

# CMOS

貴方が働けば私は休む  
貴方がパラなら私はシリ

CMOS(Complementary MOS)はNMOSとPMOSを相補的に接続する方式

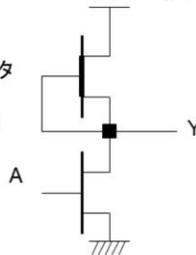
現在使われているデジタル回路の8-9割はCMOS  
80年代に急速に発達し、年間1.5倍に利用可能な素子数が増えた→ムーアの法則  
これによってコンピュータ、デジタル機器の大発達時代を迎えた

前回簡単に紹介したCMOSは、nMOSとpMOSを相補的に接続した回路構成です。相補的とは、pMOS,nMOSをペアにして入力を共有し、pMOSが直列接続のときはnMOSは並列接続に、pMOSが並列接続のときはnMOSは直列接続にする方法です。現在使われているデジタル回路の8-9割はCMOSです。CMOSは1980年代から急速に発達し、毎年チップ内に格納する素子数が1.5倍(18ヶ月で倍)になるという急成長を遂げました。この成長率をムーアの法則と呼びます。これによって、コンピュータは大発展を遂げ、デジタル回路はアナログ回路に置き換わって様々な用途に使われるようになりました。今回はこのCMOS回路をやや深く見て行きましょう。

## nMOSのみの回路

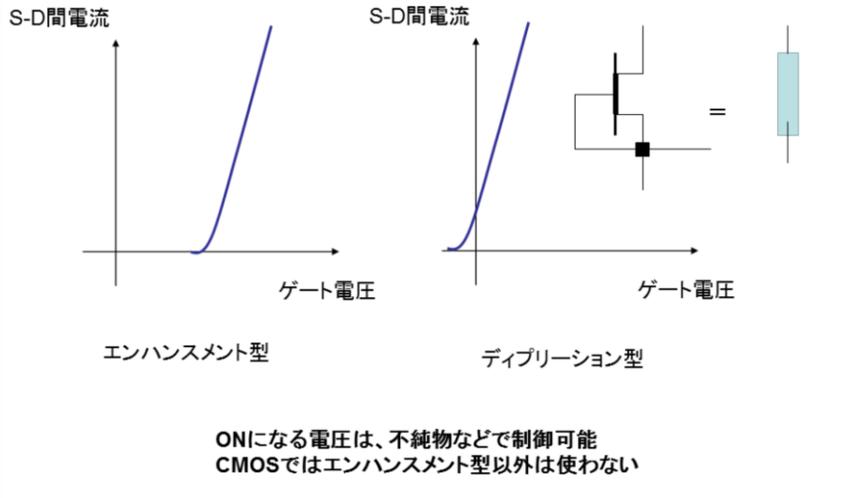
- 歴史的にpMOS→nMOS→CMOSで発展してきた
  - CMOSは両方必要なので最初作るのが大変だった
  - はじめてのマイクロプロセッサintel 4004, 8008はpMOSでできていた
- CMOSの方が高速で設計しやすい
  - メモリ回路はnMOSがかなりがんばった
  - 80年代にはほぼCMOSに置き換わった

nMOS回路のインバータ  
ディプレッション型を  
負荷抵抗の役割で使う



歴史的には、製造のしやすいpMOSのみでできた回路が最初に発達しました。世界初のマイクロプロセッサ4004をはじめとする初期のマイクロプロセッサはpMOSでできていました。しかしpMOSはマイナスの電源電圧を与える必要がありました。また、ホールは電子に比べて易動度といって動かし易さが小さいため、性能が低くなりやるいのです。このため、70年代の中ごろには、LSIの主流は全てをnMOSで作る方式に移りました。この方式は、エンハンスメント型のnMOS FETの負荷抵抗にディプレッション型のnMOS FETを抵抗として使います。

## ディプリーション型は抵抗として使える



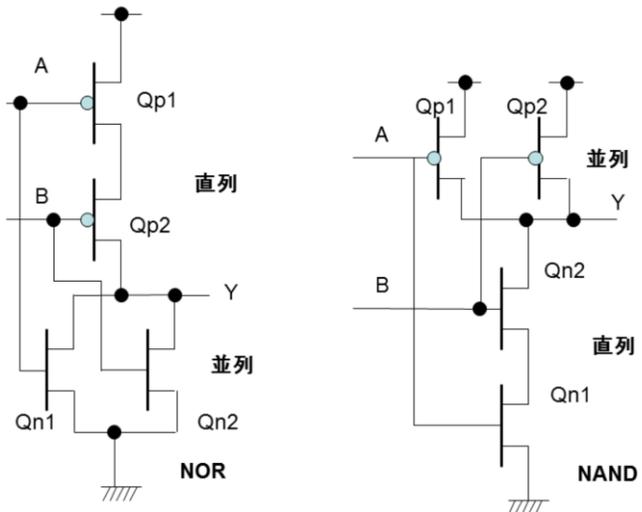
ディプリーション型はゲートとソースの電圧が0Vでもある程度電流が流れるので抵抗として使うことができます。nMOSのみで作るため、作りやすいという利点があり、80年代のはじめまで使われました。中でもメモリ回路はかなり後になってもnMOSが使われました。しかし、トランジスタを抵抗と使うために、出力がLレベルのときには常に電流が流れてしまいます。また、LからHへの変化が遅いという問題もありました。このため、序々にCMOSに置き換わりました。

## 相補的 (Complementary)

- 一つの入力がpMOS,nMOSのペアのGateに接続される
  - 片方がONならば片方はOFF
- pMOSが並列ならば、nMOSは直列
- nMOSが並列ならば、pMOSは直列
  - 電源からGNDまでのパスのどこかがOFFで切れている
- CMOSの基本的な構成法
  - Static CMOS
  - 前回までにNOT、NAND、NORはやった
  - 今日はもっと複雑なゲートの構成法を紹介する

では、CMOSの回路構成を見て行きましょう。CMOSの特徴は、ゲートがHにするとONになるnMOS、LにするとONになるpMOSという相補的(Complementary)なトランジスタを相補的に接続することにあります。まずnMOS,pMOSのペアを入力数分用意して、ゲートを共通にしてそれぞれの入力を与えます。次に回路の下半分にnMOSを、上半分にpMOSを配置し、nMOSが直列の場合、pMOSは並列に、nMOSが並列の場合、pMOSを直列に接続します。このようにすると、出力にはGNDと電源のどちらかがトランジスタを介してつながり、電源からGNDまでの通路のどこかは切れていることとなります。前回、NOT、NAND、NORを紹介しました。今回はより複雑な回路について紹介します。

