

2. コンピュータの基礎（2）－メモリ－ Memory

コンピュータはどのようにものを憶えているのでしょうか？磁気ディスクや光ディスクにデータを保存する使い方の場合に、電源を入れていない間どのように情報が保持されているかは想像しやすいでしょう。これらの記録装置は「補助記憶装置」と呼ばれます。昔は紙テープやパンチカード、磁気テープなどが補助記憶装置として使われていました。

しかし、データを補助記憶装置から読み込んだとしても、主な処理をして次に書き込むまでの間は電子回路に記憶されています。また、コンピュータの中の処理装置（プロセッサ）の動作を指定する一連の指令（コマンド）はコードと呼ばれ、これも電子回路に記憶されます。データやコードを記憶しておくための電子回路をメモリと呼びます。

原理的には、電子回路によるメモリ素子が無くても動作するコンピュータを作ることが可能ですが、一般的には補助記憶装置を使ったデータの読み書きよりメモリ素子を用いたデータの読み書きの方がずっと高速であり、實際上メモリ素子はコンピュータにとって必要不可欠な部品です。

ふつうに使われるメモリ素子には2種類あります。

(1) スタティック・メモリ；Static memory

(2) ダイナミック・メモリ；Dynamic memory

これら2種類のメモリ素子は全然違うしくみになっています。詳しいしくみは後述するとして、これらのメモリ素子の特徴を比較すると表2.1のようになります。

表2.1 スタティック・メモリとダイナミック・メモリの比較

素子の種類	読み書きの速度	構造の複雑さ	高集積密度化	単価
スタティック	高速	やや複雑	やや困難	やや高価
ダイナミック	やや低速	単純	容易	安価

2-1 スタティック・メモリ Static memory

スタティック・メモリは論理素子の組み合わせで実現されます。典型的な例は以下のように2つの NAND ゲートを組み合わせた回路です。

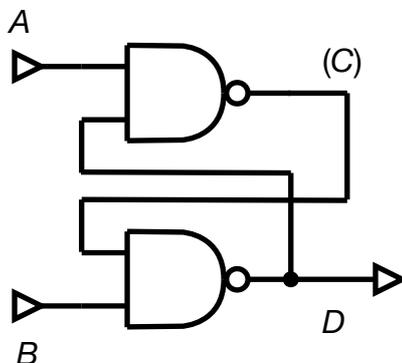


図 2.1 スタティック・メモリ回路

このような回路の出力 D は、2つの入力 A, B によってどのように変化するでしょうか？

NAND ゲートのはたらきから、以下の2つの式が成り立ちます。

$$\begin{cases} C = \overline{A \wedge D} \\ D = \overline{B \wedge C} \end{cases}$$

この1番目の式を2番目の式に代入すると、

$$\begin{aligned} D &= \overline{B \wedge \overline{A \wedge D}} \\ &= \overline{B} \vee (A \wedge D) \end{aligned}$$

という関係が得られます。2番目の式の変形にはド・モルガンの法則を使いました。さらに、 A と B の値に応じて、 D の値がどのような値を取るかを調べると、

以下のようになります。

$$A = 0, B = 0 \text{ のとき, } D = 1 \vee (0 \wedge D) = 1 \vee 0 = 1$$

$$A = 0, B = 1 \text{ のとき, } D = 0 \vee (0 \wedge D) = 0 \vee 0 = 0$$

$$A = 1, B = 0 \text{ のとき, } D = 1 \vee (1 \wedge D) = 1 \vee D = 1$$

$$A = 1, B = 1 \text{ のとき, } D = 0 \vee (1 \wedge D) = 0 \vee D = D$$

つまり、

$$D = \begin{cases} 1 & (A = B = 0) \\ 0 & (A = 0, B = 1) \\ 1 & (A = 1, B = 0) \\ \text{uncertain} & (A = B = 1) \end{cases}$$

