



ロジックセル・ライブラリ、 寄生素子とサインオフ検証

サクセスインターナショナル株式会社 小川公裕
<https://www.success-int.co.jp/>

Ogawa-Kimihiko@success-int.co.jp

<https://www.success-int.co.jp/ogawa-kimihiro/>

目次

ページ

1. MOSデジタル基本回路	003
1. 概略設計フロー	008
2. 論理基本ゲート	010
3. 順序回路	019
2. MOSデジタルIP	033
1. プロセッサ	035
2. メモリ	054
3. その他IP	090
3. デジタルLSIサインオフ検証	094
1. セル特性抽出	097
2. 寄生素子抽出	102
3. 遅延計算とパワー解析	105
4. PVTWばらつきと設計基準	128
5. MOSの経年変化	149

サンプル

SUCCESS
Intellicon

1.2 1ビット全加算器 計算例

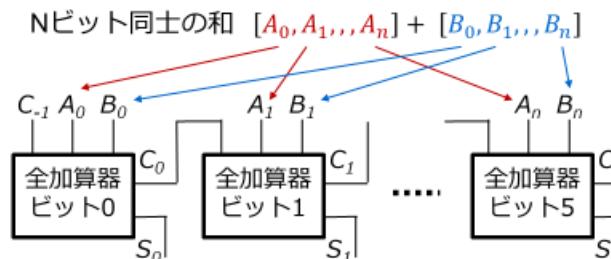
ページ

4. AI	152
5. 日本の半導体の状況	168
最後に	190
付録1 最新プロセス技術	191
付録2 配線構造	196
付録3 3次元積層	201

- 全加算器を6ビット分接続した場合の、計算例

n [b]	-1	0	1	2	3	4	5
An		1	0	1	1	0	1
Bn		0	1	1	0	0	1
Cn	1	1	1	1	1	0	1
Sn	0	0	1	0	1	0	0

上位ビット

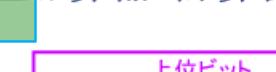
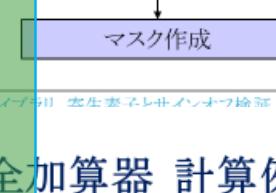
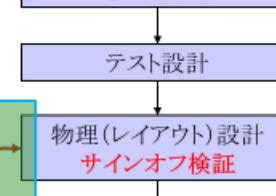
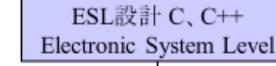
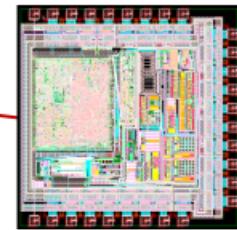
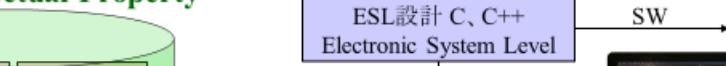


