

# An Efficient Timing Optimization Method in ASIC Design

Huang Xu<sup>1,a</sup>, He Xin<sup>1,b</sup>, Yang Wu<sup>1,c</sup> and Li Yujing<sup>1,d</sup>

<sup>1</sup>Sichuan Institute of Solid State Circuits, Chongqing, P.R. China

<sup>a</sup>hxtt103@163.com, <sup>b</sup>letter1988@163.com, <sup>c</sup>yangwu@cetc.cn, <sup>d</sup>liyujing@cetc.com

**Keywords:** Clock tree synthesis, Clock skew, Useful clock skew.

**Abstract:** In the physical design of integrated circuits, the traditional clock tree synthesis is difficult to meet the timing convergence requirements at high frequency. This paper takes a data processing chip design based on TSMC 65nm 1P8M process as an example, proposes a clock tree synthesis method combining bottom-up timing convergence and useful clock skew, efficient to solving the setup time violations of chip at high frequency applications, and meeting the setting time of 0.5ns, holding time 0.15ns, effectively guarantee the timing convergence of the chip.

## 1. INTRODUCTION

In large scale digital integrated circuit design, the timing performance of a design is determined by its data path and clock path. The clock signal is the datum of data transmission, which plays a decisive role in the function, performance and stability of the synchronous digital system. Therefore, the processing of clock signal is a very important part in the design of high performance chip [1].

At present, the design of clock distribution network mainly uses clock buffer tree structure. The starting point of the clock signal is called the root node. The clock signal passes through a series of distributed nodes, and then ends to register clock inputs or other clock ends, which are called leaf nodes. The clock network is inserted driving buffers from root nodes, which can balance the delay of the arrival of different leaf node registers, to meeting the timing requirements, and to building a reasonable clock distribution network. The construction of clock distribution network is called clock tree synthesis.

The main goal of the traditional clock tree synthesis is to minimize the clock skew [2], so as to meet the critical timing requirements of the chip, which is usually judged by no violation of the setup time and hold time. As the chip size is larger and larger, work frequency is more and more high, the traditional clock tree synthesis is difficult to meet the timing convergence requirements at high frequency, which will cause a large amount of setup time violation, resulting in chip timing without convergence, cannot do subsequent routing. Then the clock tree synthesis puts forward new requirements.

This paper takes a data processing chip design based on TSMC 65nm 1P8M process as an example, proposes a clock tree synthesis method combining bottom-up timing convergence and useful clock skew, efficient to solving the setup time violations of chip at high frequency applications.

## 2. TIME SEQUENCE

### 2.1. TIME SKEW

At present, most of the digital circuits are designed into synchronous sequential logic circuits. The basic timing model is composed of the registers and the combinational logic circuit between registers, as shown in Figure 1(a).

The ideal circuit system needs the clock signal to be as possible as consistent on each register, making behavior synchronize of each register. However, in the actual synchronization circuits, because physical connection lengths from the clock source to each register are different, and the

physical parameters of the connection on the clock path are different, the number of clock buffer into different clock on the path is different, cause clock signal arriving the various parts of the circuit at different time. There is a time difference between the different register clocks in the clock domain from a same clock, which is called clock skew, as shown in Figure 1(a), the waveforms are shown in Figure 1(b). The existence of clock skew makes the setup time and hold time of the register unable to meet the timing requirements.

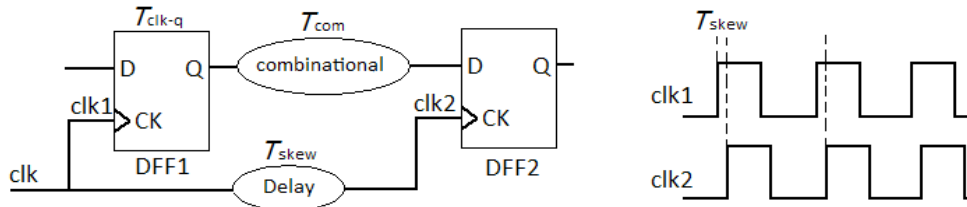


Figure 1 (a) The clock skew

(b) The waveforms of clock skew

For setup time,  $T_{setup}$  must be satisfied

$$T_{cycle} + T_{skew} \geq T_{clk-p\_max} + T_{com\_max} + T_{setup} \quad (1)$$

If Formula 1 is not satisfied, it is called setup time violation.

For hold time,  $T_{hold}$  must be satisfied

$$T_{hold} + T_{skew} \leq T_{clk-p\_min} + T_{com\_min} + T_{setup} \quad (2)$$

If Formula 2 is not satisfied, it is called hold time violation.

$T_{skew}$  is clock skew,  $T_{cycle}$  is clock cycle,  $T_{clk-p\_max}$  is maximum register device delay,  $T_{clk-p\_min}$  is minimum register device delay,  $T_{com\_max}$  is maximum combinational logic propagation delay,  $T_{com\_min}$  is minimum combinational logic propagation delay.