となります。A=1, B=1 のときに D の値が不定となることに注目してください。

この回路の D の値をメモリ（記憶）だとみなすことができます。A と B はメモリを書き換えるための信号線です。どういうことかということについて、以下に説明します。

通常の状態では A=1, B=1 とします。はじめは D の値は不定です。初期化のために「0」というデータを「書き込む」ためには、B=1 に保ったまま A=0 とします。すると、図 2.2 のように D の値が 0 に変化します。

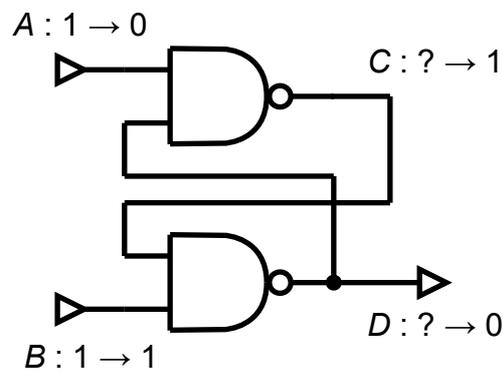


図 2.2 スタティック・メモリ回路に 0 を書き込む。

B を 1 に保ちながら A=0 とすれば、必ず D=0 となる。

はじめに、2 つある NAND ゲートの上の方を注目してください。この NAND ゲートの入力の一方が 0 になるので、出力 C は必ず 1 になります。すると、下の方の NAND ゲートは 2 つの入力が両方とも 1 になるので、出力 D が 0 になります。

さて、この A=0, B=1 の状態から、はじめと同じ A=1, B=1 の状態に戻したらどうなるでしょうか？図 2.3 のように、出力は D=0 のままで変化しないことに注目してください。上の方の NAND ゲートの入力のうちの 1 つ D は、今 D=0 になっているので、出力は C=1 のままです。一方で、下の方の NAND ゲートの 2 つの入力は両方とも 1 なので、その出力は D=0 でつじつまが揃っていることになります。

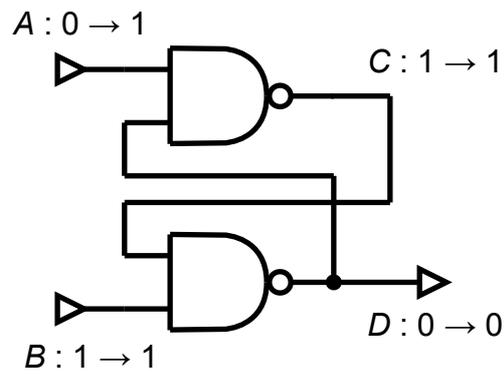


図 2.3 スタティック・メモリ回路に 0 を書き込んだあと、

$A=0$ から $A=1$ に戻しても、 $D=0$ のままである。

(書き込まれた値を憶えている！)

次に、このメモリに「1」という値を「書き込む」には、 $A=1$ に保ったまま $B=0$ とします。すると、図 2.4 のようになります。下の NAND ゲートの入力の一方が $B=0$ なので、その出力は $D=1$ になります。上の NAND ゲートの入力は $A=1, D=1$ となるので、その出力は $C=0$ となります。

