An efficient methodology for model extraction using waveform analysis

Naresh Kumar and Sneh Sourabh
Cadence Design Systems

cadence®

Overview

- Introduction
- Waveform propagation (WFP)
- Waveform invariant node
- Efficient waveform propagation
- Compact representation of timing model arcs
- Results
- Conclusions

Introduction

- **Extracted Timing Model (ETM)**
  - encapsulates the timing information of the interface-paths of a given block in Liberty Format (.LIB)
- At advanced process nodes, ETM generation process must take into account full waveforms in its delay computation

Problem Statement:

- To generate the ETMs by efficiently computing and propagating waveforms through various stages of the interface paths.
- To reduce the size of the generated ETMs, whose complexity increase at advanced processes due to modeling non-ideal effects

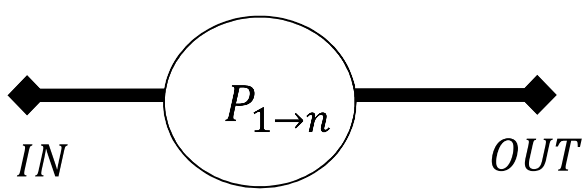
Introduction (Cont'd)

Timing Path

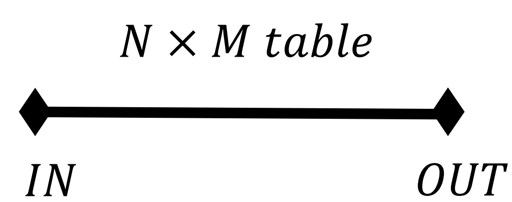
– Timing path consisting of n pins $\{p_1, p_2, \ldots, p_n\}$:

$$P_{1\to n} = \{p_1, p_2, \ldots, p_n\}.$$

Representation of ETM Arcs



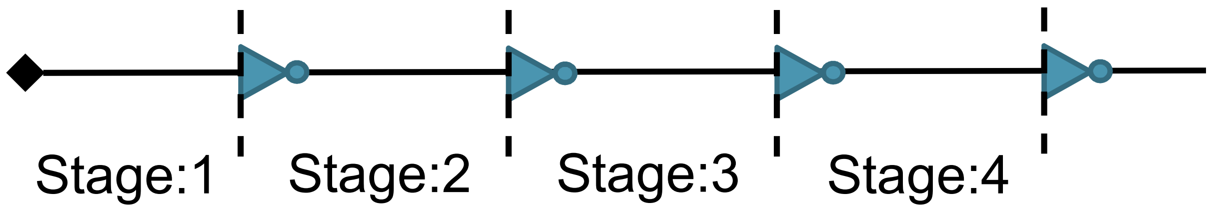
Given Block

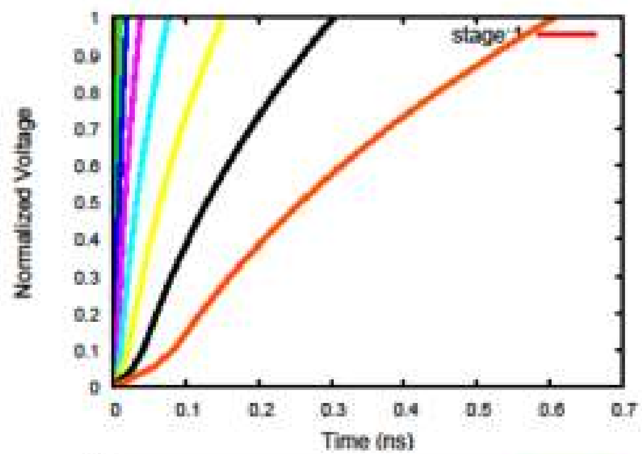


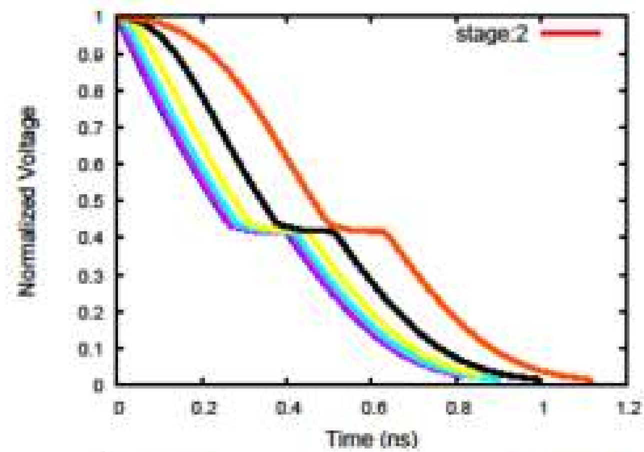
ETM: N Slews and M Loads

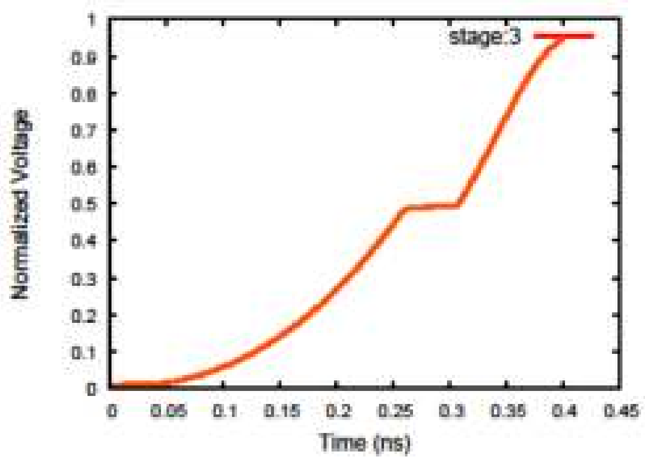
cādence

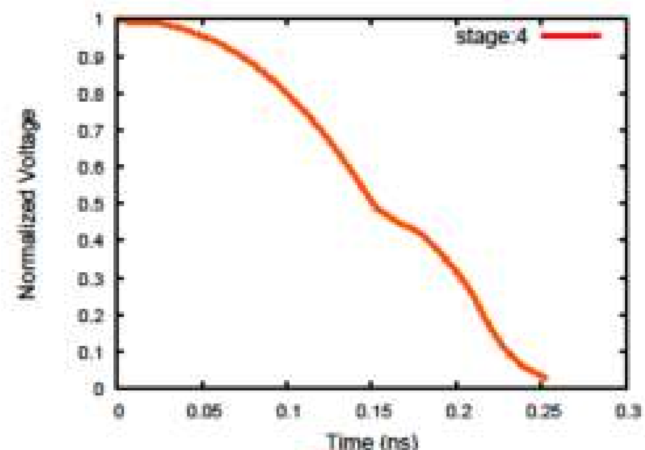
Waveform Propagation (WFP)











Waveform Invariant Node (WIN)

Definition: For a ***given path*** and a ***given set of waveforms*** at the input of the path, the stage closest to the input where ***no variation in the output waveforms*** are observed is termed as *Waveform Invariant Node*

Identification of WIN:

– Given two waveforms W_1 and W_2, ***Waveform Variation Measure*** is defined as:

$$WVM(W_1, W_2) = \frac{1}{|W_1|} \sum_{i=1}^{|W_1|} \{time(W_1, voltage(W_1, i)) - time(W_2, voltage(W_1, i))\}^2$$

– ***Waveform Variation Measure*** for a stage is defined as***:***

$$WVM(W_{Stage}) = \frac{1}{N-1} \sum_{i=1}^{N-1} WVM(W_1, W_{i+1})$$

– WIN is the stage closest to the input where:

$$WVM(W_{Stage}) < S_T \qquad S_T = Stabilization\ Threshold$$

Efficient Waveform Propagation

- Waveform Propagation is done ***topologically*** from input to output
- At each stage, for a given set of N input waveforms, $\boldsymbol{N}$ ***output waveforms*** W_{stage} are computed
- Each stage is checked whether it is a WIN
- If a stage is found to be a WIN, ***only one waveform*** needs to be computed and stored subsequently
  - Original Number of Computation = $\{N \times (n-2)\} + \{N \times M\}$
  - If a stage k is found to be WIN, number of computation = $\{N \times (k-2)\} + \{n-k-1\} + \{M\}$

cādence®

Compact Representation of ETM Arcs

Path Decomposition:

- Delay is two-dimensional (2D) function:

$$Delay(Arc_{p_1 \to p_n}) = f(s, l)$$ [f is a 2D functions]

- For any intermediate pin p_k:

$$Delay(Arc_{p_1 \to p_n}) = Delay(Arc_{p_1 \to p_k}) + Delay(Arc_{p_k \to p_n})$$

$$f(s, l) = \phi'(s, load(p_k)) + \psi'(slew(p_k), l)$$ [where ϕ' and ψ' are 2D functions]