1.1 デジタルLSI設計フロー 概略図

IP: Intellectual Property



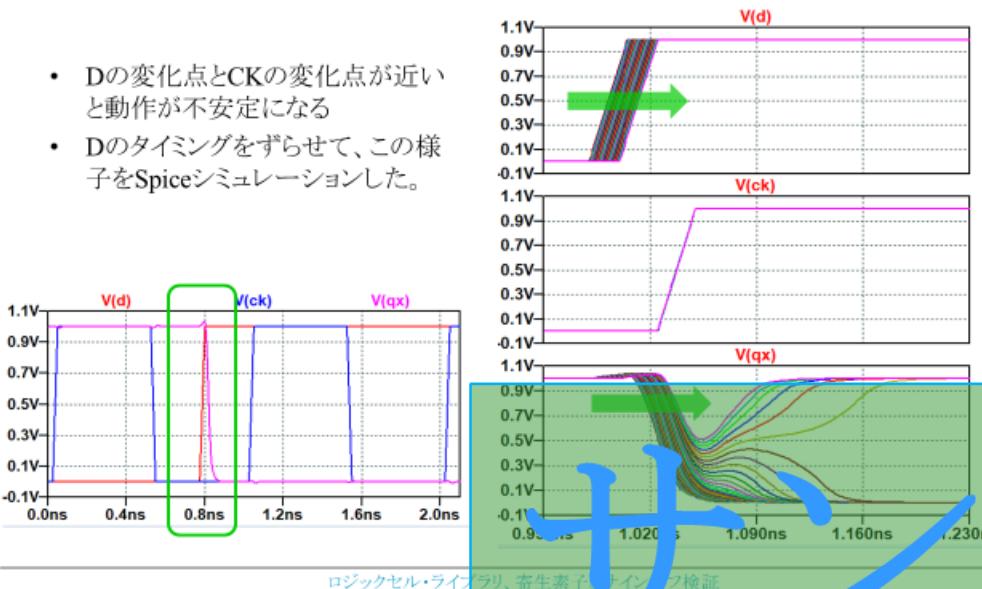
論理Lib



1.3 順序回路 D-Latch

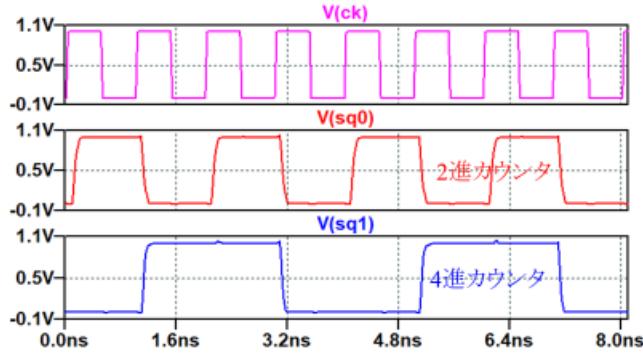


- Dの変化点とCKの変化点が近いと動作が不安定になる
- Dのタイミングをずらせて、この様子をSpiceシミュレーションした。



1.3 同期2進カウンタ

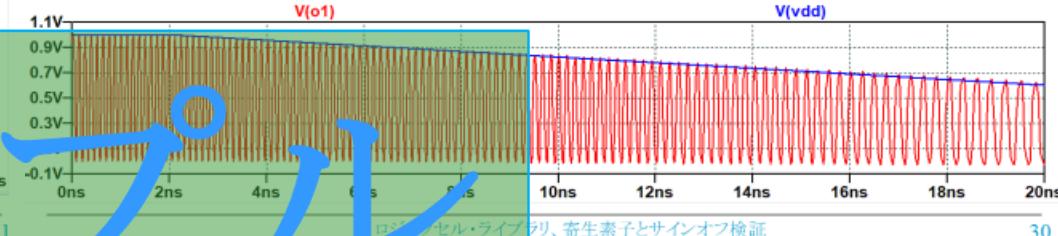
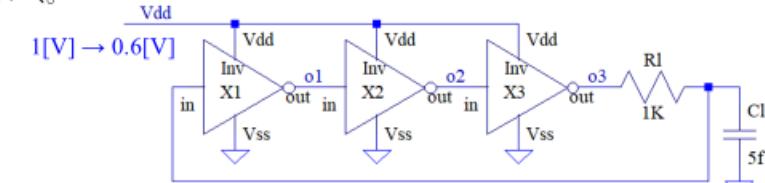
- 同期2進カウンタの例を示す。
- 右図の真理値表から D_0, D_1 と Q_0, Q_1 の関係、つまり $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$ の状態遷移を実現する回路を DFF の前に付けると右下図になり、下のようなSpiceの結果が得られる。sQ1は2進、sQ2は4進カウンタになっている。
- $V(sQ0), V(sQ1)$ 共に CK の立ち上がりで変化するので、クロックに同期している。



1.3 VCO 電圧制御発信器 Voltage Controlled Oscillator

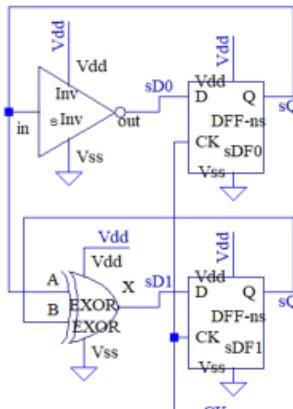


- クロック生成の中心回路としてVCOがある。
- 最も簡単な例として、3段リングオシレータの電源電圧を変化させた発振波形を示す。
- 下のグラフで分かるように V_{dd} を $1[V]$ から $0.6[V]$ まで下げるにつれ、次第に発振周期が長くなっていく。



