

加算器

山本昌志*

2004年2月13日

1 本日の授業の内容と到達目標

本日は、加算の回路の学習を行います。講義の内容と諸君が修得すべき到達目標をいかに示します。

[内容]

- 1桁の加算回路
 - － 繰り上がってくる桁の加算を考慮しない半加算器の動作と回路を示す。
 - － 繰り上がってくる桁の加算を考慮する全加算器の動作と回路を示す。
- N桁の加算回路
 - － 全加算器を直列に並べる並列加算回路を示す。
 - － 累算器と全加算器による直列加算器を示す。

[目標]

- 半加算器の真理値表が書けて、回路図を描くことができる。
- 全加算器の真理値表が書けて、回路図を描くことができる。
- N桁を計算する並列加算器と直列加算器の動作の内容が理解できる。

2 組み合わせ回路と順序回路とは

いよいよ今まで学習してきた知識を利用して、電子計算機の回路の設計を行います。これまで、苦勞してブール代数を学習したのは、電子計算機の回路の設計を行うために必要な基礎知識を習得するためです。

論理回路は、大別すると

組み合わせ回路 出力がそのときの入力の状態のみで決まる回路です。これは、出力が以前の動作に依存しないことを言ってます。要するに情報を記憶しない回路です。入力の組み合わせにより出力が決まるので、組み合わせ回路と呼ばれます。

* 国立秋田工業高等専門学校 電気工学科

順序回路 出力は、そのときの入力と、その以前の状態で決まる回路です。これは、出力が以前の状態にも依存することを言っています。以前の状態に依存するということは、以前の情報を記憶していると言うことです。これは入力の順序により、出力が決まるので順序回路と呼ばれます。フリップフロップ回路で構成されています。

に分けられます。今までの学習で、組み合わせ回路のどんな順序回路でも設計できる準備が整いました。順序回路は、4年生で学習することになっていますので、それまでのお楽しみです。そんなに難しくないので、興味のある人は自分で学習してください。

ただし、後で出てくる累算器¹には順序回路が使われています。しかし、ここでは順序回路を意識する必要は無く、累算器の動作と、加算とのかかわりについて、理解できれば十分です。

3 1桁の加算器

論理回路を用いた1桁の加算器には、下からの桁上げを考慮しない半加算器 (Half Adder) とそれを考慮した全加算器 (Full Adder) があります。それぞれについて、学習します。

3.1 半加算器

入力 A と B を加算する回路を考えよう。もちろん、それぞれの入力は $0, 1$ のいずれかです。1桁の2進数の加算回路です。入出力をブラックボックスで書くと、図1のようになります。入力 A と B に対して、出力が S と C です。 S は和を表し英語の Summation から、 C は桁上りを表し Carry の頭文字です。このように、入出力だけ示して、中身が分からないものブラックボックスと言います。この1ビットの加算のブラックボックスの機能は、

- 加算する各1ビットを入力 A と B に入れる。実際の回路では、 A と B に対応する端子に $5V$ を印加²する。
- A と B の加算結果の1桁目を S から、桁上りを示す2桁目を C から出力する。実際の回路では、 S と C に対応する端子に $5V$ が出力される。

です。これが、2進数の1桁の加算に必要な入出力の全てです。

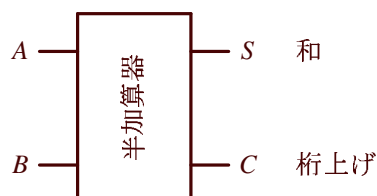


図 1: 半加算器のブラックボックス

¹通常はアキュムレーターとカタカナで書かれることが多い。演算用のメモリーと考えてよい

²TTL の場合。

さて、中身の回路はどうなっているのでしょうか？。それを考えるには、真理値表を書いてみるのが最も良いでしょう。入出力から、真理値表は表1のようになります。

表 1: 半加算器の真理値表

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

残る問題は、この真理値表を実現する回路を設計することだけです。真理値表から、論理回路を求める方法はいろいろ有りますが、この程度で有れば主加法標準形に直すのが簡単でしょう。それぞれは、

$$S = \bar{A} \cdot B + A \cdot \bar{B} \quad (1)$$

$$C = A \cdot B \quad (2)$$

となります。式 (1) の方は排他的論理和 (Exclusive OR) と呼ばれ

$$S = A \oplus B \quad (3)$$

と書かれることも多いです。排他的論理和については、付録を参照のこと。論理式ができたので、それを回路に直すだけです。これらを表す論理回路は、図2や3になります。1桁の半加算器はこれでおしまい。

