

集積回路設計

7. レジスタとメモリ

一色 剛

工学院情報通信系

isshiki@ict.e.titech.ac.jp

7. レジスタとメモリ

■ ラッチ回路:スタティック型、ダイナミック型

- 情報を1/2クロック周期だけ記憶
- スタティックラッチ : CMOS伝送ゲート型、クロックドインバータ型、スタティックMUXゲート型

■ フリップフロップ回路:スタティック型、ダイナミック型

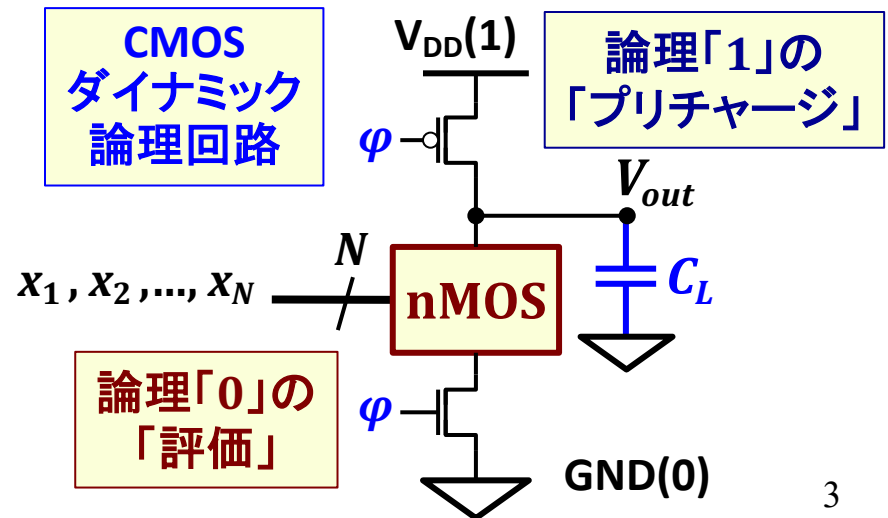
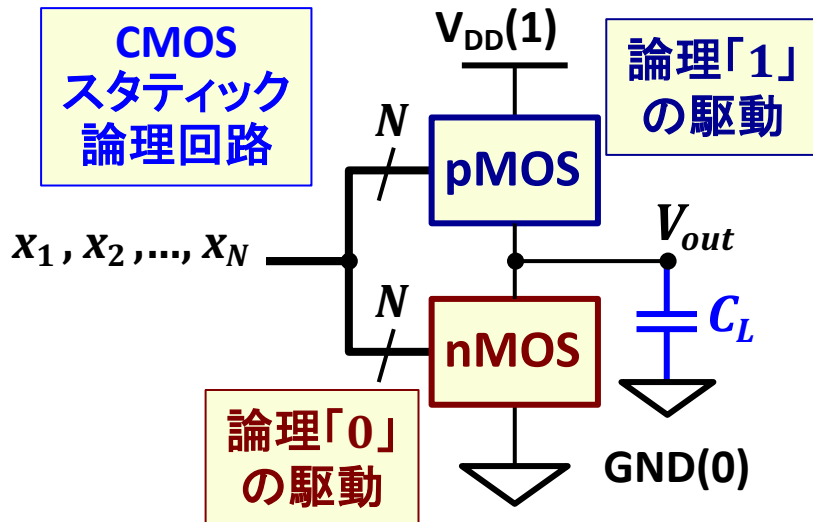
- 情報を1クロック周期だけ記憶
- 2つのラッチで構成

■ メモリ回路 :スタティック型(SRAM)、ダイナミック型(DRAM)

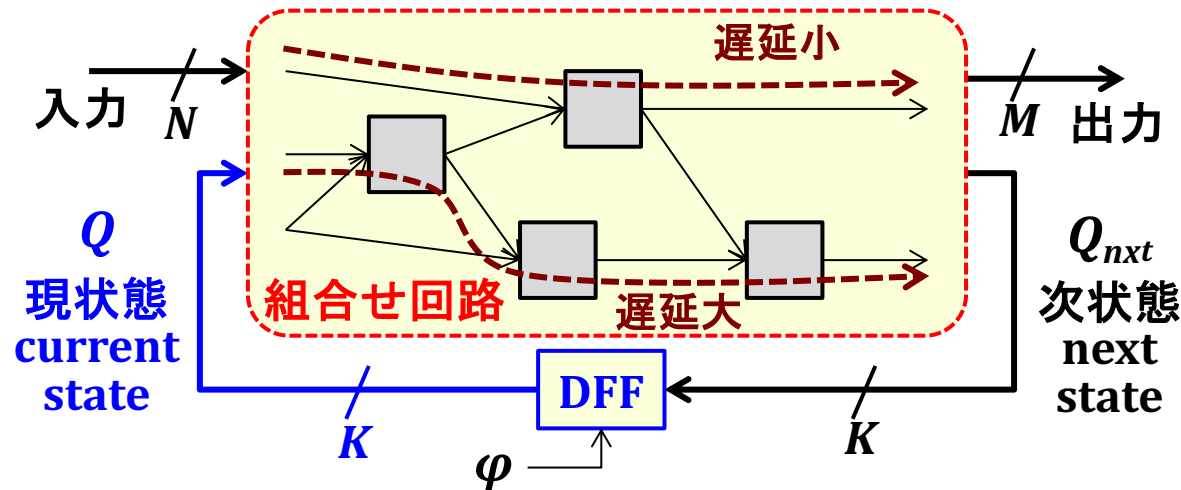
- アドレスデコード回路、ワードセレクト回路
- メモリセル回路
- 非同期メモリ回路、同期メモリ回路

スタティク論理回路とダイナミック論理回路 (復習)

- **スタティク論理回路** : pMOSブロック(論理「1」の駆動)とnMOSブロック(論理「0」の駆動)が相補的に動作
 - ❖ 出力 V_{out} は常に V_{DD} または GND と導通
- **ダイナミック論理回路** : pMOSTランジスタによる「プリチャージ」とnMOSブロックによる論理関数の「評価」
 - ❖ 論理「1」の V_{out} 出力は、 V_{DD} からの駆動ではなく出力負荷容量の電荷に頼る(V_{out} はハインピーダンス状態)

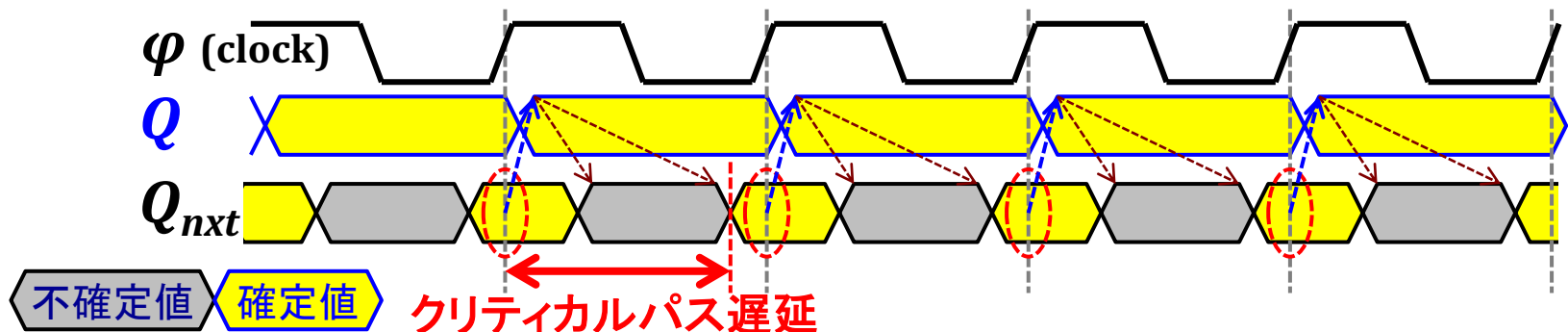


順序回路構成(単相クロック方式)(復習)



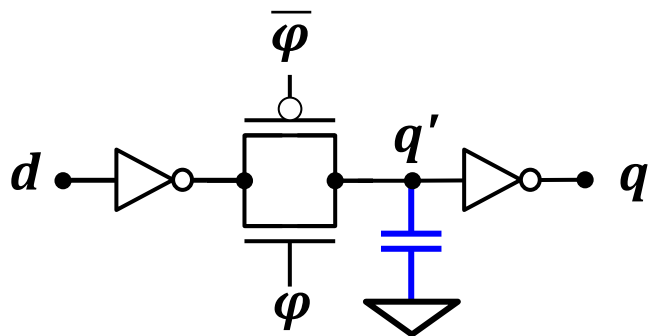
単相クロック方式：
最も一般的な順序
回路構成 → 設計自
動化技術が成熟

- ❖ DFF(Dフリップフロップ)出力：クロック立上り後、安定した値を保持
- ❖ パス遅延：信号伝搬経路上のゲート遅延の総和 → 組合せ回路の出力が安定するまで時間が掛かる
- ❖ 最大動作周波数：最大パス遅延(クリティカルパス遅延)で決まる