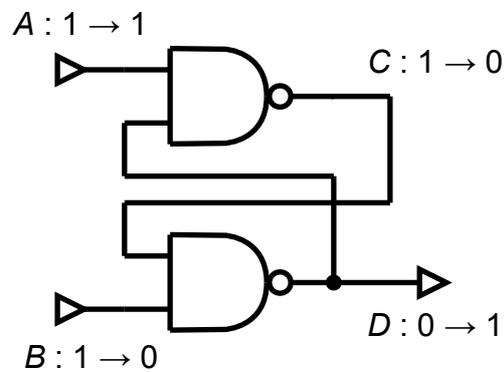


図 2.4 スタティック・メモリ回路に 1 を書き込む。

B を 0 にすれば、必ず D が 1 になる。

この後 $B=0$ から $B=1$ に戻しても、図 2.5 のように、 $C=0$ なので下の NAND ゲートの出力は $D=1$ のままです。

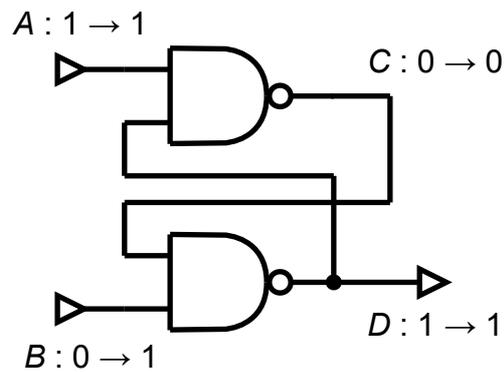


図 2.5 スタティック・メモリ回路に 1 を書き込んだあと、

$B=0$ から $B=1$ に戻しても $D=1$ のままになっている。

(書き込まれた値を憶えている！)

このような回路のことをフリップ・フロップ (flip-flop) 回路とかラッチ (latch) 回路と呼ぶことがあります。フリップ・フロップとは「ひょいとひっくり返す」といった意味合いの言葉です。ラッチ latch とは「掛け金」という意味の言葉です。

2-2 ダイナミック・メモリ Dynamic memory

ダイナミック・メモリ素子は、キャパシタ (コンデンサ) に静電気をためることでデータを記憶します。キャパシタの構造は、図 2.6 のように 2 枚の金属板を向い合せに近づけて配置したものです。

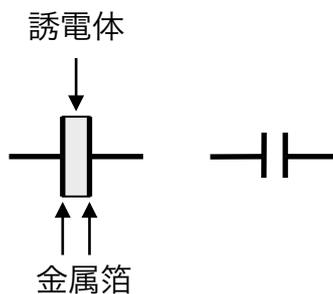


図 2.6 キャパシタ (コンデンサ) の構造 (左) と回路記号 (右)

このキャパシタと NMOS を図 2.7 のように組み合わせれば、これがダイナミック・メモリ回路として働きます。キャパシタに電荷がたまっているかどうかで 1 か 0 の値が表されます。

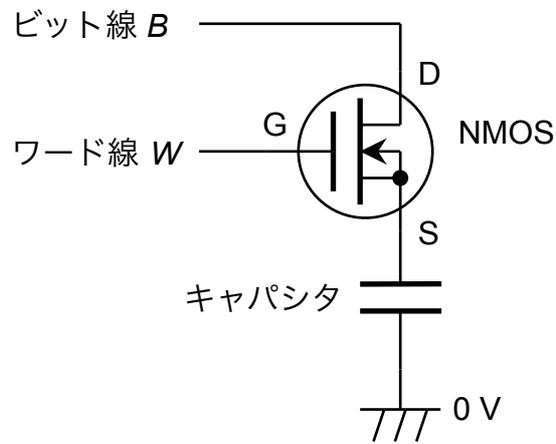


図 2.7 ダイナミック・メモリ回路

実際のメモリ素子ではワード線とビット線はマトリックス（行列）状に配線されます（図 2.8）。たとえば、4本のワード線と4本のビット線を使って、 $4 \times 4 = 16$ 個のメモリ素子を制御することができます。この一つ一つのメモリ素子のことは「メモリセル」と呼ばれます。