2.1 デジタルLSI 簡単なCPUの例

現出力		次入力	
Q_1	Q_0	D_1	D_0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



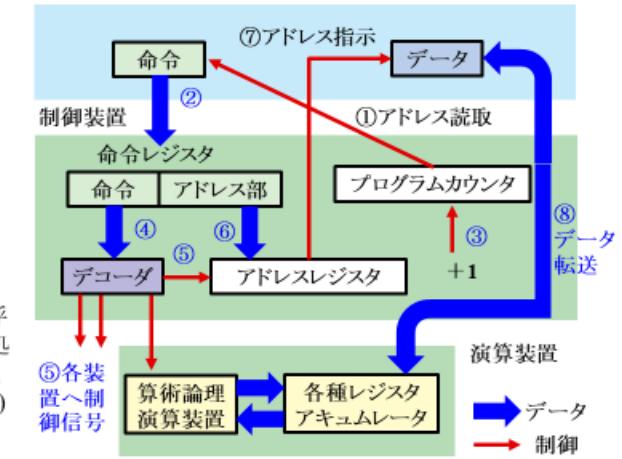
CPUは様々なバリエーションがあり一口には語れないが、基本の処理の流れは

- ① プログラムアドレスを主記憶から読む
- ② アドレスの命令をレジスタにしまう
- ③ プログラムの次の行にカウンタを移す(アドレス+1)
- ④ 命令を解釈する(デコード)
- ⑤ データの準備やその他各種制御指令を送る
- ⑥ アドレスの取り込み
- ⑦ 主記憶にデータのアドレスを指示
- ⑧ 主記憶からデータを移し演算処理

この様な逐次処理方式をノイマン型と呼び長らく主流であったが、最近は並列処理など非ノイマン型が多く登場している。(ニューラルネット、量子コンピュータ等)

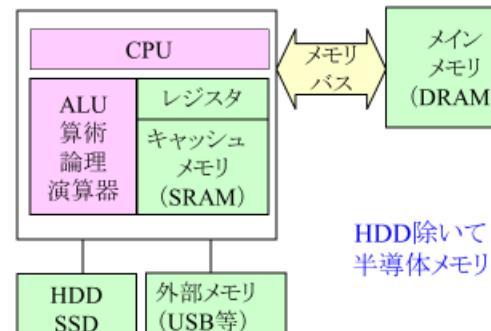
アセンブリ言語 $a = b + c;$
 00411A2C mov eax, dword ptr [b] ; b → eax
 00411A2F add eax, dword ptr [c] ; eax + c → eax
 00411A32 mov dword ptr [a], eax ; eax → a

主記憶装置



2.2 CPUとメモリ

- 速度／コスト／容量／消費電力／耐久性／サイズ 等のトレードオフで様々なメモリ装置が使い分けられている
- メインメモリ：プログラムやデータを一時的に蓄えておくCPUの命令で直接アクセスできる装置
- レジスタ：演算や実行状態の保持に用いる最高速な記憶装置
- キャッシュメモリ：高速メモリで使用頻度の高いデータを蓄積して低速なメインメモリへのアクセスを減らす
- HDD、SSD：保存用大容量メモリ。SSDの方が高速で耐久性が高いため、低価格化と共に置き代わりつつある



CPU: Central Processing Unit
ALU: Arithmetic and Logic Unit
DRAM: Dynamic Random Access Memory
SRAM: Static Random Access Memory
HDD: Hard Disk Drive
SSD: Solid State Drive
USB: Universal Serial Bus