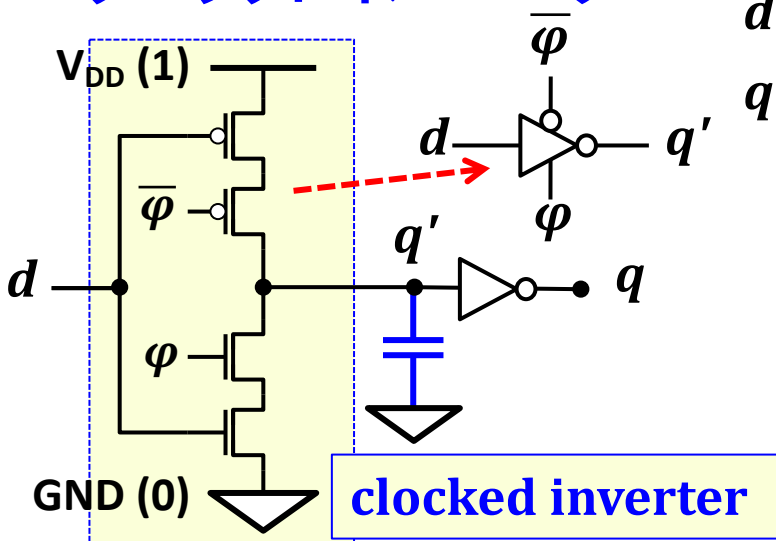


ダイナミックラッチとクロックドインバータ(復習)

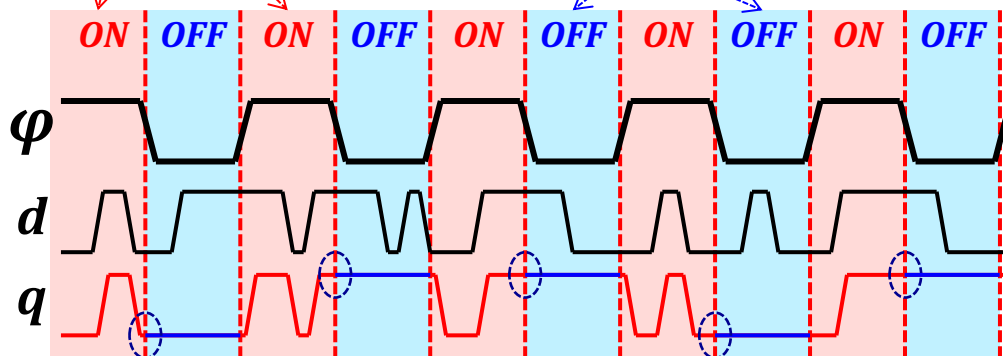
■ ダイナミックラッチ



■ クロックドインバータ


$$\varphi = 1 : q = \overline{q'} = d \text{ (書込み)}$$

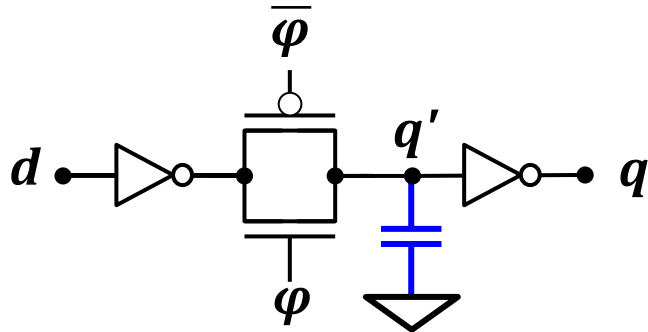
$\varphi = 0 : q' = \bar{q} \rightarrow q$ の値を保持



❖ 低いクロック周波数で誤動作
→ ダイナミック論理回路と同じ原因
(漏れ電流による電荷の消失)

ダイナミックラッチとスタティックラッチ

■ ダイナミックラッチ: 情報を寄生容量に記憶

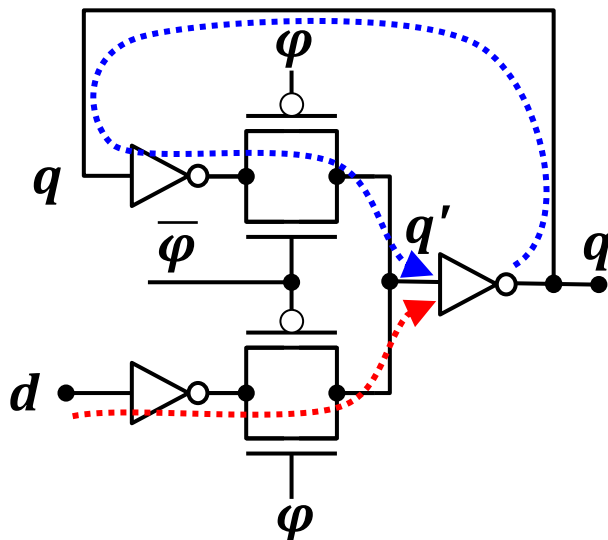


$\varphi = 1 : q = \bar{q}' = d$ (書込み)

$\varphi = 0 : q' = HiZ \rightarrow q$ の値を保持

❖ 低いクロック周波数で誤動作
(漏れ電流による電荷の消失)

■ スタティックラッチ: 情報をインバータループ構造に記憶



$q' = (\bar{q} \cdot \bar{\varphi} + \bar{d} \cdot \varphi) : 2$ 入力MUX

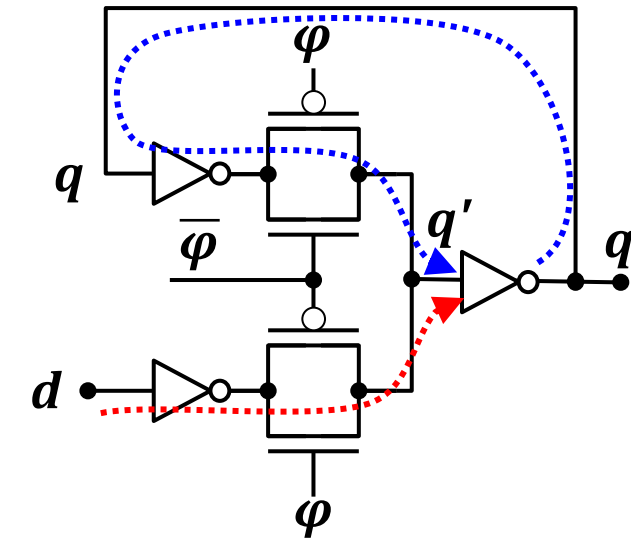
$\varphi = 1 : q = \bar{q}' = d$ (書込み)

$\varphi = 0 : q' = \bar{q} \rightarrow q$ の値を保持

❖ 情報は永久に(電源電圧が供給されている限り)保存される

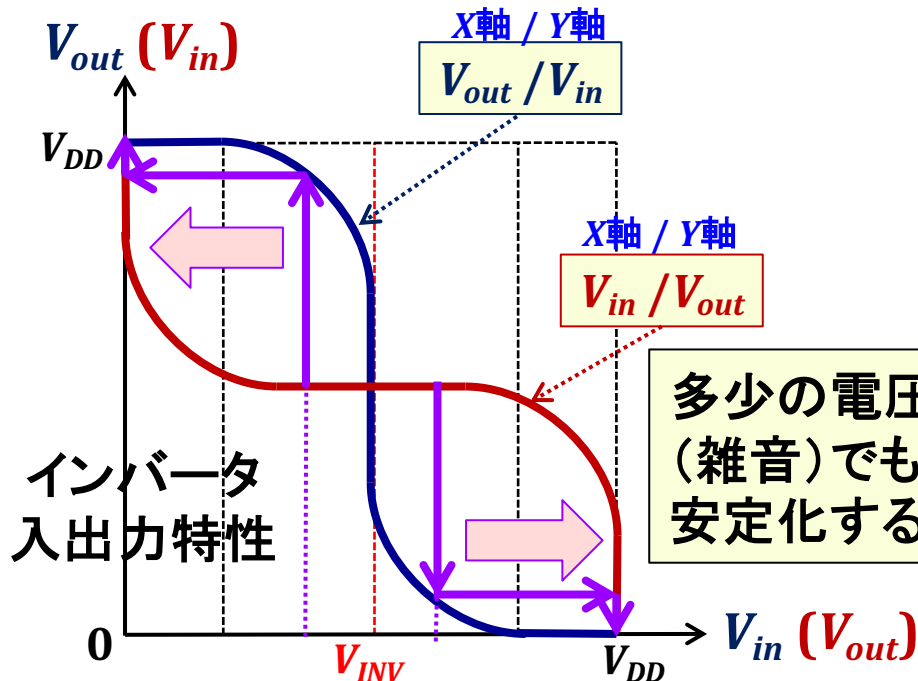
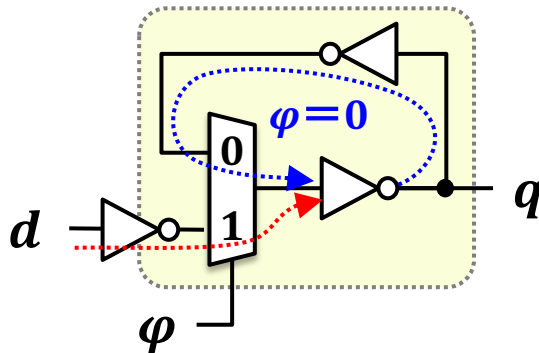
インバータループ構造による記憶素子

