A Low Power Shift Register Design with Clock Gating

Abhijeet R. Pathak
Department of Electronics and Telecommunications
University of Mumbai, India
Abhip.9294@gmail.com

Abstract:
In this paper clock gating technique is presented for low power VLSI (very large scale integration) circuit design. Clock in digital circuits is used for synchronization of various components. Clock power is a major source of dynamic power consumed in synchronous circuits. Clock-gating is a well-known technique to reduce clock power. In clock gating clock to an idle block is disabled. Thus significant amount of power consumption is reduced by employing clock gating. In this paper a 8-bits serial shift register is designed using clock gating. Power consumption simulation is performed on Hspice. Experimental result shows that the clock gating technique significantly improves total dynamic power consumption.

Keywords: Clock gating, low power design, shift register.

I. INTRODUCTION
Reducing power consumption in very large scale integrated circuits (VLSI) design has become an interesting research area. Most of the portable devices available in the market are battery driven. These devices impose tight constraint on the power dissipation. Reducing power consumption in such devices improves battery life significantly. Due to lesser advancement in battery technology, low power design has become more challenging research area. Power consumed in a digital circuit is of two types. (1) Static power and (2) Dynamic power. Static power consists of power dissipated due to leakage currents whereas dynamic power consists of capacitive switching power and short circuit power. In VLSI circuit clock signal is used for the synchronization of active components. Clock power is a major component of power mainly because the clock is fed to most of the circuit blocks, and the clock switches every cycle. Thus the total clock power is a substantial component of total power dissipation in a digital circuit. Clock-gating is a well known technique to reduce clock power. In a sequential circuit individual blocks usage depends on application, not all the blocks are used simultaneously, giving rise to dynamic power reduction opportunity. By clock gating technique, clock to an idle portion is disabled, thus avoiding power dissipation due to unnecessary charging and discharging of the unused circuit. In clock gating clock is selectively stopped for a portion of circuit which is not performing any active computation. Local clocks that are conditionally enabled are called gated clocks, because a signal from the environment is used to gate the global clock signal. In this paper a CMOS clock gated shift register has been proposed. Section II explains various types of power dissipated in a synchronous circuit. In section III clock gating technique and structure knowledge is been discussed. Section IV mainly talks about the shift register information and structure. In section V, it shows the result of shift register power consumption of gated and non-gated clock circuit. Using Hspice to get results in this section.

II. POWER CONSUMPTION IN SEQUENTIAL CIRCUITS
Average power dissipated in a digital circuit is given as.

\[ P_{\text{average}} = P_{\text{dynamic}} + P_{\text{short-circuit}} + P_{\text{leakage}} + P_{\text{static}} \] (1)

P average is the average power dissipation, P dynamic is the dynamic power dissipation due to switching of transistors, P short-circuit is the short circuit current power dissipation when there is a direct current path from power supply down to ground, P leakage is the power dissipation due to leakage currents, P static and is the static power dissipation

Fig. 1 Sources of power consumption in digital circuits

a) Static Power
Static power is the power dissipated by a gate when it is inactive or idle. Ideally, CMOS (Complementary Metal Oxide Semiconductor) circuits dissipate no static (DC) power since in the steady state there is no direct path from Vdd to ground.

b) Dynamic Power
Dynamic power is the power dissipated during active state due to switching activity of input signal. In other words, dynamic power dissipation is caused by the charging. Since

Research Article
Volume 6 Issue No. 6
DOI 10.4010/2016.1742
ISSN 2321 3361 © 2016 IJESC

an input can change without necessarily resulting in logic transition in the output, dynamic power can be dissipated even when an output doesn’t change its logic state. This component of dynamic power dissipation is the result of charging and discharging parasitic capacitances in the circuit. Dynamic power dissipation in a circuit is given as:

\[ P_D = \alpha CL V_{DD}^2 f \]

Where \( \alpha \) is the switching activity, \( f \) is the operation frequency, \( CL \) is the load capacitance, \( VDD \) is the supply voltage.

c) Short-Circuit Power
The short-circuit power consumption, \( P_{short-circuit} \), is caused by the current flow through the direct path existing between the power supply and the ground during the transition phase.

d) Leakage Power
The PMOS and NMOS transistors used in a CMOS logic circuit commonly have non-zero reverse leakage and subthreshold currents. These currents can contribute to the total power dissipation even when the transistors are not performing any switching action. The leakage power dissipation, \( P \) leakage is caused by two types of leakage currents. a) Reverse-bias diode leakage current. b) Sub threshold current through a turned-off transistor channel.

III. CLOCK GATING TECHNIQUE

Clock power is a major component of power mainly because the clock is fed to most of the circuit blocks, and the clock switches every cycle. Thus the total clock power is a substantial component of total power dissipation in a digital circuit. Clock-gating is a well-known technique to reduce clock power. By clock gating technique, clock to an idle portion is disabled, thus avoiding power dissipation due to unnecessary charging and discharging of the unused circuit. In clock gating clock is selectively stopped for a portion of circuit which is not performing any active computation. This is done by using a signal from the environment. An example of gated clock is shown in figure.

