

# Static Timing Analysis (STA) 기법을 이용한 Clock Tree Synthesis (CTS) 최적화에 관한 연구

## Pre-layout Clock Analysis with Static Timing Analysis Algorithm to Optimize Clock Tree Synthesis

박주현\* · 류성민\* · 장명수\* · 최세환\*\* · 최규명\*\* · 조준동\*\*\* · 공정택\*\*

(Joo-Hyun Park · Seong-Min Ryu · Myung-Soo Jang · Sea-Hawon Choi · Kyu-Myung Choi · Jun-Dong Cho · Jeong-Taek Kong)

**Abstract** - For performance and stability of a synchronized system, we need an efficient Clock Tree Synthesis(CTS) methodology to design clock distribution networks. In a system-on-a-chip(SOC) design environment, CTS effectively distributes clock signals from clock sources to synchronized points on layout design. In this paper, we suggest the pre-layout analysis of the clock network including gated clock, multiple clock, and test mode CTS optimization. This analysis can help to avoid design failure with potential CTS problems from logic designers and supply layout constraints so as to get an optimal clock distribution network. Our new design flow including pre-layout CTS analysis and structural violation checking also contributes to reduce design time significantly.

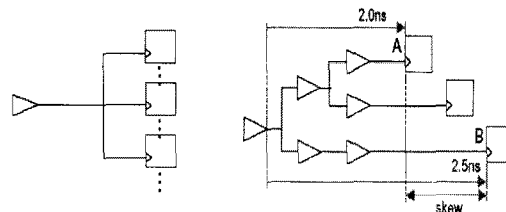
**Key Words** : Clock Tree Synthesis, CTS Optimization, Pre-layout Clock Analysis, Structural Violation Checking

### 1. 서론

최근 반도체 공정기술의 발달로 feature 크기가 작아지고 이에 따른 높은 집적도를 사용한 system-on-a-chip(SOC) 설계가 보편화되고 있다. 이러한 환경 하에서 수백MHz이상 높은 성능과 신뢰성의 확보를 위한 중요한 요소가 clock의 동기화이다[1][2]. Digital 회로는 clock 신호에 동기 되어 작동되므로 clock 신호의 빠른 전이가 system의 성능 향상으로 이어진다. Clock distribution network이라고 표현되는 clock 신호는 system내에서 수백 수천 개의 flip-flop들로 구성된 load들을 구동하며 chip 전체에 걸쳐서 배치된다.

Clock 신호를 chip의 전체 flip-flop에 distribute시키기 위해 trunk, mesh, symmetric H-tree, buffered tree등 다양한 clock network structure가 제안되었다[3][4]. 이중 buffered clock tree방식은 chip size가 늘어나는 단점에도 불구하고 buffer를 삽입하는 방식은 손쉽게 자동화 할 수 있어 ASIC/SOC 설계에 적합하다[1].

회로의 Layout시 자동으로 clock network을 구성하고 적절한 위치에 buffer를 삽입하는 작업을 Clock Tree Synthesis(CTS)라 한다. 회로의 layout 수행 시 CTS는 clock source와 receiver 간의 buffer distribution을 통하여



(a) CTS 전 (b) CTS 후  
그림 1 logic 측면의 CTS의 예

network의 load balance를 얻는 기술이다. 이때 source pin을 clock root라 하며 receiver pin은 일반적으로 flip-flop 혹은 latch의 clock pin이 된다.

CTS의 제약 조건은 source pin에서 특정 receiver pin까지의 지연시간(insertion delay), 그 값 중 가장 큰 값과 가장 작은 값의 차이 값(clock skew)이 있다. 이 제약 조건은 설계의 race분해와 chip 성능을 결정한다. CTS의 quality를 높이기 위해서는 clock scheme이 복잡한 SOC설계를 고려하기 위해 다양한 CTS 제약 조건의 설정이 필요하다. 또한 function clock과 test mode clock을 동시에 고려하는 경우 흔히 발생하는 buffer의 과도 삽입을 막아야 한다. 그림1은 CTS의 한 예를 보였다.

본 논문에서는 이러한 문제를 예방하기 위한 pre-layout 단계의 clock network analysis, test mode CTS 최적화를 위한 gate level의 clock network diagnosis, CTS 제약 조건 생성, structural rule checking 방법을 제안한다.