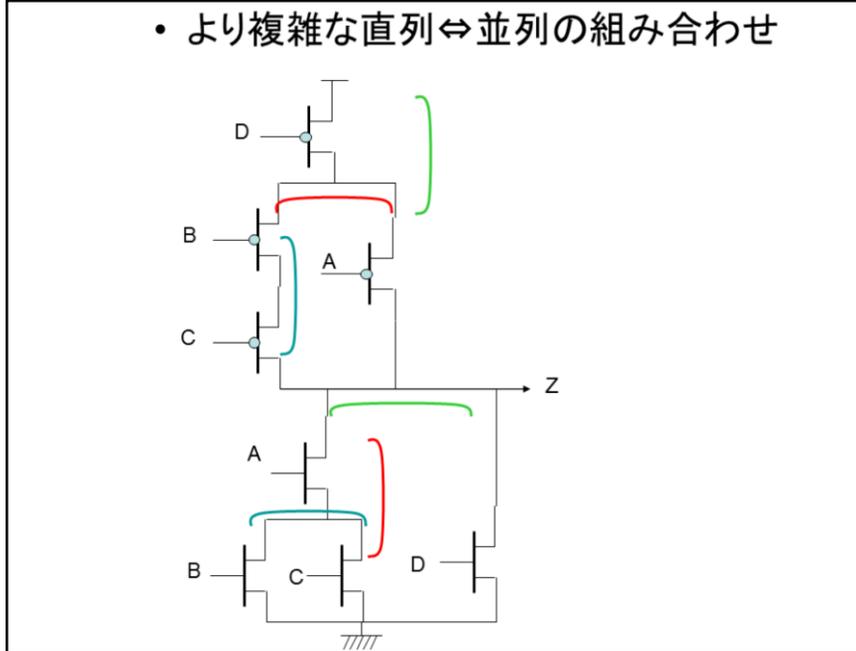
• 相補的の意味→ 直列⇔並列



直並列を組み合わせればもっと複雑なゲートができるのでは？

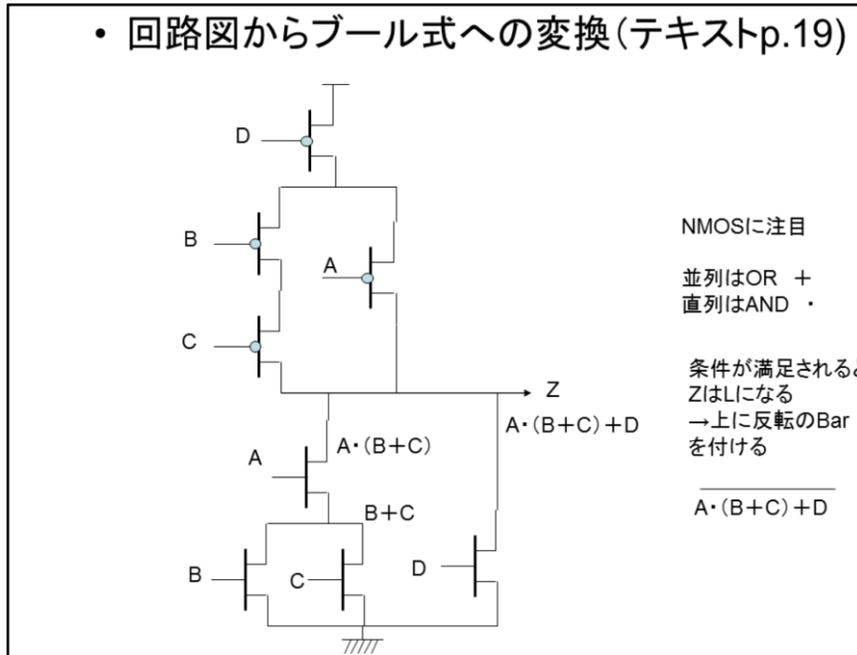
前回紹介したNORとNANDの回路を見てください。NORの方はnMOSが並列に繋がっているので、どちらかの入力がHになると出力YはGNDに繋がってLが出力されます。両方の入力がLのときだけYは電源と繋がってHが出ます。一方、NANDはnMOSが直列なので、両方HのときだけYがLになり、pMOSが並列なのでどちらかがLのときにはYがHになります。この方法を利用すればもっと複雑な回路も実現可能ならずです。

• より複雑な直列⇔並列の組み合わせ



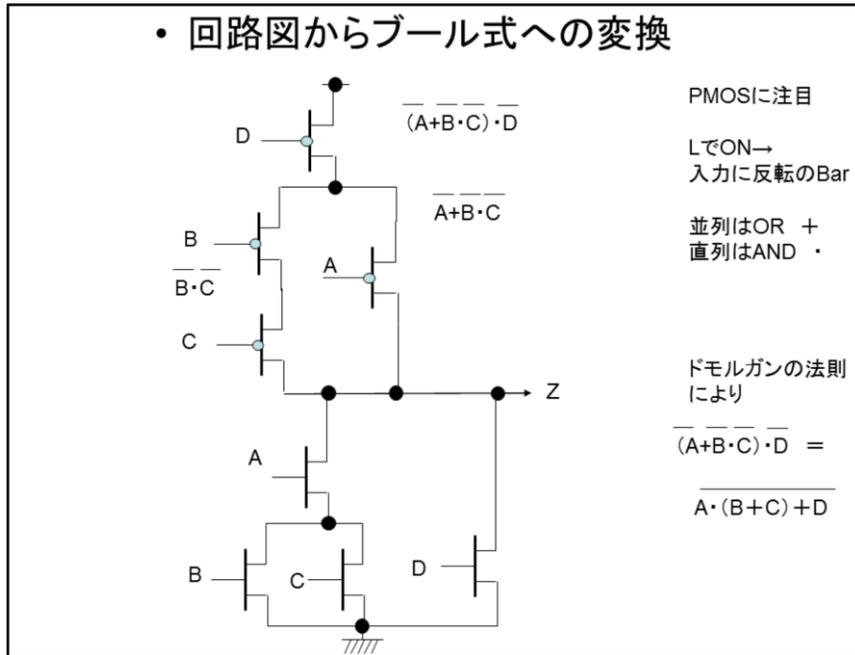
例えば、直並列を組み合わせるとこの図に示すような回路が実現できます。このようなゲートをCMOS複合ゲートと呼びます。この種の回路を解析する場合は、下半分のnMOSに着目する方法と、上半分のpMOSに着目する方法があります。ここでは下半分に着目しましょう。ZがGNDに落ちるためにはどのトランジスタがONになれば良いかを考えます。

• 回路図からブール式への変換(テキストp.19)