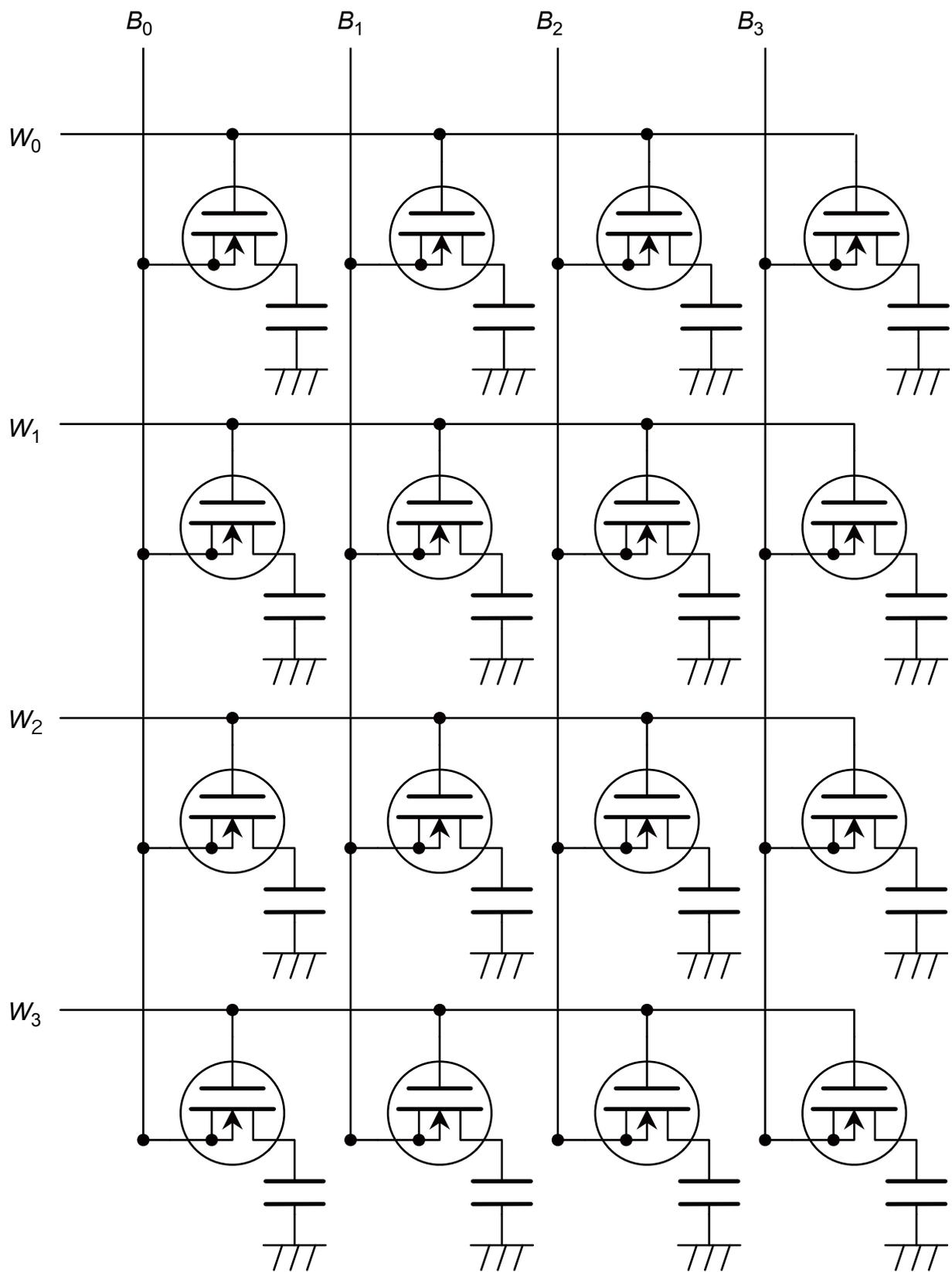


図 2.8 ダイナミックメモリ

ダイナミック・メモリに「0」という値を書き込むにはゲート電圧をかけた状態でビット線の電圧を下げてキャパシタから放電させます(図2.9)。この後にワード(ゲート)電圧を下げればビット線からキャパシタへは電流が流れにくくなるので、ビット線の値にかかわらずキャパシタは放電したままです。つまり、しばらくは書き込んだ値を憶えていることとなります(図2.10)。

