



## No.10 CMOSLSI の構造と動作(ロジック LSI)

講師：小川 公裕 Email: Ogawa-Kimihiro@success-int.co.jp

サクセスインターナショナル株式会社

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



### CMOSLSI の構造と動作(ロジック LSI)

#### 目次

|                                   |    |
|-----------------------------------|----|
| CMOSLSI の構造と動作(ロジック LSI) .....    | 1  |
| 1 前書き .....                       | 1  |
| 2 CMOS 基礎知識 .....                 | 1  |
| 2.1 MOSFET とは .....               | 1  |
| 2.2 CMOS の利点 .....                | 2  |
| 2.3 2進数のおさらい .....                | 4  |
| 2.4 SI 接頭文字 .....                 | 6  |
| 3 LSI の基本知識 .....                 | 6  |
| 3.1 LSI の構造 .....                 | 6  |
| 3.2 アナログとデジタル .....               | 8  |
| 4 半導体デバイス .....                   | 10 |
| 4.1 半導体デバイスの例 .....               | 10 |
| 4.1.1 半導体を使っている商品例 .....          | 10 |
| 5 電気・電子製品 中身 .....                | 12 |
| 5.1 スマートフォン .....                 | 12 |
| 5.2 自動車 .....                     | 13 |
| 5.3 コンピュータ .....                  | 14 |
| 6 半導体の微細化とドライブ製品の変遷 .....         | 15 |
| 6.1 半導体デバイス開発の牽引役 .....           | 16 |
| 7 システム LSI の構成デバイス .....          | 17 |
| 7.1 標準ロジック (Standard Logic) ..... | 17 |
| 7.2 ロジック機能回路の例 ~ 全加算器 .....       | 18 |
| 7.3 クロック .....                    | 20 |
| 7.4 プロセッサ (Processor) .....       | 21 |
| 7.4.1 CPU .....                   | 21 |
| 7.4.2 GPU .....                   | 22 |
| 7.4.3 AI 向けプロセッサ .....            | 23 |

## CMOSLSI の構造と動作(ロジック LSI)

抵抗をオープン(断線)状態、低抵抗をショート(短絡)状態と見做せば、簡単に言ってスイッチの役割を果たすことが出来る。現在 MOSFET の使い道は、ほとんどの場合 CMOS 回路を構成する事であり、CMOS の用途にはアナログ回路もあるが、その主たる用途はロジック回路であり、その機能はスイッチとして理解すればよい。デジタル (Digital) 回路、ロジック (Logic) 回路、論理回路 等の言葉は全て同じものを指すと考えて構わない。

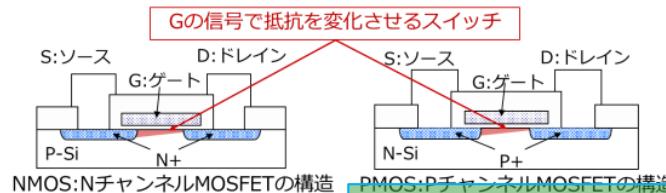


図 2.1-1 MOSFET トランジスタ

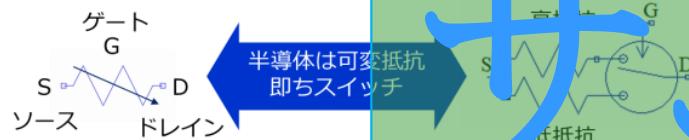


図 2.1-2 MOSFET はスイッチ

## 2.2 CMOS の利点

冒頭でも触れたように CMOS が勝ち残った最大の理由は 電源→グラウンド 間に直流電流が流れず無駄な消費電力がほとんど無いことである。LSI の素子数があまり多く無く、かつ携帯用機器が全盛では無かった頃は問題にならなかったが、現在の様に 1 チップに 1 千億個ものトランジスタが搭載されるようになった状況では、トランジスタ 1 個ずつの僅かな無駄消費電力も全体としては膨大なものとなってしまう。仮に 1 トランジスタ当たり  $1n[A] = 10^{-9}[A]$  と言う極小電流でも  $1n[A] \cdot 千億 = 10^{-9} \cdot 10^{11} = 100[A]$  という、とんでもない電流量になってしまふ。さらに携帯機器が数多く出回っている現在では、バッテリーがすぐに上がつてしまい使い物にならない。

## CMOSLSI の構造と動作(ロジック LSI)

次は 図 5.2-1 に自動車の制御系の機能一覧を示す。これら制御系は全て LSI が機能を実現している。この図を見ると、どうして、2022 年の様に半導体不足で自動車が造れなかつたのかがよく分かるであろう。