IV. Shift register
In digital circuits, a shift register is a cascade of flip flops, sharing the same clock, in which the output of each flip-flop is connected to the "data" input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the "bit array" stored in it, shifting in the data present at its input and shifting out the last bit in the array, at each transition of the clock input.

More generally, a shift register may be multidimensional, such that its "data in" and stage outputs are themselves bit arrays: this is implemented simply by running several shift registers of the same bit-length in parallel. Shift registers can have both parallel and serial inputs and outputs. These are often configured as 'serial-in, parallel-out' (SIPO) or as 'parallel-in, serial-out' (PISO). There are also types that have both serial and parallel input and types with serial and parallel output. There are also 'bidirectional' shift registers which allow shifting in both directions: L→R or R→L. The serial input and last output of a shift register can also be connected to create a 'circular shift register'. In my design, I use 8-bit shift-left register with positive-edge clock, serial In, and serial Out. First I write Vhdl code for 8-bit Shift-Left Register and then synthesize by Leonardo spectrum. The image below is used Design Architect IC software to create.

![Fig.2 Gated clock](image1)

![Fig.3 8-bit SISO Shift-Left Register](image2)

![Fig.4 8-bit SISO shift register with clock gating Clock gating and D flip-flops are in the blocks.](image3)
V. Simulation results by Hspice
Technology: 45nm
Clock frequency: 50MHz

<table>
<thead>
<tr>
<th>Voltage (v)</th>
<th>Power without clock gating (uw)</th>
<th>Power with clock gating (uw)</th>
<th>Power Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>13</td>
<td>1.12</td>
<td>91</td>
</tr>
<tr>
<td>1</td>
<td>129.35</td>
<td>74.7</td>
<td>42</td>
</tr>
<tr>
<td>1.3</td>
<td>622</td>
<td>257</td>
<td>58</td>
</tr>
<tr>
<td>1.5</td>
<td>1882.4</td>
<td>355</td>
<td>81</td>
</tr>
</tbody>
</table>

VI. CONCLUSION
In conclusion clock gating technique significantly reduces dynamic power of shift register.

REFERENCES


**VHDL codes:**

8-bits serial shift register:
library ieee;
use ieee.std_logic_1164.all;

entity shift is
port(C, SI : in std_logic;
SO : out std_logic);
end shift;

architecture archi of shift is
signal tmp: std_logic_vector(7 downto 0);
begin
process (C)
begin
if (C'event and C='1') then
for i in 0 to 6 loop
  tmp(i+1) <= tmp(i);
end loop;
  tmp(0) <= SI;
end if;
end process;
SO <= tmp(7);
end archi;

8-bits shift register with clock gating:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity gatedclk is
port( clk : in std_logic;
op : in std_logic;
i : in std_logic;
oclk : out std_logic);
end gatedclk;
architecture Behavioral of gatedclk is
signal gclk : std_logic;
BEGIN
gclk <= op xor i;
oclk <= gclk nand clk;
end Behavioral;

library ieee;
use ieee.std_logic_1164.all;

entity shift is
port(C, I : in std_logic;
O : out std_logic);
end shift;
architecture archi of shift is
signal tmp: std_logic;
begin

process (C)
begin
if (C'event and C='1') then
O<=I;
end if;
end process;
end archi;

library ieee;
use ieee.std_logic_1164.all;

t entity reg is
port(Clk, SI : in std_logic;
SO : out std_logic);
end reg;
architecture regist of reg is

component shift is
port(C, I : in std_logic;
O : out std_logic);
end component;

component gatedclk is
port( clk : in std_logic;
op : in std_logic;
i : in std_logic;
oclk : out std_logic
);
end component;
signal
C0,C1,C2,C3,C4,C5,C6,C7,d0,d1,d2,d3,d4,d5,d6,d7,
s: std_logic;

begin
bit0c : gatedclk port map ( clk, d0, SI, C0);
FF0: shift port map (C0, d0, d1);
bit1c : gatedclk port map ( clk, d1, d0, C1);
FF1: shift port map (C1, d1, d2);
bit2c : gatedclk port map ( clk, d2, d1, C2);
FF2: shift port map (C2, d2, d3);
bit3c : gatedclk port map ( clk, d3, d2, C3);
FF3: shift port map (C3, d3, d4);
bit4c : gatedclk port map ( clk, d4, d3, C4);
FF4: shift port map (C4, d4, d5);
bit5c : gatedclk port map ( clk, d5, d4, C5);
FF5: shift port map (C5, d5, d6);
bit6c : gatedclk port map ( clk, d6, d5, C6);
FF6: shift port map (C6, d6, d7);
bit7c : gatedclk port map ( clk, d7, d6, C7);
FF7: shift port map (C7, d7, s);
SO<=s;

end regist;