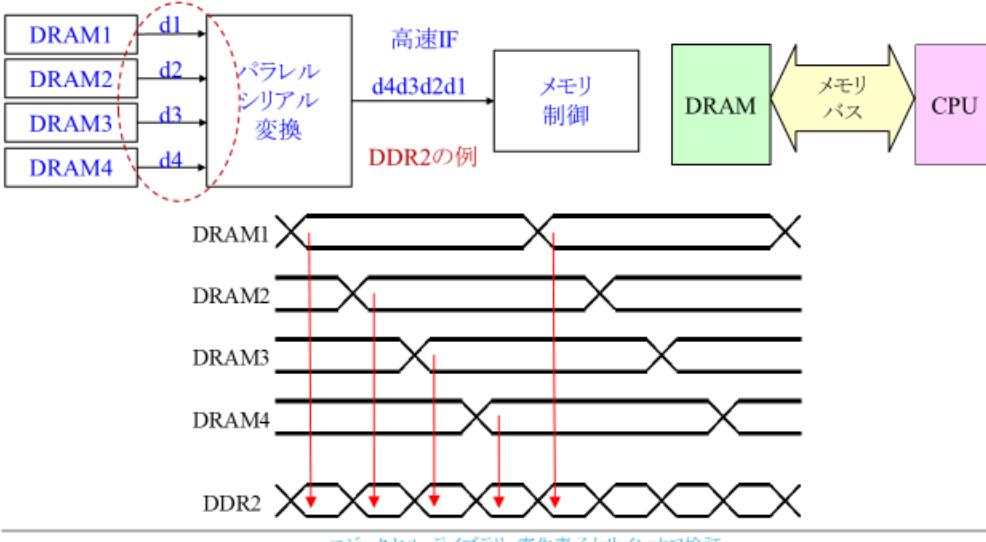
HDD除いて
半導体メモリ

ロジックセル・ライブラリ、寄生素子とサイントラブル検証

55
SUCCESS

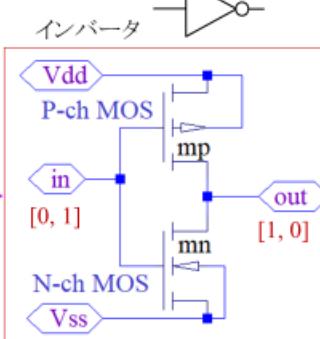
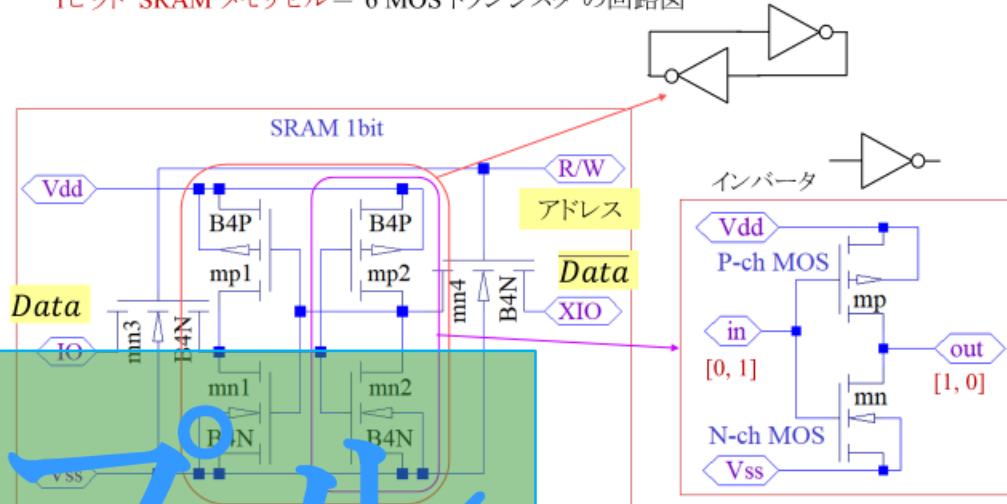
2.2 DDR2 DRAM 4倍転送