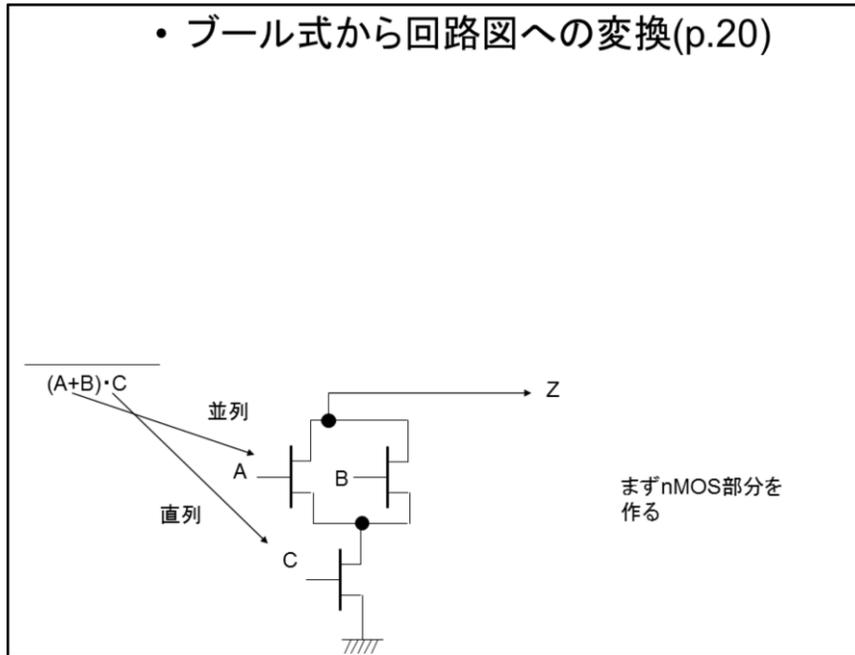
ZがGNDに繋がる条件は、以下のように求めます。並列の場合はORと考え、直列の場合はANDと考えます。この場合、BとCのORで、これに対してAがANDされることとなります。さらにDがORされます。この条件が満足されれば、ZはGNDになります。すなわち論理式は、ここに示すようになります。

## 回路図からブール式への変換



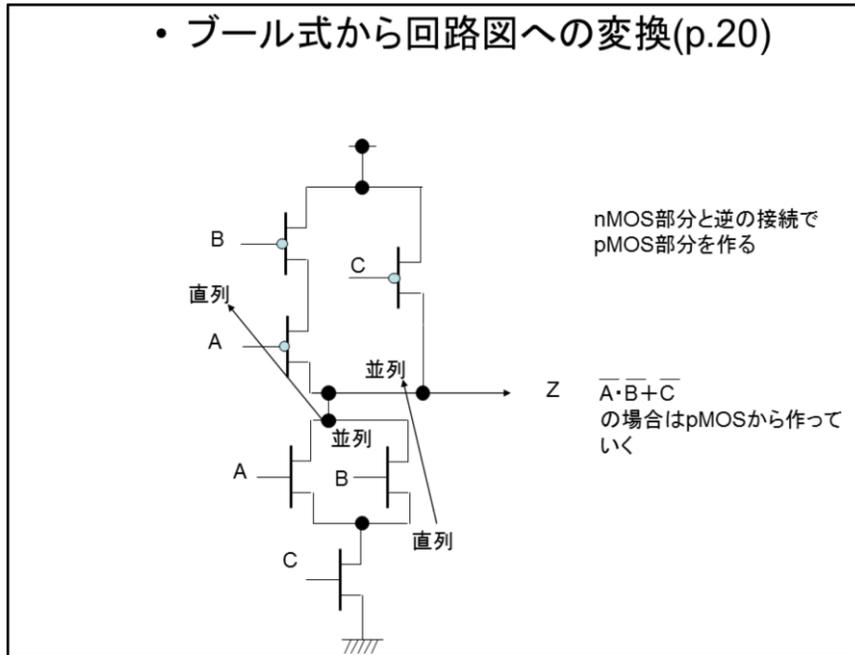
今度は上半分のpMOSに着目します。pMOSはLの時ONになるので、入りにバーが付くこととなります。あとは同じで、直列がAND、並列がORです。出来上がった式はさきほどと違うように見えますが、ドモルガンの法則により同じことがわかります。

• ブール式から回路図への変換(p.20)

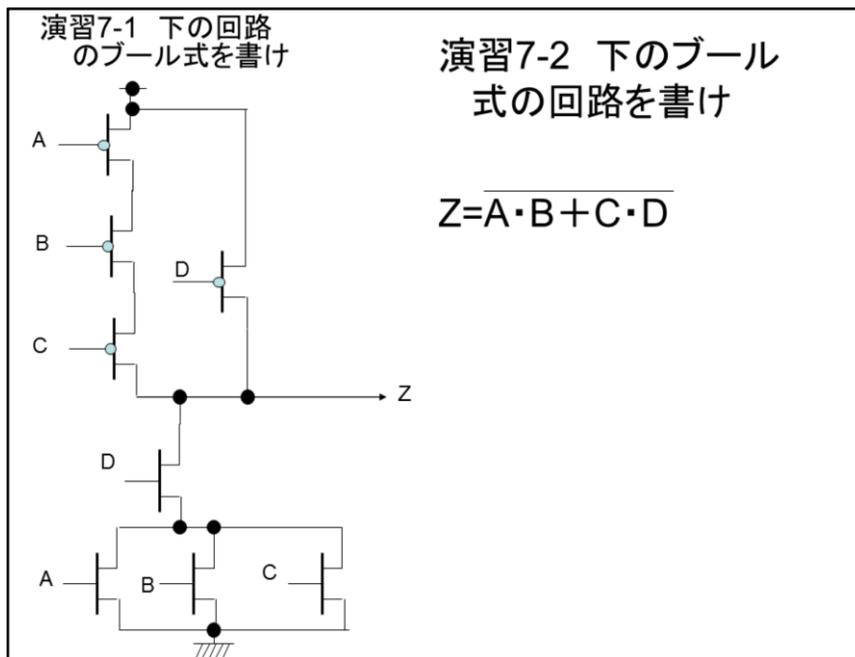


次にブール式から回路図に変換する方法を紹介しましょう。式の形が全体にバーが付いている場合は、nMOSから作っていきます。AとBはORなので並列に繋がります。これにCがANDされているので、直列にnMOSを繋がります。ここで、CとA、Bの並列接続の上下関係は逆でもかまいません。この図ではCがGNDに接続されていますが、AとBの並列接続がGNDに接続されていてもかまいません。

• ブール式から回路図への変換(p.20)



次に、nMOSの部分と逆の接続関係でpMOS部分を作ります。AとBは並列なので、pMOSを直列に接続します。Cは直列なので、AとBの直列に対して並列に繋がります。入力のバーに対するブール式が与えられた場合はpMOS部分をまず作って、これに対応したnMOS部分を作っていきます。