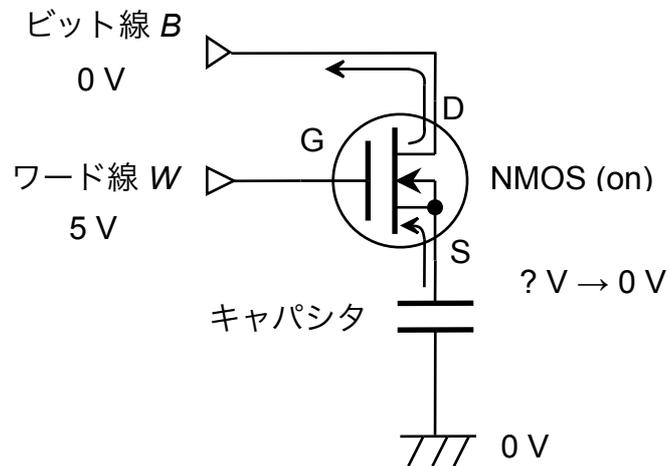


図2.9 ダイナミック・メモリ回路に「0」を書き込む。

ワード線 W を 5V とすれば NMOS が on 状態になり、
ビット線 B を 0V にすれば、矢印の向きに電流が流れて
キャパシタが放電する。

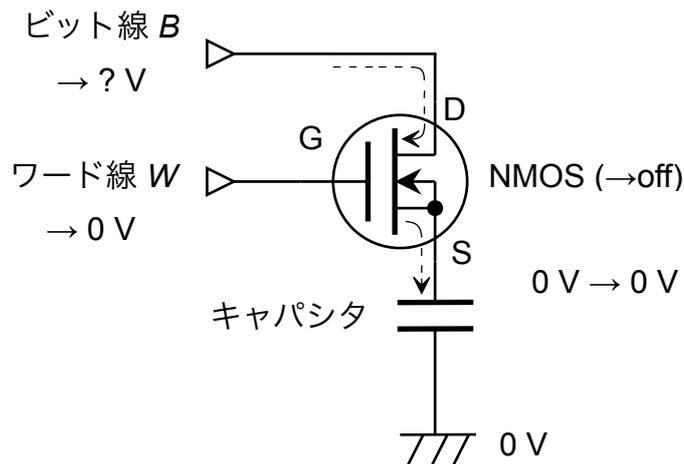


図 2.10 ダイナミック・メモリ回路は「0」という値を憶えている。

ワード線 W を $5V$ から $0V$ に戻してやれば、ビット線が $5V$ になったとしても、矢印の向きには電流が流れにくいので、キャパシタが放電された状態がしばらく保たれる。

逆に「1」という値を書き込むには、ワード線の電圧を上げてゲート電圧をかけた状態にし、ビット線の電圧の電圧を上げてキャパシタに充電します (図 2.11)。その後ゲート電圧を下げればキャパシタからビット線へは電流が流れにくくなり、やはりしばらく書き込んだ値を憶えていることとなります (図 2.12)。

