Design a 32-bit adder with reduced supply and parallelism for power saving

**ELEC 6270 Final Report**

**Mingzi Duanmu**

**May 1, 2015**

***Abstract--In this paper, a new architecture for low-power design of parallel adder is proposed. For this design, a reference design and a low power design were first created in VHDL, then converted to Verilog files with Leonardo spectrum, the the SPICE netlist and circuit schematic can be created by using the Design Architect, then use the Hspice to simulation the SPICE netlist , the value of power and delay can created.***

**KEYWORDS: parallel architecture, reduce supply, 32 bit adder.**

# I.TRODUCTION

Low power dissipation is very critical in today's electronic designs. There are two major methodologies to improve adder’s performance. One is architecture viewpoint. In this approach it is required to find the longest critical paths in the multi-bit adders and then shortening the path to reduce the total critical path delay. The other approach is circuit design viewpoint on transistor level. In this approach, designers proposed high performance full adder core based design on transistor level. Many design considerations including the minimum transistor counts, low power consumption, high throughput, full-swing output, driving capability, chip area, and layout regularity are focused.

The parallel adder is the most important element used in arithmetic operations of many processors. With the rising popularity of mobile devices, low power consumption and high performance integrated circuits has been the target of recent research. However, the two design criteria are often in conflict and that improving one particular aspect of the design constrains the other.

In this paper, reference design and a low-power design were first created in VHDL. These designs were both verified functionally using ModelSim, and then converted to Verilog files with Leonardo. Supply voltage calculation was then performed to determine which values would give meaningful results for power analysis performed using Design Architect and Hspice

# II.reference adder design

## 2.1 ripple carry adder

A 32 bit ripple carry adder was first simulated as reference. This is the simplest design in which the carry out of one bit is simply connected as the carry of the next bit. It can be implemented as a combination circuit using n full adder in series and is called ripple carry adder

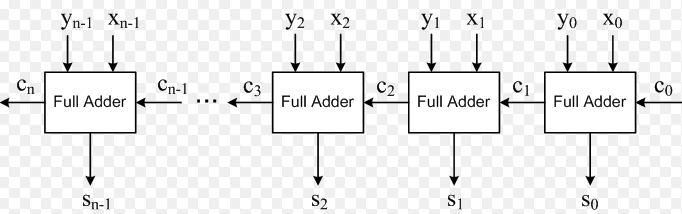


Figure 1

The latency of K-bit ripple carry adder can be derived by considering the worst case signal propagation path. As shown in fig.1 the critical path usually begins at the x0 or y0 input proceeds through the carry propagation chain to the leftmost FA and terminates at sk-1 output.

2.2.design procedure

The 32 bit ripple carry adder was created in VHDL, after writing the HDL code in Model Sim, the code should be compiled after check for errors. Then the VHDL models were optimized in Leonardo Spectrum (Level 3) and converted to Verilog, the SPICE netlist and circuit schematic can be created by using the Design Architect, then use the Hspice to simulation the SPICE netlist , the value of power and delay can be created.

# III. Low-power Design

## 3.1 parallelism

There are two commonly used architectural approaches for decreasing circuit power consumption: first, apply the standard speed optimisation techniques, only more so; second, use parallelism.

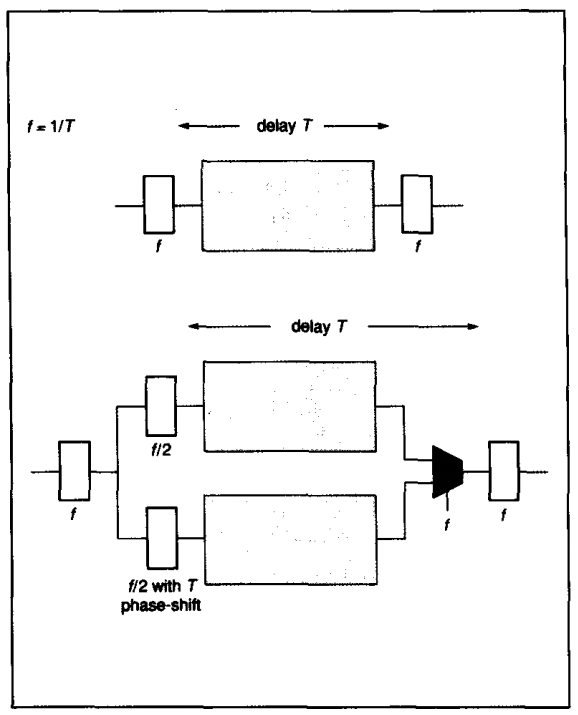
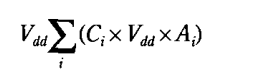