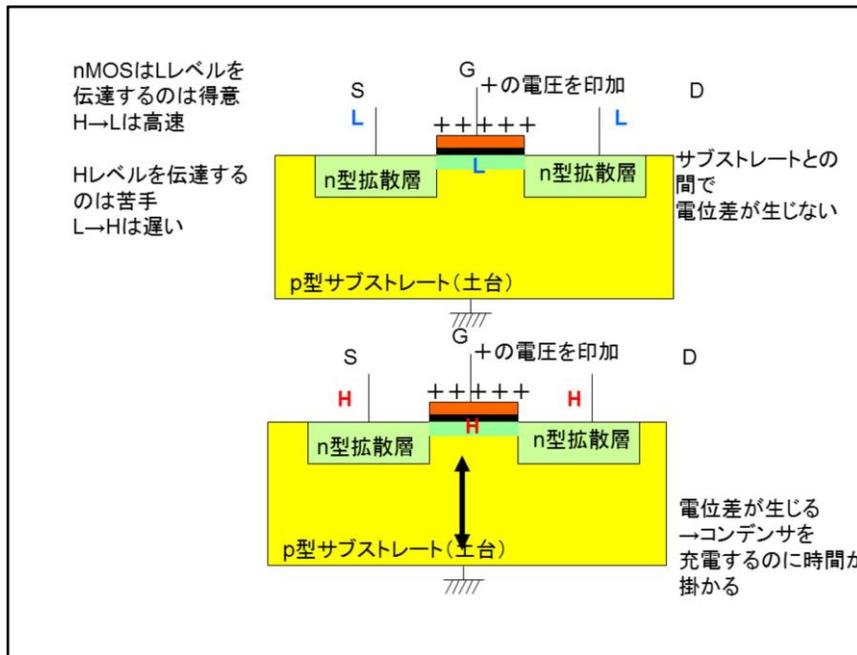


ではこの辺で演習をやってみましょう。演習7-1は回路からブール式に変換する問題、演習7-2はブール式から回路を作る問題です。

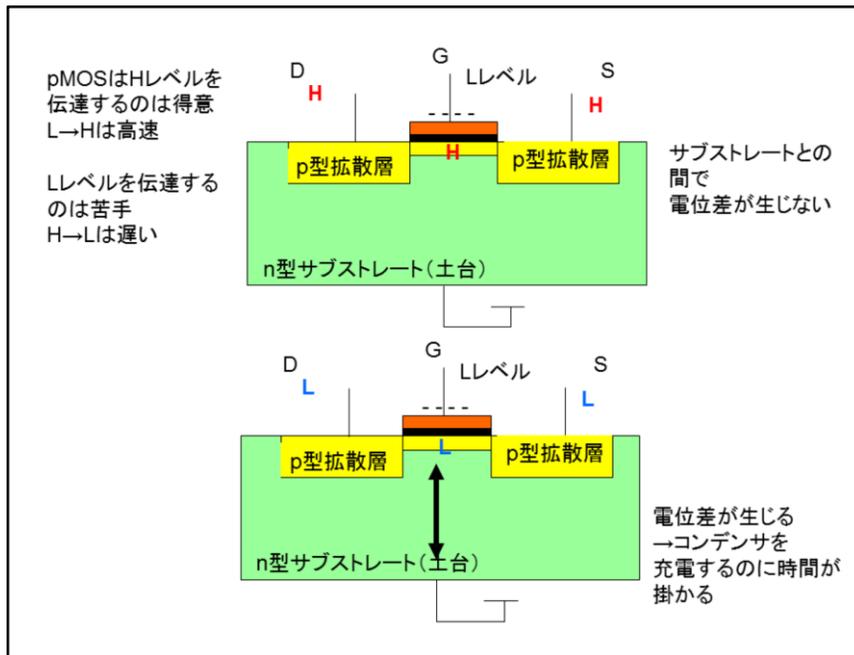
## なぜPMOSを上半分にNMOSを 下半分に使うのか？

- このため、必ず出力は反転になってしまう
- NAND、NORは構成可能だがAND、ORは構成できない
  
- 理由
  - PMOSはHレベルの伝達(L→H)が得意だがLレベルの伝達が苦手
  - NMOSはLレベルの伝達(H→L)が得意だがHレベルの伝達が苦手

さて、今まで紹介したCMOS回路は必ず上半分にpMOSを使い、下半分にnMOSを繋ぎました。このため、入力と出力の関係は必ずNOTが入ることになり、今までの複合ゲートは全体の式にバーが付いた形か、個々の入力にバーが付いた形のどちらかの論理式でなければ実現できませんでした。AND、ORなどは実現できません。これはなぜでしょう？理由はpMOSはHレベルの伝達、細かく言うとLからHへの変化の伝達は得意ですが、HからLへの変化の伝達は苦手です。逆にnMOSは、LレベルつまりHからLへの変化の伝達は得意で、LからHへの伝達が苦手という特徴があるからです。

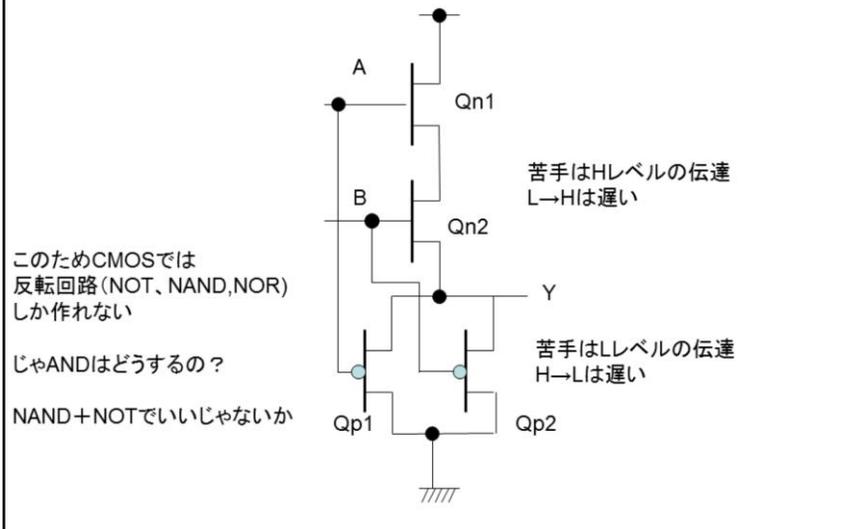


これは動作原理を考えると理解できます。nMOSのp型サブストレートはGNDレベルになっています。このため、S-DがLレベルになるときは電位差が生じません。一方、S-DがHレベルになる際は、電位差が生じます。この時、S-Dとサブストレートの間の容量(コンデンサ)に電荷を充電しなければS-D間はHレベルになることができず、このために時間が掛かってしまいます。



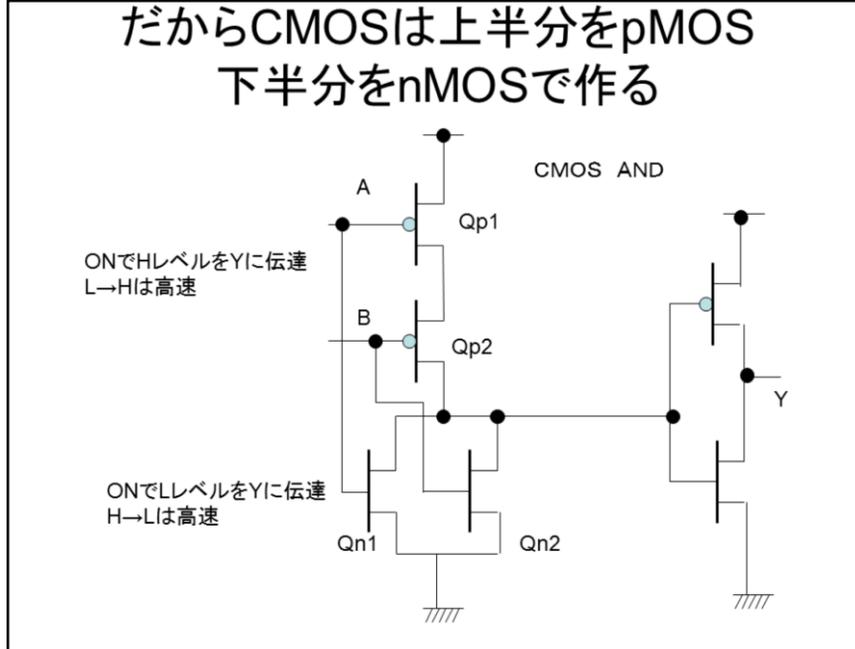
pMOSの場合、nMOSとは逆にサブstrateは電源電位になっているので、S-D間  
 がHレベルのときには電位差が生じません。一方で、S-D間がLレベルになる場合  
 は、容量に充電する必要があり時間が掛かってしまいます。