下図は×4ビットプリフェッチ *個々のメモリの速度は同じでデータ転送のみが高速化



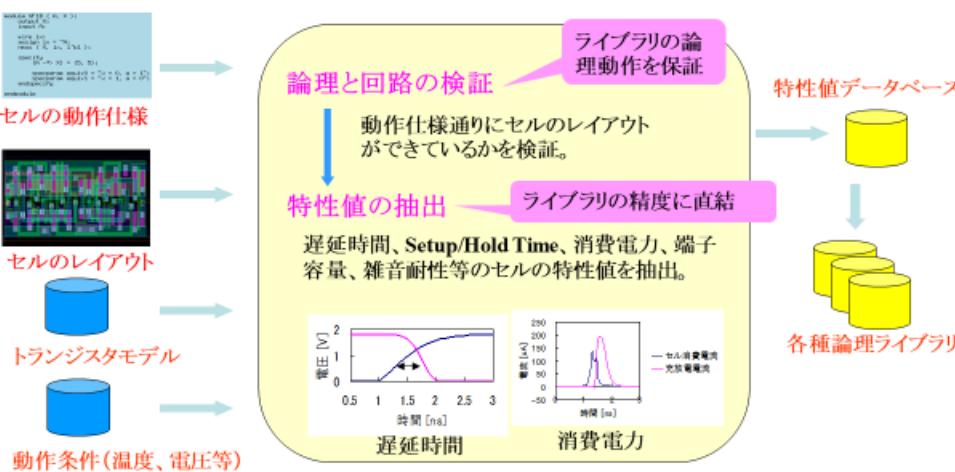
2.2 SRAM回路

- SRAMは回路としてはDlatchと同じで、アドレス入力とデータ入出力線が付く。
1ビット SRAM メモリセル = 6 MOSトランジスタ の回路図



3.1 セル特性抽出

- 論理と物理情報からセルの特性値を抽出しライブラリを生成
- 現在はライブラリ専業会社が主で、数万CPU×数か月の莫大な計算が必要

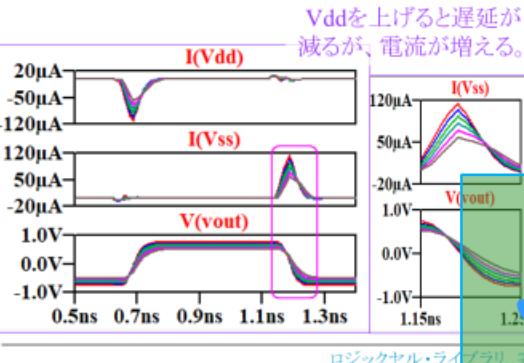


3.3 周波数パワー積

- パワー解析と遅延計算は同じ現象の異なる観点
 - ✓ 信号に着目すれば ⇒ 遅延計算
 - ✓ 電源に着目すれば ⇒ パワー解析
- デジタルの遅延計算とパワー解析の基本データは同じライブラリ上にある。
- 遅延とパワーは常にトレードオフ、即ち 周波数×パワー積の最小化

Vdd [V]	f [GHz]	Pavr [uW]
0.8	0.97	15.1
1.0	1.45	35.5
1.2	1.85	65.9
1.4	2.16	107.7
1.6	2.40	163.2
2.0	2.5	10.8

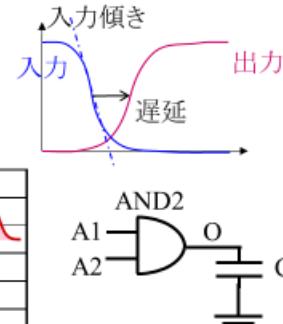
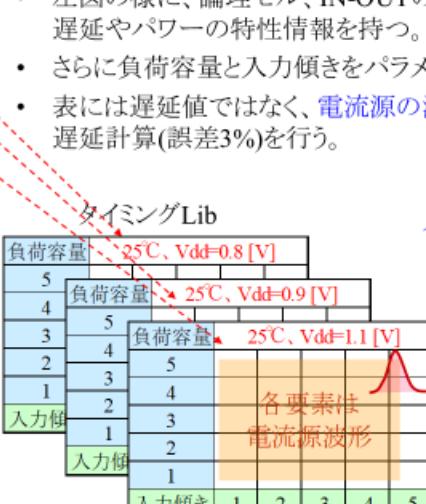
電圧2倍で速度2.5倍だが、パワーは10倍



