

卒業論文

題目

DFF 及び スキャン FF の低消費電力
化の研究

指導教員

近藤 利夫 教授

2015 年

三重大学 工学部 情報工学科
計算機アーキテクチャ研究室

宿里 美緒 (411854)

内容梗概

DFE はスタンダードセル部における占有面積，消費電力が大きく，その低電力化はチップ全体の消費電力に大きく貢献する．そのため，近年も当研究室の DSSTSPC DFE をはじめ，低電力遅延積の実現をねらった新構成の DFE がいくつも提案されている．しかし，ASIC には，テストのためにスキャン対応が必要不可欠にもかかわらず，スキャン対応構成での比較はほとんど行われていない．そこで本研究では，DSSTSPC と，その当研究室の提案以降に登場した DFE との比較評価を，スキャン対応構成も含めて行った．また，低消費電力化に効果が大きく，広く利用されている DVFS 技術への適用性を確認するために，通常電圧に加えて低い電圧での比較評価も行った．その結果，DSSTSPC を利用することで約 29% 電力削減ができるということと，複数種類の DFE の混載をすることで低電力かつ高速な回路を実現できることが分かった．

Abstract

DFFs play important roles in the CMOS circuits because the performance of DFFs significantly affects the area and the power consumption of VLSI chips. The Scan FFs are an essential part of the Scan test. However, little has reported on evaluations of Scan FFs. In this paper, DFFs and their Scan FFs are evaluated the power consumption. DSSTSPC achieves 29% less power consumption than conventional DFFs.

目次

| | | |
|-------|----------------------------|----|
| 1 | はじめに | 1 |
| 2 | DFFの概要および構成 | 2 |
| 2.1 | マスタースレーブ型 | 2 |
| 2.2 | パルス型 | 3 |
| 2.3 | 回路構成による分類 | 4 |
| 3 | シミュレーション内容について | 5 |
| 3.1 | 比較対象のDFFについて | 5 |
| 3.2 | 比較対象DFFの改良と構成トランジスタのサイズ最適化 | 9 |
| 3.3 | スキャンFFの概要 | 10 |
| 3.4 | DVSの適用性について | 12 |
| 3.5 | シミュレーション評価 | 12 |
| 3.5.1 | DFFの評価 | 12 |
| 3.5.2 | スキャンFFの評価 | 14 |
| 4 | シミュレーション結果 | 14 |
| 4.1 | DFFの消費電力評価結果 | 14 |
| 4.2 | スキャンFFの消費電力評価結果 | 17 |
| 5 | おわりに | 21 |
| | 参考文献 | 22 |
| A | ネットリストについて | 24 |
| A.1 | CLKのような波形の表現 | 24 |
| A.2 | CLKについて | 24 |
| A.3 | spiceの電流計測 | 25 |
| A.4 | 寄生容量など | 25 |
| B | HSPICEの使い方 | 27 |
| B.1 | 前提など | 27 |
| B.2 | 使い方 | 27 |

| | | |
|----------|-----------------------|-----------|
| C | 回路図エディタについて | 27 |
| C.1 | エディタの起動 | 28 |
| C.2 | ファイルの新規作成 | 28 |
| C.3 | 1. ディレクトリの作成 | 29 |
| C.4 | 2. 回路図データの作成 | 30 |
| C.5 | 3. 回路図の描画 | 31 |
| | C.5.1 PMOS と NMOS の描画 | 31 |
| | C.5.2 Vdd と Vss の描画 | 34 |
| | C.5.3 入力と出力の描画 | 35 |
| | C.5.4 ワイヤの描画 | 36 |
| | C.5.5 描画したものを消したいとき | 36 |
| C.6 | 4. ネットリストの出力 | 37 |
| D | 評価結果の算出について | 38 |
| D.1 | トグルとは | 38 |
| D.2 | 消費電力 | 39 |
| D.3 | トグル率に対する消費電力 | 39 |
| D.4 | 最低駆動電圧 | 40 |
| D.5 | 最高動作周波数 | 40 |
| D.6 | PDP | 40 |
| E | 回路の性能向上の工夫 | 41 |
| E.1 | PMOS のチャネル幅を大きくする | 41 |
| E.2 | NMOS の抵抗成分を大きくする | 42 |

目 次

| | | |
|------|--------------------------------------|----|
| 3.1 | C^2 MOSDFF の回路図 | 7 |
| 3.2 | DSSTSPCDFF の回路図 | 7 |
| 3.3 | LPMSDFF の回路図 | 8 |
| 3.4 | DETSDFF の回路図 | 8 |
| 3.5 | C^2 MOSDFF の Muxed-D 型スキャン FF | 11 |
| 3.6 | C^2 MOSDFF の clocked-scan 型スキャン FF | 11 |
| 3.7 | DSSTSPC の clocked-scan 型スキャン FF | 15 |
| 3.8 | LPMS の clocked-scan 型スキャン FF | 16 |
| 3.9 | IBM のラッチを使用した C^2 MOSDFF のスキャン FF | 16 |
| 4.10 | 各 DFF のトグル周波数 | 18 |
| 4.11 | 1.8V における DFF の実効 PDP | 18 |
| 4.12 | 1.8V における DFF の消費電力 | 19 |
| 4.13 | 1.4V における DFF の消費電力 | 19 |
| 4.14 | 1.8V におけるスキャン FF の消費電力 | 20 |
| 4.15 | 1.1V におけるスキャン FF の消費電力 | 20 |
| 3.16 | icfb のウィンドウ | 28 |
| 3.17 | New Library のウィンドウ | 30 |
| 3.18 | Create New File のウィンドウ | 31 |
| 3.19 | 回路図エディタのウィンドウ | 32 |
| 3.20 | Add Instance のウィンドウ | 33 |
| 3.21 | Library Browser のウィンドウ | 34 |
| 3.22 | PMOS の設定をしている Add Instance のウィンドウ | 43 |
| 3.23 | NMOS の設定をしている Add Instance のウィンドウ | 44 |
| 3.24 | トランジスタ設置後の回路図エディタのウィンドウ | 45 |
| 3.25 | Add Instance のウィンドウ | 46 |
| 3.26 | 電源とアース設置後の回路図エディタのウィンドウ | 47 |
| 3.27 | 入力ピンの作成 | 48 |
| 3.28 | 出力ピンの作成 | 48 |
| 3.29 | 入力と出力設置後の回路図エディタのウィンドウ | 49 |
| 3.30 | 結線後の回路図エディタのウィンドウ | 50 |
| 3.31 | Analog Design Environment ウィンドウ | 51 |
| 3.32 | Choosing Design ウィンドウ | 52 |
| 3.33 | Choosing Simulator ウィンドウ | 52 |
| 3.34 | 描画した回路図のネットリスト | 53 |

| | |
|----------------------------|----|
| 4.35 トグルしている DFF | 54 |
|----------------------------|----|

表 目 次

1 はじめに

携帯端末などの高機能化，高性能化に伴う消費電力を抑えるために多方面から LSI の低電力化がはかられている．その中で基本回路の DFF とそれへのクロック分配用信号線の消費電力は LSI 全体の 4 割 [1] も占めることから，DFF の低消費電力化は LSI の電力低減に効果的であり，種々の低電力回路が提案されてきた．当研究室でも新たな TSPC 型 DFF を提案した [2]．その際比較対象は C^2 MOSDFF と HSTSPCDFF という従来のマスタースレーブ型の DFF との比較評価を行った．近年，これまでの主流であったマスタースレーブ型に加え，パルス型の DFF が数多く提案されており，当研究室提案の DFF がいまだに優位性があるか否かの評価が必要である．

一方，ASIC の良品選別に標準的に用いられるスキャンテストのサポートに，スキャン機能付き DFF(スキャン FF) を構成する場合の優位性が示せていないだけでなく，低電力化の方法として広く利用されている電源電圧と周波数を動的に変更する DVFS への適用性を示せていない．

そこで本研究では，ベースの C^2 MOS 型に加えて最近提案された低電力 DFF との比較評価を回路構成の最適化をはかりながら DFF とスキャン FF の両方について行う．その際，DVFS の適用性もみるために，低電

圧側まで比較範囲を広げて行う。

2 DFF の概要および構成

DFF には様々な種類が存在する。まず、マスタースレーブ型とパルス型に分ける。それぞれについて、回路構成の違いによりスタティック回路とセミスタティック回路、ダイナミック回路が存在する。

2.1 マスタースレーブ型

マスタースレーブ型とは、ラッチを2つ直列につなげたものである。前段のラッチをマスター、後段のラッチをスレーブと呼ぶ。マスターは入力されたデータを保持し後段のスレーブラッチへと伝播し、スレーブはマスターの出力を保持する。マスターラッチとスレーブラッチは、一方がクロックが0の時に動作し、他方はクロックが1の時に動作する。マスターとスレーブの動作するタイミングが異なることで立上がり時または立下り時にスレーブ側への書き込みが行われる。エッジトリガで駆動するためレーシング耐性がある。ただし、2つのラッチにより駆動するため、速度性能が劣る傾向がある。

2.2 パルス型

パルス型は、微小なクロックパルスの間のみ入力を受け付けて駆動する。クロックの立上がり、立下りのそれぞれのエッジでクロックパルス生成すれば (Double Edge Triggerd:DET)，クロック周波数が半分で済み、クロックネットワークでの消費電力の半減が可能になる。さらに動作が高速で、遅延も小さくできる。しかし、レーシング耐性は低く、クロックスキュー制約が厳しくなるため、ホールドタイムが確保されない部分にはそれを補償するための遅延回路を余分に挿入しなければならない。その部分にのみ、マスタスレーブ型を用いても良いが、その分設計が複雑化する。なお、DET 型は、パルス生成部までのクロック給電配線上のクロックの周波数が半分になるため、その部分で消費される消費電力 (ASIC では web ページ [1] から全電力の 10%程度と推定される) が半分になる利点を有する。しかし、サイクルタイムのデューティ比が静的、動的に変動する可能性があり、その分、等価的にサイクルタイムが伸びることとなり、遅延の小ささが打ち消される。従って、消費電力より速度を優先する場合には適用困難である。

2.3 回路構成による分類

本節では、回路構成による分類について述べる。保持機能によって3種類に分類することができる。保持機能をマスター、スレーブの両方にもつものをスタティック回路、一方のみにもつセミスタティック回路、全くもたないダイナミック回路が存在する。ダイナミック回路は速度や消費電力に優れるが、定常的な保持機能を持たないためクロックを停止させた場合などにデータが消失する問題がある。スタティック回路は定常的な保持機能を持つが、遅延や消費電力がダイナミック回路よりも生じる。セミスタティック回路は両者の中間的な性能ながらも値の保持が可能である。また、クロックの使用方法によってもクロックの反転と非反転を利用する一般的な方式と、TSPC回路のようにクロックの反転、非反転のいずれか一方のみを利用する方式に分類できる。TSPC回路は、クロックで駆動されるノード数が少なくできるため、クロック系の消費電力を削減することができる。

3 シミュレーション内容について

3.1 比較対象の DFF について

本研究で対象とする DFF は C^2 MOSDFF , DSSTSPCDFF , LPMSDFF , DETSFF の 4 種類である .

図 3.1 に C^2 MOSDFF の回路図を示す . C^2 MOSDFF は TGFF と呼ばれ , 消費電力の低さとレーシング耐性の高さを両立していることから , 現在まで最も広く利用されてきたマスタースレーブ型の DFF である . 比較対象としては他の DFF の構成に合わせ , セミスタティック型の構成を採った .