## このAND回路は使い物にならない



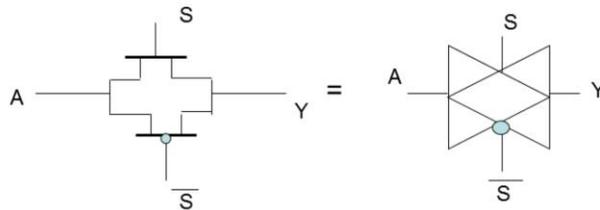
したがって、このようにnMOSに電源電位を伝達させ、pMOSにGND電位を伝達させる方法は、互いに苦手な電位を伝達することになり動作速度が遅くて使い物になりません。

## だからCMOSは上半分をpMOS 下半分をnMOSで作る



では、CMOSでANDやORはどのようにして作れば良いのでしょうか？NANDやNORの後にNOTゲートを付ければよいのです。CMOSのNOTゲートはトランジスタ2個でできますし、高速です。このようにCMOSは、NOTゲートを惜しげなく使って論理を合わせることで、nMOSでLレベルを伝達し、pMOSでHレベルを伝達する原則を維持します。

• トランスミッションゲート (p.20)

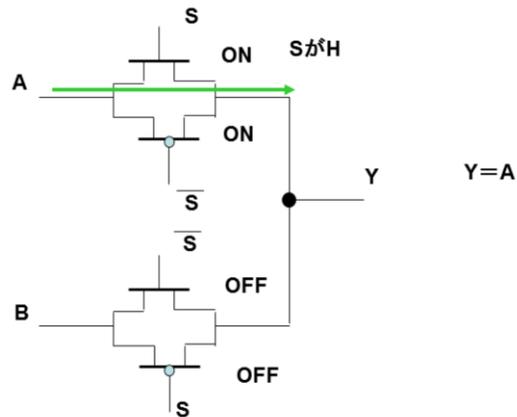


pMOSがONのときnMOSもON  
 pMOSがOFFのときnMOSもOFF → ON/OFFのスイッチ  
 相補的なCMOSと全く逆の動きをする

なぜ二つ共ON? → pMOSはHを通すのが得意、nMOSはLを通すのが得意  
 力を合わせれば両方共うまく通過できる  
 A→Y、A←Yの両方向の転送が可能

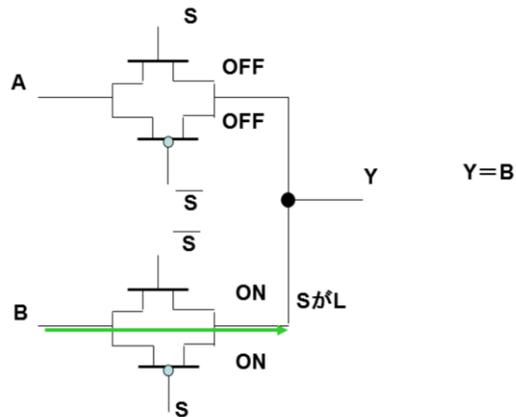
さて、ここまでで通常の相補的な使い方を紹介しました。これと全く逆な利用法があります。nMOSとpMOSのソースドレイン同士を接続し、ゲートには片方にSを与えたとすると、もう片方にはSの反転信号を与えます。この場合、pMOSがONのときはnMOSもONになり、nMOSがOFFのときはpMOSもOFFになります。つまり、両方が必ず同じ状態になるのです。このことにより、ONになった時はAとYが接続され、OFFの時は、AとYが切り離されます。これをトランスミッションゲート、あるいはトランスファゲートと呼びます。なぜnMOSとpMOSの両方必要か?という点、先ほど解説したとおり、それぞれ通すのが得意なレベルが違うからです。HレベルはpMOSが、LレベルはnMOSが主に動作することで、両方のレベルを高速に通すことができます。トランスミッションゲートは、双方向である点にご注意ください。この双方向性を利用してFPGAの配線用スイッチとして使うことができます。

## マルチプレクサ



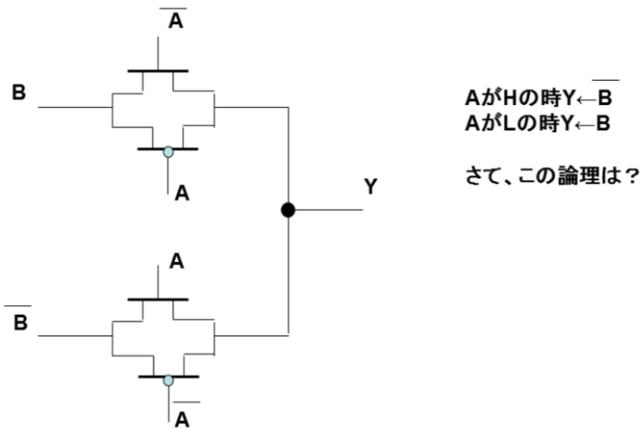
トランSMissionゲートを使うとマルチプレクサ(データセレクタ)を簡単に作ることができます。図に示すようにSをHにすると上のトランSMissionゲートがONになって、AがYに出力されます。

## マルチプレクサ



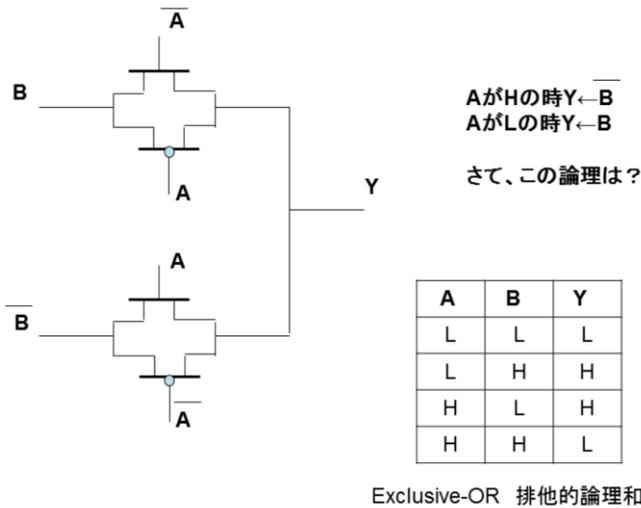
一方、 $S$ をLにすると $B$ が $Y$ に出力されます。 $S$ の反転信号を作るNOTゲートを含めてもトランジスタ6個で実現できるのが特徴です。後の授業で紹介するD-FFの中身などに使います。