3.3 nm時代論理タイミングLibと遅延計算

膨大な組み合わせ

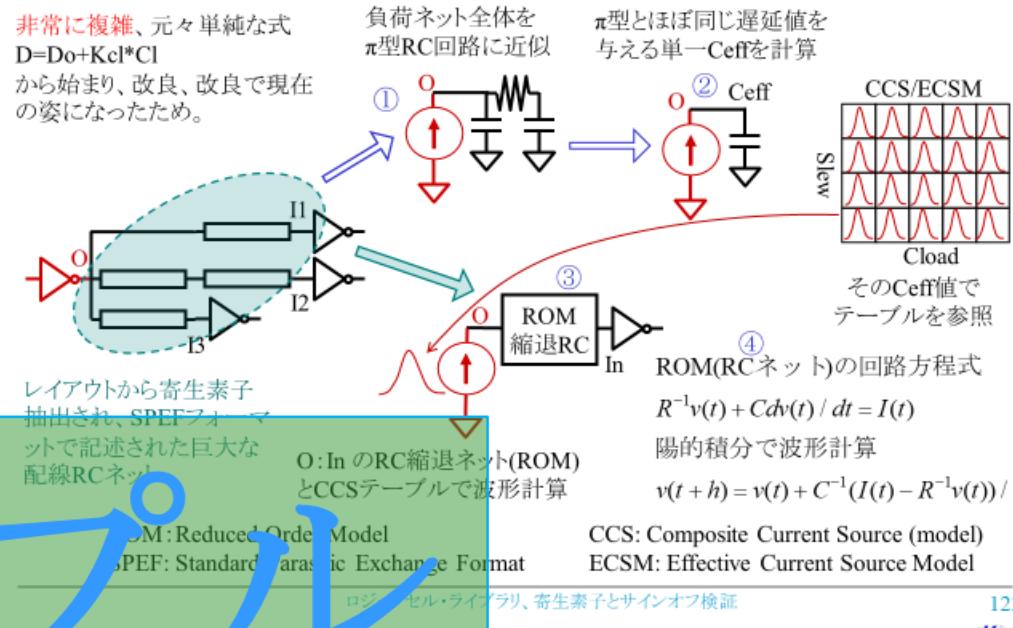
- 入力どれかが変化した場合の出力の遅延を計算する。
- 左図の様に、論理セル、IN-OUTの組、温度、電圧 毎に遅延やパワーの特性情報を持つ。
- さらに負荷容量と入力傾きをパラメータとする。
- 表には遅延値ではなく、電流源の波形を入れ、高精度の遅延計算(誤差3%)を行う。



121

3.3 STA nm時代のRC遅延計算

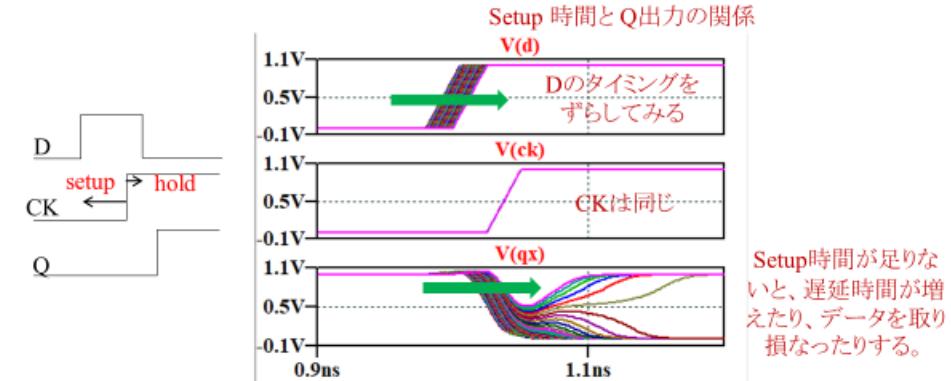
非常に複雑、元々単純な式
 $D = D_0 + K_{el} * C_l$
 から始まり、改良、改良で現在の姿になったため。