DSSTSPCDFF[2] は当研究室で提案した DFF で , 図 3.2 に回路図を示す . セミスタティック型の TSPC マスターラッチ 2 個回路で , 差動入力 of スタティックスレーブラッチをプッシュプル駆動することで書き込み速度の高速化を図る TSPCDFF である .

図 3.3 に LPMSDFF[3] の回路図を示す . LPMS はセミスタティック型のマスタースレーブ型 DFF で , クロックの立下り時に駆動するネガティブエッジトリガである . フィードバックに PMOS が組み込まれており , CLK が 1 の間のみフィードバックが作動する . そうすることで CLK が 0 の間の電力を削減する .

図 3.4 に DETSFF[4] の回路図を示す。DETS はセミスタティック型で、今回唯一のパルス型である。クロックの立ち上がり時と立下り時に駆動するデュアルエッジトリガであることで実質 2 倍の動作速度となる。マスター側のインバータの前後に CLK と CLKN でスイッチングされる NMOS を配置し、マスター側のインバータの寄生容量を一時的なストレージとして利用することでパルス FF として動作する。

LPMS と DETS は共にフィードバックを駆動能力の小さい PMOS トランジスタのみで行うようにして書き込み時の遅延を低減すると共に貫通電流による電力消費を抑えている。しかし、フィードバックのローレベルが PMOS トランジスタの閾値電圧分上昇しうるため、サブスレッシュホールドリーク電流が大幅に増加する欠点がある。

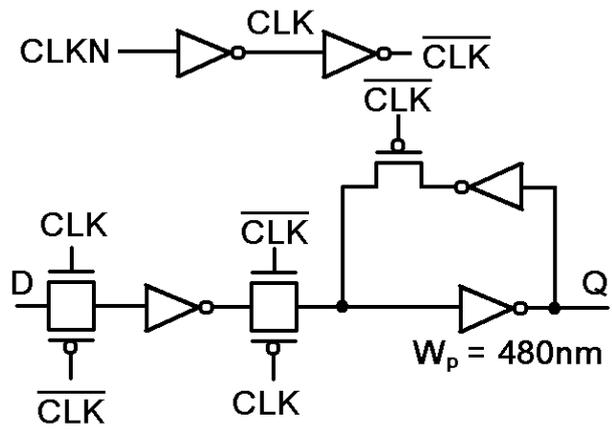


図 3.3: LPM SDF の回路図

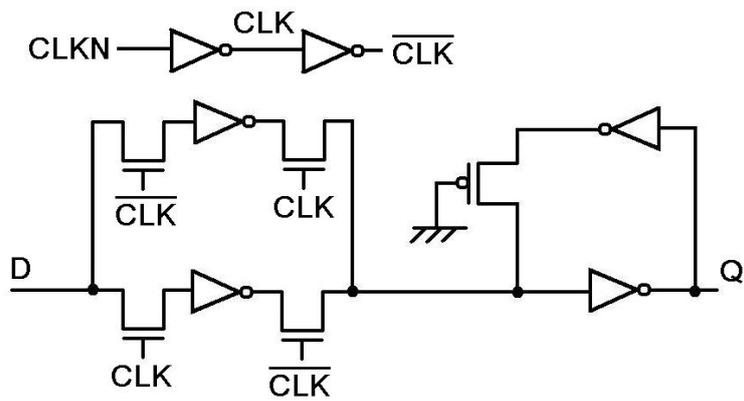


図 3.4: DET SDF の回路図

3.2 比較対象 DFF の改良と構成トランジスタのサイズ最適化

トランジスタのチャンネルの長さとは最小幅は Rohm のデザインルールに従い、それぞれ 180nm と 440nm とした。チャンネル幅は個々のトランジスタで、最低駆動電圧と消費電力が最小となる値にした。一般に、PMOS は NMOS の約 2 倍の抵抗値を持つため、PMOS のチャンネル幅は NMOS の 2 倍に設定することで遅延を低減でき、結果として電力遅延積の削減につながる。しかし、単純に全ての PMOS のサイズを大きくしたところ、それによって得られる遅延低減効果以上にチャンネル幅の拡大による消費電力が増加してしまう結果となった。そこで、大きくするのを一部の PMOS に限定すると共に、その大きくするサイズも調整することとした。具体的には、 C^2MOS と DETS では PMOS と NMOS のサイズを共に 440nm とした。DSSTSPC では、図 3.2 の $W_p = 586nm$ と書かれているインバータの PMOS のみを 586nm にする以外は、PMOS、NMOS 共に 440nm のままとした。LPMS では、図 3.3 の $W_p = 480nm$ と書かれているインバータの PMOS のみ 480nm にした。

3.3 スキャン FF の概要

スキャン FF とは、通常の DFF の動作に加え、スキャン信号によって DFF のデータの書き込みと読み出しを可能にしたスキャンテスト機能付き DFF である。スキャンテストとは、組み合わせ回路の入力と出力をスキャン FF に書き込み、外部の I/O 端子からスキャン FF の制御と観測を行うことで設計の正しさを確認するものである。スキャン FF には、マルチプレクサを利用した Muxed-D 型と、スキャン用のラッチを DFF に組み込む Clocked-scan 型の 2 種類が存在する。図 3.5 に Muxed-D 型を適用した C^2 MOS のスキャン FF を示す。DFF の前にマルチプレクサを接続し、Scan 信号によってスキャンの入力 S_{in} とデータの入力 D_{in} を使い分ける。設計が容易であるものの、スキャンをしない場合でも入力がマルチプレクサを経由するため、性能が下がる欠点がある。

図 3.6 に Clocked-scan 型を適用した C^2 MOS のスキャン FF を示す。これはスキャン用のクロック SCK0 と SCK1 で制御されるスキャン用ラッチを DFF に付加した形になっている。Muxed-D 型と比較して、スキャンをしない場合は通常の DFF と同じパスを利用するため性能が下がり難い。一方で、スキャンクロックをクロックとは別で配る必要があり設計が複雑になる。

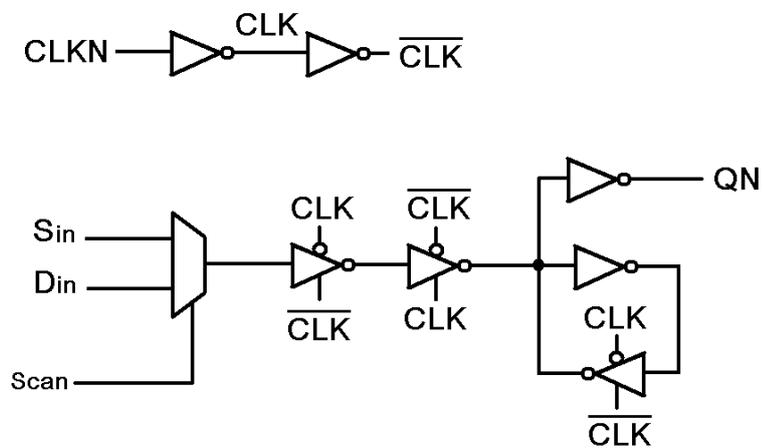


図 3.5: C^2 MOSDFF の Muxed-D 型スキャン FF

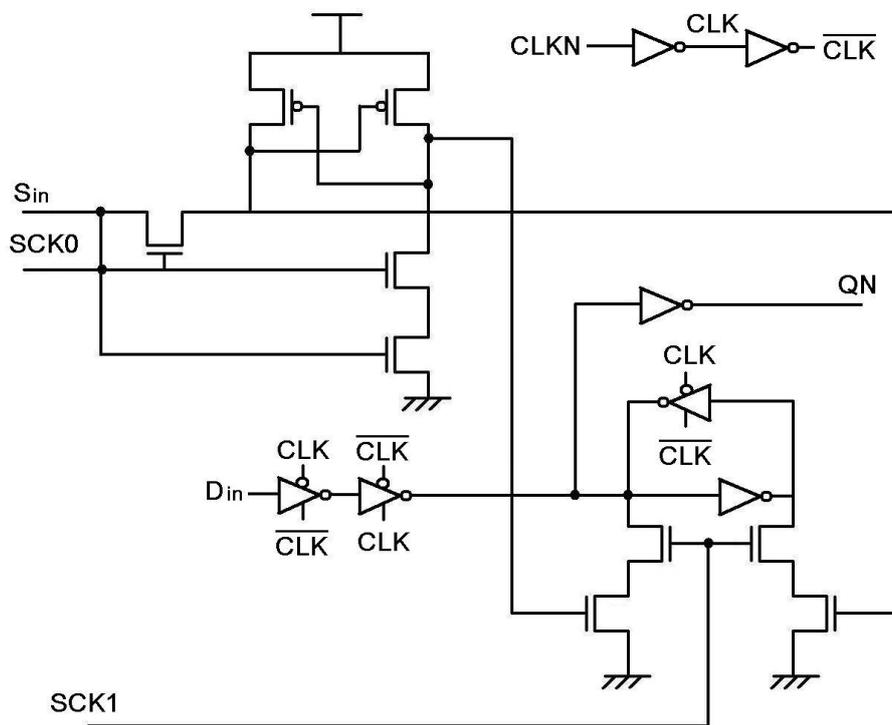


図 3.6: C^2 MOSDFF の clocked-scan 型スキャン FF

3.4 DVS の適用性について

DVS(Dynamic Voltage Scaling) 技術は、性能が必要とされない期間に LSI を低電圧動作させることによって、その期間の消費電力を削減する手法である [5]。電圧を下げても、ある程度の速度は必要であり、かつ低消費電力である必要がある。そこで本研究では速度と低電力の両立を示す電力と遅延の積 (電力遅延積 PDP) で適用性を判断した。ここで、PDP は値が小さいほど高速で低電力であることを示す。

3.5 シミュレーション評価

DFF の性能比較を、 C^2 MOS、DSSTSPC、LPMS、DETS の 4 種類について行った。それぞれ、回路図を SPICE のネットリストで記述し、Synopsys HSPICE を用いて回路シミュレーションをした。また、それぞれの DFF を元にスキャン FF を設計し、DFF と同様に計測した。計測は、DFF およびスキャン FF に加えて、クロック波形形成用のインバータを含めた回路全体で行った。

3.5.1 DFF の評価

電圧や出力の遷移率 (トグル率) に応じて消費電力が変化するため、DFF の低電力性能の優劣は典型的条件におけるトグル率 100%での消費電

力だけでは判断できない。

また、トグル率の違いにより消費電力の小さい DFF が入れ替わるならば、トグル率に応じて消費電力最小の DFF を用いることにより、LSI 全体の消費電力低減が見込める。これらより、比較はティピカル条件だけでなく様々な電圧やトグル率で行う必要のあることが分かる。

ここではじめに動作速度を評価するため電源電圧 1.8V から 1.0V まで 0.2V 刻みの電圧に対してトグル周波数を計測した。DVS 技術の適用性を判断するために、1.8V の電圧でクロック周波数 2GHz の条件と、1.4V の電圧でクロック周波数 1GHz の条件でのトグル時の消費電力を計測した。ただし、DETS についてはデュアルエッジトリガで実質動作速度が 2 倍であり、トグル回数を他の DFF と同じにするために他の DFF の半分のクロック周波数で計測した。

また、1.8V と 1.4V の電圧での消費電力とトグル周波数の結果から、実効 PDP を算出した。消費電力は、トグル時の消費電力とトグルさせない場合のクロック系の消費電力を別々に計測し、それからトグル率と DFF 全体の消費電力の関係を求めた。