By Formula 1, you can see that when  $T_{cycle}$ ,  $T_{clk-p\_max}$  and  $T_{com\_max}$  are fixed, if  $T_{skew}$  is positive, That is, the time delay of the clock path of the register DFF2 is greater than DFF1, and the setup time is easy to be satisfied. In this case, formula 2 is not easy to satisfy, and it is easy to hold time violations, leading to design errors. In physical design, buffers can usually be inserted into a data path, so that  $T_{com\_min}$  is increasing to satisfy formula 2 is used to repair the hold time violation so that the time series converges. If  $T_{skew}$  is negative, setup time violation is prone to occur. There are usually three methods to fix a time violation and to make a sequential convergence: increases the clock cycle, reduces  $T_{com\_max}$ , and increases  $T_{skew}$ . However, the first method results in reduced chip performance. The second method is completed by the front-end staff rewriting code, but it will lead to an increase in the number of clock cycles, affect the performance of the chip, and increase front-end design staff workload, and extend the chip design cycle. The third method, by changing the path of the clock, may cause other parts to be violated, so it must be handled with care and accuracy.

## 2.2. USEFUL SKEW

Studies have shown that clock skew that affects the timing performance of the system is usually only related to each other of the two sequential units adjacent to the time sequence. If this clock bias can be effectively used, the performance and stability of the circuit can be improved and the design of high performance can be achieved. This clock skew is also known as a useful clock skew[3].

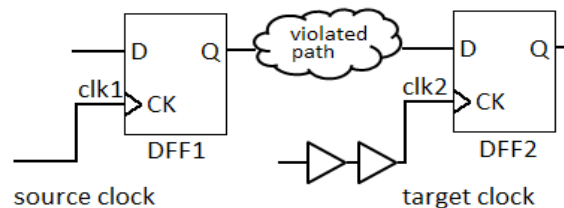


Figure 2 Useful skew timing diagram

When designing a chip, the delay time of the combinational logic circuit of the clock tree structure is usually larger, while the clock delay in the register in the same clock tree is relatively small. At this time, in order to meet the combinational logic circuit long timing delay, we can increase the register

clock delay, and does not affect the function of the clock tree, namely in the sequential path establishing particularly tense time on the appropriate clock path while receiving data register on the increased delay, increasing  $T_{skew}$ . This method is called a useful skew of the clock tree[4]. As shown in Figure 2, in the light of setup time violation, we can speed up the source clock signal to reach of first registers, slow down the target clock signal to reach of second registers, namely using differential clock signal delay to improve the circuit timing, so as to make the circuit work in the best performance.

By Formula 1, it can be seen that increasing  $T_{skew}$  is easier to meet the requirements of the setup time, and thus a better time margin value is obtained. But formula 2 shows that increasing  $T_{skew}$  is not good for hold time. Therefore, in the integrated design of higher frequency, it is necessary to sacrifice the hold time properly to meet the more and more severe setup time requirements, and then repair the violation of the hold time.

### 3. DESIGN AND ANALYSIS

This paper takes the data processing chip for example, the design scale is about 2 million gates, and the operating frequency includes two modes, 125MHz and 250MHz. It is taped out in the TSMC 65nm 1P8M technology successfully, the clock structure mode as shown in Figure 3.

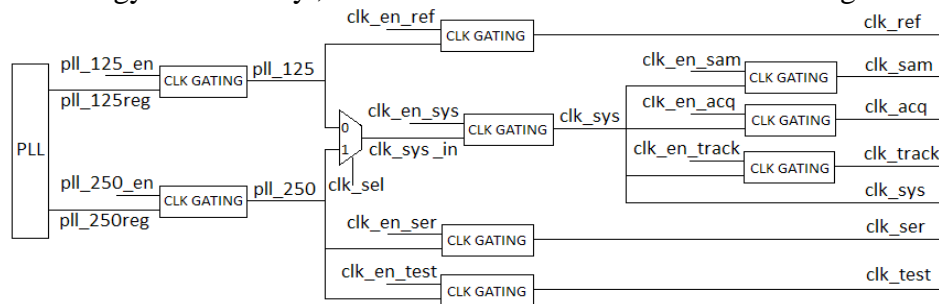


Figure 3 The clock structure mode

The traditional clock tree synthesis, first, clock is synthesized from bottom to top, although you can successfully decrease the clock offset, but in order to ensure the performance of the chip is better, to achieve the desired design goals, we use a clock tree synthesis method combining bottom-up timing convergence and useful clock skew.

