/>
```

Fig 7(e). Clock gating Asic Design Delay in Cadence 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] S. Jang, K. Chung, A. Mishchenko, and R. Brayton, "A power optimization toolbox for logic synthesis and mapping," in Proc. IEEE Int. Workshop Logic

[2] V. Tiwari, S. Malik, and P. Ashar, "Guarded evaluation: Pushing power management to logic synthesis/design," IEEE Trans. Comput.-Aided Des., vol. 17, no. 10, pp. 1051–1060, Oct. 1998.

[3] J. Anderson and Q.Wang, "Improving logic density through synthesis-inspired architecture," in Proc. IEEE Int. Conf. Field-Programmable Logic Applicat., Aug.–Sep. 2009, pp. 105–111.

[4] A. Mishchenko. (2009). ABC: A System for Sequential Synthesis and Verification [Online].

[5] Clock-Gating and Its Application to Low Power Design of Sequential Circuits Qing WU Department of Electrical Engineering-Systems, University of Southern California Los Angeles, CA 90089, USA. Available: http://www.eecs.berkeley.edu/~alanmi/abc

[6] Altera Corporation. (2009). Quartus-II University Interface Program [Online].Available:http://www.altera.com/education/univ/research unv-quip.html

[7] Automatic synthesis of clock gating logic with controlled netlist perturbation Hurst, A.P. Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE Publication Year: 2008.

[8] Symbolic synthesis of clock-gating logic for power optimization of control-oriented synchronous networks Benini, L. ; De Micheli, G. ; Macii, E. ; Poncino, M. ; Scarsi, R. European Design and Test Conference, 1997. ED&Tc97.Proceedings Publication Year: 1997.

[9] D. Howland and R. Tessier, "RTL dynamic power optimization for FPGAs," in Proc. IEEE Midwest Symp. Circuits Syst., Aug. 2008, pp. 714–717. Synthesis, Aug. 2009, pp. 1–8.

[10] L. Shang, A. Kaviani, and K. Bathala, "Dynamic power consumption of the Virtex-II FPGA family," in Proc. ACM Int. Symp. Field Programmable Gate Arrays, 2002, pp. 157–164.

A.RAMKUMAR M.Tech (VLSISD) M. Tech. degree in swarnandhra college of engineering and technology narasapur, Andhra Pradesh, India . Presently he is working as assistant professor in Department of Computer Science & Engineering, in Sir C R Reddy College of Engineering Eluru, Andhra Pradesh, India.

G.SURESH received B.Tech Degree in Electronics and Communication Engineering from Rao and Naidu in Engineering College 2010.Currently pursuing M.Tech in Sir C R Reddy College of Engineering Eluru. His areas of interest are Low Power VLSI Design.

G.SANATH KUMAR Mtech (MICRO ELECTRONICS)Presently he is working as VLSI Engineer Consultant at Central Institute of Tool Design, Hyderabad Andhra Pradesh, India

.