3.5.2 スキャン FF の評価

比較対象の 4 種類の DFF それぞれについて Clocked-scan 型のスキャン FF を設計し、電源電圧 1.8V と 1.1V の実効 PDP と、1.8V クロック周波数 2GHz、1.1V クロック周波数 1GHz のそれぞれの条件での消費電力を計測した。ただし、スキャン用のラッチの接続による駆動能力低下が見られたため、すべてのトランジスタのチャネル幅を 3.2 で示した大きさの 2 倍に広げている。図 3.7 に DSSTSPC のスキャン FF、図 3.8 に LPMS のスキャン FF の回路図を示す。C²MOS に関しては、スキャン用のラッチとの相性が悪く、最低駆動電圧が 1.92V と高くなってしまったため、図 3.9 に示すように一般によく利用される IBM のスキャン機能付きラッチを使用した。DETS の最低駆動電圧が 1.66V と高かったため、DVS の適用性が低いと判断し、1.1V での計測は、DETS を除いた 3 種類に対して行った。

4 シミュレーション結果

4.1 DFF の消費電力評価結果

図 4.10 に電圧ごとの各 DFF のトグル周波数を示す。DETS については、電源電圧 1.4V 未満では動作しなかった。すべての電圧に対して、LPMS

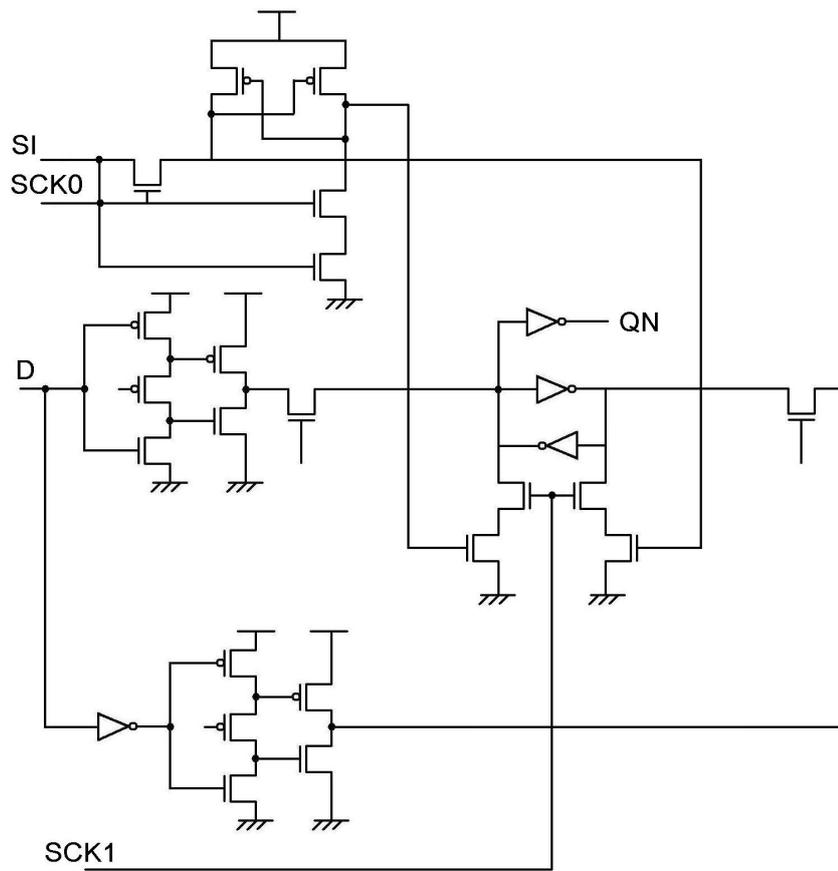


図 3.7: DSSTSPC の clocked-scan 型スキャン FF

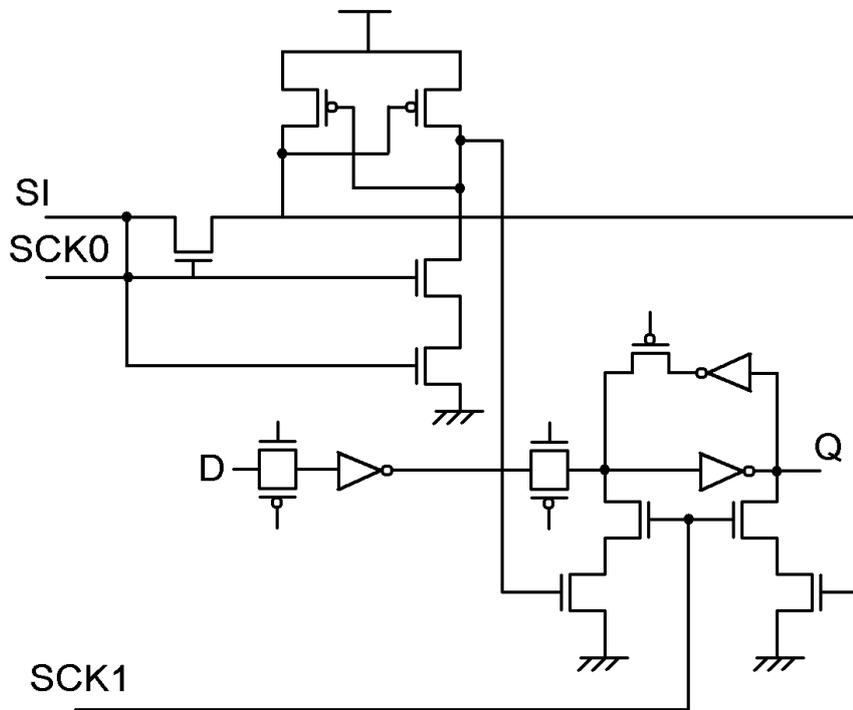


図 3.8: LPMS の clocked-scan 型スキャン FF

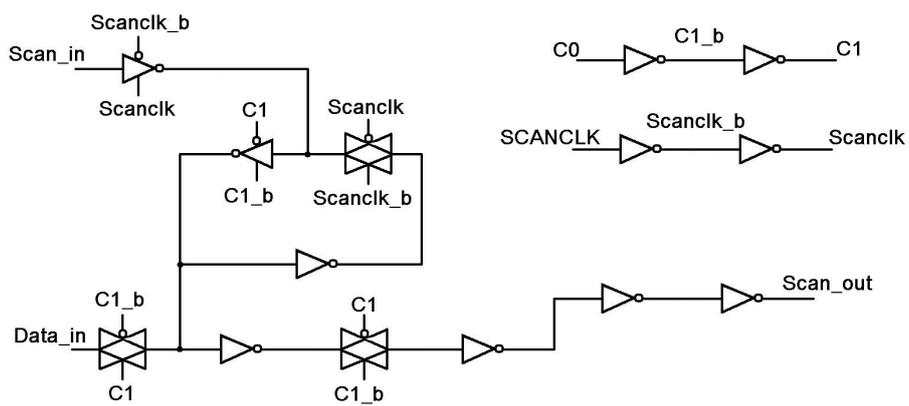


図 3.9: IBM のラッチを使用した C^2 MOSDFF のスキャン FF

が最もトグル周波数が高い。この結果から、クリティカルパスのような速度を重視するところでは高速な LPMS を使用するとよいと考えられる。

図 4.11 に 1.8V における DFF の実効 PDP を示す。電圧や DFF とスキャン FF の種類に変化があっても全てについて実効 PDP は同様の傾向が見られたため、他の条件の計測結果は割愛している。DETS はトグル率 5%以上で 4 種類の中で最も実効 PDP が高い。トグル率 40%未満では DSSTSPC、トグル率 40%以上では LPMS の実効 PDP が最も低いことが分かった。一般に入力データの遷移率は 12.5%と低い[6]ため、DSSTSPC が最も DVS への適用性が高いと考えられる。

図 4.12 に 1.8V における消費電力、図 4.13 に 1.4V における消費電力を示す。1.8V の全てのトグル率において、DSSTSPC は最も消費電力が低い。1.4V においてはトグル率 70%以下で DETS の消費電力は最も低い。実効 PDP が著しく大きいため DETS は DVS の適用性が低いと考えられる。