■ スタティクラッチ：情報をインバータループ構造に記憶

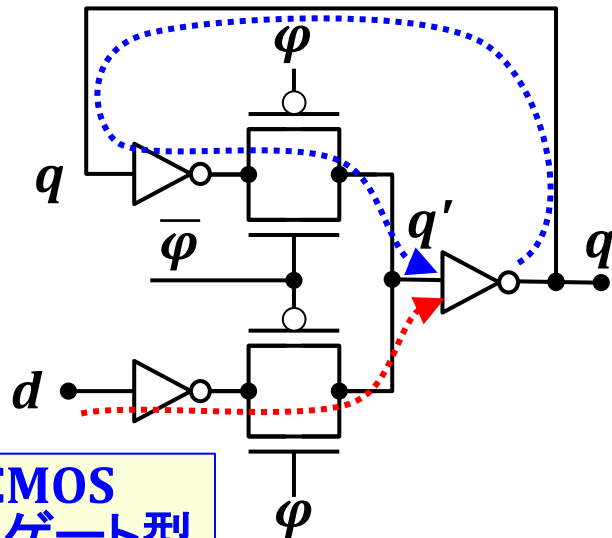
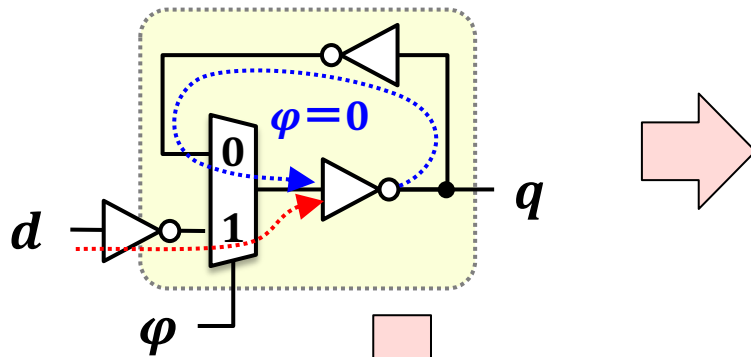


$\phi = 0 : q' = \bar{q} \rightarrow q$ の値を保持

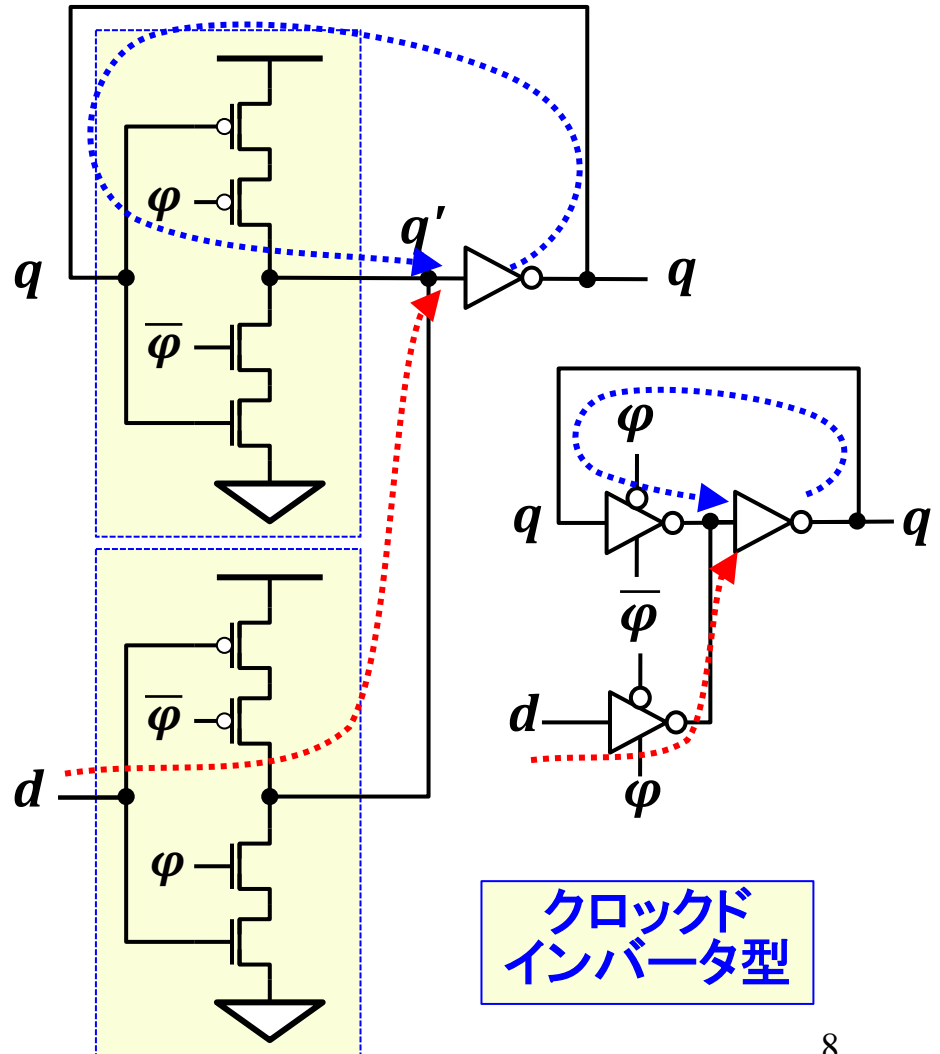
❖ 情報は永久に(電源電圧が供給されている限り)保存される



スタティックラッチ回路構成 (1)