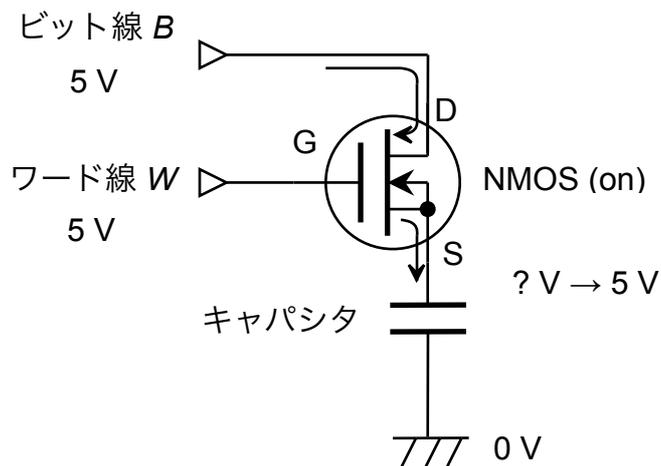


図 2.11 ダイナミック・メモリ回路に「1」を書き込む。

ワード線 W を $5V$ とすれば NMOS が on 状態になり、ビット線 B を $5V$ にすれば、矢印の向きに電流が流れてキャパシタが充電される。

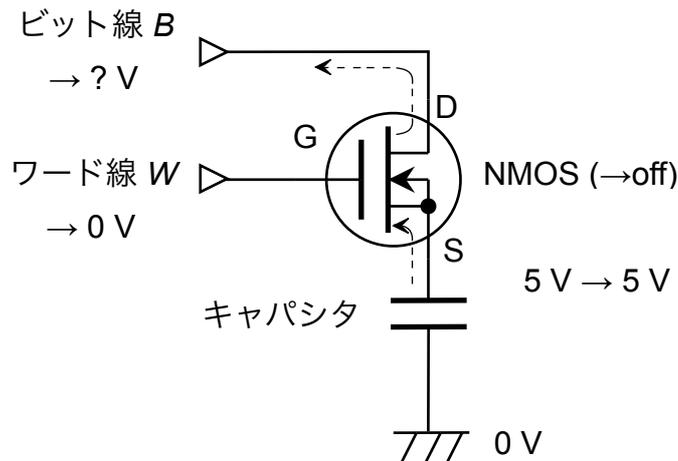


図 2.12 ダイナミック・メモリ回路は「1」という値を憶えている。

ワード線 W を $5V$ から $0V$ に戻してやれば、ビット線が $0V$ になっ
ても、矢印の向きには電流が流れにくいので、キャパシタの充電さ
れた状態がしばらく保たれる。

ダイナミック・メモリ回路から値を読み出すためには、ワード線電圧を上げて、その間にビット線に流れ出る電流を検出してやります。キャパシタに電荷がたまっていれば（記憶が「1」であれば）ビット線に電流が流れ出し（図 2.13）、キャパシタに電荷がたまっていなければ（記憶が「0」であれば）ビット線には電流が流れ出ません。ただしこのときにキャパシタは放電してしまうので、読み出したデータを書き戻す操作が必要になります。この動作のことをプリチャージ precharge と呼びます。

また読み出しを行わない場合でもキャパシタは徐々に放電するので、データを書き込んで一定の時間が経過したらデータを書き込み直す操作が必要になります。この操作のことをリフレッシュ refresh と呼びます。通常は 1 秒間に数万回 (数十 μs に一回) ていどリフレッシュ動作が行われます。

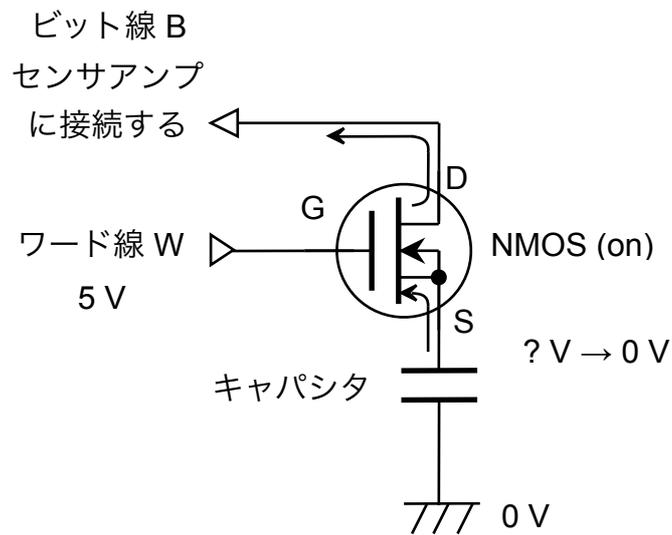


図 2.13 ダイナミック・メモリ回路からデータを読み出す。
 ワード線 W を $5V$ としてビット線 B を検出回路に接続する。
 キャパシタからビット線に電荷が流れ出せば
 記憶が「1」だったということがわかる。