4.2 スキャン FF の消費電力評価結果

図 4.14 に 1.8V におけるスキャン FF の消費電力を示す。DETS は全てのトグル率で最も消費電力が大きく、トグル率 100%の時、他の DFF の

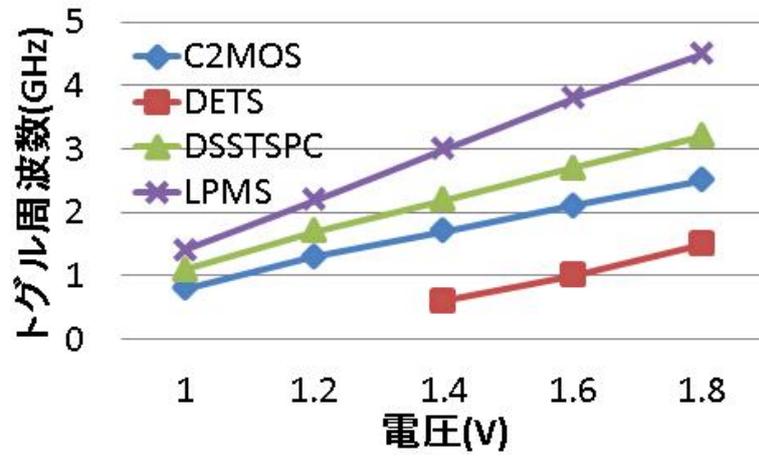


図 4.10: 各 DFF のトグル周波数

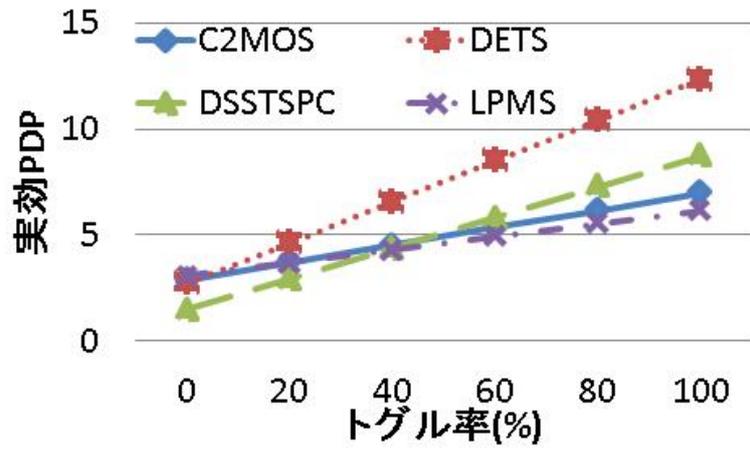


図 4.11: 1.8V における DFF の実効 PDP

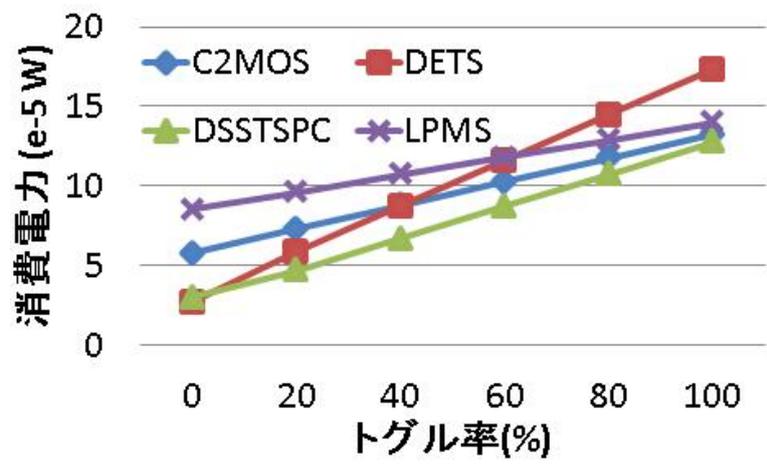


図 4.12: 1.8V における DFF の消費電力

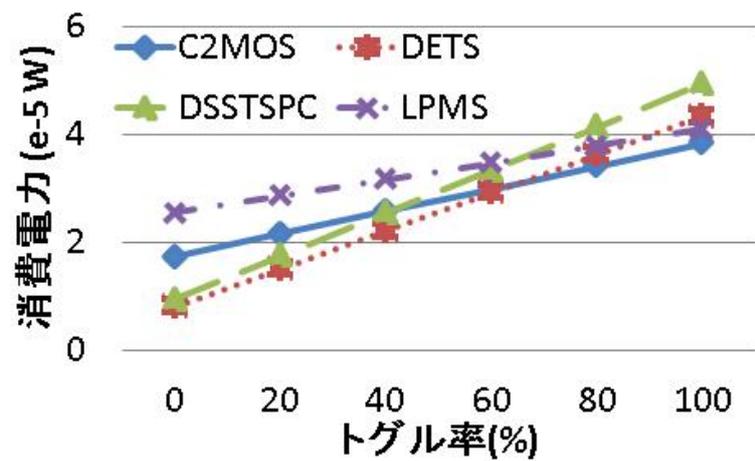


図 4.13: 1.4V における DFF の消費電力

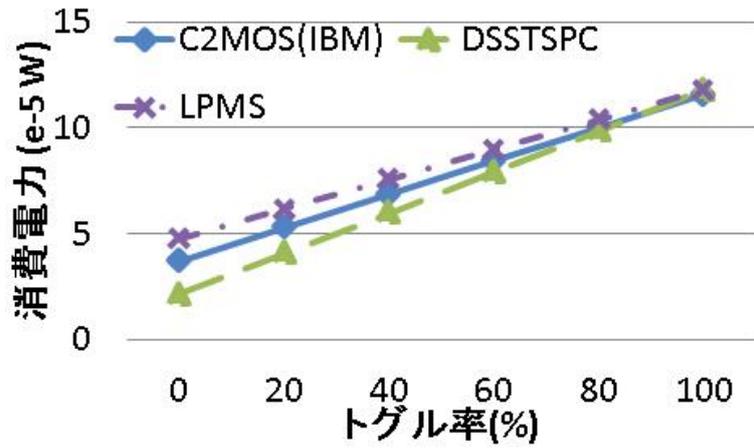


図 4.14: 1.8V におけるスキャン FF の消費電力

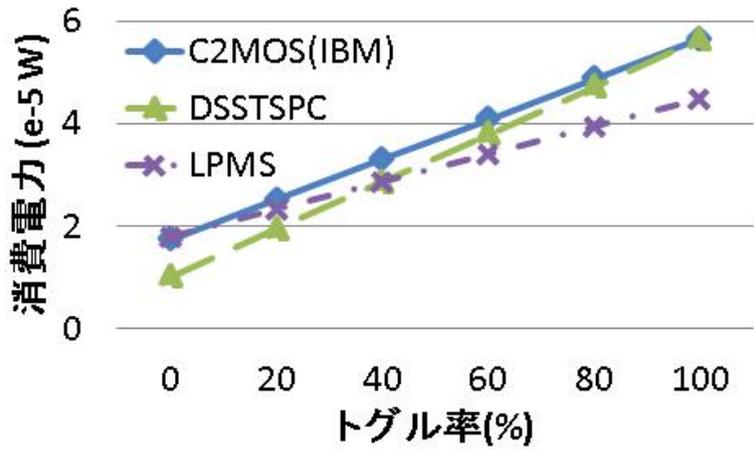


図 4.15: 1.1V におけるスキャン FF の消費電力

約 3.7 倍の値であり，グラフの見やすさのため省略した．トグル率 90%未満では DSSTSPC が最も消費電力が低い．

図 4.15 に 1.1V におけるスキャン FF の消費電力を示す．トグル率 45%未満では DSSTSPC ，トグル率 45%以上では LPMS の最も消費電力が低い．電源電圧 1.1V ，クロック周波数 1GHz でトグル率が 12.5%の時， C^2 MOS を TSPC に置き換えると約 29%の電力削減が可能である．またトグル率が 100%の時， C^2 MOS を DSSTSPC に置き換えてもほとんど電力が削減できないが， C^2 MOS を LPMS に置き換えると約 21%の電力削減が可能である．

5 おわりに

DFE とスキャン FF の計測より，トグル率 40%未満の大半の DFE の動作条件では DVS を適用したとしても DSSTSPC が最も低消費電力であることを明らかにできた．

トグル率が高い場合は LPMS ，トグル率が低い場合は DSSTSPC が消費電力が低いということが分かった．この結果から，クリティカルパスで高速動作が必要不可欠な箇所には LPMS ，それ以外には DSSTSPC を搭載することでより低消費電力と高速動作が両立できると考えられる．今

後は、LPMSのような高速 DFF と混載をする場合に使い分けの比率がどの程度で、低消費電力化の効果がどれだけ得られるかを明らかにしていく必要がある。

参考文献

- [1] 牧井 徹, “クールダウン！ EDI System を用いた最新 Low Power 技術でチップを低消費電力化！”, 日本ケイデンス・デザイン・システムズ, <https://www.cadence.co.jp/soconline/vol7/tec1/all.html>, 2012.
- [2] T. Nakabayashi, *et al.*, “Low power semi-static TSPC D-FFs using split-output latch,” IEEE Proc. ISOCC., pp.17–18, Nov. 2011.
- [3] I. A. Khan, *et al.*, “Design and Analysis of Low Power Master Slave Flip-Flops,” Vol.43, No.1, pp.41–49, 2013.
- [4] G. Singh, *et al.*, “High Performance Low Power Dual Edge Triggered Static D Flip-Flop,”
- [5] Pouwelse, J., *et al.*, “Dynamic voltage scaling on a low-power,” microprocessor, ACM Int. Conf. on Mobile Computing and Networking, 2001.

- [6] XPower Estimator User Guide, XILINX 社, UG440 (v13.1), pp.11,
March 2011.

A ネットリストについて

A.1 CLK のような波形の表現

V[信号線名 1] [信号線名 1] [信号線名 2] PULSE ([電圧 A の値] [電圧 B の値] [開始時間] 1ps 1ps [電圧 A が持続する時間] [周期])

と書けばよい。

例えば、ネットリスト中で CK という名前で、最低電圧が 0V、最高電圧が 1.8V、開始時刻が 3ns、周波数が 2GHz のクロックを作るとする。

周波数から周期を計算する。

$$\text{周期} = \frac{1}{2\text{GHz}} = 0.5\text{ns}$$

周期の半分が一方の電圧が占める時間なので、

$$\text{電圧 A が持続する時間} = \frac{\text{周期}}{2} = 0.25\text{ns}$$

よって

VCK CK VSS PULSE (0v 1.8v 3ns 1ps 1ps 0.25ns 0.5ns)

と記述すればよい。(VSS はネットリスト中に.global VDD VSS で定義したとする)

A.2 CLK について

ネットリストで

VCK CK VSS PULSE (0v 1.8v 2ns 1ps 1ps 0.25ns 0.5ns)

と記述すれば遅延がなく綺麗な CLK の波形が得られるが、実機では綺麗な波形の CLK は供給されない。そこで、今回は各回路図に示したように、CLK とその反転は一度以上インバータを經由して各トランジスタへ供給される形をとった。そうすることで、遅延が加わったり、波形の角が丸くなるなどの実際の CLK の波形に近づけることができる。

A.3 spice の電流計測

ネットリストに

```
.meas tran i_vdd avg i(vdd)
```

の一行を追加することで、HSPICE でシミュレーションをすると回路全体の電流が mt0 ファイルに出力される。

本研究での消費電力は、ネットリストで定義した電源電圧に、mt0 ファイルに出力された電流をかけることで算出している。

A.4 寄生容量など

寄生容量を含めてシミュレーションする場合は、トランジスタのチャンネル長 L やチャンネル幅 W に加えて、 PS (ソース周囲長)、 PD (ドレイン周囲

長), AS (ソースの拡散面積), AD (ドレインの拡散面積) を定義する必要がある。 L はチャンネル長, W はチャンネル幅で, a はソースまたはドレインの長さ, b はソースまたはドレインの幅とする。

今回, a の値はデザインルールに従って 480nm とした。また, 今回はソースとドレインの大きさは同じである。従って PS , PD , AS , AD は以下の式で算出できる。

- $a = 480$
- $b = W$
- $PD = PS$
- $AD = AS$
- $PD = 2 * (a + b) = 2 * (480 + W)$
- $AD = a * b = 480W$

算出した PS , PD , AS , AD は以下のようにネットリストで記述すればよい。

```
M01 OUT IN VSS VSS NMOS L=180e-9 W=440e-9 AD=211e-15
AS=211e-15 PD=184e-8 PS=184e-8
```

B HSPICE の使い方

B.1 前提など

- ネットリスト (拡張子が.cirのもの) を作る。(仮に netlist.cir とする)
- EDA/CAD は moule0-2 か , hp2-6, scylla0-3 上で動かす .

B.2 使い方

端末上で

1. ssh -X moule0 (サーバーに入る)
2. hspice netlist.cir (ネットリストをシミュレーションし , さまざまな
ファイルを出力)
3. simvision netlist.tr0 (ネットリストの波形図を見ることができる . 波
形図を見ない場合は省略可)

を上から順に入力して実行する .

moule0 , netlist は適宜変更する .

C 回路図エディタについて

今回は使用しなかったが , 回路図エディタを使用して回路図を描くと ,
その回路図のネットリストを出力させることができる .

C.1 エディタの起動

CAD用のサーバ (moule, scylla など) に入った後, 回路図のデータの入ったディレクトリを作成したいディレクトリまで移動する. そして端末上で

```
icfb &
```

と入力すると, 回路図エディタが起動する. 回路図エディタを起動すると, ウィンドウが2つ出てくる. 以下では図 3.16 に示す icfb - Log (略) と書いてある方のウィンドウ (What's New でない方) を操作していく.



図 3.16: icfb のウィンドウ

C.2 ファイルの新規作成

ファイルの新規作成をする手順を示す.