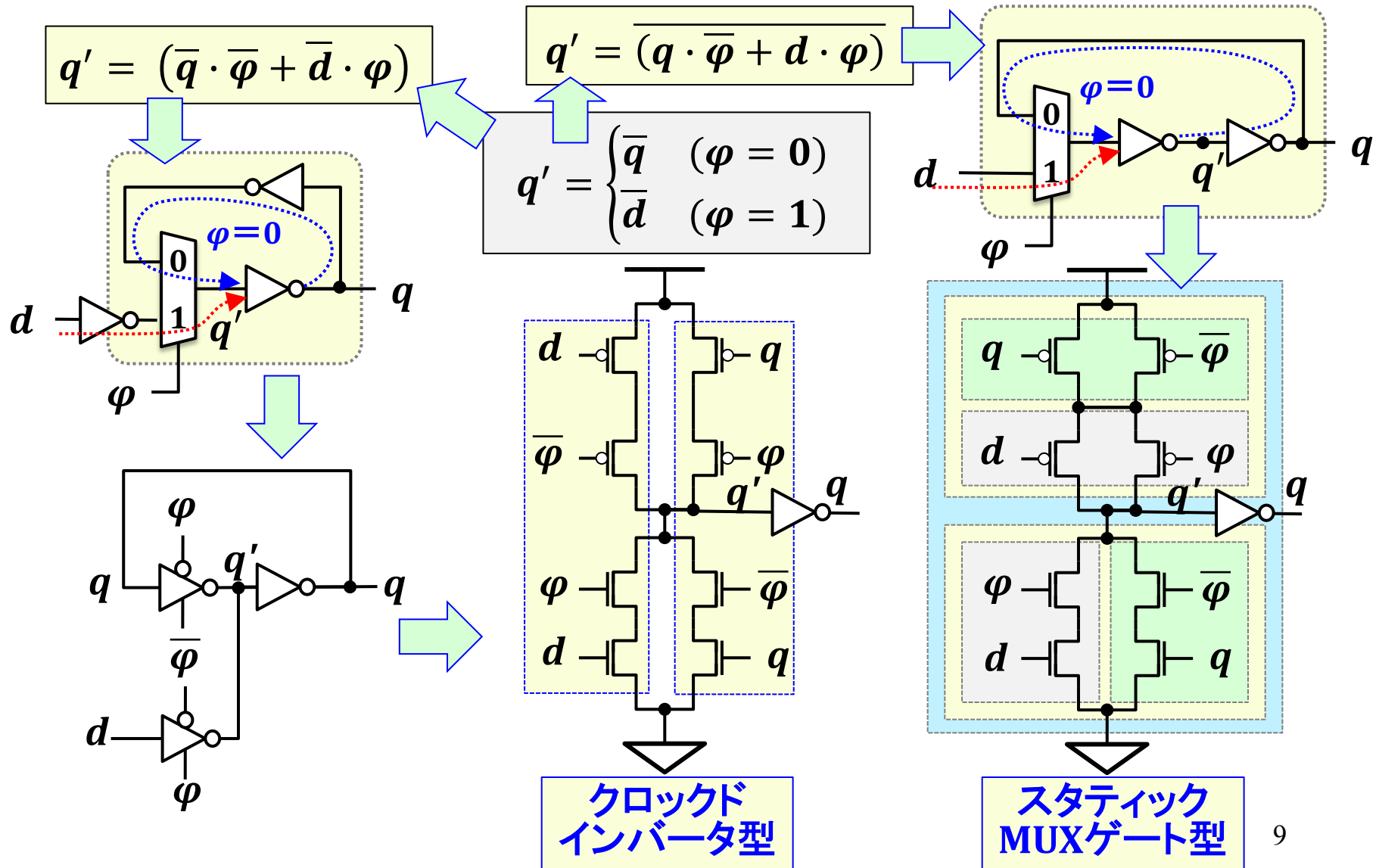
CMOS
伝送ゲート型



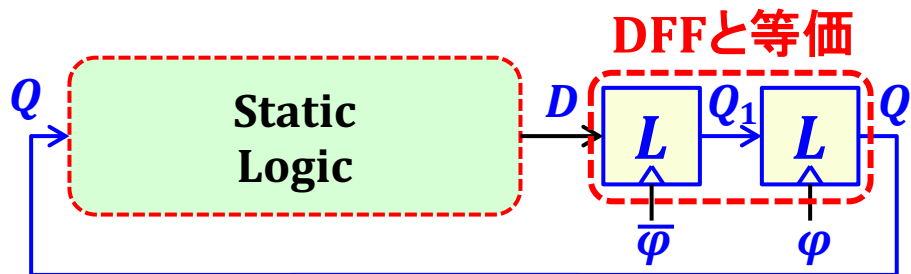
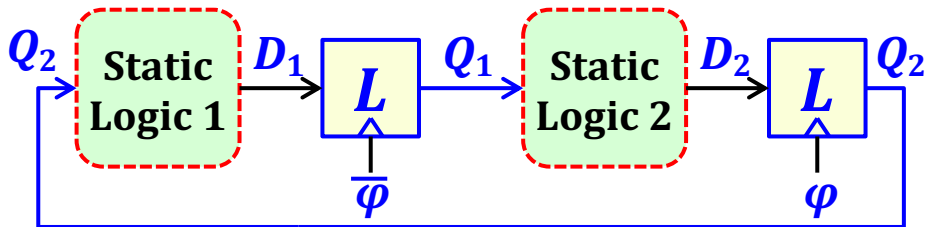
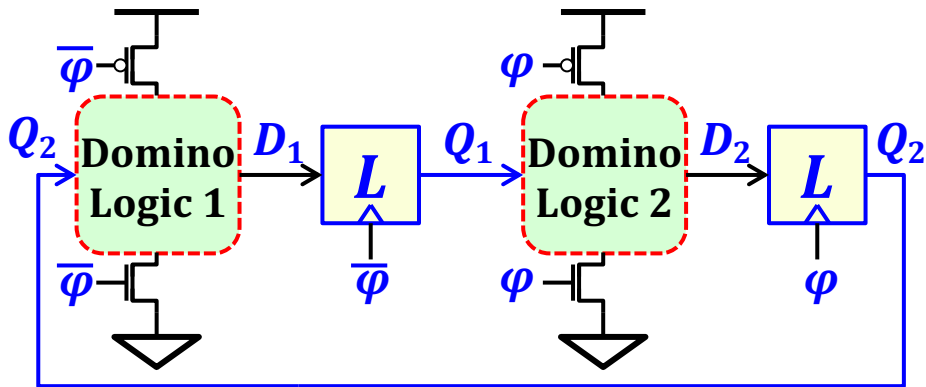
クロック
インバータ型

スタティックラッチ回路構成 (2)

訂正



クロック方式の比較(復習)



■ ダイナミック論理回路：

- ❖ プリチャージ期間のクロック半周期が必要 → 2相クロック方式

■ スタティック論理回路：

- ❖ 2つのクロック方式を柔軟に選択可能

■ 単相クロック方式：

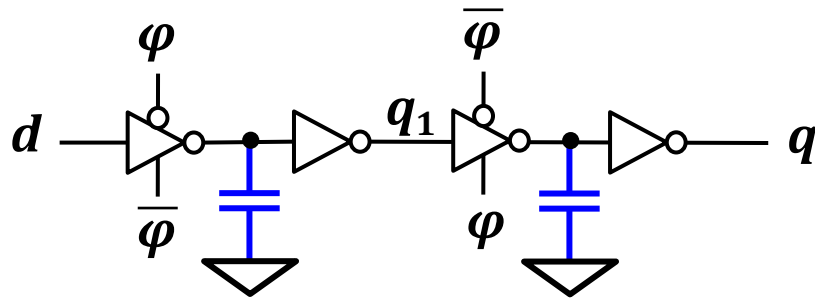
- ❖ 設計が簡単、設計自動化技術が成熟

■ 2相クロック方式：

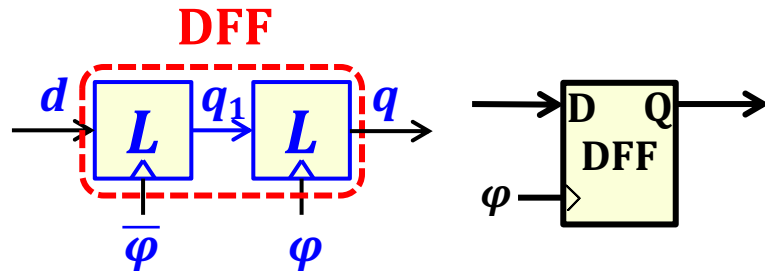
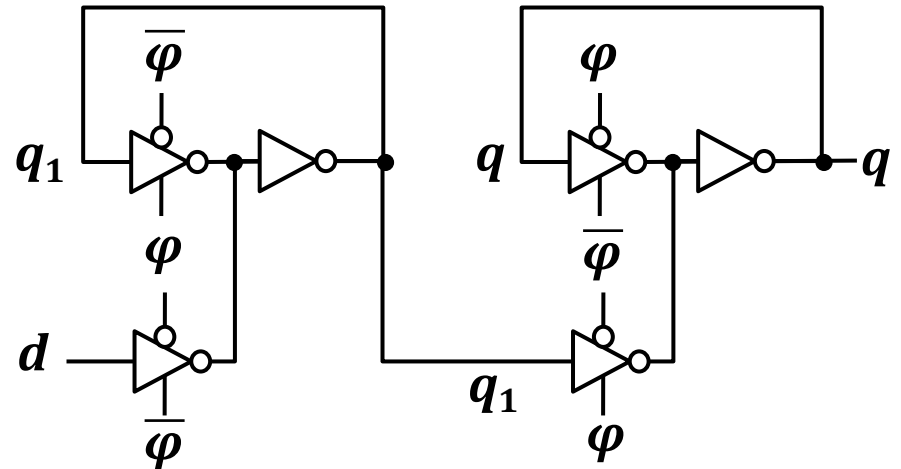
- ❖ ピーク電流が分散されるため電源ノイズが低い
- ❖ 設計が複雑(組合せ回路の分割)

Dフリップフロップ(DFF)