저자 소개

\* 삼성電子 半導體 SOC研究所

\*\* 삼성電子 半導體 SOC研究所 正博

\*\*\*成均館大學 電氣電子工學科 正教授 · 正博

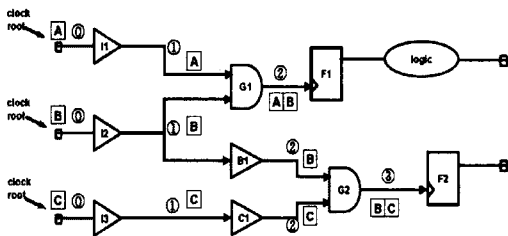


그림 2 clock propagation의 예

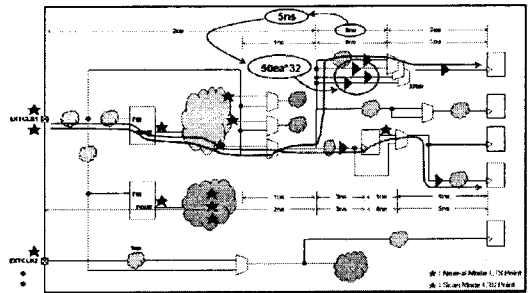


그림 3 test mode CTS 시 과도한 buffer 삽입의 예

## 2. 본론

### 2.1 Clock Network Analysis Technique

Clock network analysis 기능은 설정된 clock의 범위와 다른 clock 간의 관계 파악함으로써 CTS가 곤란한 clock scheme을 layout 전에 detect하고 CTS가 가능한 구조로 바꿀 수 있는 능력이다. Clock root의 개수가 많을 경우 모든 gate에 대하여 clock간 관계 분석에 빠른 수행을 위해 static timing analysis (STA)와 유사한 방식을 적용하였다[5].

그림2는 clock 사이의 상관관계를 인기 위한 clock propagation 방법을 적용 예이다. Logic leveling을 통해 clock network상의 모든 net에 order(①, ②, ③)를 정하고 clock propagation을 수행한다. Propagation시 clock root의 정보를 order가 정해진 net에 fan-in되는 instance의 input net에 clock 정보를 받아 전달한다. 그 결과 gate G1에는 clock root A와 B에서 정의된 clock 정보가 저장되며 gate G2에는 clock root B와 C의 clock이 도달함을 알 수 있다.

Clock propagation 결과 특정 gate에 1개 이상의 clock이 도달해 있다면 CTS 기능이 처리 할 수 없는 경우가 될 수 있으며 그 상관관계는 다음과 같다.

- Clock merge : 두 개 이상의 clock 신호가 하나의 gate에서 만나 동일한 경로를 공유하는 경우
- Clock conflict : 하나의 clock 신호가 분기되어 서로 다른 path로 진행하다 하나의 gate에서 다시 만나는 경우
- Clock overlap : 하나의 clock root가 다른 clock 신호에 의해 drive되는 경우

이중 clock conflict의 경우 CTS algorithm이 어떤 것을 기준으로 skew를 최소화할지 알 수 없으므로 설계자의 guide가 없다면 CTS 중 원치 않는 결과를 얻을 수 있다.

Clock network의 leaf pin들은 CTS시 skew 최소화의 고려 대상이 되는 sync pin과 무시되는 ignore pin으로 나뉘며 분류는 다음과 같다[6].

- Implicit pin : implicit sync pin에는 flip-flop의 clock pin이나 latch의 control pin, implicit ignore pin에는 output floating pin
- User pin : gated pin중 설계자의 필요에 따라 설정 가능한 sync 혹은 ignore pin

Clock analysis 후 clock conflict, merge, overlap등의 문제

가 발생할 경우 올바른 CTS를 위하여 설계자가 user sync, ignore pin을 설정하고 pin setting 결과는 layout의 CTS module의 제약 조건을 생성한다.

### 2.2 Test Mode CTS Optimization

Clock 경로의 logic depth 차이가 큰 normal clock과 test mode scan clock에 대해 단일 CTS 수행 시 삽입된 과도한 buffer가 timing closure의 iteration, area overhead 및 취약한 process variation 문제를 야기 시킨다. 그림3은 test mode CTS시 과도한 buffer가 삽입된 예이다.

이러한 buffer의 과도 삽입을 방지하기 위해서는 function mode clock 경로와 test mode clock 경로를 최대한 분리하여 각 mode간의 CTS에 간섭을 주지 않아야 하며 부득이한 경우에는 각 mode의 선택을 위한 mux 설계를 clock receiver에 근접하게 삽입함으로써 공유 경로를 최소화해야 한다.

실제상의 normal clock과 scan clock network간 공유 경로의 최소화를 위해 RTL 설계 시에 정해진 rule을 바탕으로 설계를 수행하고 logic synthesis 수행 이후에 생성된 gate level net-list를 checking하여 최적화된 CTS 결과를 얻을 수 있는 방법을 제안한다.

Test mode CTS를 위한 RTL coding rule을 정리하면 다음과 같다.

- Inter-scan clock을 위한 clock 설계 rule
- Intra-scan clock을 위한 rule
- MUX gate를 위한 RTL coding rule

RTL 설계에서 정해진 rule이 올바르게 지켜졌는지 여부를 확인하기는 힘들기 때문에 gate level에서 checker를 개발하였다.

Synthesis 이후에는 gate level net-list를 이용하여 clock source와 receiver사이에 존재하는 logic의 종류와 연결 관계를 정리하여 체계적인 rule로 정리하고 이에 따라 clock 정의 과정, clock propagation, receiver pin에서 clock source 방향으로의 tracing 과정을 거쳐 structural checking을 수행한다.