## 5.3 コンピュータ

図 5.3-1 に従来からの一般的なコンピュータの構成を示す。「一般の」と書いたのは 2020 年前後から新しいタイプのプロセッサが続々と提案され誕生しているからである。後で説明するが、これらの新しいタイプは主に HPC (High Performance Computing) や AI (Artificial Intelligence) を志向したものである。

従来のコンピュータは以下列挙するデバイスから構成されており、HDD (Hard Disk Drive) 以外は LSI である。また HDD 自体にもモーター等機械的部品以外に多くの半導体デバイスが使われている。

- CPU (Central Processing Unit) : 中央制御装置
- ALU (Arithmetic Logic Unit) : 算術論理演算装置

対応のため、下位ビットからの桁上げを受けて、上位ビットへの桁上げを出力するよう設計されている。 $C_{n-1}$  がある一つ下のビットからの桁上げ入力、 $C_n$  がそのビットから一つ上のビットへの桁上げ出力である。真理値表は入力夫々に 1, 0 が入った場合に出力がどうあるべきかを示すものである。少々面倒だが、真理値表の各行で 2 進数の加算が正しく行われているか確認することは難しくないだろう。例えば、図 7.2-1 の黄色くした行に関して言えば、 $1+1 \& \text{ 桁上げ } 0 \Rightarrow \text{ 和が } 0 \& \text{ 桁上げ } 1$  という事を示している。

この全加算器を論理回路で組む場合、排他的論理和を定義するのが便利である。図

7.2-2 にそのシンボルと基本論理回路で組んだ内部回路を示す。

$$\text{排他的論理和 } S = A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$$

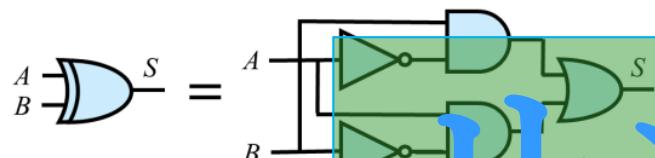


図 7.2-2 排他的論理和のシンボルと論理回路

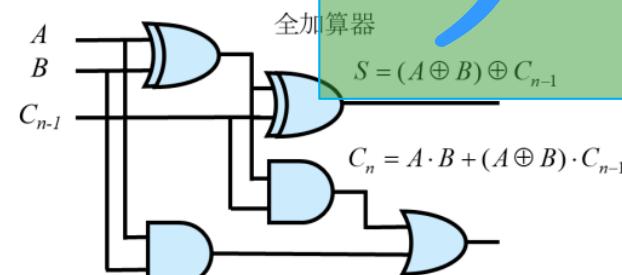


図 7.2-3 全加算器

排他的論理和を使って構成した全加算器を 図 7.2-3 に示す。この論理回路が実現する論理関数を図中に示しているので、図 7.2-1 の真理値表などを参考に各自で正しいことを確認して欲しい。

さて、図 7.2-3 は 1 ビット分の回路であるが、実際には 16[b]とか 32[b] 等の加算器が必要なので、これを  $n$  ビット数分繋げて実現する。この様子を 図 7.2-4 に示す。

的には同じものである。

さて、3 つの DFF を 図 7.3-1 のように接続すると、説明は省略するが（面倒だが、丹念に信号を追いかければ理解出来るはず）、Q1 が 2 進、Q2 が 3 進、Q3 が 5 進のカウンターになっている。カウンターとは CK の数を数えるものである。下のタイムチャートを見れば、夫々が CK を 2 個、3 個、5 個ずつ数えている事が分かる。

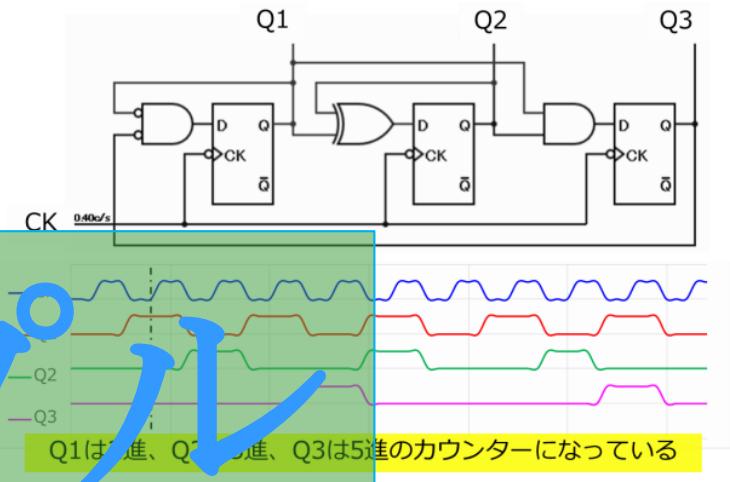


図 7.3-1 同期式カウンターの例

ここまで簡単な例で説明したが、同期式デジタル LSIにおいては、CK で刻まれる各時刻毎に論理関数で演算を進めて行くことで様々な機能を実現している。7.2 で説明した入出力が静的な関係で決まる論理回路を組み合わせ論理 (Combinational Logic) 7.3 で説明した CK で次々と状態を変えて行く論理を 順序論理 (Sequential Logic) と呼ぶ。