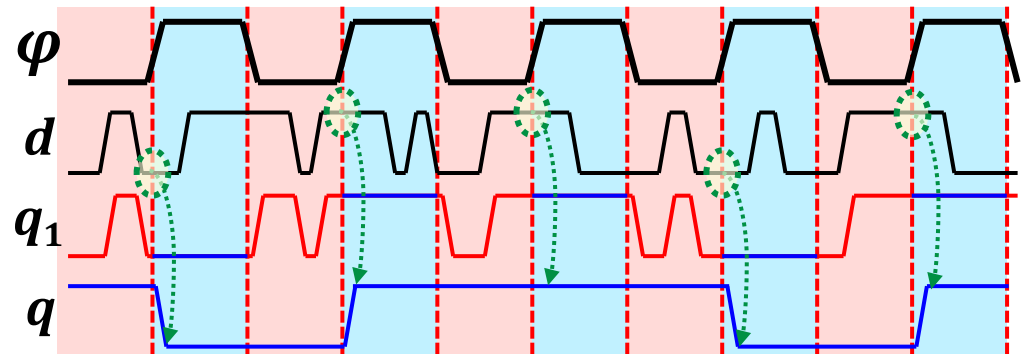
■ ダイナミックDFF



■ スタティックDFF

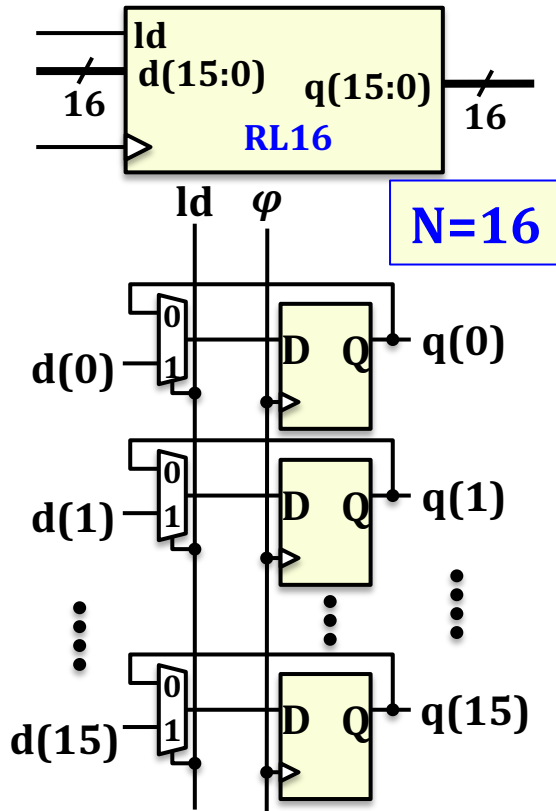


ϕ の立上り遷移時の
 d の値を q に出力
→情報を1クロック
周期だけ保持



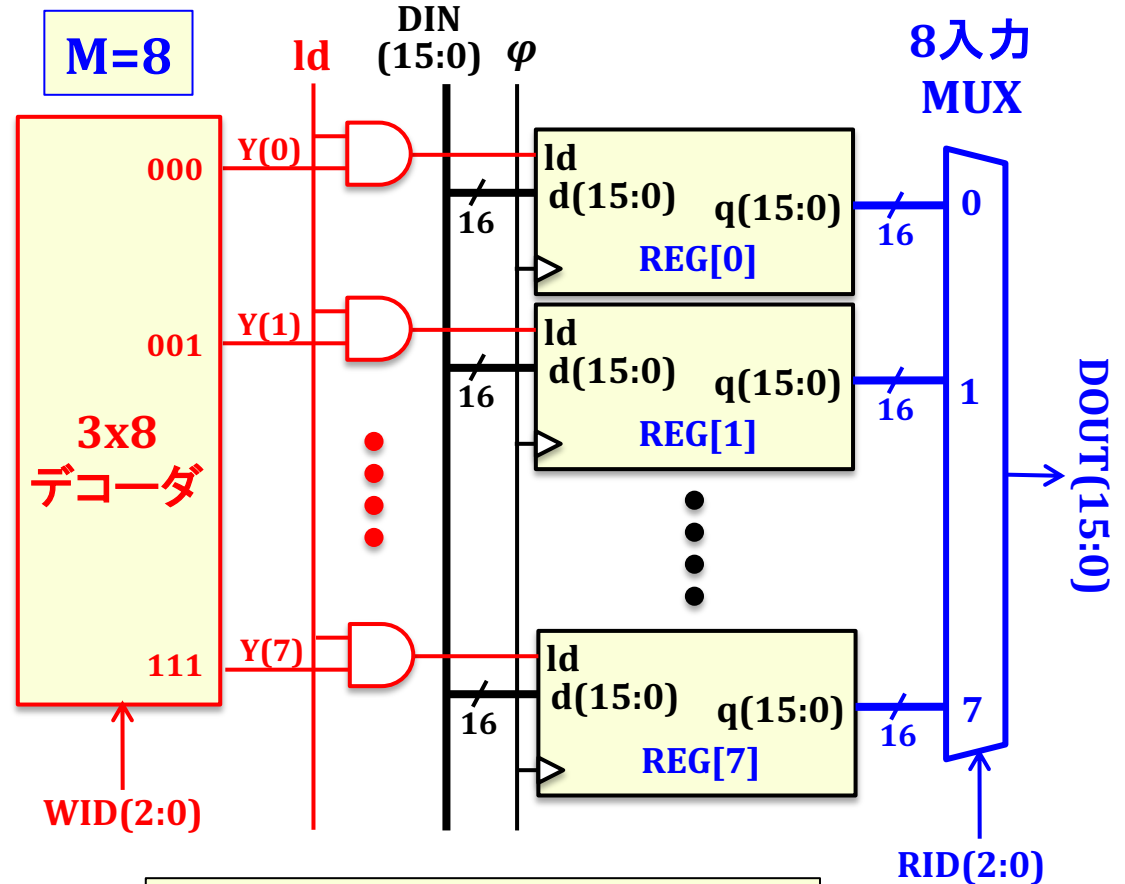
Nビットレジスタとレジスタファイル

■ Nビットレジスタ



ld : 書込み制御信号
情報を任意期間保持

■ MxN ビットレジスタファイル

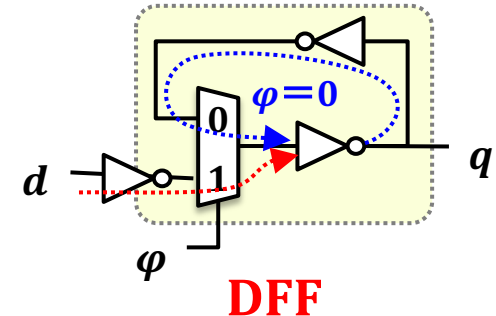


WID(2:0) : 書込みレジスタID
RID(2:0) : 読出しレジスタID

レジスタファイル回路構成

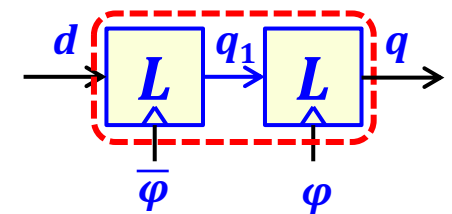
■ ラッチ回路: 情報を1/2クロック周期だけ保持

- ❖ 書込み状態: 1/2クロック
- ❖ 保持状態: 1/2クロック
- ❖ 2相クロック方式順序回路の状態を保持



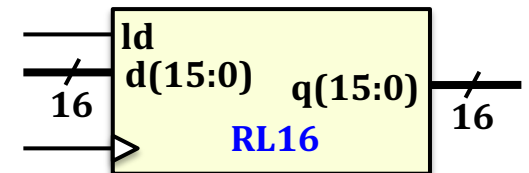
■ フリップフロップ回路: 情報を1クロック周期だけ保持

- ❖ 2つのラッチ回路で構成
- ❖ クロックの立上り時の入力値を出力値として保持する
- ❖ 単相クロック方式順序回路の状態を保持



■ Nビットレジスタ: 情報を任意期間保持

- ❖ DFF+MUX構成
- ❖ 制御信号 (ld) で書込み状態と保持状態を制御



■ NxMビットレジスタファイル: 小容量高速メモリ(プロセッサ内部用)

- ❖ 書込みレジスタID (WID) → M出力デコーダで書込みレジスタを指定
- ❖ 読出しレジスタID (RID) → M入力MUXで読出しレジスタを選択
- ❖ 通常は、N=16~32ビット、M=8~32ワード程度 → 大容量化が困難

メモリ回路構成

■ RAM (Random Access Memory) : レジスタファイルと同機能

- ❖ 任意メモリ番地(アドレス)をアクセス
- ❖ (通常)1ポートメモリ : Read/Write共通アドレス、排他的R/W動作

■ メモリセル : 小面積記憶素子、2次元平面に配置

- ❖ Static RAM (SRAM) : インバータループ + nMOSパストランジスタ
- ❖ Dynamic RAM (DRAM) : キャパシタ + nMOSパストランジスタ