3.2 Setup/Hold Time

- デジタル回路の記憶素子に関して、安定動作を保証するための データ入力とクロックのタイミング関係に制約がある。
- Setup 時間制約 クロックよりもデータが早く届いている条件
- Hold 時間制約 クロック後も暫くデータが安定している条件

これらが満たされないと、右下図の様な現象が起こる。

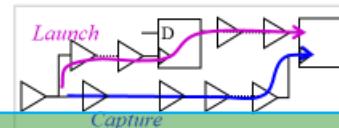


ロジックセル・ライブラリ、寄生素子とサインオフ検証

123

3.3 STAによるタイミング検証

- 信号のタイミングは遅延付き論理シミュレータで検証出来ない訳ではない(昔は使っていた)が
 - ✓ 100%信号パスを網羅する入力ベクトルは作成困難(不可能)
 - ✓ シミュレーション時間が膨大・非現実的(論理は2進整数だが、遅延は実数)
- 論理動作とタイミングの検証をサイクルベース論理シミュレータ(遅延なし)と STA (Static Timing Analysis)に役割分担した。
 - ✓ STAは信号パスの遅延を全網羅して検証可能
 - ✓ 出力→入力 のパス毎の計算なので高速で並列処理も簡単
 - ✓ 調べるのは
 - 右下式の Slack > 0
 - 最小パルス幅 > P_{width}^{min}
 - あらゆる入力依存遅延
 - VT条件
 - 寄生素子を加味した遅延
 - Xtalkや雑音を加味した遅延



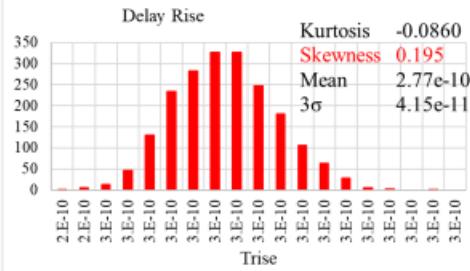
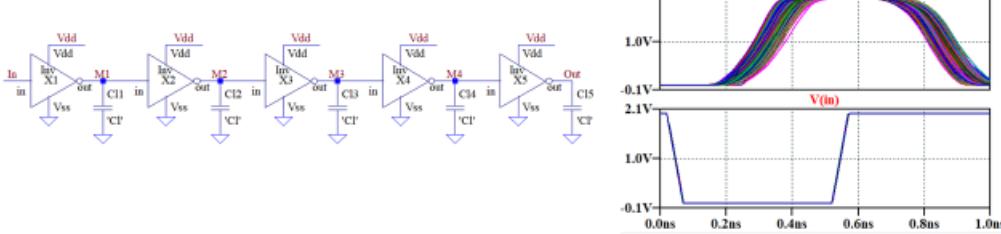
$$\begin{aligned} Slack_{Setup} &= T_{clock} + D_{early} - D_{late} - T_{setup} - Uncertainty > 0 \\ Slack_{hold} &= D_{late} - D_{capture} - T_{hold} - Uncertainty > 0 \\ D_{early,late} &= D_{max,min} - Uncertainty \end{aligned}$$