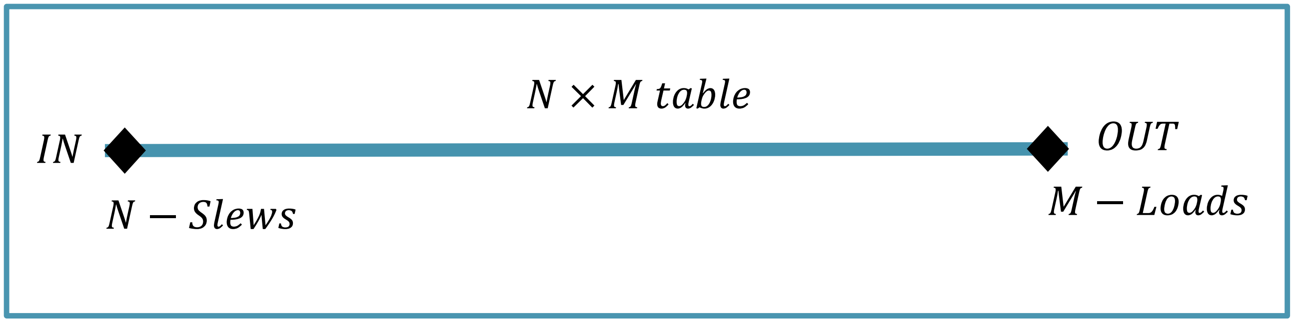
If pin p_k is a WIN (exploit the property of WIN):

- $slew(p_k)$ and $load(p_k)$ are fixed, irrespective of the environment of ETM
- 2D functions ϕ' and ψ' can be represented as one-dimensional (1D) functions:
  - $\phi(s) = \phi'(s, load(p_k))$ $\psi(l) = \psi'(slew(p_k), l)$
- 2D Delay function can be written as sum of two 1D functions:

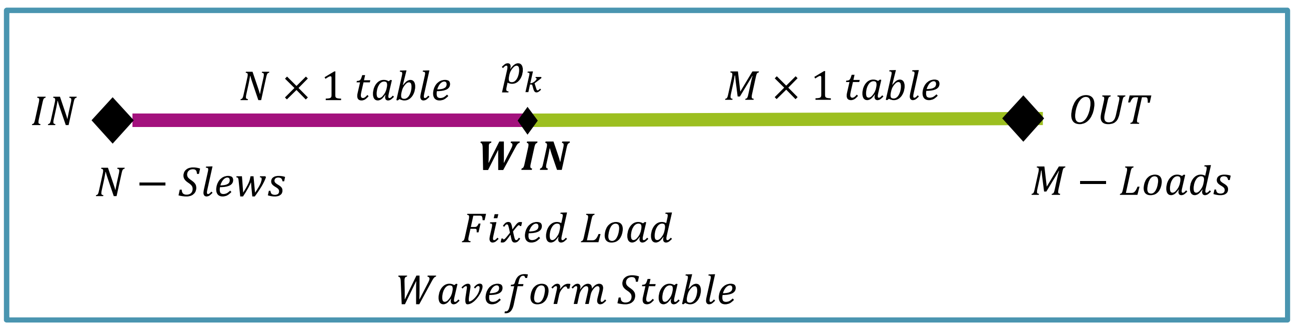
$$Delay(Arc_{p_1 \to p_n}) = \phi(s) + \psi(l)$$

Compact Representation of ETM Arcs

$$(\boldsymbol{N} \times \boldsymbol{M}) \longrightarrow (\boldsymbol{N} + \boldsymbol{M})$$