例題 (p.22)



ではここで、テキスト22ページの例題をやってみましょう。これは先ほどのマルチプレクサの応用です。

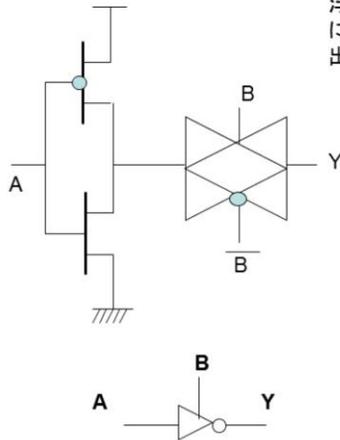
## 例題2.2 (p.22)



A=Hの時にはYにBが、A=Lの時にはYにBの反転信号が表れます。これを真値表にしてみると、Exclusive-OR(排他的論理和)ができてくるのがわかります。排他的論理和は複合ゲートで作ることができず、普通のゲートの組み合わせで作ると多数のゲートが必要であることから、この方法にはメリットがあります。このようにトランミッションゲートを使って論理回路を作ることパストランジスタロジックと呼びます。パストランジスタロジックは、排他的論理和の場合はコストの小さいメリットが際立っているため、使われることがありますが、伝達レベルが劣化する問題点があります。通常のCMOSは、出力には電源、GNDレベルがトランジスタを通じて表れます。しかし、パストランジスタロジックの出力は入力そのまま出てくるため、複数段を経過するとレベルが劣化してしまうのです。

## 3ステート出力

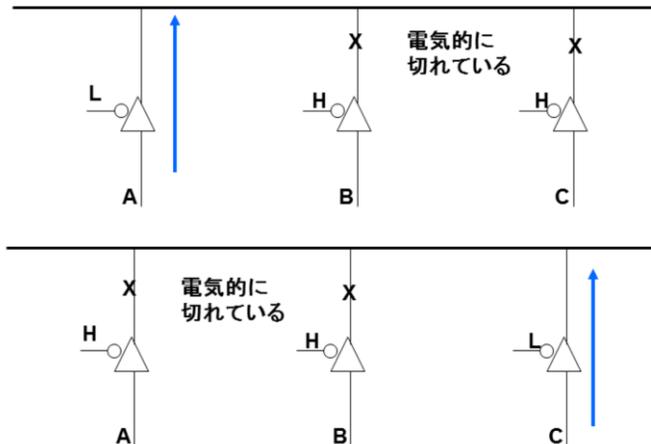
B=Lの時、Yは電氣的に浮いた状態(ハイインピーダンス状態)になる。L/Hのほかはこの状態を持つ出力を3ステート出力と呼ぶ



A	B	Y
L	L	Hi-Z
L	H	H
H	L	Hi-Z
H	H	L

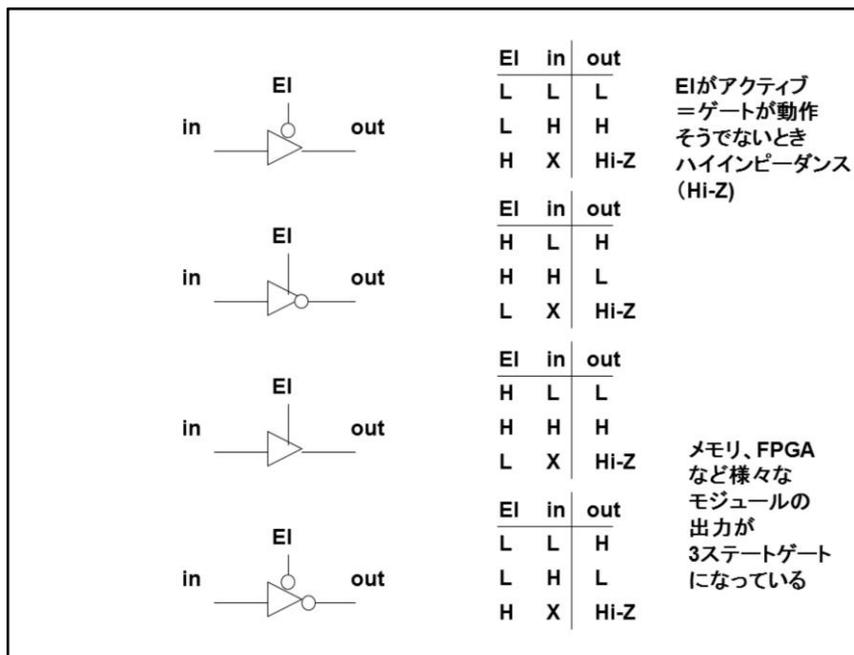
トランSMissionゲートのpMOS, nMOSが両方OFFの状態では、出力は電氣的に浮いた状態になります。この状態は、出力インピーダンスが高くなることからハイインピーダンス状態とも呼びます。にこれはLとHのほかの第3の出力ということで3ステート出力あるいはトリステート出力と呼びます。この図のようにNOTゲートの出力にトランSMissionゲートを付けることで、3ステートゲートを作ることができます。この例ではB入力がLの場合は、出力はハイインピーダンス(真理値表ではHi-Zと表します。Zはインピーダンスをしめすことを思い出しましょう)になり、B入力がHの時は普通のNOTゲートとして働きます。MIL記号法ではこのように横から3ステート状態にするかどうかを切り換える制御線を付けて示します。

### 3ステートゲートのバス



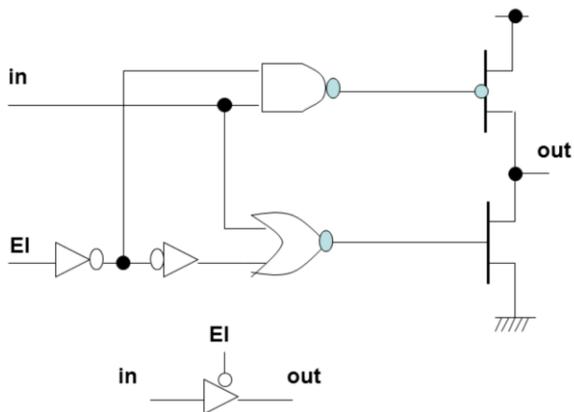
EIをLにした入力バスにデータを送ることができる  
同時に二つのゲートのEIをLにするとデータが衝突して過電流が流れる

3ステートゲートを利用するとバスを作ることができます。通常のデジタル回路は一つの信号線に複数の出力を繋げることはありませんが、場合によっては、複数の出力を繋いで、時分割で信号を載せることにより、信号線を共有して効率良く利用することがあります。このような信号線は通常束にして一定の大きさのデータを送ることが多く、バス (Bus) と呼ばれます。3ステートゲートの出力を複数接続し、どれか一つを除いて全てハイインピーダンス状態にします。この場合、Aのみ制御線をLにし、B,CはHにします。このことにより、信号線上にはAからの信号のみが載ります。今度はCの制御線をLにして、他をHにすれば、Cの入力がバス上に載ります。このようにバスはA,B,Cのどの信号を載せることもできるのですが、複数同時に制御線をLにすると出力同士が衝突して、トランジスタ間に過電流が流れてしまいますので注意が必要です。