ロジックセル・ライブラリ、寄生素子とサインオフ検証



124

3.4 設計基準 5段インバータ遅延の分布

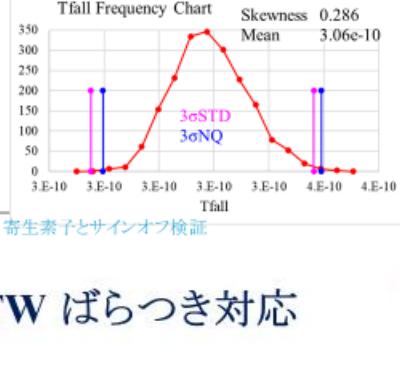
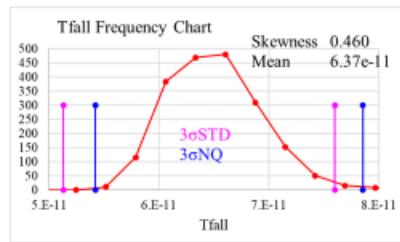
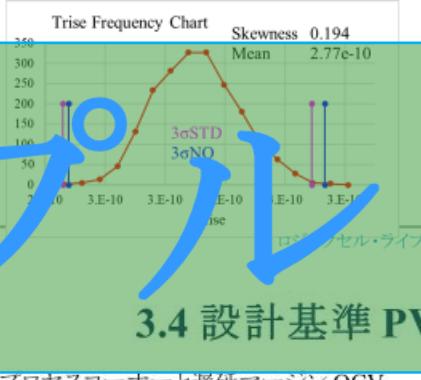
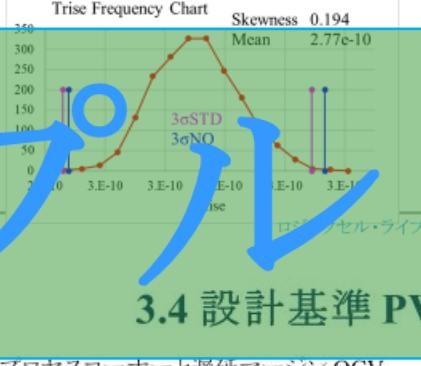
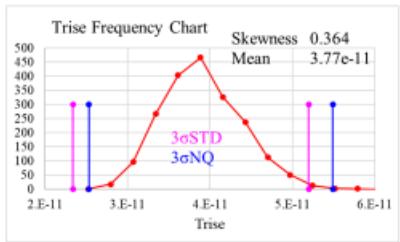


ロジックセル・ライブラリ、寄生素子とサインオフ検証

137

3.4 設計基準 $3\sigma_{STD}$ と $3\sigma_{NQ}$

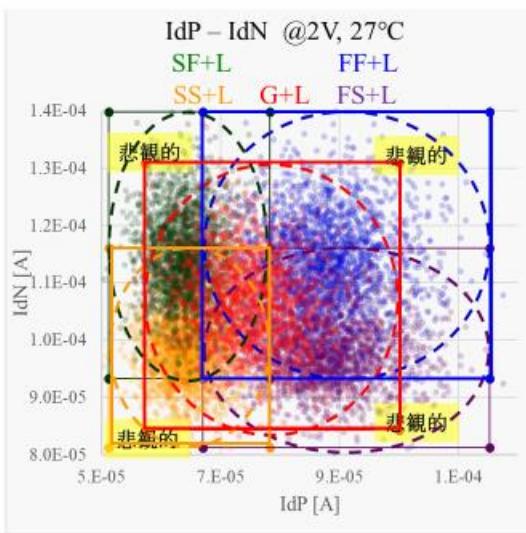
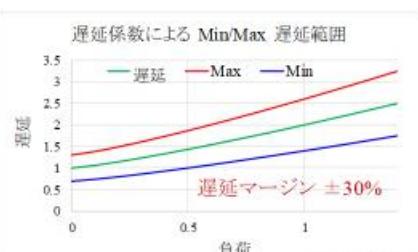
- インバータ5段のばらつきで見てみる。1段に比べると $3\sigma_{STD}$ と $3\sigma_{NQ}$ のずれが縮んで、より正規分布に近づいていることが分かる。デジタル回路で、ある程度MOSの段数があれば、ばらつきは正規分布と見做しても構わない。標準偏差 $3\sigma_{STD}$ が使える。



141

3.4 設計基準 PV TW ばらつき対応

- プロセスコーナーと遅延マージン OCV
 - ✓ MOSと配線の 3σ プロセスばらつき内での動作を保証。検証コーナーと遅延マージンで検証。
- 比較的古いやり方。
- プロセスばらつきに対して、速度電力視点から SS と FF コーナーを建てる。
- 3σ ばらつきは遅延値に Min/Max の遅延係数を掛けた上でカバーする。
- 当然角の所では過剰評価になる。



ロジックセル・ライブラリ、寄生素子とサインオフ検証

144