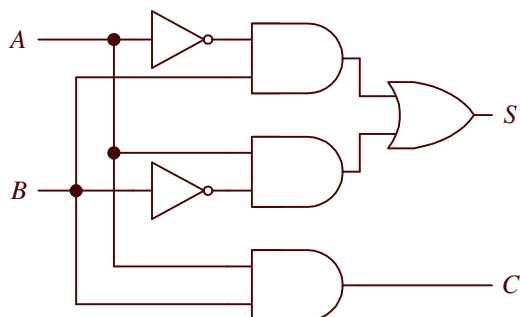


図 2: OR と AND、NOT による半加算器。式 (1) と (2) の回路

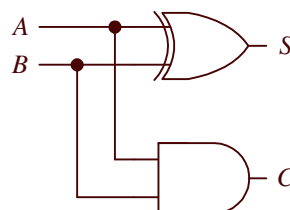


図 3: XOR と AND による半加算器。式 (3) と (2) の回路

3.2 全加算器

先ほどの半加算器は、入力演算の対象の A と B だけで、1桁の2ビットの加算しかできません。実際、複数の桁の加算を行う場合、下位からの桁上がりも考慮する必要があります。この下位からの桁上が

りを考慮した回路が全加算器です。全加算器の入出力をブラックボックスで書くと、図4のようになります。入力は3個の1桁の2進数なので、ブラックボックスの入力端子は3個になります。一方、出力はその和をあらわし、その最大は2桁の $(11)_2$ となります。従って、出力端子は2個必要になります。

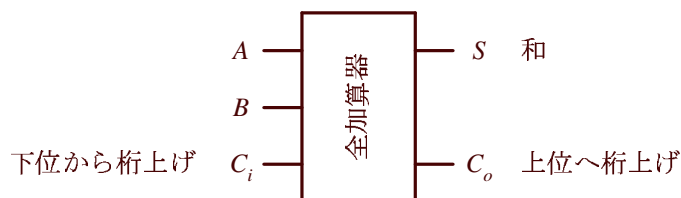


図 4: 全加算器のブラックボックス

先ほどと同じように、真理値表からこのブラックボックス内部の論理回路を考えます。演算の対象が2つから3つに変わっただけです。3つの1ビットの和を考えればよいのです。その加算の演算の真理値表を表2に示します。

表 2: 全加算器の真理値表

A	B	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

真理値表ができたので、次はこれから論理式を作ります。標準展開を用いて論理式を書くこともできますが、ここではカルノー図を使うのが適当でしょう。この真理値表の出力 S と C_o それぞれのカルノー図は、図5と6のようになります。これから、論理式を導き出しましょう。論理式は、いろいろな形に展開可能ですが、半加算器の結果の式 (1) や (2)、(3) に近い形に変形することを考えます。これは後で分かるように、全加算器は半加算器と OR ゲートで可能であることを示すためです。

最初に述べたように、全加算器は A と B 、 C_i の和と桁上がりを計算している回路です。従って、 A と B 、 C_i には区別は全くありません。従って、出来上がった論理式は、それらを入れ替えても成り立つ必要があります。このようなことを考えながら、式の展開を行うと計算が上手になります。

A \ B \ C _i	0	1
0	0	1
0	1	1
1	1	1
1	0	1

A \ B \ C _i	0	1
0	0	
0	1	1
1	1	1
1	0	1

図 5: 全加算器の和 S のカルノー図。

図 6: 全加算器の桁上がり C_o のカルノー図。

まず初めに和 S の論理式を求めますが、そのとき以下の排他的論理和の否定の式を使います。

$$\begin{aligned}
 \overline{A \cdot B} + A \cdot \overline{B} &= (\overline{A \cdot B}) \cdot (\overline{A \cdot \overline{B}}) \\
 &= (A + \overline{B}) \cdot (\overline{A} + B) \\
 &= A \cdot \overline{A} + A \cdot B + \overline{A} \cdot \overline{B} + B \cdot \overline{B} \\
 &= A \cdot B + \overline{A} \cdot \overline{B}
 \end{aligned} \tag{4}$$

これに注意しながら、カルノー図から求められた主加算標準形の S を以下のように変形します。

$$\begin{aligned}
 S &= \overline{A} \cdot \overline{B} \cdot C_i + \overline{A} \cdot B \cdot \overline{C}_i + A \cdot B \cdot C_i + A \cdot \overline{B} \cdot \overline{C}_i \\
 &= (\overline{A} \cdot \overline{B} + A \cdot B) \cdot C_i + (\overline{A} \cdot B + A \cdot \overline{B}) \cdot \overline{C}_i \\
 &= (\overline{A \cdot B} + A \cdot \overline{B}) \cdot C_i + (\overline{A} \cdot B + A \cdot \overline{B}) \cdot \overline{C}_i \\
 &= (\overline{A \oplus B}) \cdot C_i + (A \oplus B) \cdot \overline{C}_i \\
 &= (A \oplus B) \oplus C_i
 \end{aligned} \tag{5}$$