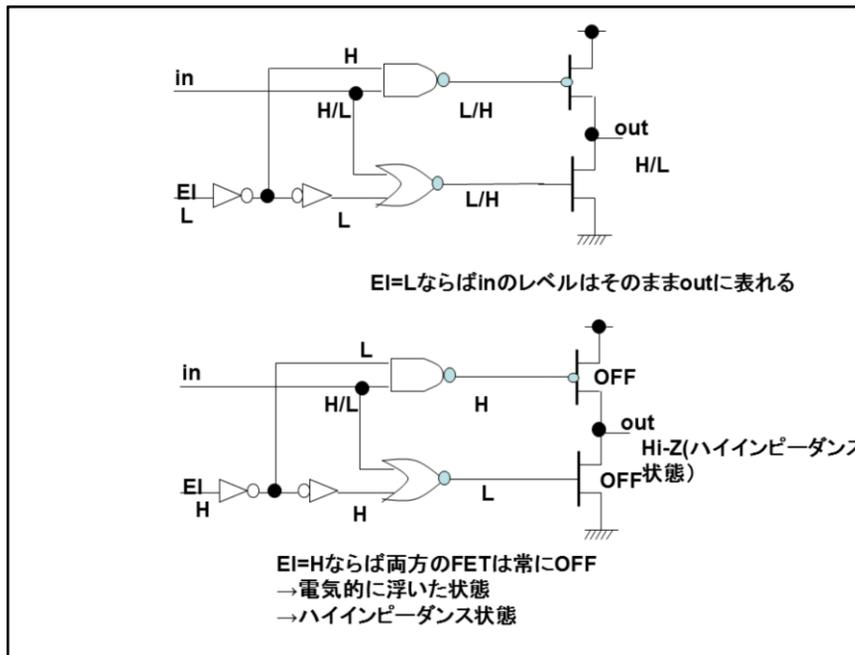


様々なスリーステートゲートの例を示します。制御線はEI(Enable Input)という名前を使っています。EIをアクティブにするとゲートは働き、そうでない場合は出力がハイインピーダンス状態になります。

## 3ステートゲート



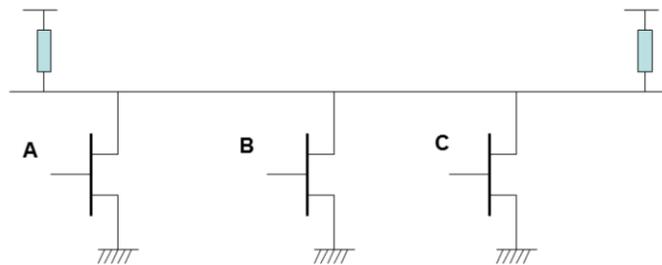
3ステートゲートは、NAND、NOR、NOTなど通常のゲートを使ってつくることもできます。これがその例です。



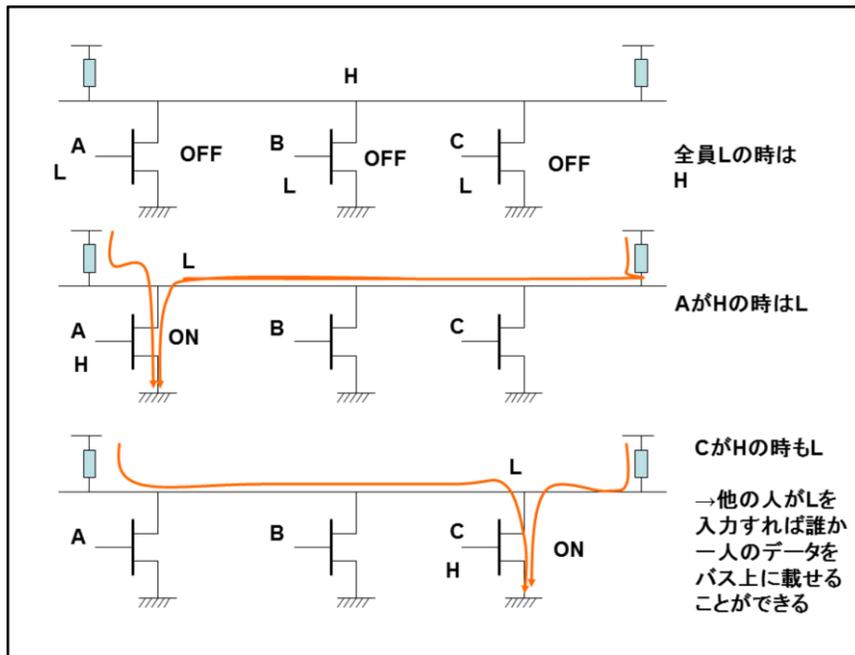
この例ではEI=Lならば、NAND、NORは共にNOTの働きをするので、inのレベルはそのままoutに繋がります。一方、EI=Hの場合、両方のトランジスタがOFFになり、電氣的に浮いた状態、すなわちハイインピーダンス状態になります。この方式はトランスミッションゲートを使う方法に比べてトランジスタ数が多く必要ですが、出力レベルが電源、GNDから経由するトランジスタ数が1なので、電氣的に強いです。トランスミッションゲートを使うとどうしても複数個のトランジスタを経由することになります。

## オープンドレイン(オープンコレクタ)

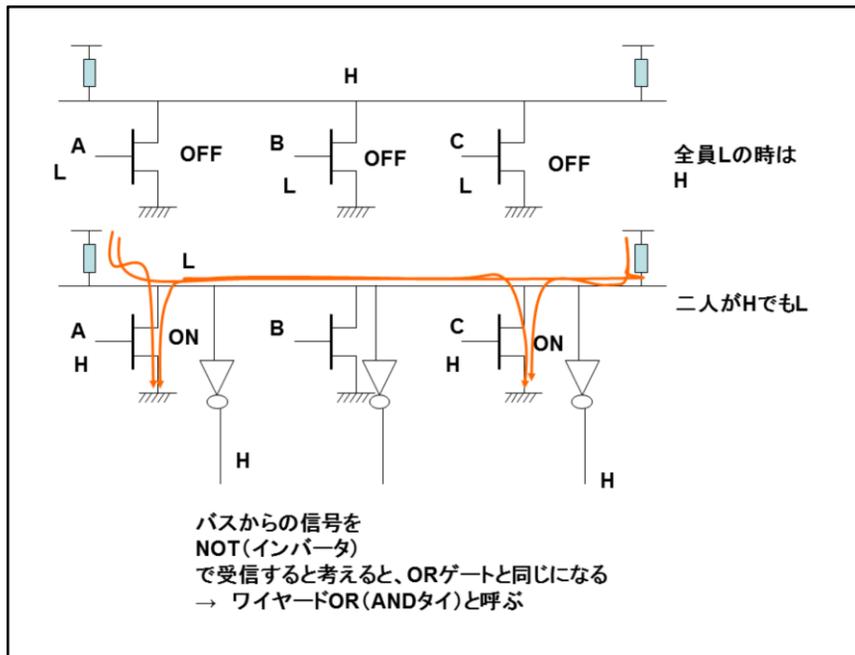
- pMOSをとっばらってしまったってドレインを開けっ放しにする
- 負荷抵抗を付ける→しかし、バスはどっちみちターミナル抵抗が必要



