

デジタル回路 チートシート

第1部: デジタルの基礎と数値表現

重要公式

- **n 進数から10進数への変換:**

$$(x_m x_{m-1} \dots x_1 x_0)_n = (n^m x_m + n^{m-1} x_{m-1} + \dots + n^1 x_1 + n^0 x_0)_{10}$$

- **2進数と8進数の関係:** 2進数3桁が8進数1桁に対応する。
- **2進数と16進数の関係:** 2進数4桁が16進数1桁に対応する。

主要アルゴリズム

- **10進数から n 進数への変換:** 変換したい10進数を n で割り、その余りを下から順に並べる。商が0になるまで繰り返す。
- **負数の表現:**
 - **1の補数 (One's Complement):** ある数の全てのビットを反転させて表現する。
 - **2の補数 (Two's Complement):** 1の補数を計算し、それに1を加える。コンピュータで最も一般的に用いられ、加算器で減算を実現できる。
- **算術シフト (Arithmetic Shift):** 符号付き数の2のべき乗での乗算・除算に用いる。
 - **左シフト:** k ビット左シフトは 2^k 倍に対応。空いた下位ビットには0が入る。
 - **右シフト:** k ビット右シフトは 2^k での除算（床関数）に対応。符号を維持するため、空いた上位ビットには元の数の符号ビットがコピーされる（符号拡張）。

基本用語

- **アナログ:** 連続的に変化する量。情報の読み間違いや信号劣化の可能性があるが、表現できる情報量は無限大。
- **デジタル:** 離散的（飛び飛び）な量。0と1のように2値で表現されることが多く、情報の読み間違いや劣化がない。コンピュータはデジタルで動作する。
- **2進数 (Binary):** 「0」と「1」の2種類の数字のみを使い、「2」で桁上がりする記数法。
- **16進数 (Hexadecimal):** 0-9とA-Fの16種類の記号を使い、「16」で桁上がりする記数法。2進数4桁と対応する。
- **2の補数 (Two's Complement):** コンピュータで負の数を表現するための方法。加算器で減算を実現できる。

第2部: 論理演算とブール代数

重要公式（ブール代数の公理と定理）

- **交換則:** $A \cdot B = B \cdot A, A + B = B + A$
- **結合則:** $A \cdot (B \cdot C) = (A \cdot B) \cdot C, A + (B + C) = (A + B) + C$
- **分配則:** $A \cdot (B + C) = (A \cdot B) + (A \cdot C), A + (B \cdot C) = (A + B) \cdot (A + C)$
- **吸収則:** $A + (A \cdot B) = A, A \cdot (A + B) = A$
- **補元則:** $A + \overline{A} = 1, A \cdot \overline{A} = 0$

- ド・モルガンの法則: $\overline{A \cdot B} = \overline{A} + \overline{B}, \overline{A + B} = \overline{A} \cdot \overline{B}$

基本用語

- 論理演算:** 真 (1) と偽 (0) の2値に対する演算。
 - NOT (否定):** \overline{A}
 - AND (論理積):** $A \cdot B$
 - OR (論理和):** $A + B$
 - NAND (否定論理積):** $\overline{A \cdot B}$
 - NOR (否定論理和):** $\overline{A + B}$
 - XOR (排他的論理和):** $A \oplus B$ 。入力が異なるときに1になる。
- 論理ゲート:** 論理演算を実現する電子回路の基本素子。
- ブール代数:** 2値の論理を扱うための代数系。論理回路の等価性の確認や単純化に用いられる。
- 完全性 (完備性):** ある論理ゲートの集合だけで、全ての論理関数を表現できる性質。
AND, OR, NOTやNAND、NORの集合は完全性を持つ。

第3部: 組み合わせ回路

主要アルゴリズム/回路構成

- シャノン展開:** $n + 1$ 変数の論理関数を、特定の1変数 (例: x_{n+1}) が0のときと1のときの n 変数関数に場合分けして表現する手法。

$$f(x_1, \dots, x_{n+1}) = (\overline{x_{n+1}} \cdot f_0(x_1, \dots, x_n)) + (x_{n+1} \cdot f_1(x_1, \dots, x_n))$$
- 加算器 (Adder):**
 - 半加算器 (Half Adder):** 1ビットの2入力の和(S)と桁上げ(Co)を計算する。 $S = A \oplus B, Co = A \cdot B$ 。
 - 全加算器 (Full Adder):** 下位からの桁上げ(C_i)を含む3入力の和(S)と桁上げ(Co)を計算する。
 $S = A \oplus B \oplus C_i, Co = A \cdot B + C_i(A \oplus B)$ 。
 - 桁上げ先見加算器 (Carry-Lookahead Adder):**
 - 桁上げ生成: $g_i = a_i \cdot b_i$
 - 桁上げ伝搬: $p_i = a_i \oplus b_i$
 - 桁上げ出力: $c_i = g_i + p_i \cdot c_{i-1}$
- 乗算器 (Multiplier):**
 - 2ビット乗数 $X = X_1 X_0$ と被乗数 $Y = Y_1 Y_0$ の積 $Z = Z_3 Z_2 Z_1 Z_0$ は、部分積を生成し、それらを加算することで実現される。
 - 各ビットの論理式例: $Z_0 = X_0 Y_0, Z_1 = X_1 Y_0 \oplus X_0 Y_1$ など、ANDゲートと加算器 (XOR, AND) で構成。
- 回路の拡張:**
 - 既存の回路をブラックボックス (部品) として扱い、それらを組み合わせてより複雑な機能を持つ回路を設計できる。例: 3入力の多数決回路と追加のゲートで4入力の多数決回路を構成。