1. 回路図のデータを格納するディレクトリを icfb 上で作成
2. 回路図データを作成

3. 回路図を描画

4. ネットリストの出力

今回の例ではインバータを回路図で描画してから，ネットリストへ出力する．

C.3 1. ディレクトリの作成

図 3.16 のウィンドウ上で

1. ツールバーの”File” をクリック
2. でてくるメニューの”New” をクリック
3. その中の”Library” をクリック

この手順を終えると，図 3.17 に示す New Library のウィンドウが表示される．

Name には作成する回路図のデータを格納するディレクトリの名前を入力する．(回路図のデータの名前ではないので注意) 図 3.17 では DataDirectory という名前にしている．名前を入力したら，左上の”OK” をクリックする．その後，新しいウィンドウ (Load Technology File) が出るが消す．



図 3.17: New Library のウィンドウ

C.4 2. 回路図データの作成

図 3.16 のウィンドウに戻り、

1. ツールバーの”File” をクリック
2. でてくるメニューの”New” をクリック

すると、図 3.18 に示すウィンドウが開く。

Library Name は先ほど作成したディレクトリを選択し、Cell Name は自由につける。図 3.18 では INV という名前にしている。左上の”OK” をクリックすると、図 3.19 に示す回路図エディタのウィンドウが出る。

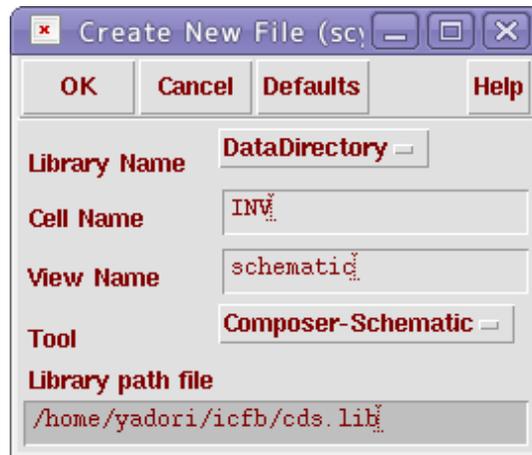


図 3.18: Create New File のウィンドウ

C.5 3. 回路図の描画

図 3.19 のウィンドウ上で回路図を描画していく。

C.5.1 PMOS と NMOS の描画

1. ツールバーの”Add” をクリック
2. “Instance” をクリック

すると、図 3.20 に示すウィンドウが開く。

Browse ボタンを押すと図 3.21 に示すウィンドウが開く。ここで、

Library は analoglib

Cell は pmos4 または nmos4

View は symbol

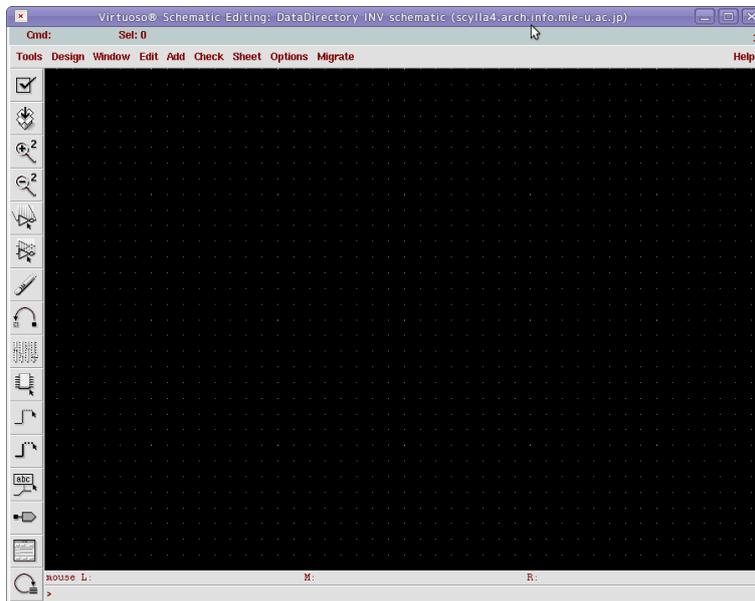


図 3.19: 回路図エディタのウィンドウ

を選択する。Cell では、pmos4 と nmos4 以外に pmos と nmos があるが、どちらも 4 のつく方を選択する。

Cell を選択すると図 3.22 に示すウィンドウが開く。このウィンドウでは PMOS や NMOS のチャンネル長やチャンネル幅、各モデルの名前を決定する。

Model name は PMOS または NMOS

Width はチャンネル幅

Length はチャンネル長

を入力する。図 3.22 では、チャンネル幅を 1.0u、チャンネル長を 0.18u と入

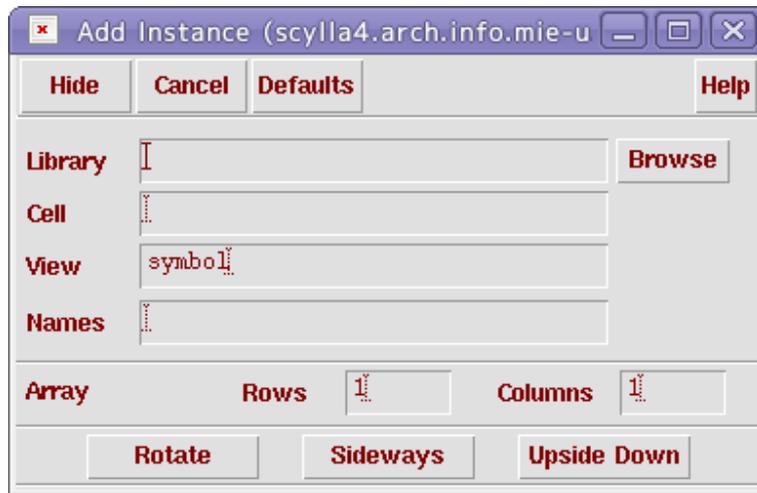


図 3.20: Add Instance のウィンドウ

力した。(自動でそれぞれ 1.0u M と 180.00n M に変換される)

それに加えて Names に P1 と入力している。以上を入力した後は、Hide ボタンを押すか、ウィンドウを横に追いやればよい。

図 3.19 のウィンドウに戻ると、マウスカーソルに合わせて黄色のトランジスタがついてくる。クリックすると先ほど入力した情報の PMOS や NMOS が描画される。

同様にして、Add-Instance を選択すると、再度図 3.22 のウィンドウが開く。連続して PMOS や NMOS を設置したい場合は、また Model name などを入力し直せばよい。PMOS を設置した後 NMOS を設置するなど、別の種類を設置したい場合は Browse ボタンを押して、Library や Cell や View を選択し直せばよい。

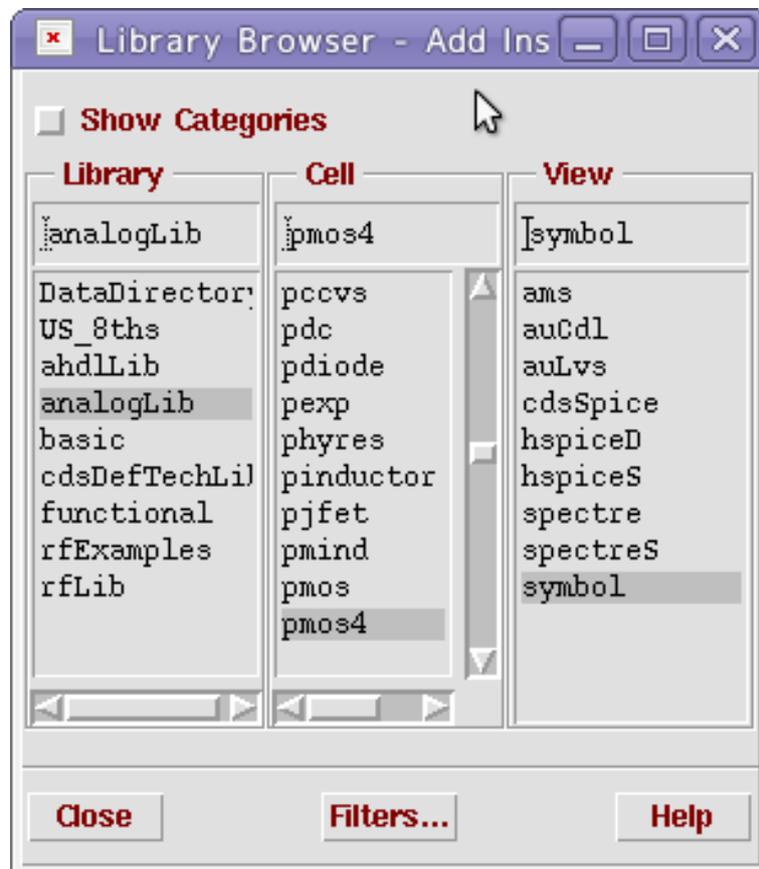


図 3.21: Library Browser のウィンドウ

図 3.22 と図 3.23 で設定した PMOS と NMOS をエディタ上に設置した様子を図 3.24 に示す。

C.5.2 Vdd と Vss の描画

1. ツールバーの”Add” をクリック
2. “Instance” をクリック

3. 出てきたウィンドウで”Browse” をクリック

この手順を終えるとウィンドウ 3.25 が開く .

Library は basic

Cell は vdd または vss

View は symbol

を選択することで , Vdd または Vss を設置することができる .

Vdd と Vss をエディタ上に設置した状態を図 3.26 に示す .

C.5.3 入力と出力の描画

1. ツールバーの”Add” をクリック

2. “Pin...” をクリック

すると図 3.27 や図 3.28 に示すウィンドウが開く .

Pin Names には入力や出力の名前を入力する .

Direction は入力なら input , 出力なら output を選択する .

入力と選択が終わったあとは Hide ボタンを押すかウィンドウを横に追い

やればよい . 回路図エディタ上でクリックすれば , 設定した内容の入力

ピンまたは出力ピンが描画できる .

IN と OUT をエディタ上に設置した状態を図 3.29 に示す .

C.5.4 ワイヤの描画

1. ツールバーの”Add” をクリック
2. “Wire(narrow)” をクリック

すると、ワイヤ描画モードになり、トランジスタなどの赤い四角を結線することができる。

赤い四角の上でクリックすると、別の赤い四角の上でクリックをするまではワイヤが描画される。赤い四角の上でクリックした後、赤い四角の上以外でクリックすると、その点から方向転換ができる。

赤い四角の上以外でクリックした場合は、ダブルクリックをすることでワイヤの描画を終了することができる。

結線した状態を図 3.30 に示す。この例ではこれでインバータの回路図が完成した。

C.5.5 描画したものを消したいとき

1. ツールバーの”Edit” をクリック
2. “Delete” をクリック

するとデリートモードになる．消したいものの上にカーソルを重ねると，黄色の点線で囲まれるので，クリックすると黄色の点線で囲まれたものが消される．

C.6 4. ネットリストの出力

回路図が完成したら，エディタのウィンドウ 3.30 で

1. ツールバーの”Tools” をクリック
2. “Analog Environment” をクリック

すると，Analog Design Environment ウィンドウ 3.31 が開く．

そのウィンドウで

1. ツールバーの”Setup” をクリック
2. “Design” をクリック

すると，新しいウィンドウ 3.32 が開くので，Library Name を最初に作成したディレクトリ名，Cell Name を今回作った回路図のデータ名にする．その後 OK ボタンを押す．図 3.32 では，Library Name は DataDirectory，Cell Name は INV になっている．

次に

1. ツールバーの”Setup” をクリック
2. “Simulator/...(作成したディレクトリ名)” をクリック

するとウィンドウ 3.33 が開く . Simulator の部分を hspiceS に選択したら , OK ボタンを押す . 解析をするのでやや時間がかかる .

最後に

1. ツールバーの”Simulation” をクリック
2. “Netlist” をクリック
3. “Create Raw” をクリック

すると , ネットリストが 2 種類出力される . そのうちの一方 3.34 が描画した回路図のネットリストである . この内容をサブサーキットとして利用することでネットリストとして使えるようになる .