もう一つバスを作る方法を紹介します。CMOSの原則を崩して、pMOSを取ってしまった、その代わりに抵抗で電源と接続します。ドレインが空くので、オープンドレインと呼ばれます。複数のドレインを共通の抵抗に繋ぐことにより、複数の出力を繋いでバスを作ることができます。筐体の背面などに使うバックプレーンバスでは、バス上を伝搬するデジタル波形が反射によって乱れるのを防ぐため、抵抗を付けます(この辺の話は分布定数線路という理論があるのですが情報工学科ではやらなくてもいいだろう)。このような抵抗を終端抵抗(ターミナル抵抗)と呼びますが、これを流用することができます。



オープンドレインのバスは全トランジスタがOFFの時は抵抗から電源のレベルが伝わりHとなります。どれか一つのトランジスタのONになる(つまりゲートがH)と、そこに電流が流れてレベルがLになります。すなわち、他の全員がゲートをLにしておいて、一人だけゲートにレベルを与えると、それがバスに反映されます。



このバスのレベルをNOTゲートを介して受け取る場合は、バス全体が、どれか一つがHの時、出力がHになる、ORゲートとして働きます。そこで、このようなバスをワイヤードOR(またはANDタイ)と呼びます。ワイヤードORはバスの制御線(ハンドシェイク線)や分散的な優先順位決定回路(アービタ)に用いられます。

## オープンドレイン vs. 3ステート

- オープンドレインの良い所
    - 出力がぶつかってもワイヤードORになるだけ
    - 負荷抵抗はターミナル抵抗と兼用にできる
  - 3ステートゲートの良い所
    - pMOSを使うのでL→Hが高速
    - nMOSがONでもpMOSがOFFなので低消費電力
    - 負荷抵抗が不要
- バックプレーンバスにはオープンドレイン、他は3ステートゲートが使われる

ではバスを作る際、どちらの方法を使えばいいでしょう？両者の利点をまとめておきます。オープンコレクタの良い所は、制御を誤って出力が競合しても、ORが行われるだけで、不具合が生じない点で、これは何が刺さるか分からないバックプレーンバスに向いています。負荷抵抗は必要ですが、終端抵抗と兼用にできるのでバックプレーンバスに有利です。一方、3ステートゲートは、pMOSを使ってCMOS構造にするため、高速で消費電力も小さく、負荷抵抗も必要ないです。ただし、制御を誤って出力を衝突させ、片方がLレベル、片方がHレベルを出そうとすると過大電流が流れてしまいます。このため、3ステートゲートは、基板上のバスなど、動作が完全に制御できる場合に使われます。

## 今日のポイント

複合ゲートの論理式を求める方法

nMOSに注目

並列接続はOR +、直列接続はAND ・で結んで、式を作って上に反転のBarを付ける

ブール式から複合ゲートの回路を描く方法

論理式の上にBarが付く形に変換

nMOSのゲートをANDは直列、ORは並列に繋ぐ。

これと逆の関係になるようにpMOSを繋ぐ。

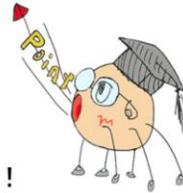
トランスマッションゲート

ONの場合は接続、OFFの場合は切れる。

3ステートゲートにも使うことができる。

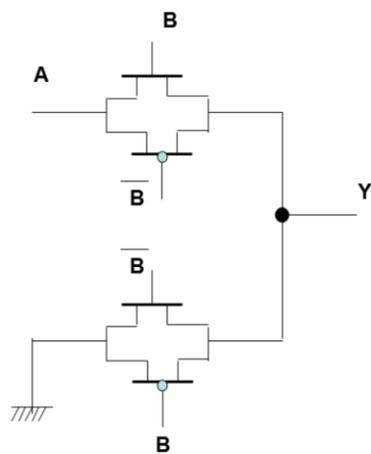
バスが作れる。

バスを作る方法としてはオープンドレインもあるよ！



今日のポイントをインフォ丸が示します。

### 演習7.3



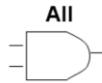
真理値表を書け

ではこの演習問題をやってみましょう。

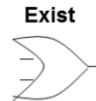
## おまけ

- 計算機基礎で習った基本ゲートとMIL記号法

ゲート:論理回路の基本素子  
論理回路はLとHの二値  
ゲートの種類も基本的には2種類



全ての入力がアクティブならば  
出力がアクティブ



入力のどれか一つでも  
アクティブならば  
(アクティブが存在すれば)  
出力がアクティブ



論理的な意味はない  
論理レベルを伝達

以降はおまけです。計算機基礎で習ったはずです。忘れた人はスライドを見て思い出してください。

## アクティブLとアクティブH

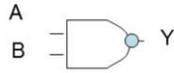
アクティブL: Lレベルに注目している(意味があると考える)

アクティブH: Hレベルに注目している(意味があると考える)

MIL記号法ではアクティブLの信号線に○印を付けて示す

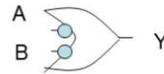
# MIL記号法

この二つのゲートは実体は同じだが意味が  
違う→これを書き分けるのがMIL記号法



入力が共にHならばLを出力

A	B	Y
H	H	L
他		H

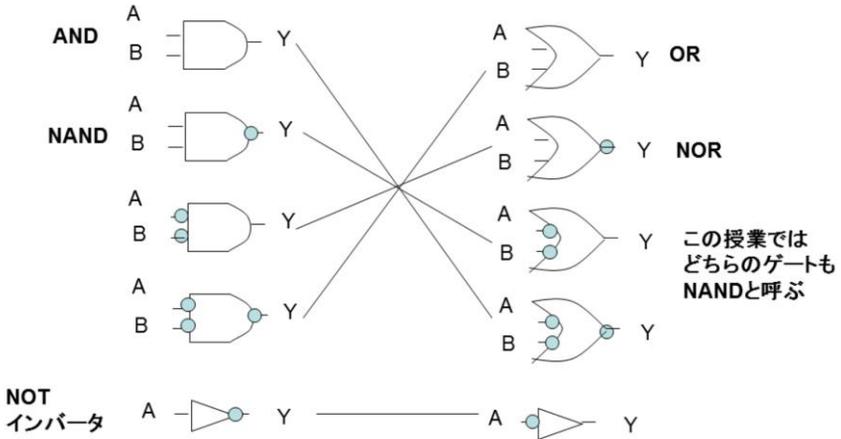


入力がどちらかがLならばHを出力

A	B	Y
L	-	H
-	L	H
他		H

A	B	Y
H	H	L

# 基本ゲート



All⇔Exist、入出力のアクティブL ⇔ アクティブHを入れ替えると同じゲートになる: ド・モルガンの法則