基本用語

- 組み合わせ回路:** 現在の入力のみによって出力が一意に決まる回路。記憶素子を持たない。
- 標準形:**
 - 積和標準形 (Sum of Products):** 真理値表で出力が1になる最小項の論理和で関数を表現する形式。

- **和積標準形 (Product of Sums):** 真理値表で出力が0になる最大項の論理積で関数を表現する形式。
- **カルノー図:** 真理値表を2次元の図に表現したもの。隣接するセルは1変数のみが異なるように配置（グレイコード順）されており、論理式の簡単化に用いる。
- **デコーダ (Decoder):** n ビットの2進数入力を、対応する 2^n 本の出力線のうちの1本だけをアクティブにする回路。
- **マルチプレクサ (Multiplexer):** 制御入力によって、複数のデータ入力から1つを選択して出力する回路。

第4部: 順序回路

主要アルゴリズム/回路構成

- **SRラッチ:** S (セット) 入力で出力を1に、 R (リセット) 入力で出力を0にする基本的な記憶素子。 $S = R = 1$ は禁止入力。
- **Dフリップフロップ:** クロック信号の立ち上がり（または立ち下がり）のタイミングで、入力 D の値を記憶し出力 Q に反映させる。
- **カウンタ:** クロックパルスが入るたびに、記憶している値を1ずつ増加（または減少）させる回路。
- **レジスタ:** 複数のビットを一時的に記憶する回路で、複数のフリップフロップを並べて構成される。

基本用語

- **順序回路:** 現在の入力と内部の状態（過去の入力履歴）によって出力が決まる回路。記憶素子を持つ。
- **状態遷移図/状態遷移表:** 回路の状態と、入力による状態の変化および出力を図や表で示したもの。
- **ラッチ (Latch):** 制御信号がアクティブな間、入力の变化が出力に反映される記憶素子。
- **フリップフロップ (Flip-Flop):** クロック信号のエッジ（立ち上がり/下がり）の瞬間に入力値を取り込み、出力を変化させる記憶素子。
- **同期式回路:** 全ての記憶素子が共通のクロック信号に同期して動作する回路。
- **シフトレジスタ:** クロックが入るたびに、内部のデータを一方向に1ビットずつ移動（シフト）させるレジスタ。

第5部: メモリ

主要アルゴリズム/回路構成

- **SRAMセル:** 6つのトランジスタ（CMOSインバータ2つとアクセス用2つ）で構成され、フリップフロップとして1ビットを記憶する。電源が供給されている限りデータを保持する。
- **DRAMセル:** 1つのトランジスタと1つのコンデンサで構成される。コンデンサの電荷の有無で1ビットを記憶するが、自然に放電するため定期的なリフレッシュ（再書き込み）が必要。
- **NAND型フラッシュメモリ:** 複数のメモリセルを直列に接続した構造。大容量化に適しているが、動作は比較的低速。
- **NOR型フラッシュメモリ:** 複数のメモリセルを並列に接続した構造。動作は高速だが、大容量化には不向き。

基本用語

- **RAM (Random Access Memory):** 任意のアドレスに対して高速に読み書きが可能なメモリ。揮発性であり、電源を切ると内容が消える。
 - **SRAM (Static RAM):** 高速だが、構造が複雑で高価。キャッシュメモリなどに使用。
 - **DRAM (Dynamic RAM):** 構造が単純で高密度・安価。メインメモリなどに使用。
 - **ROM (Read-Only Memory):** 原則として読み出し専用のメモリ。不揮発性であり、電源を切っても内容が消えない。
 - **フラッシュメモリ:** 電氣的に内容を消去・書き換え可能な不揮発性メモリ(EEPROMの一種)。USBメモリやSSDに利用される。
 - **CAM (Content-Addressable Memory):** アドレスではなく、記憶しているデータの内容を指定して検索できる特殊なメモリ。
-

第6部: CMOS論理回路

基本用語

- **MOSFET (電界効果トランジスタ):**
 - **nMOS:** ゲート入力が'1' (High) のときにON (導通) するスイッチ。
 - **pMOS:** ゲート入力が'0' (Low) のときにON (導通) するスイッチ。
- **プルアップ/プルダウンネットワーク:**
 - **プルアップ:** pMOSで構成され、出力を電源 (V_{cc}) に接続する部分。
 - **プルダウン:** nMOSで構成され、出力を接地 (GND) に接続する部分。
 - CMOSでは、入力の組み合わせに対してプルアップがプルダウンのどちらか一方のみが導通するように設計される。

主要回路構成

- **CMOSインバータ (NOT):** pMOSとnMOSを1つずつ直列に接続した基本回路。入力Aが'0'ならpMOSがONして出力は'1'、入力Aが'1'ならnMOSがONして出力は'0'になる。
- **CMOS NANDゲート:**
 - プルアップ部: pMOSが**並列**接続。
 - プルダウン部: nMOSが**直列**接続。
- **CMOS NORゲート:**
 - プルアップ部: pMOSが**直列**接続。
 - プルダウン部: nMOSが**並列**接続。
- **CMOS回路の論理式導出:**
 1. プルダウンネットワーク (nMOS側) に着目する。
 2. nMOSの直列接続を**AND (論理積)**、並列接続を**OR (論理和)** **とみなして論理式を立てる。
 3. 最終的に、回路全体の出力はプルダウンネットワークの論理の**否定**となる。
 - 例: プルダウンが「AとBの直列」と「C」の並列なら、論理は $(A \cdot B) + C$ 。回路全体の出力Zは $\overline{(A \cdot B) + C}$ となる。