D 評価結果の算出について

D.1 トグルとは

まず , 頻繁に出てくる”トグル” という単語について説明する . 図 4.35 にトグルしている DFF を示す . DFF の出力を反転させたものを入力と

して利用することで、DFE の出力が切り替わるごとに毎回値が反転する。

このことをトグルと呼ぶ。

D.2 消費電力

消費電力は消費電力を P 、電圧を V 、電流を I とすると

$$P = VI$$

の式で表される。今回はネットリストで定義した電源電圧と、mt0 ファイルに出力された電流を使って算出をした。

D.3 トグル率に対する消費電力

1. 毎回入力に変化する場合 (トグル時の消費電力)
2. 入力を一切与えない場合 (CLK 系での消費電力)

の 2 種類について消費電力の計測と算出をする。

トグル率に対する消費電力は、

$$(\text{トグル時の消費電力}) * (\text{トグル率}) + (\text{CLK 系での消費電力}) * (1 - (\text{トグル率}))$$

で算出することができる。

D.4 最低駆動電圧

最低駆動電圧は，トグルさせた DFF の電圧を徐々に下げていき，毎回値が反転する最低限の電圧のことである．ネットリストで定義した電圧の値を変化させることで電圧の上げ下げが可能．クロック周波数ごとに最低駆動電圧は変わってくるため，クロック周波数は固定する必要がある．

D.5 最高動作周波数

最高動作周波数は，トグルさせた DFF のクロック周波数を徐々に上げていき，毎回値が反転する上限のクロック周波数のことである．

```
VCK CK VSS PULSE (0v 1.8v 2ns 1ps 1ps 0.25ns 0.5ns)
```

ネットリストで定義したクロックの周期 (上の例の 0.25ns と 0.5ns) を変更することで，クロック周波数の上げ下げが可能．

最高動作周波数は，電源電圧ごとに変化するため，電源電圧を固定する必要がある．

D.6 PDP

PDP とは，電力遅延積のこと．消費電力を P，遅延を T とすると，

$$PDP = P * T$$

で表される．

今回は，遅延によって最高動作周波数は決定するため，

$$PDP = \frac{P}{(\text{最高動作周波数})}$$

で算出した．

E 回路の性能向上の工夫

PMOS は NMOS に比べて約 2 倍の抵抗値を持つ．そこで，PMOS または NMOS に工夫をすることで回路の速度向上，電力遅延積の改善を図ることができる．

E.1 PMOS のチャンネル幅を大きくする

PMOS のチャンネル幅を大きくすることで，NMOS よりも大きな抵抗値を補償する．今回は，回路全ての PMOS を単純に大きくした時，チャンネル幅の拡大による電力遅延積の改善よりも電力増加の影響が上回ってしまった．そこで一部の PMOS，特にインバータの PMOS に限定してチャンネル幅を大きくすると効果があった．

E.2 NMOS の抵抗成分を大きくする

NMOS とアースの間に、チャンネル長が大きく、チャンネル幅を小さくした抵抗代わりの NMOS を接続することで、NMOS の抵抗成分を大きくし、PMOS との抵抗値に近づける。フィードバックの NMOS に限定して抵抗代わりの NMOS を接続したところ、最低駆動電圧が低下し、消費電力の低下が見られた。