## 7.4 プロセッサ (Processor)

### 7.4.1 CPU

5.3 でコンピュータの一般的な構成を説明した。ここで言うプロセッサは CPU (Central Processing Unit) でコンピュータの司令塔である。そしてこれは、同期式回路の典型的なものである。

し、誘電率の高い材料が用いられる。

図 7.5.1-1 に DRAM のセル 1 ビット分を示す。図 7.5.1-1 の右のグラフに示すように、DRAM は微細化の恩恵を受け難く、最近の性能向上は CPU の性能向上について行けない。その理由は

- ・ キャパシタ容量をある程度確保しなければいけないため、微細化に限界がある
- ・ 特殊なプロセスなので最先端 CMOS ロジックとは別に開発する必要がある

DRAM は相対的に見て CPU よりも動作が遅く足を引っ張る。そこで DRAM と CPU 間のデータのやり取りを少しでも速くするため、以下の工夫がされてきた。

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

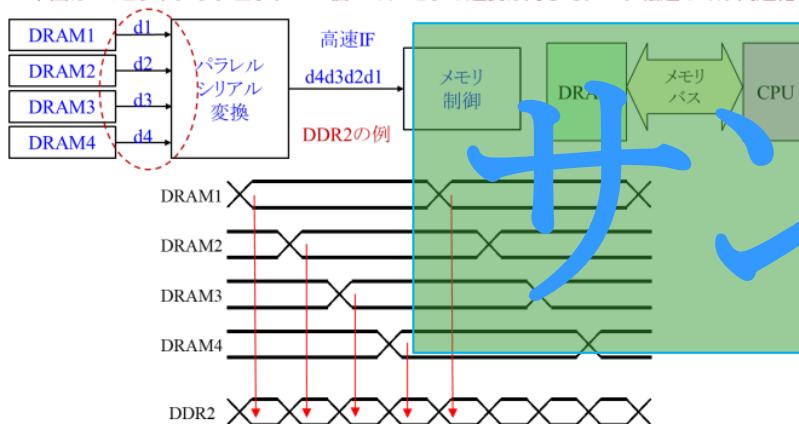


図 7.5.1-2 DRAM の速度向上策 DDR2

- ・ SDRAM: Synchronous DRAM
  - 外部クロックに同期してカラムの読み出しを行い回路設計も容易となった。
- ・ DDR: Double Data Rate SDRAM
  - 外部クロックの Rise/Fall 両エッジを利用 +2 ビットのプリフェッチ(予めデータを読んでおく)で 2 倍のデータ転送速度。
- ・ DDR2 4 ビットのプリフェッチと 4 倍転送クロックで 4 倍の転送速度。

の認知は少しづつ進んでいる。日本では富士ソフト（システム開発）やエン・ジャパン（人材紹介）、HIKKY（ヒッキー、VR 事業）など従業員がアバターを利用して働く企業が増えている。銀行や医薬品メーカーなど従来型の企業でも、メタバースプラットフォーム『クラスター』を活用してイベントを開き、テクノロジー人材の確保に努めている。メタバースはゆっくりだが着実に日本人の生活文化に浸透しつつある。

### 8.2.5 AI チップの動向

7.4.3 で AI 向けプロセッサのおおまかな分類を説明した。ここでは各 AI プロセッサの近況を示す。

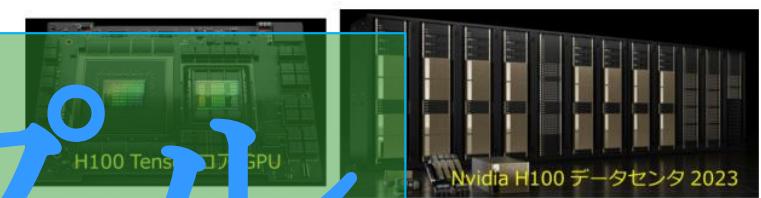


図 8.2.5-1 Nvidia の H100 と AI クラスター



図 8.2.5-2 Google の TPU クラウド

まず GPU に関しては Nvidia が ARM の CPU と自社の GPU を大量に組み合わせた AI クラスターを開発しサービスを行っている。(図 8.2.5-1) AI 用サーバーに関しては Nvidia の様に外販向けに販売している企業と、自らが使うために LSI 開発している企業に分かれる。外販向けとしては 2022 年時点での Nvidia が圧倒的であり、半導体全体売上が前