Figure 2

The idea of using parallelism is simply to have more operations being conducted at the slower speed to achieve the same overall performance. This is essentially a tradeoff between circuit area and throughput. The use of parallelism is illustrated in Fig. 2 Here we assume that the critical path delay, T, through the combinatorial logic block has (nearly) doubled due to a reduction in the power supply voltage. To achieve the same throughput, the data is interleaved so that new data is presented to one block while the previous data is still being processed by the other. The outputs of the two blocks are selected by a multiplexer so that the valid data is latched at the original frequency. Notice that although the total capacitance of the circuit has been (approximately) doubled , the term A (in eqn. 1)has been halved because of the speed reduction: these two effects compensate for each other in the dynamic power equation.

(1)

Of course, this strategy may sound attractive in the context of rapidly increasing levels of integration, but in terms of commercial viability it must be remembered that doubling the circuit area can have a large impact upon component cost. Although many design specifications may demand this approach for the resulting speed, many will also preclude it on the grounds of cost.

Parallel architecture of this project is show in Figure 3. A duplicated 32 bit ripple carry adder unit was added to the original design. Two input registers instead of one have been clocked at half the frequency of Fref. A multiplexer is added at output to help keeping the throughput of the parallel design same to the reference design.

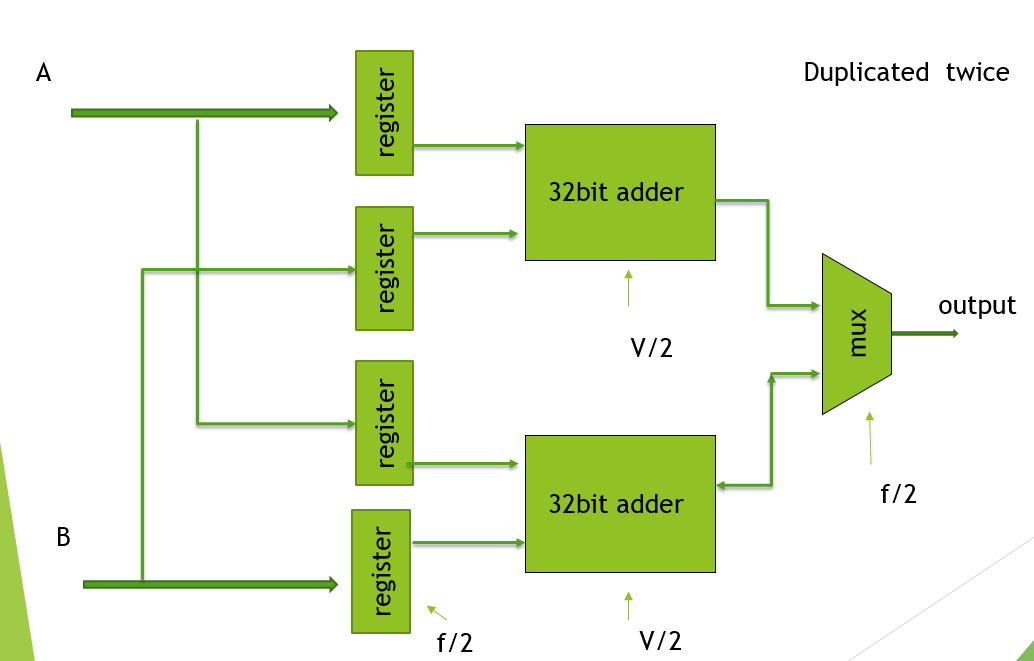
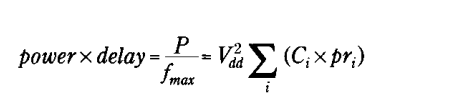