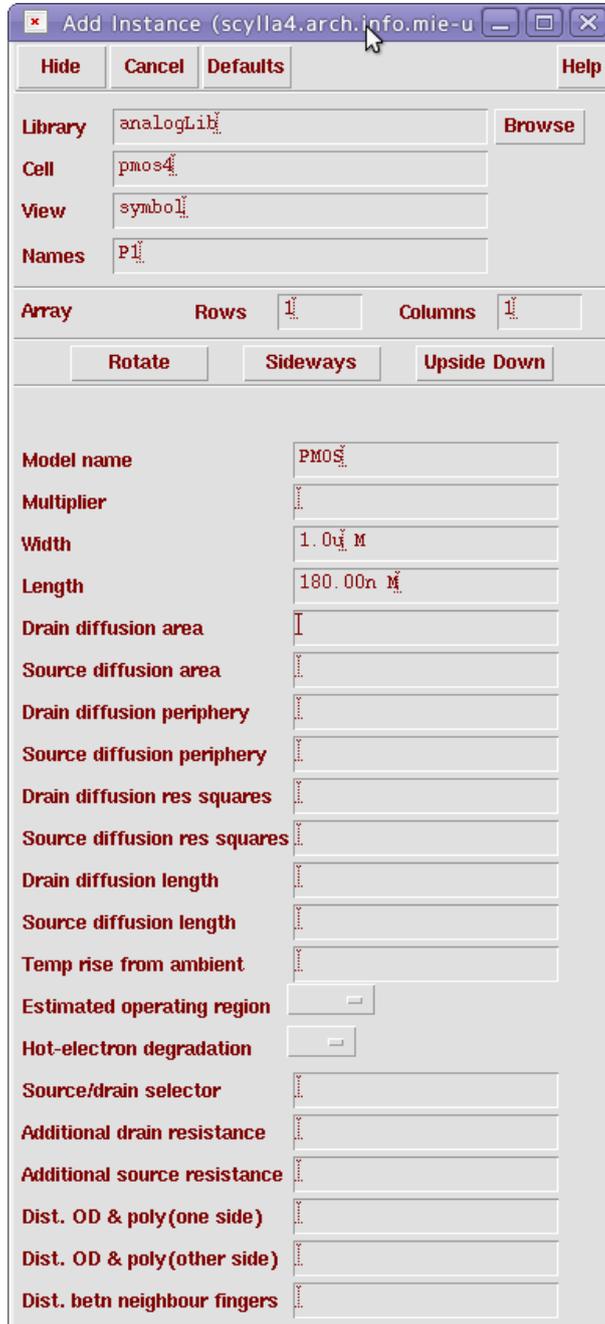


図 3.22: PMOS の設定をしている Add Instance のウィンドウ

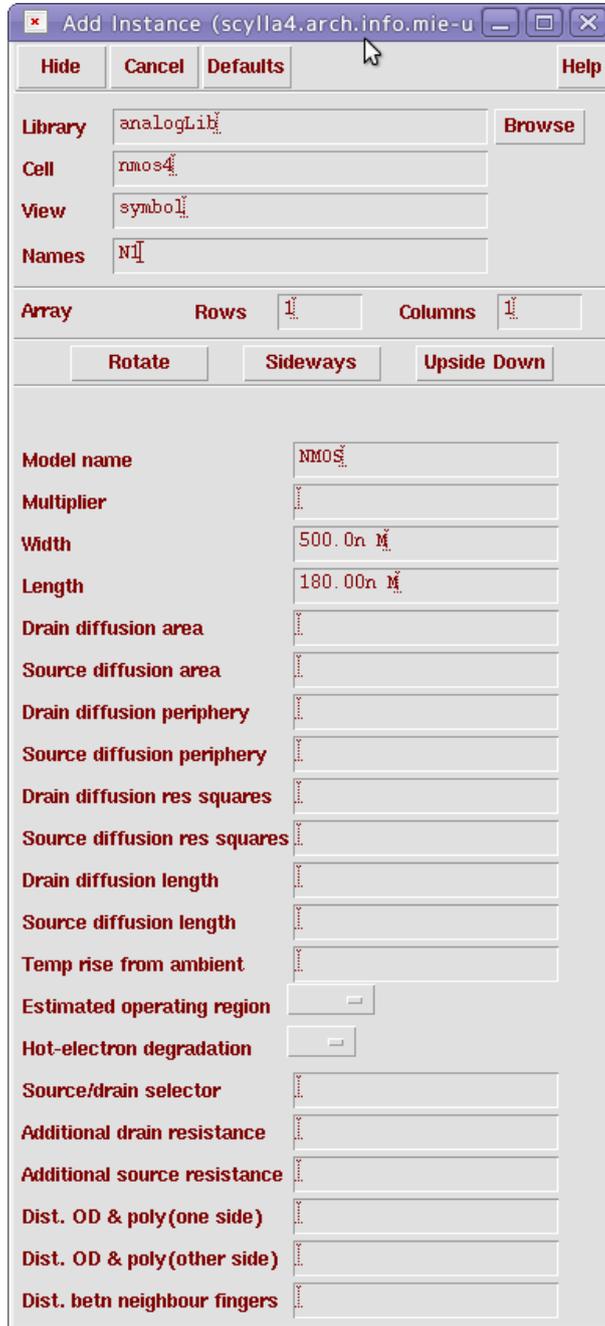


図 3.23: NMOS の設定をしている Add Instance のウィンドウ

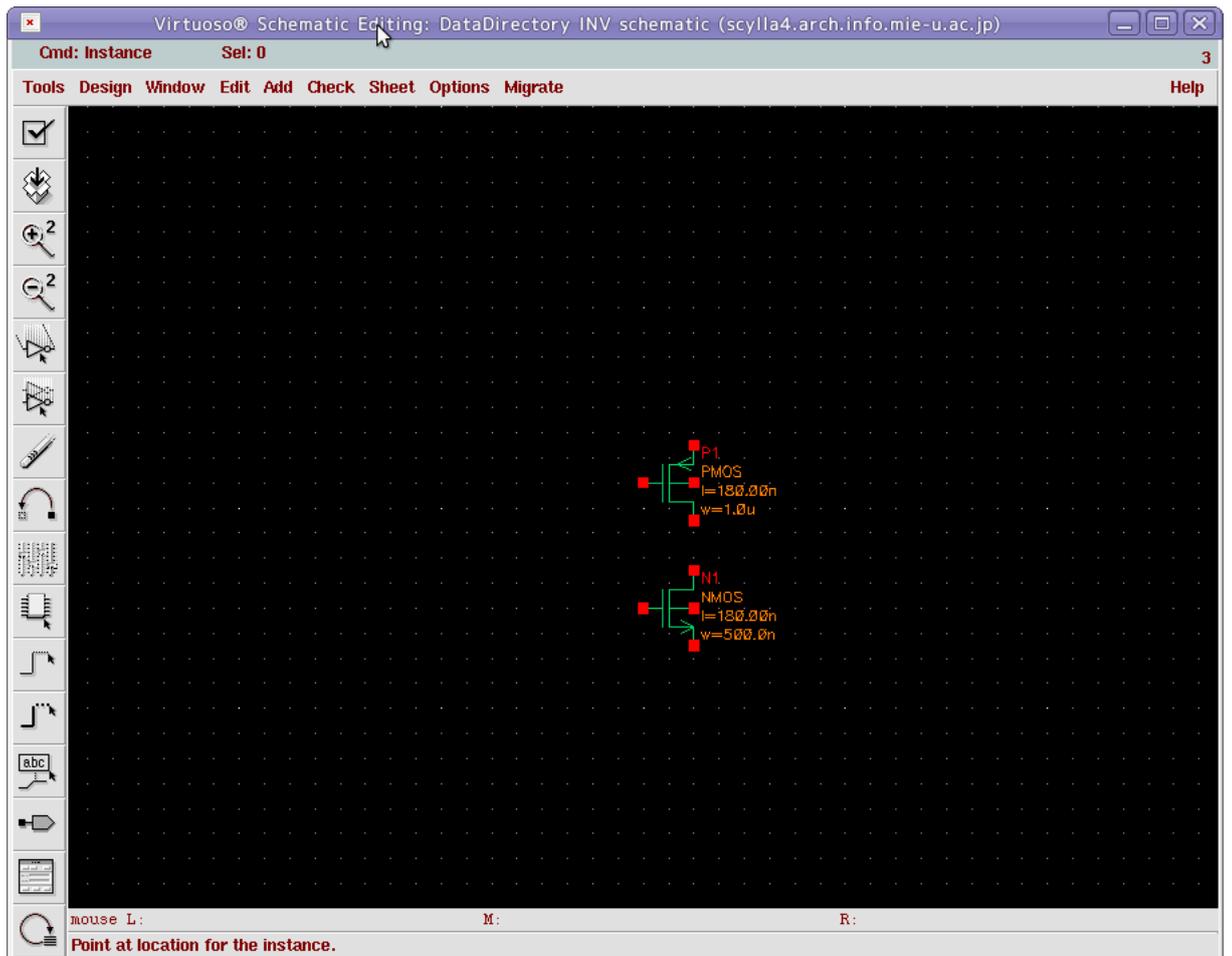


図 3.24: トランジスタ設置後の回路図エディタのウィンドウ

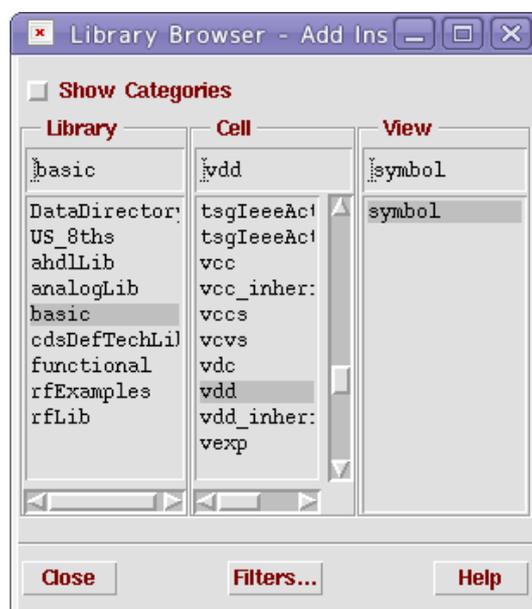


図 3.25: Add Instance のウィンドウ

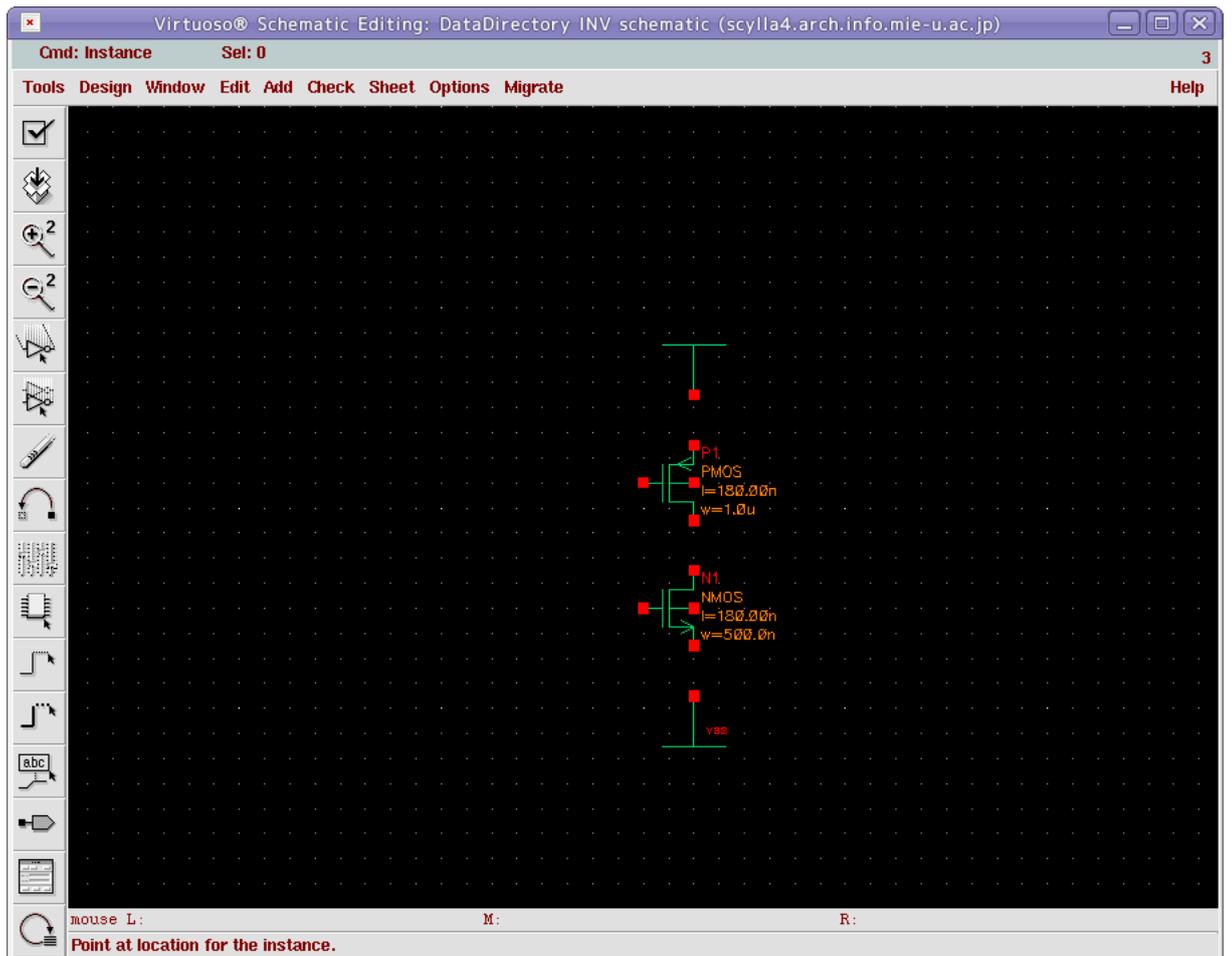


図 3.26: 電源とアース設置後の回路図エディタのウィンドウ

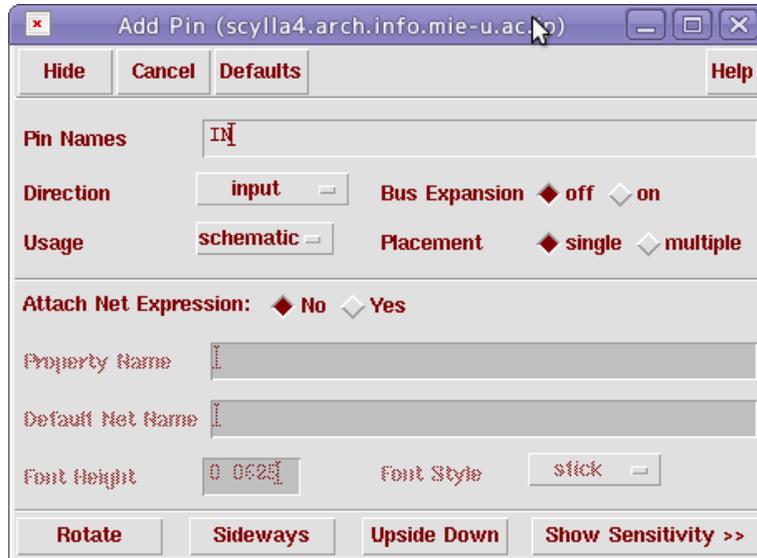


図 3.27: 入力ピンの作成

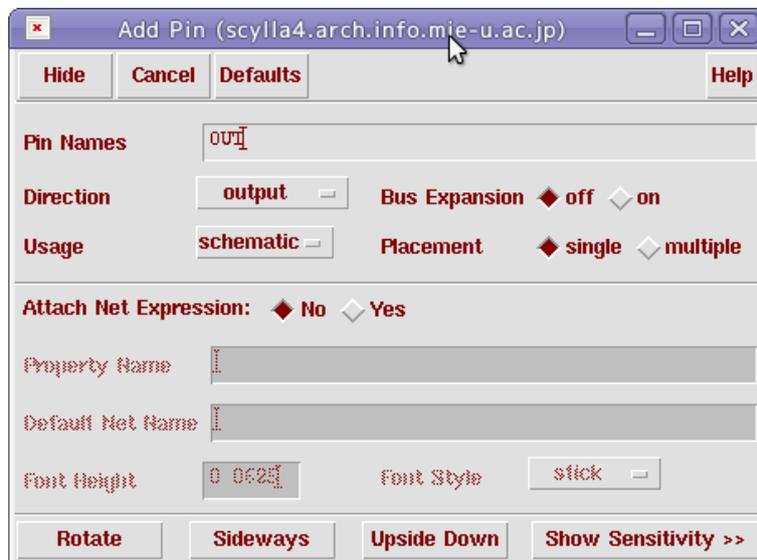


図 3.28: 出力ピンの作成

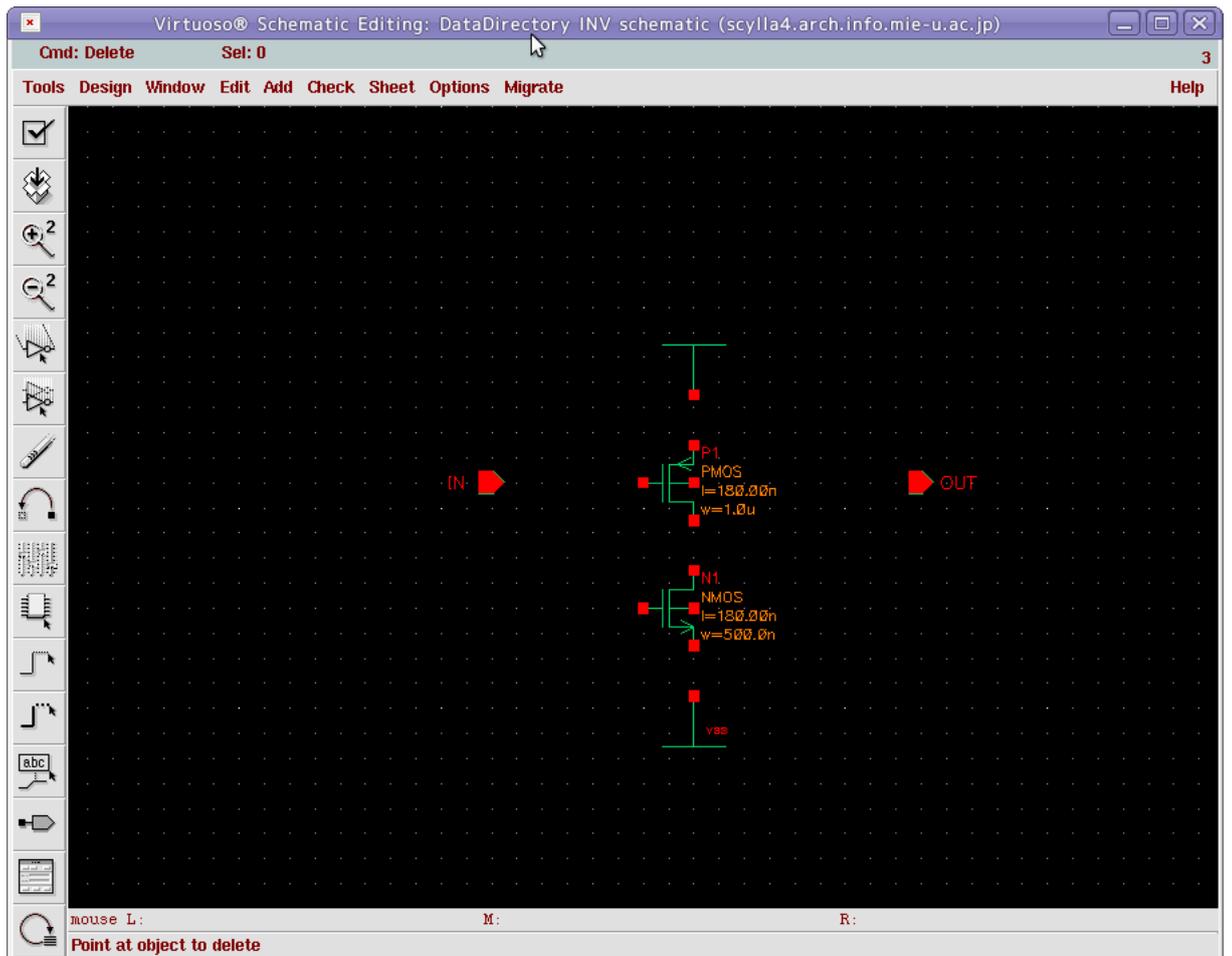


図 3.29: 入力と出力設置後の回路図エディタのウィンドウ

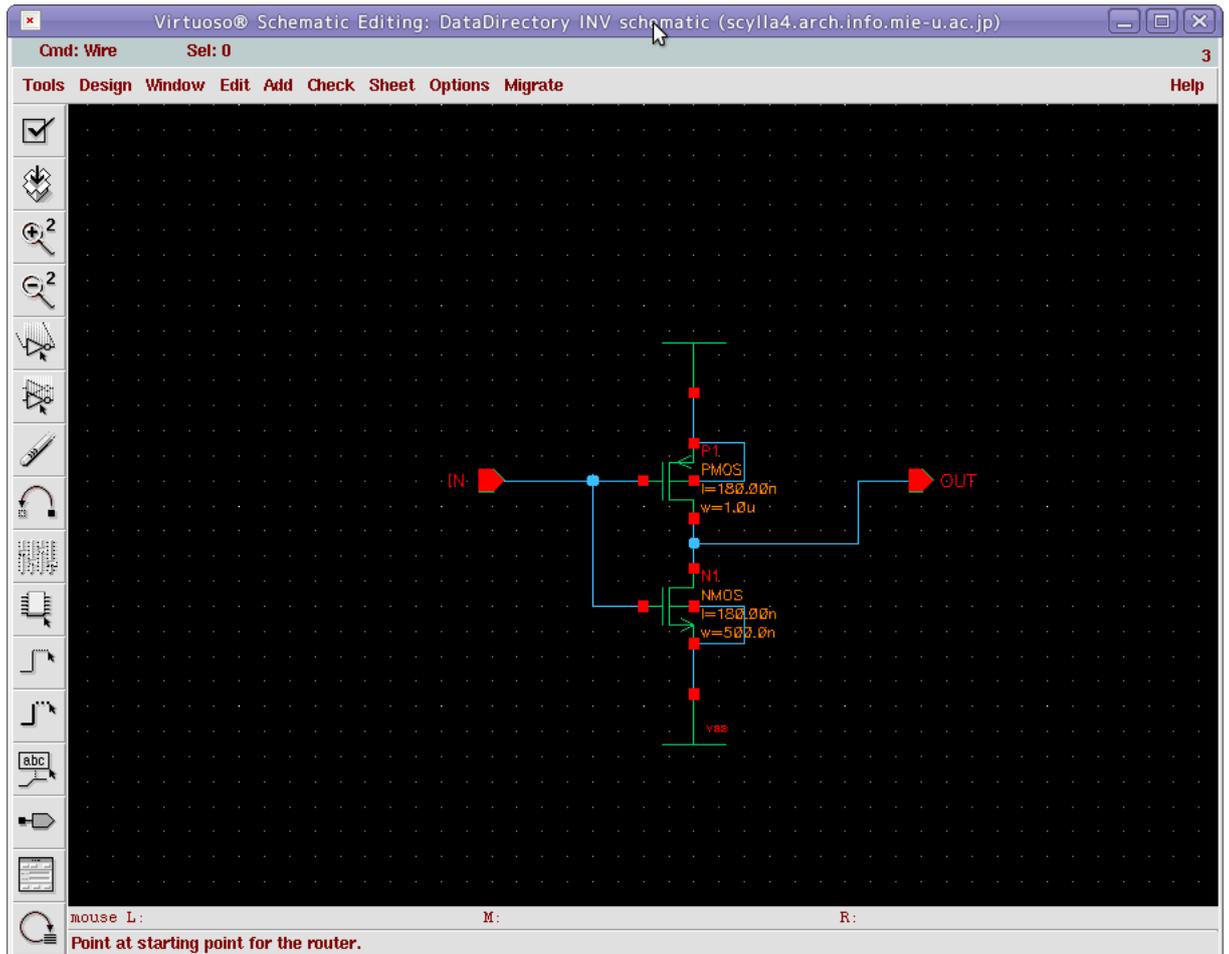