■ グローバル配線 : 大きな寄生容量

- ❖ word線、bit線

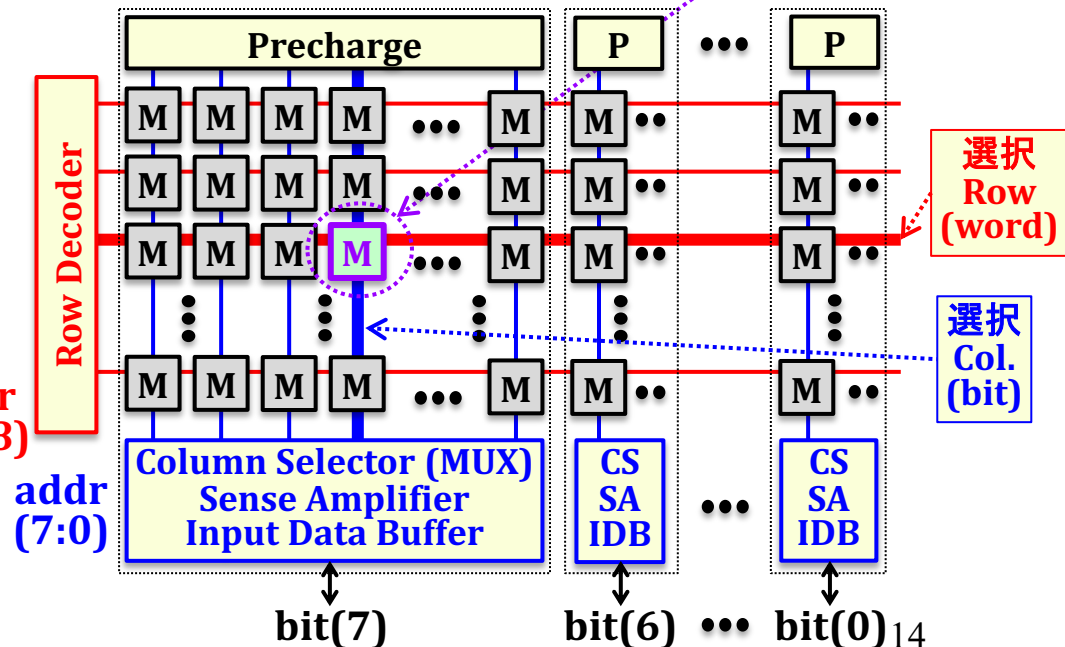
■ 周辺回路

- ❖ Rowデコーダ (word線)
- ❖ Columnセレクト (bit線MUX)
- ❖ 入力データバッファ
- ❖ センスアンプ : 高速読出し用
- ❖ プリチャージ回路 : bit線用

#row #col
512x256x8ビット
メモリ構成

addr
(16:8)

addr
(7:0)



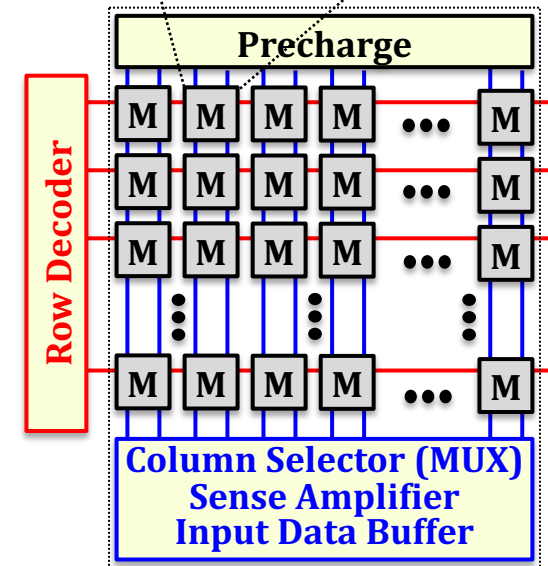
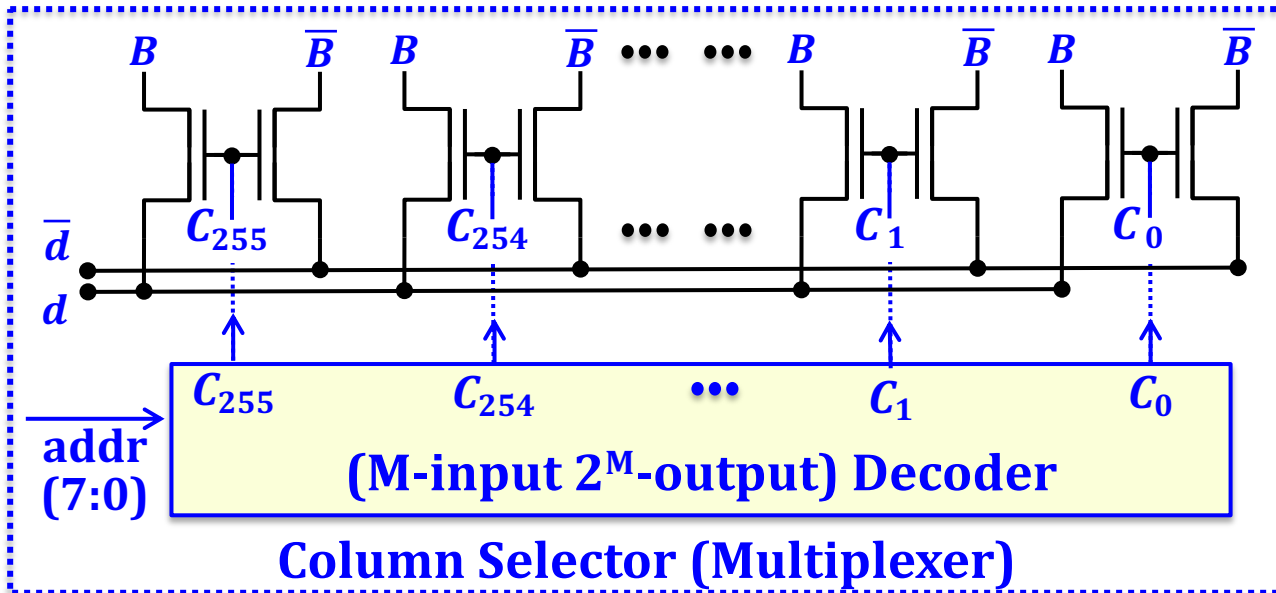
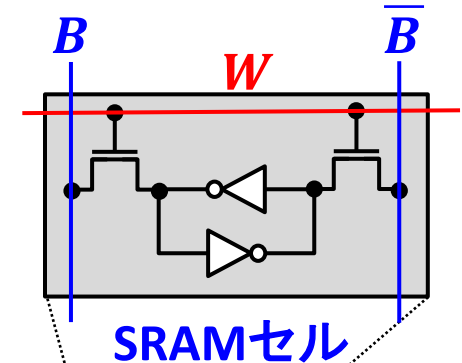
SRAM回路構成

■ SRAMセル : インバータ+nMOSパストランジスタ

- ❖ Bit線 (B, \bar{B}) : 2つのnMOSによってインバータループと接続
- ❖ Word線 (W) : nMOSパストランジスタのゲートに接続

■ 周辺回路 (1)

- ❖ Rowデコーダ (word線) : $N \times 2^N$ デコーダ
- ❖ Columnセクタ (bit線) : $M \times 2^M$ デコーダ + nMOSパストランジスタ