이러한 과정을 거친 설계의 clock 구조는 삽입된 buffer 수의 balance 유지하고 clock gating의 지연 시간을 최소화하여 on chip variation이나 공정 변화에 따른 variation의 영향을 최소화할 수 있다.

## 2.4 New CTS Design Flow

그림4에서 제안하는 CTS 최적화 설계 flow를 보였다. 설계자는 pre-layout단계에서 먼저 test mode CTS 최적화를

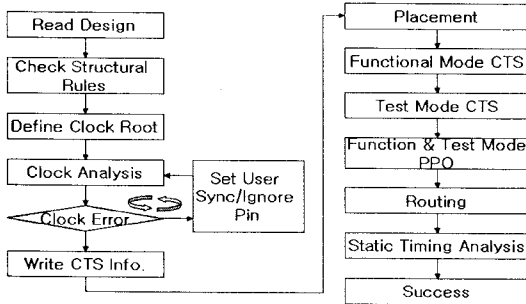


그림 4 clock 분석 기반의 CTS 최적화 design flow

위한 structural rule checking을 수행한다. 그리고 clock root를 지정하고 clock network analysis를 수행한다. Analysis 결과 detect된 CTS 문제는 user sync, ignore pin을 setting하고 analysis를 수행하는 방식으로 제거 할 수 있다. Logic 설계자가 설정한 각종 sync, ignore pin들이 CTS 제약 조건 file안에 동일하게 기술되어 있으므로 logic 설계 단계와 physical 설계 단계의 CTS correlation을 보장하여 최적화된 CTS가 가능하다.

## 2.5 Experimental Results

개발한 기능의 유용성을 검증하기 위해서 다음과 같이 실험을 실시하였다. Example은 scan clock을 포함하여 5개의 clock root의 복잡한 clock scheme을 갖는다. 본 device는 과거 비슷한 clock scheme을 가지는 다른 chip의 설계에서 CTS 문제로 인하여 약 4주간 layout 시간을 소비한 것으로 표1에서 각 clock별 clock analysis 수행 결과를 보였다. 발견된 14개의 clock conflict와 82개의 merge point에 적절한 CTS 제약 조건을 가하여 layout시 one pass CTS를 수행할 수 있었다. 표2에서는 test mode 최적화를 위해 pre-layout에서 structural rule checking을 통해 buffer가 최소한으로 삽입된 예이다. 실험 결과 receiver단에 삽입된 mux를 고려하더라도 약 8300개의 buffer가 약 200개로 줄어든 결과를 보여 본 flow가 실제 설계 환경에 사용될 수 있음을 보였다.

표 1 clock network analysis 결과

clock root	# of sync pins	# of ignore pins	# of gated pins	# of clock conflict	# of clock merge
clock1	27,851	4	704	2	11
clock2	5,393	1	381	2	34
clock3	10,006	2	477	10	6
clock4	1,949	0	82	0	0
clock5	101	1	5	0	1
total	45,300	8	1,649	14	82

표 2 scan test mode CTS optimization 결과

design	삽입 buffer 수 (적용 전)	삽입 buffer 수 (적용 후)
test design	8348	196

## 3. 결론

최근 SOC 환경에서의 design closure를 위해 pre-layout 단계의 RTL 설계 guide를 적용한 후 gate level rule checking을 통해 normal clock과 test clock간의 conflict에 의한 redundant CTS를 예방하고, STA algorithm을 이용한 clock network analysis와 이를 근거로 한 설계자의 설정을 통해 CTS 최적화 할 수 있는 방법을 제안한다. 또한 이 방법이 적용된 새로운 설계 flow를 제시한다. 실험 결과는 구현한 기능에 대한 유용성을 보였으며 실제 설계에 적용할 경우 설계 기간 단축에 기여할 것으로 예상된다.

## 참 고 문 헌

- [1] E. G. Friedman, "Clock Distribution Networks in Synchronous Digital Integrated Circuits", Proc. of the IEEE, Volume:89 Issue:5, pp. 665-692, May2001
- [2] A. Takahashi and Y. Kaiitani, "Performance and Reliability Driven Clock Scheduling of Sequential Logic Circuits", Proc. of ASP-DAC, pp. 37-42, 1997
- [3] A. Balboni, C. Costi, M. Pellencin, A. Quadri and D. Sciuto, "Clock Skew Reduction in ASIC Logic Design: A Methodology for Clock Tree Management", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Volume:17 Issue:4, April 1998
- [4] S. Rusu and S. Tam, "Clock Generation and Distribution for The First IA-64 Microprocessor", Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International, pp. 176-177, 2000
- [5] 박주현, 장명수, 임철, "CubicTime : High Speed Gate Level Static Timing Analyzer for Multi-million Gate Design", 삼성반도체기술학회, Nov 2000
- [6] Aollo User Guide, Avant! Inc., 2001