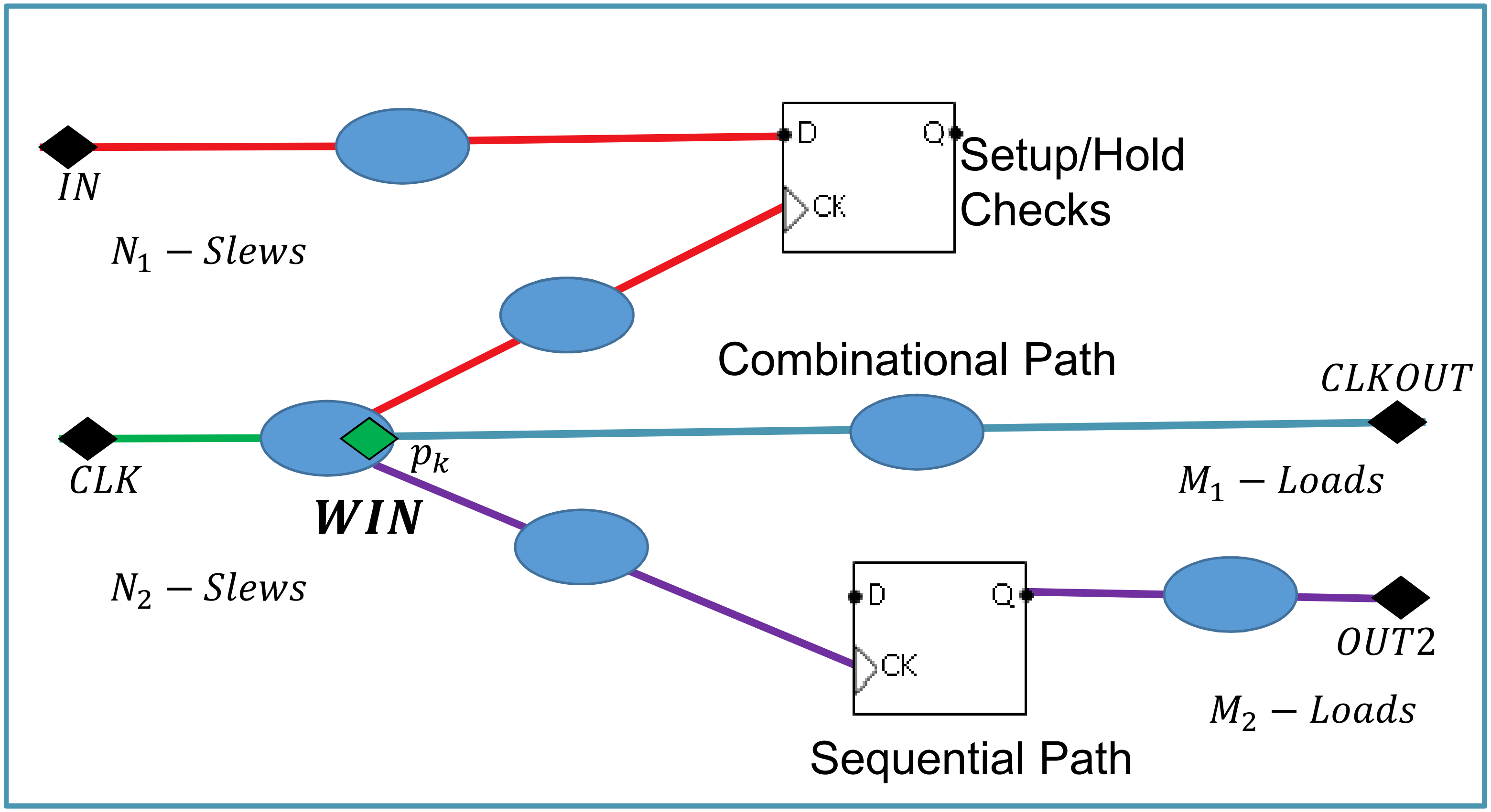


Traditional Model



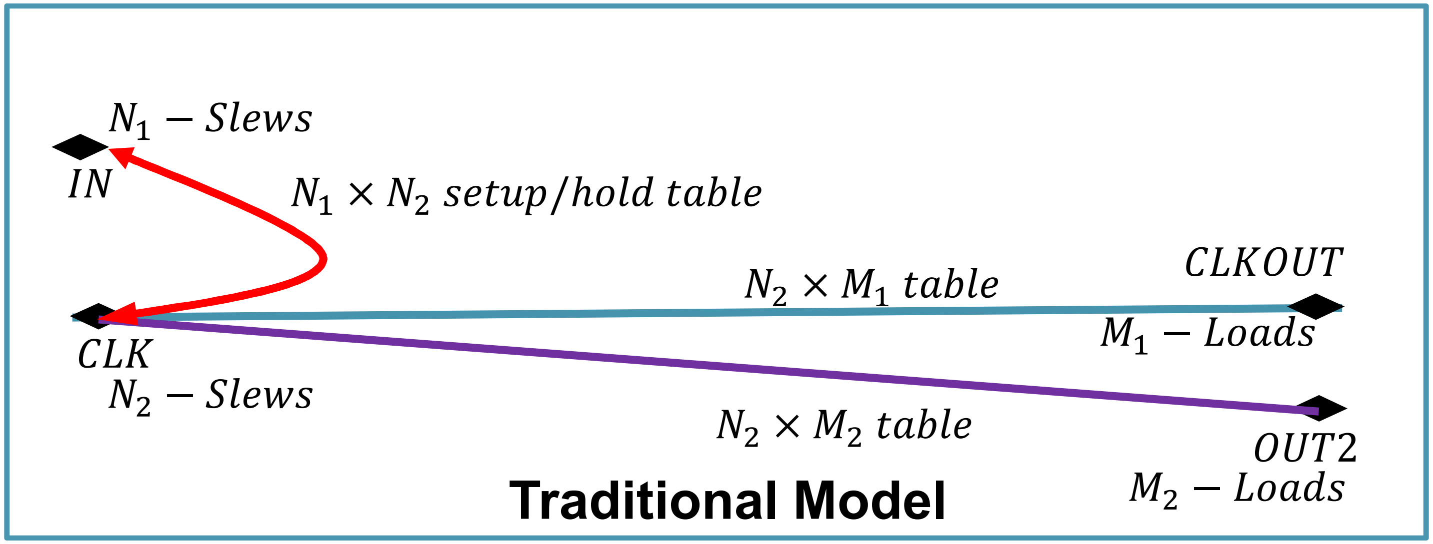
Compact Model

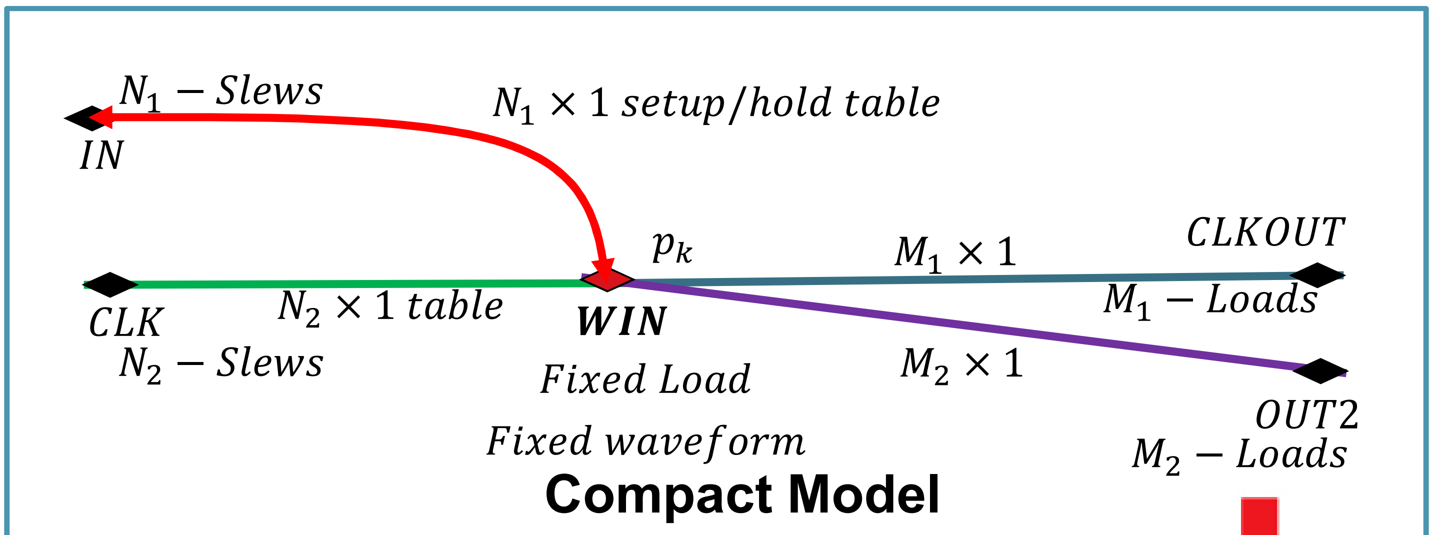
Compact Representation of ETM Arcs



Given Circuit

Compact Representation of ETM Arcs





Performance Results

#	Traditional Flow		Efficient Flow			
	Runtime (s) R_1	Memory (MB) M_1	Runtime (s) R_2	Memory (MB) M_2	$\frac{R_1}{R_2}$	$\frac{M_1}{M_2}$
1	19	39	11	29	1.7	1.3
2	43	98	18	32	2.4	3.1
3	322	609	166	203	1.9	3.0
4	453	875	218	395	2.1	2.2
5	568	1323	529	1249	1.1	1.1
6	1014	2452	431	1047	2.4	2.3

Compaction Results

#	Traditional ETM Size (KB) S_1	Compact ETM Size (KB) S_2	Number of WINs	%age Arc Decomposed	$\frac{S_1}{S_2}$
1	1226	778	23	63	1.6
2	826	530	1	86	1.6
3	22922	16162	255	40	1.4
4	14402	5562	230	81	2.6
5	1688	972	23	92	1.7
6	18882	6290	116	74	3.0

Conclusions

- Efficient Waveform Propagation:
  - Exploits waveform invariance at WINs
  - Achieves a $2\times$ reduction in the runtime and memory usage
  - No accuracy loss

- Compact Representation of ETM Arcs:
  - Transforms 2D arcs to a sum of two 1D arcs
  - Obtains $2\times$ reduction in the ETM disk size
  - No accuracy loss