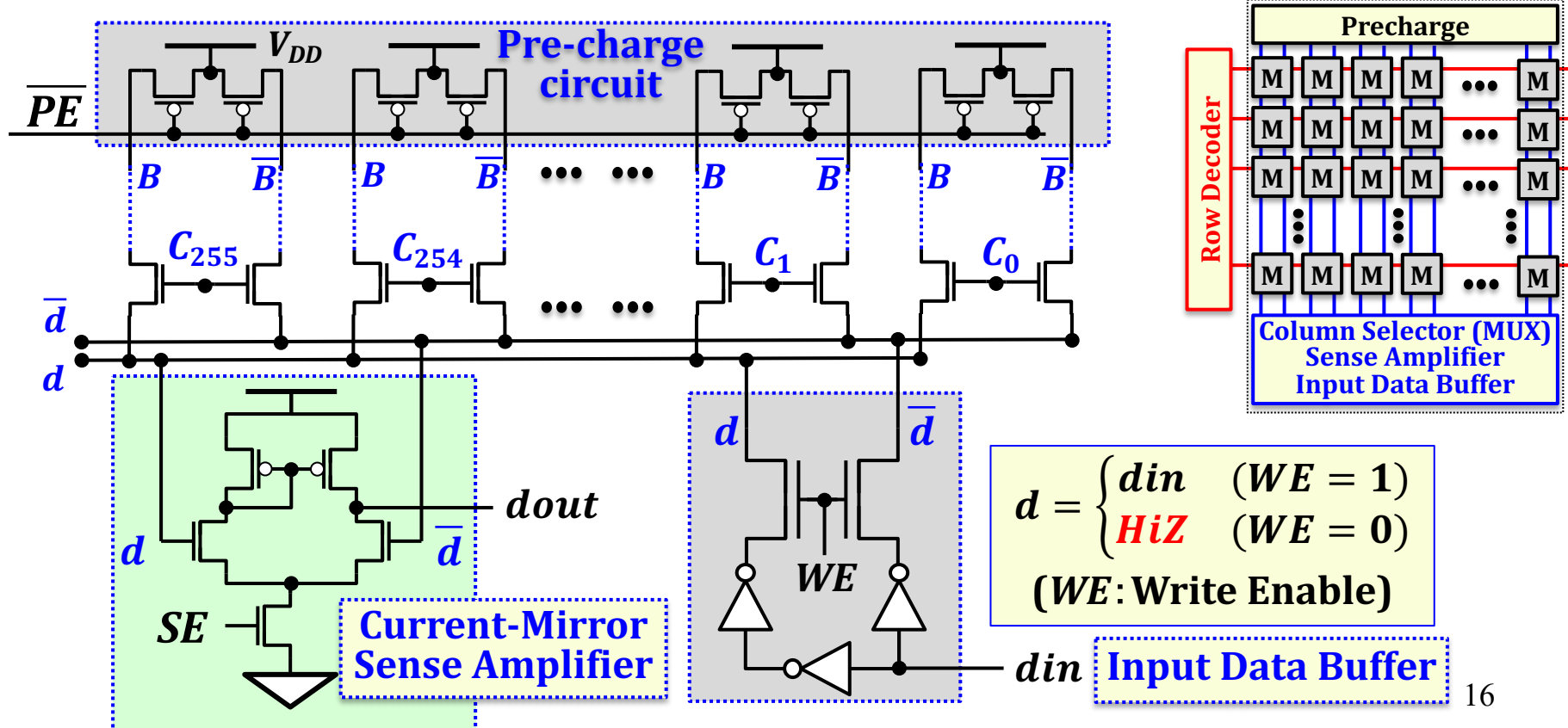


2^M 組みのBit線対 (B, \bar{B}) から1組みを選択し d, \bar{d} に接続

SRAM回路構成

■ 周辺回路 (2)

- ❖ 入力データバッファ: 書込み時 ($WE=1$) に入力データを選択ビット線対 d, \bar{d} に接続
- ❖ センスアンプ (カレントミラー型): 読出し時 ($SE=1$) に選択ビット線対 d, \bar{d} の電圧差を増幅すること (差動増幅) で高速読出しを実現 (アナログ回路)
- ❖ プリチャージ回路: 読出し前 ($PE=1$) に全てのビット線を V_{DD} に充電



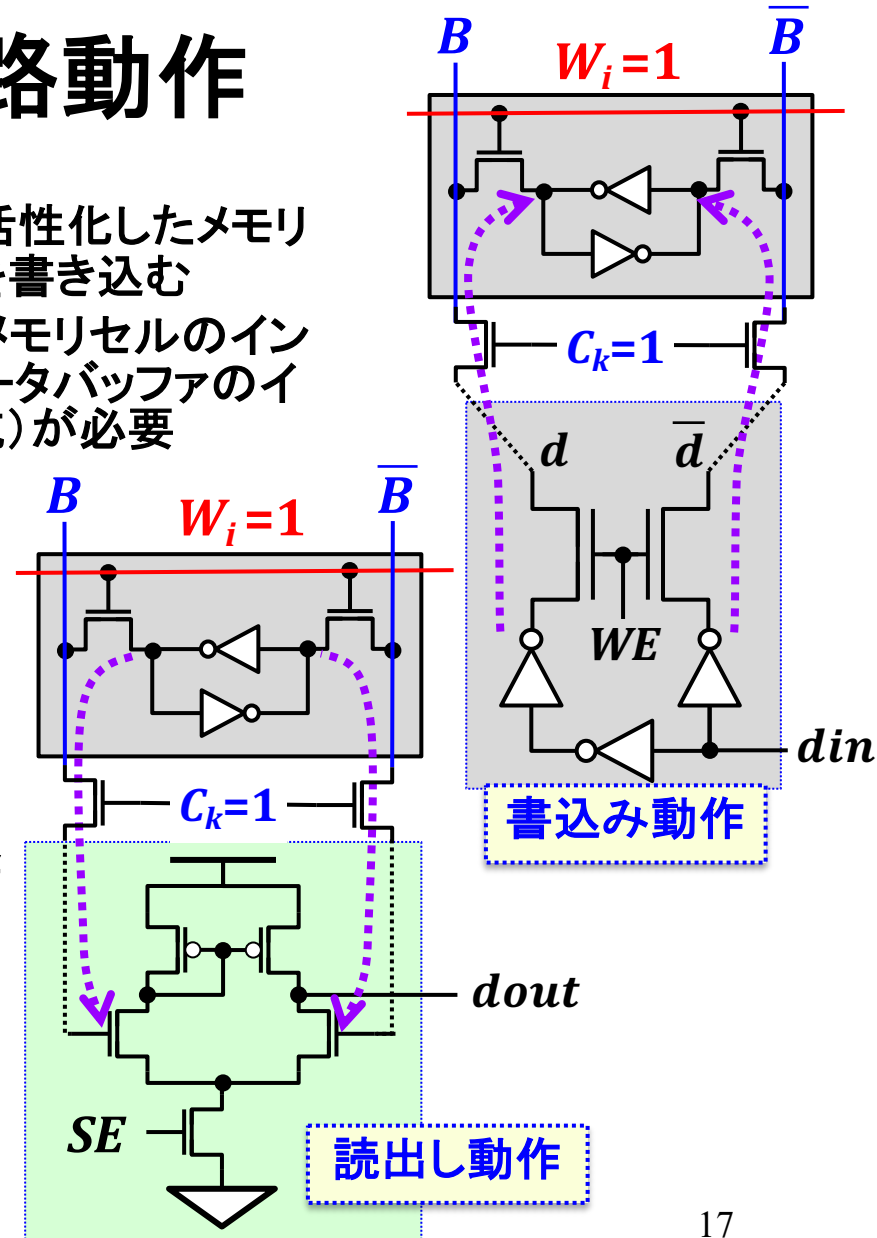
SRAM回路動作

■ 書込み動作 : $WE=1, SE=PE=0$

- ❖ RowデコーダとColumnセクタによって活性化したメモリセル ($W_i=1, C_k=1$) に入力データ(din)を書き込む
- ❖ Bit線の大きな寄生容量を充電し、さらにメモリセルのインバータの状態を反転させるため、入力データバッファのインバータは大きな駆動力(小さな等価抵抗)が必要

■ 読出し動作 : $WE=0$ (入力データバッファは不活性)

- ❖ プリチャージ期間 ($SE=0, PE=1$, 全 $W_i=0$) : B, \bar{B} を V_{DD} に充電
- ❖ 読出し期間 ($SE=1, PE=0, W_i=1$) : B, \bar{B} のうち、「0」となるbit線は活性化したメモリセルのインバータにより徐々に電圧が低下していくが、差動増幅の作用により出力 $dout = B$ が高速に確定する