非常にきれいな式が出来上がりました。

つぎに、 C_o の論理式を作ります。これもカルノー図から、

$$\begin{aligned}
 C_o &= A \cdot B + B \cdot C_i + A \cdot C_i \\
 &= A \cdot B + (A + \overline{A}) \cdot B \cdot C_i + A \cdot (B + \overline{B}) \cdot C_i \\
 &= A \cdot B + A \cdot B \cdot C_i + \overline{A} \cdot B \cdot C_i + A \cdot B \cdot C_i + A \cdot \overline{B} \cdot C_i \\
 &= A \cdot B \cdot (1 + C_i + C_i) + \overline{A} \cdot B \cdot C_i + A \cdot \overline{B} \cdot C_i \\
 &= A \cdot B + (\overline{A} \cdot B + A \cdot \overline{B}) \cdot C_i \\
 &= A \cdot B + (A \oplus B) \cdot C_i
 \end{aligned} \tag{6}$$

となります。

以上で全加算器の論理式が完成したわけですが、もうひとひねりしておきます。それは、半加算器の式 (2) と (3) を用いて、全加算器の式 (5) と (6) を書き直します。ちょっと記号の問題がありますので、半加算器の出力を

$$S_1 = A \oplus B \tag{7}$$

$$C_1 = A \cdot B \tag{8}$$

と置き換えます。すると、

$$S = S_1 \oplus C_i \quad (9)$$

$$C = C_1 + S_1 \cdot C_i \quad (10)$$

となります。ここで、最後のひねりとして、 $S_1 \cdot C_i = C_2$ を加えます。すると

$$S = S_1 \oplus C_i \quad (11)$$

$$C = C_1 + C_2 \quad (12)$$

となります。これで準備は完了です。

これらの式をゲートで組み立てる前に、最後の式 (11) と (12) から、全加算器は半加算器 2 個と OR ゲートでできることが分かります。即ち、図 7 の通りです。この半加算器と OR ゲートを使った動作は、全加算器として動作することが直ぐに理解できると思います。

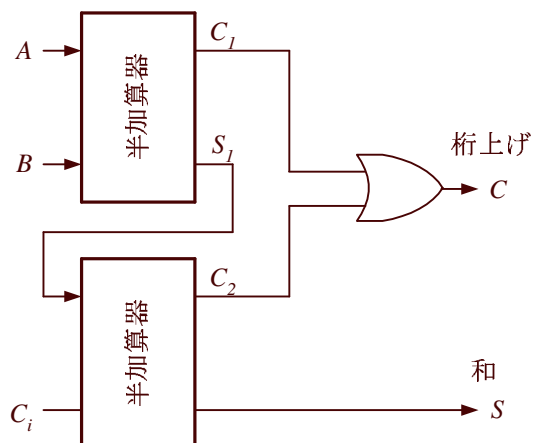


図 7: 2 個の半加算器と OR ゲートで構成される全加算器

全加算器が半加算器と OR ゲートで出来ることが分かったので、論理回路も同じことです。やはり半加算器 2 個と 1 個の OR ゲートで出来ます。図 8 と 9 の通りです。それぞれの論理回路がどの式と対応しているかは分かりますよね。考えてください。全加算器はこれで終わり。

