

まえがき

本書では、「コンピュータサイエンス」を情報科学，情報工学，計算機科学，計算機工学などの総称として用いている．

そして，本書は，大学学部，高等専門学校，専修学校のコンピュータサイエンス系学科における「論理回路」と「論理設計」の教科書として書き下ろした．

その内容は（社）情報処理学会が策定した「大学の理工系学部情報系学科のためのコンピュータサイエンス教育カリキュラム J97」の U-1 論理回路；U-6 論理設計；の 2 科目に準拠している．

本書では，コンピュータサイエンスを支える論理代数とそのハードウェアによる実現である論理回路との関係について，電気に関する専門知識がなくても理解できるように，解き明かしている．また，古典的な知識や理論だけではなく，最新の理論や実用的な手法についても平易に解説している．各所で，コンピュータハードウェアの基本原則である「論理回路」を実例として紹介し，理論と実際との関連に興味をつなげるようにしている．

J97 では，この分野の講義を，① 論理回路：数学的な概念（ソフトウェア）による組み合わせ回路や順序回路といった論理回路（ハードウェア）の実現；② 論理設計：論理回路の効率の良い設計手法の理論と実際；の 2 科目に分けて学習することを提案している．

本書の構成では，①には，1 章，2 章の 2.1 と 2.2 節，3 章の 3.1，3.2 節が，②には，2 章の 2.3～2.5 節，3 章の 3.3～3.5 節，4 章が，それぞれ対応している．また，J97 では，それぞれの講義科目に，1.5 時間×15 回（週 1 回で半年）をあてることも推奨している．この分類を参考にして，各機関での実現可能性に合わせて，本書を教科書として活用してもらえば幸いである．

本書についてのコメントや意見は著者（kiyoshi-s@h9.dion.ne.jp）への E メール

2. 1 の操作を最前段の入力端子(「論理変数」に対応)から最後段の出力端子(「論理関数」に対応)に向かって順に行う。
3. NOT は(途中経過の)論理式を複雑にするので, 適当に, ド・モルガンの定理によって, 論理式全体にかかる NOT を展開してリテラルにする。

 例題

2.3 図 2.27 の AND/OR 回路を解析しなさい。

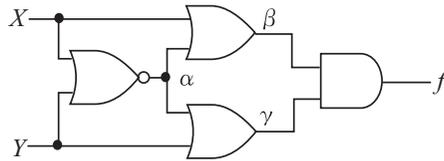


図 2.27 AND/OR 回路例(例題 2.3)

[解]

$$\alpha = \overline{X + Y} = \overline{X} \overline{Y}$$

$$\beta = X + \alpha = X + \overline{X} \overline{Y} = X + \overline{Y}$$

$$\gamma = \alpha + Y = \overline{X} \overline{Y} + Y = \overline{X} + Y$$

$$f = \beta \gamma = (X + \overline{Y})(\overline{X} + Y) = XY + \overline{X} \overline{Y}$$

解析の結果, AND-OR 回路は積和形の, OR-AND 回路は和積形の, それぞれ論理関数となる。それら以外の AND/OR 回路からは積和形や和積形を含む一般的な AND/OR 形式の論理関数を得る。

NAND 回路や NOR 回路の解析も AND/OR 回路と同様の手順でできる。X と Y の 2 入力の NAND ゲートの出力は $\overline{X \cdot Y}$, 同じく 2 入力 NOR ゲートの出力は $\overline{X + Y}$, のそれぞれの論理式で表せる。ただし, NAND や NOR の定義そのものが NOT(“ $\overline{\quad}$ ”)を含んでおり, それに NAND 回路も NOR 回路もそれぞれ NAND あるいは NOR ゲートだけで表されている。したがって, NAND 回路や NOR 回路の解析では, AND/OR 回路の解析よりも NOT が