Figure 3

*3.2 Power supply reduction*

One of the main motivations in technology development has been to increase the levels of integration by reducing feature sizes. However, as gate lengths are reduced (without reducing voltage levels)the electric field strength increases in the gate region. This leads to reliability problems as the high electric field strengths accelerate the conducting electrons to such speeds that they cause substrate current (by dislodging holes on impact in the drain area) and actually penetrate the gate oxide. The latter effect gradually alters the characteristics of the device and leads eventually to latch-up and so to destruction. There are three approaches to enabling further feature size reduction. The first is drain engineering in which the doping profile is crafted in the channel region to reduce the degradation due to hot-electrons; the lightly doped drain (LDD) technique allows the smallest gate length.6The second approach is to use new circuit techniques which avoid the high electric fields across individual transistors. The third approach is to reduce the supply voltage; this solution is by far the simplest for circuit designers but acceptance has been delayed as the industry wished to maintain compatibility with existing products.

The reduction in Vdd does not lead to a quadratic reduction in power as might be thought from eqn. 1since some the other terms are dependent upon the supply voltage. To understand the actual effect, consider the activity level of each gate, A. This can be re-expressed as the product of the frequency, f; with which new inputs are presented to a whole circuit (for synchronous circuits, the clocking frequency) and a probability for each node, pri, that it will change on any given cycle. The maximum possible frequency of a circuit, fmax, presents the fastest throughput of data and this is limited by the critical path or longest delay: thus fmax is inversely proportional to circuit delay. This brings us to a common measure of circuit quality: the power-delay product. By re-arranging eqn. 1 we have:  
  
Thus variation in Vdd actually leads to a quadratic change in the power-delay product.

In this design, the 32um technology was be chosen, the supply voltage of this technology is 0.9v and the threshold voltage of this technology is 0.49v, so the low power voltage should between 0.49V-0.9V.

*3.3 design procedure*

In this low power design, the architecture was constitutive of four laths , two 32 bit adders, one multiplexer. The parallel adder was created in VHDL, after writing the HDL code in Model Sim, the code should be compiled after check for errors. Then the VHDL models were optimized in Leonardo Spectrum (Level 3) and converted to Verilog, the SPICE netlist and circuit schematic can be created by using the Design Architect, then use the Hspice to simulation the SPICE netlist , the value of power and delay can created. When simulation in hspice, the value of the voltage and CLK can be set in .vec file. The voltage was 0.53v

and 0.55V, the CLK was 10ns.

# IV. Experimental result

Compare the power of reference design and low power design. The result in Figure 4.

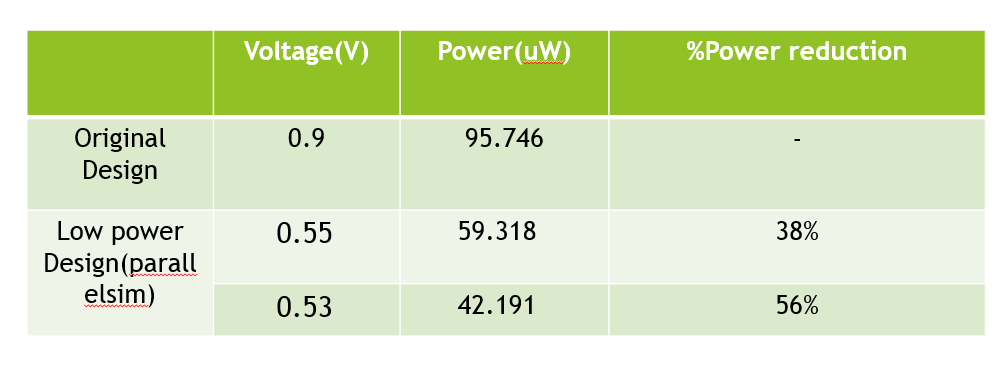


Figure 4

# V. Conclusion

This project proved through experimentation that implementing a parallel scheme for the functional components of a design and reducing the supply voltage to each parallel component can significantly reduce the power dissipation.