図 3.30: 結線後の回路図エディタのウィンドウ

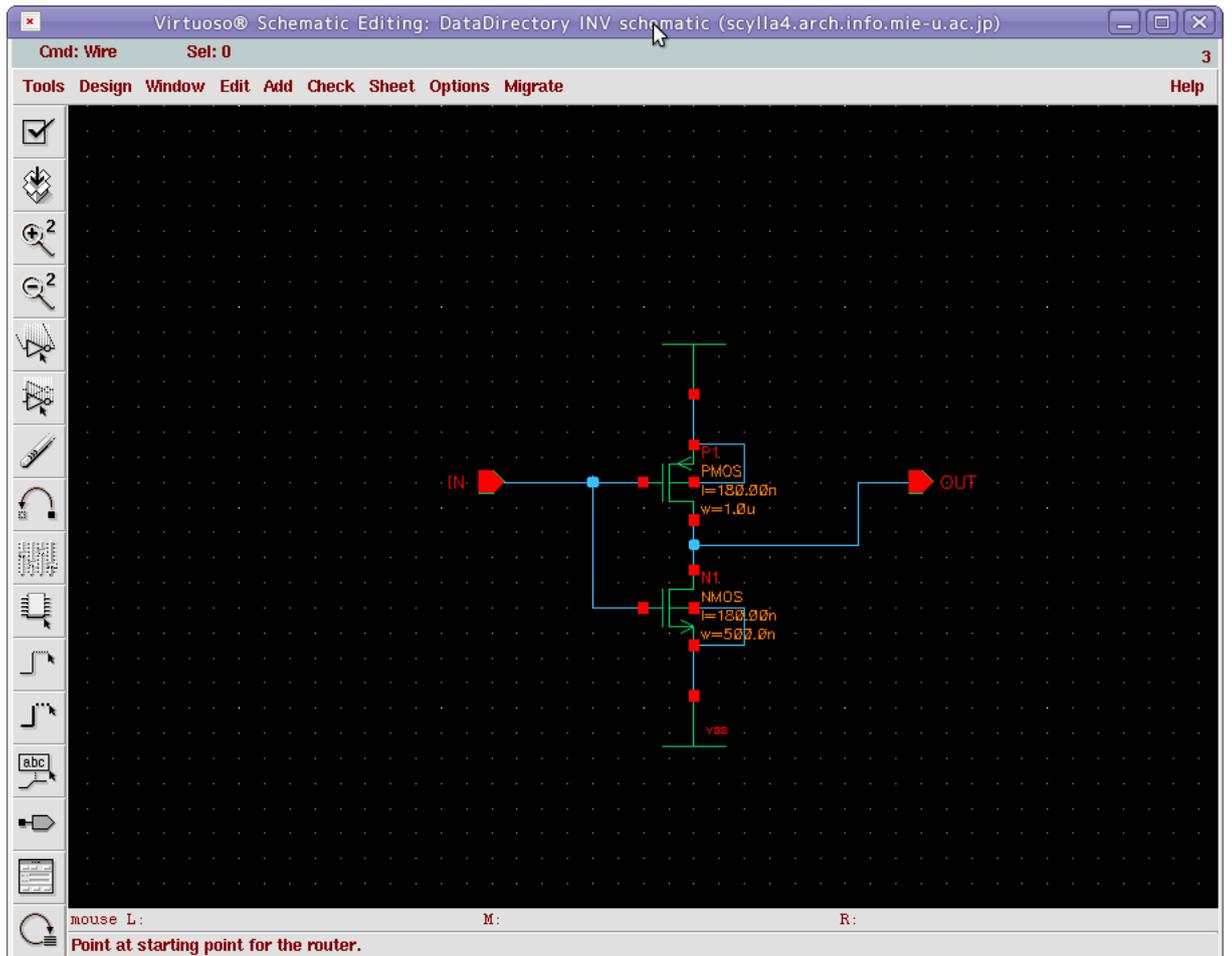


図 3.31: Analog Design Environment ウィンドウ

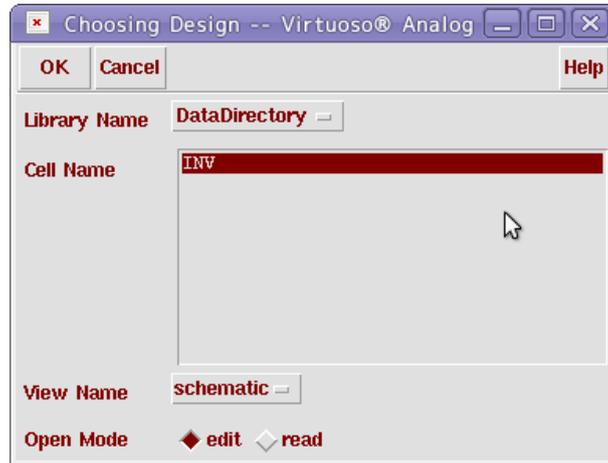


図 3.32: Choosing Design ウィンドウ

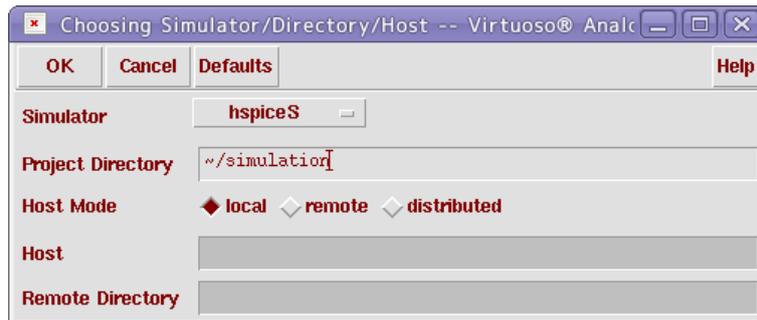


図 3.33: Choosing Simlator ウィンドウ



```
# File name: DataDirectory_INV_schematic.S.  
# Subcircuit for cell: INV.  
# Generated for: hspiceS.  
# Generated on Mar  9 18:01:47 2015.  
  
MN1 OUT IN vss! vss! NMOS L=180e-9 W=500e-9  
MP1 OUT IN vdd! vdd! PMOS L=180e-9 W=1e-6
```

図 3.34: 描画した回路図のネットリスト

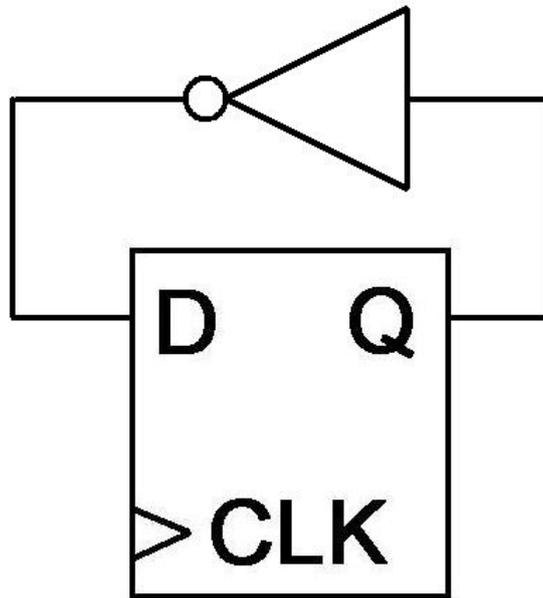


図 4.35: トグルしている DFF