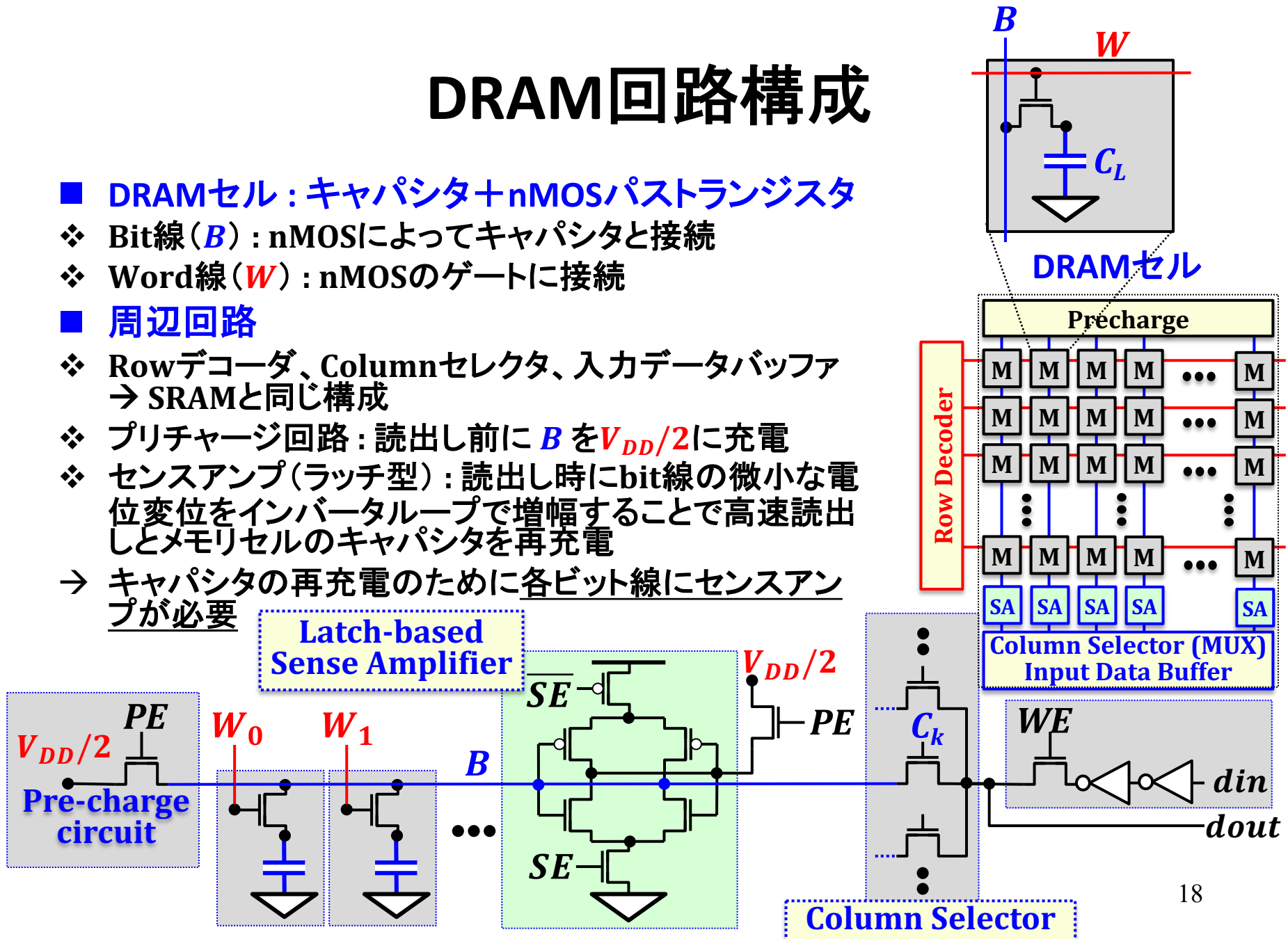
DRAM回路構成

■ DRAMセル : キャパシタ+nMOSパストランジスタ

- ❖ Bit線 (B) : nMOSによってキャパシタと接続
- ❖ Word線 (W) : nMOSのゲートに接続

■ 周辺回路

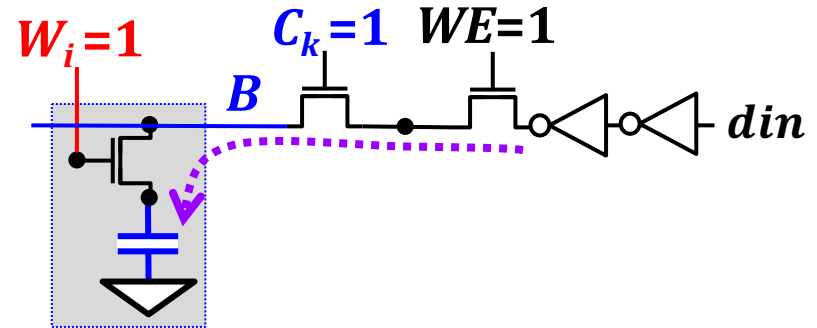
- ❖ Rowデコーダ、Columnセレクタ、入力データバッファ
→ SRAMと同じ構成
 - ❖ プリチャージ回路 : 読出し前に B を $V_{DD}/2$ に充電
 - ❖ センスアンプ(ラッチ型) : 読出し時にbit線の微小な電位変位をインバータループで増幅することで高速読出しとメモリセルのキャパシタを再充電
- キャパシタの再充電のために各ビット線にセンスアンプが必要



DRAM回路動作

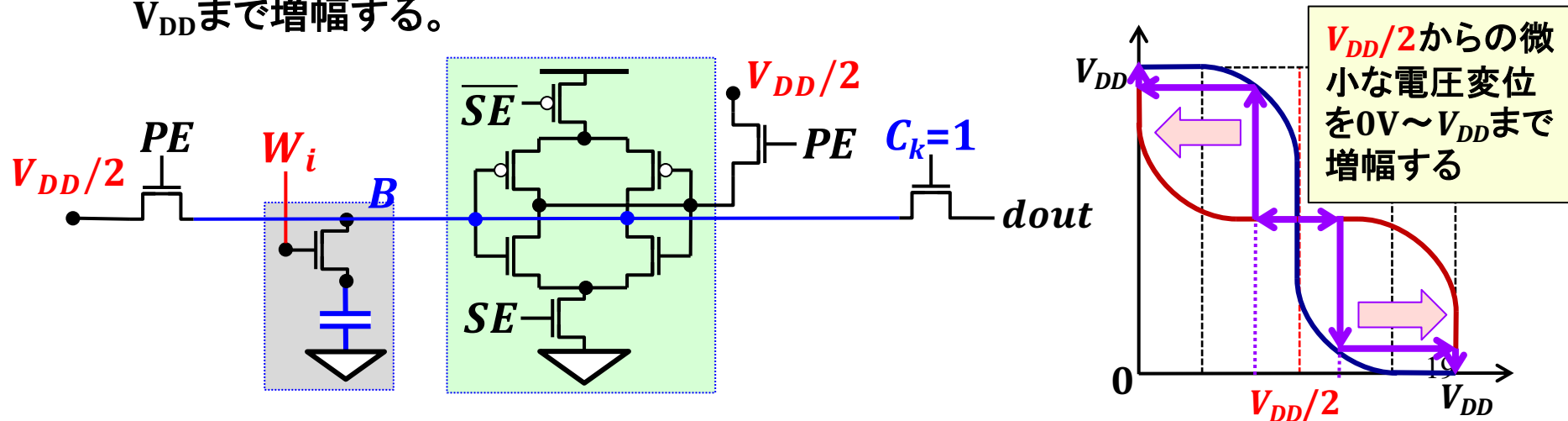
■ 書込み動作 : $WE=1, SE=PE=0$

- ❖ RowデコーダとColumnセクタによって活性化したメモリセルに入力データ(din)を書き込む



■ 読出し動作 : $WE=0$ (入力データバッファは不活性)

- ❖ プリチャージ期間 ($SE=0, PE=1$, 全 $W_i=0$) : B を $V_{DD}/2$ に充電
- ❖ 読出し期間 ($PE=0, W_i=1$) : $SE=0$ のままで活性化したメモリセルの電荷を B に転送するが、 B には大きな寄生容量があるため、 B の $V_{DD}/2$ からの電圧変移は微小である。その後、 $SE=1$ とし、インバータループによって、微小な電圧変移を $0V$ または V_{DD} まで増幅する。



不揮発性メモリ (FLASH)

■ FLASHメモリセル : nMOS + Floating Gate

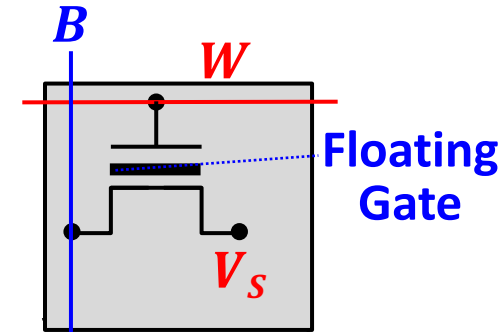
- ❖ Floating Gate : 電氣的に絶縁した電極
- ❖ Floating Gateに電荷がある場合 : nMOSの閾値電圧が上がり、制御ゲートに V_{DD} を印加してもOFF状態のままになる
- ❖ Floating Gateに電荷がない場合 : 通常のnMOS動作

■ 書込み動作 :

- ❖ 電荷注入 : V_S を0Vにし、 W に高電圧(12V)を印加することで、Floating Gateに電荷(電子)を注入する。
 - ❖ 電荷消去 : V_S に高電圧(12V)を印加し、 W を0Vにすることで、Floating Gateの電荷(電子)を V_S から吸い出す。
- Floating Gateの電荷は絶縁しているため、電源電圧を落としても保存される(不揮発性)

■ 読出し動作 : $V_S=0V$

- ❖ Floating Gateに電荷があるセルが活性化した場合 : nMOSはOFF状態のままなので、Bit線(B)はPullup回路によって V_{DD} になる
- ❖ Floating Gateに電荷がないセルが活性化した場合 : nMOSはON状態になり、Bit線(B)は0V(近く)になる



不揮発性メモリセル