***Appendix A:***

Adder.vhd

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.std\_logic\_unsigned.ALL;

entity adder\_32 is

port

(

a, b : in std\_logic\_vector(31 downto 0);

sum : out std\_logic\_vector(31 downto 0);

carry\_out : out std\_logic

);

end entity adder\_32;

architecture Behavioral of adder\_32 is

signal temp : std\_logic\_vector(32 downto 0);

begin

temp <= ('0' & a)+('0' & b);

sum <= temp(31 downto 0);

carry\_out <= temp(32);

end architecture Behavioral;

latch.vhd

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity d\_latch is

Port ( EN : in STD\_LOGIC;

D : in STD\_LOGIC\_VECTOR(31 downto 0);

Q : out STD\_LOGIC\_VECTOR(31 downto 0));

end d\_latch;

architecture Behavioral of d\_latch is

begin

process (EN, D)

begin

if (EN = '1') then

Q <= D;

end if;

end process;

end Behavioral;

MUX.vhd

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity mux is

port ( a,b: in std\_logic\_vector(32 downto 0);

y: out std\_logic\_vector(32 downto 0);

s:in std\_logic);

end mux;

architecture beone of mux is

begin

y<= a when (s='0') else b;

end;

top.vhd

library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

entity top is

port (l,p : in std\_logic\_vector(31 downto 0);

clk1,clk2,clk:in std\_logic;

y:out std\_logic\_vector(32 downto 0));

end top;

architecture one of top is

component mux

port ( a,b: in std\_logic\_vector(32 downto 0);

y: out std\_logic\_vector(32 downto 0);

s:in std\_logic);

end component;

component d\_latch

Port ( EN : in STD\_LOGIC;

D : in STD\_LOGIC\_VECTOR(31 downto 0);

Q : out STD\_LOGIC\_VECTOR(31 downto 0));

end component;

component adder\_32

port(a,b: in std\_logic\_vector(31 downto 0);

sum :out std\_logic\_vector(31 downto 0);

carry\_out :out std\_logic);

end component;

signal a1,a2,b1,b2,sum1,sum2:std\_logic\_vector(31 downto 0);

signal c1,c2:std\_logic;

signal s1,s2:std\_logic\_vector(32 downto 0);

begin

s1<=c1&sum1;

s2<=c2&sum2;

ad\_1:adder\_32 port map(a1,b1,sum1,c1);

ad\_2:adder\_32 port map(a2,b2,sum2,c2);

l1:d\_latch port map(clk1,l,a1);

l2:d\_latch port map(clk1,p,b1);

l3:d\_latch port map(clk2,l,a2);

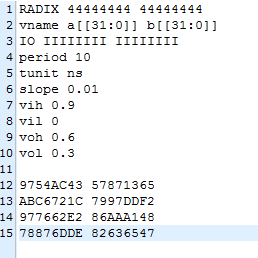
l4:d\_latch port map(clk2,p,b2);

m1:mux port map(s1,s2,y,clk);

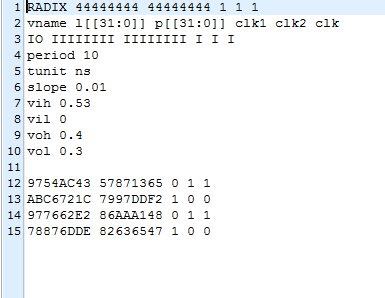
end one;

***Appendix B:***

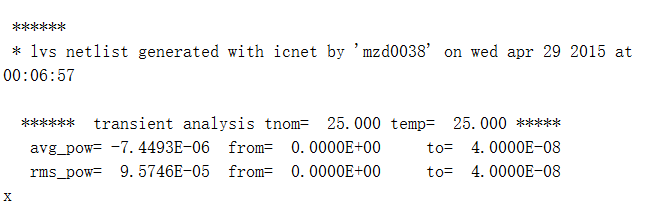
reference design .vec file



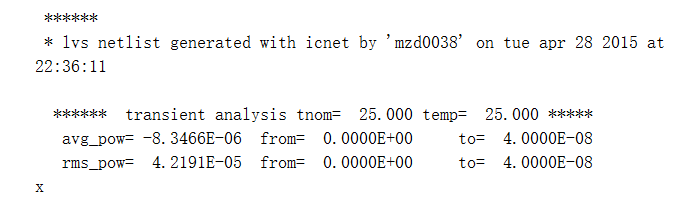
Low power design.vec file



reference design:power



Low power design:power in 0.53V



Low power design:power in 0.55V