An external passive crystal oscillator provides a reference clock source for the chip PLL, and then obtains two clocks pll\_125reg and pll\_250reg by frequency doubling and frequency down by the PLL. In the clock control module, the first gating of the two clocks is carried out, and the first stage clock pll\_125 and pll\_250 are obtained. Secondly, through gating, second levels of clock clk\_ref, clk\_ser, clk\_test are obtained. Then select the first stage clock pll\_125 and pll\_250, and get the clock clk\_sys\_in, gated to obtain third clock clk\_sys chip system clock. Finally, fourth level clock clk\_sam, clk\_acq and clk\_track are obtained by controlling signals respectively.

Firstly, do the clock tree synthesis level by level gradually, constraint all clocks, the scripts is:

```
set_clock_tree_optings - max_transition 0.2 - max_capacitance 0.3
-max_fanout 20 - target_skew 0 - buffer_relocation TRUE -buffer_sizing TRUE
- gate_relocation TRUE - gate_sizing TRUE - delay_insertion TRUE
```

And set all clocks to ideal clock.

```
remove_propagated_clock [all_fanout - clock]
remove_ideal_latency -all
remove_ideal_network -all
```

First, do the the skew\_opt implementation of the fourth level clock, involves timing analysis of different clocks, selecting the optimized path, and get the optimization strategy, writing skew\_opt.tcl, that is, different paths required different useful clock offsets. Finally, read the skew\_opt.tcl as the target clock offset of the clock tree synthesis.

```
skew_opt - path_groups [get_path_groups {clk_sam clk_acq clk_track}]
```

To do clock synthesis of the first part clocks.

```
compile_clock_tree - config_file_wrire rpts/compile.rpt - clock_tree {clk_sam clk_acq  
clk_track}
```

Fixing the first part clock tree structure, it forbidden to be changed.

```
mark_clock_tree - clock_synthesized - clock_tree {clk_sam clk_acq clk_track}
```

Then the same method is used to integrate all the clock tree synthesis of the latter three levels clocks.

After the clock tree synthesis has been fully finished, because of the optimized routing of clock skew, the WNS sequence can improve the 0.6ns of the original large violated path, improves the speed of the entire design effectively. It is compared with the QualityofResult (QoR) after the tool is automatically integrated with the clock tree, as shown in Table 1.

Table 1 QoR contrast after two different methods of CTS

Timing data	Auto CTS	Bottom-up CTS with useful skew
SetupWNS (ns)	-0.876	-0.168
SetupTNS (ns)	-35.982	-7.214
HoldWNS (ns)	-0.325	-0.199
HoldTNS (ns)	-21.698	-9.651
No.of Violated Path	575	128

Afterwards, the violation timing path is repaired and the final timing result is obtained by iterating, as shown in table 2. As can be seen, all the clock setup time to 0.5ns, hold time to reach even more than 0.15ns, to meet the design requirements for hold time.

Table 2 Comparison of two different methods after timing repair

clk name	Auto CTS		Bottom-up CTS with useful skew	
	Setup (ns)	Hold (ns)	Setup (ns)	Hold (ns)
clk_sam	0.325	0.161	0.686	0.158
clk_acq	0.483	0.157	0.963	0.161
clk_track	0.421	0.152	0.696	0.150
clk_sys	0.534	0.158	1.212	0.158
clk_ref	0.900	0.157	1.537	0.157
clk_ser	1.621	0.151	2.632	0.153
clk_test	2.865	0.150	3.965	0.152
pll_125	0.698	0.159	0.963	0.162
pll_250	0.404	0.160	0.856	0.159

#### 4. CONCLUSIONS

This paper presents a clock tree synthesis method combining bottom-up timing convergence and useful clock skew, basing on a data processing chip, elaborates the implementation process of the method. The detailed data comparison results show that the proposed method can effectively construct the clock tree, so that the chip can achieve timing convergence at 250MHz frequency, and successfully meet the clock requirements of the chip. The chip is taped out in the TSMC 65nm 1P8M process successfully, after testing, the chip can work normally in the 125MHz and 250MHz frequency, verified the feasibility and effectiveness of the clock tree synthesis.

## References

- [1] Chunzhang Chen, Xia Ai, Guoxiong Wang. Physical Design of Digital Integrated Circuits [M]. BeiJing, China.2012.In Chinese.
- [2] M S Kim, J hu. Associative Skew Clock Routing for Difficult Instances[C]. In the Proceedings of Design Automation and Test in Europe, 2006:1~6.
- [3] Xixi Zhihua. Efficient Implementation of Useful Skew Optimization for Clock Tree [D].ShangHai, China.2012.
- [4] Guirong Wu, Song Jia, Yuan Wang. An Efficient Clock Tree Synthesis Method in Physical Design[C]. IEEE International Conference of Electron Devices and Solid State Circuits, 2009.
- [5] IC Compiler Implementation User Guide [M], version D-2010.03-SP2.SYNOPSYS.2010