

# Timing violation의 효율적인 수정을 위한 closure 방법의 제안

\*서영호, \*\*박성호, \*\*\*최현준, \*\*\*김동욱

\*한성대학교, \*\*LG전자, \*\*\*광운대학교

e-mail : \*[kyhseo@hansung.ac.kr](mailto:kyhseo@hansung.ac.kr), \*\*[chipdori@lge.co.kr](mailto:chipdori@lge.co.kr), \*\*\*[chj/dwkim@kw.ac.kr](mailto:chj/dwkim@kw.ac.kr)

## A New Closure Method for Efficient Modification of Timing Violation

\*Young-Ho Seo, \*\*Sung-Ho Park, \*\*\*Hyun-Jun Choi,  
and \*\*\*Dong-Wook Kim

\*Hansung University, \*\*LG Electronic, \*\*\*Kwangwoon University

### Abstract

In this paper, we propose an efficient timing closure methodology during physical implementation. Many types of slacks and closure solutions were introduced case-by-case. The major part of violations was managed by specified tools, but the exceptionally generated minor violation which was occurred through correlation error between tools was manually corrected by ASIC engineer. From the proposed method, we identified that the best effective method is to decrease the sum of intrinsic delay in case of setup time violation.

### I. 서론

공정이 미세해질수록 ASIC 엔지니어가 physical implementation시 고려해야 하는 사항들이 점점 많아지고 있다. 그러나 Time-to-market이 강조되면서 칩 개발기간은 점점 짧아지고 있다. 따라서 ASIC 엔지니어는 효율적이고 검증된 칩 개발 과정을 구축하고 각 과정마다 신뢰할 수 있는 EDA tool을 사용해서 개발 일정을 맞추고 있다. 그러나 모든 문제를 tool이 해결해줄 수 없다. Tool마다 모델링 및 해석 방식에 차이가 있으므로 correlation 오차가 발생한다. 이러한 오차들은 결국 violation으로 남는다. 이런 문제를 극복하기 위하여 대량의 violation은 최대한 tool을 이용해서 수정되 소량의 violation은 ASIC 엔지니어가 manual

ECO (engineering change order) 작업을 수행해서 수정해야 한다. Timing violation은 tool간 correlation 차이로 발생하는 violation의 대표적인 예이다. Optimization 및 routing과정에서 모든 tool이 timing constraint를 만족시키는 것을 최우선시 하고 있지만 아직도 correlation 문제는 극복되지 않았다. 따라서 ASIC engineer는 STA(static timing analysis)결과를 참고하여 manual로 timing violation을 수정해야 한다.

### II. 제안한 Manual Timing Closure

Logic synthesis 및 physical synthesis, PPO (post-placement optimization), 및 PRO (post-route optimization)등의 노력에도 불구하고 언제나 STA결과에는 setup time violation 및 hold time violation이 존재하게 마련이다. STA결과는 RC extraction, delay calculation등 전문적인 각각의 tool에서 산출된 결과를 종합하고 해석한 결과이다. 만약 이 모든 tool과 완벽하게 호환성이 있는 tool이라면 한번에 timing violation을 수정 할 수 있다. 그러나 세상에 그런 tool은 없다. 오히려 소량의 timing violation을 수정시키기 위하여 tool에 의존했다가 더 많은 timing violation을 만드는 경우가 많다. 결국은 설계자가 직접 STA결과를 토대로 timing violation을 수정해야만 한다.

#### 3.1 제안한 Setup Time Violation 수정

Setup time이란 플립플롭 혹은 레지스터의 동작적인

안정화를 위하여 클록이 천이(transition)하기 전에 데이터가 먼저 입력되어야 하는 최소한의 시간이다. 따라서 클록이 천이한 이후 다음에 천이하는 시간보다 데이터의 전파 경로에서 소요되는 지연시간이 커서 데이터가 먼저 안정화되어 있지 못하면 setup time violation이 발생한다. 그 해결 방안은 아래와 같다.

- Data required time 증가방안
  - 클록의 주기 : chip의 spec과 연관이 있으므로 함부로 변경하지 않음.
  - EP의 클록 propagated delay 증가 : useful skew control (특정 CTS 경로에 cell을 insertion/remove하면서 phase delay 조절)
  - Library의 setup time 감소 : 빠른 sequential cell을 사용.
  - 클록 uncertainty 감소 : timing margin 조절.
- Data arrival time 감소방안
  - SP의 클록 propagated delay 감소 : useful skew control
  - data path delay 감소 : 주로 많이 사용하는 방법

### 3.2 제안한 Hold Time Violation 수정

Hold time이란 클록이 천이되는 동안 입력된 데이터가 안정화 되어야 하는 최소한의 시간이다. 따라서 이전 데이터 경로에서 거의 지연시간이 소요되지 않아 클록의 천이시간 동안 입력된 데이터가 이전 데이터로 변하여 hold time violation이 발생한다. 그 해결 방안은 아래와 같다.

- Data arrival time 증가방안
  - SP의 클록 propagated delay 증가 : useful skew control
  - Data 경로 delay 증가
- Data required time 감소방안
  - EP의 클록 propagated delay 감소 : useful skew control
  - 클록 uncertainty 감소 : timing margin 조절

### III. 구현

표 1에서 3까지는 앞에서 제안한 manual timing closure 방법들에 대한 실험 결과 및 slack의 감소량을 나타내었다. 표 5에서 표 1에 대한 2nd timing closure 결과를 나타내었다. Manual closure는 PC 수정 db를 이용하였다. 더 이상 tool이 slack을 수정하지 못함을

확인할 수 있다. 1st timing closure 후에 수동으로 직접 timing closure를 수행하면 시간은 많이 소요되지만 slack을 대폭 감소시킬 수 있으며 게이트수를 유지 또는 소량 감소시킬 수 있음을 확인할 수 있다.

표 1. 천이지연시간의 감소량 비교

	Rout. Path Ctrl		FO Split	
	Before	After	Before	After
transition	1.20ns	0.40ns	2.60ns	1.10ns
Setup slack	-0.05ns	+0.03ns	-0.07ns	+0.08ns

표 2. 일정 경로의 Intrinsic delay의 감소량 비교

	Del. Buf.Chain		F/F CG	
	Before	After	Before	After
Σ Delay	0.32ns	0.10ns	0.51ns	0.36ns
Setup slack	-0.09ns	+0.13ns	-0.04ns	+0.11ns

표 3. 2nd timing closure 결과

	After			
	Astro	PC	Man.	
gate count	2,330,099	2,150,112	2,097,003	
Setup	WNS	-0.11	-0.04	-0.02
	#WNS	411	99	19
Run Time(H)	2.5	2	6.5	

### IV. 결론

이 논문에서는 효율적으로 timing violation을 수정시키기 case-by-case로 violation 수정 방안을 제안하였다. 대량의 violation은 tool에 의존하되 tool간 correlation 오차로 인해서 수정되지 않는 소량의 violation은 ASIC 엔지니어가 직접 수정해야 한다. 아울러 setup violation 수정 시에는 cell의 intrinsic delay의 합을 감소시키는 방법이 가장 효율적임을 확인하였다.

### 참고문헌

- [1] Synopsys, Physical Compiler User Guide, 2005.
- [2] Synopsys, Astro User Guide, 2005.
- [3] Synopsys, Prime Time User Guide Advanced Timing Analysis, 2005.

### Acknowledgement

본 논문은 교육인적자원부, 산업자원부, 노동부의 출연금 및 보조금으로 수행한 최우수실험실지원사업의 연구결과입니다.