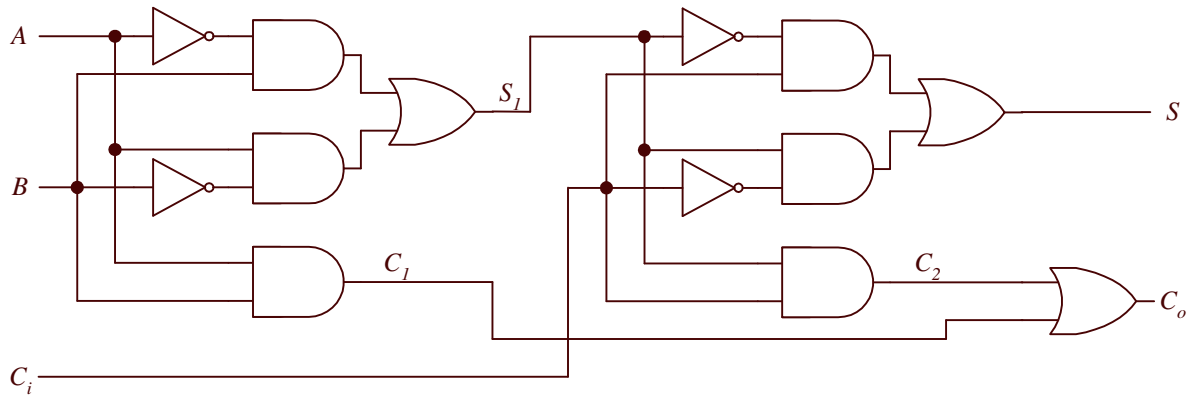


図 8: OR と AND、NOT ゲートによる全加算器

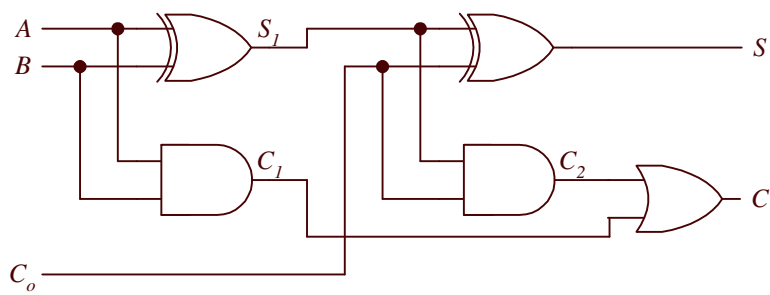


図 9: XOR と AND、OR ゲートによる全加算器

4 N桁の加算

ここでは、2進数N桁の加算の回路を考えます。N桁の加算回路の構成がどのようになっているか良く理解してください。そして、計算が回路で可能であることをよく理解してください。

4.1 並列加算回路

いままで、1桁の加算器の回路について説明しました。今度は、複数の桁の加算です。賢明な諸君であれば、N桁の加算はN個の全加算器用いればよいと直ぐに気が付くと思います。即ち、図10のようにすればよいのです。これを並列加算器と言います。全加算器は直列に接続されているのですが、加算データは並列に入力されるのでその名前がつけられと思います。

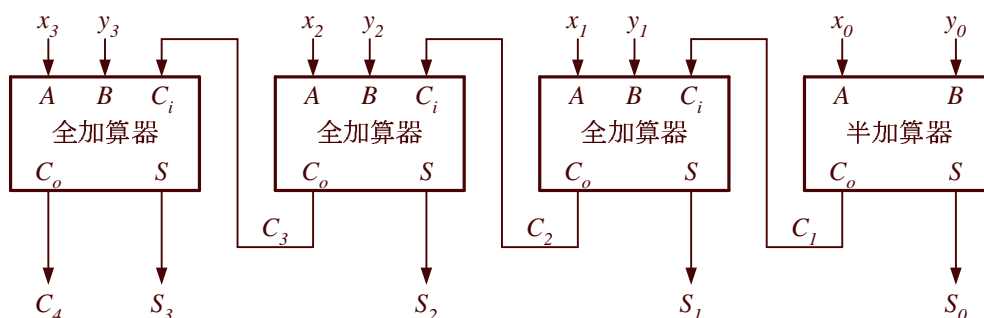


図 10: 4桁の並列加算器。2進数の $x_3x_2x_1x_0$ と $y_3y_2y_1y_0$ の加算を行う。

4.2 直列加算回路

これまでの回路は、組み合わせ回路でした。しかし、ここでは順序回路である累算器³を用いた回路です。これは、記憶するので順序回路となっています。

直列加算回路は、図11に示すように1個の全加算器と、2個のレジスタ(演算用のメモリ)と1個の桁上げメモリから構成されています。これらの回路の詳細については、時間の都合でここでは述べません。その動作を述べます。

この回路の動作の順序は、次のようになります。

1. 図に示すように計算する数 $x_3x_2x_1x_0$ と $y_3y_2y_1y_0$ を各レジスタに格納します。
2. 次に、各レジスタを1ビット右にシフトさせます。あふれ出たビットは、全加算器の入力 A , B となります。最初、 C_i は0です。したがって、最下位の和が計算され、その結果が S と C_o に出力されます。
3. 計算結果の S は累算器の最上位のビットに格納されます。 C_o は桁上げメモリに格納されます。

³アキュムレーター (accumulator) と呼ばれることが多い。演算用のメモリと考えれば良い。

4. 次の桁の計算は、先ほどとおなじで各レジスターを1ビット右にシフトさせます。あふれ出たビットは、全加算器の入力 A, B となります。同時に、桁上げメモリーから先ほどのデータを引き出し C_i の入力とします。これで次の桁の計算結果が、 S, C_o に現れます。
5. 加算すべきビットが全て出て行くまで、以上を繰り返します。
6. 全てのビットの処理が終了したならば、累算器には加算結果が残ります。そして、桁上げメモリーにはオーバーフローのデータが残ります。

以上が、直列加算回路の動作です。

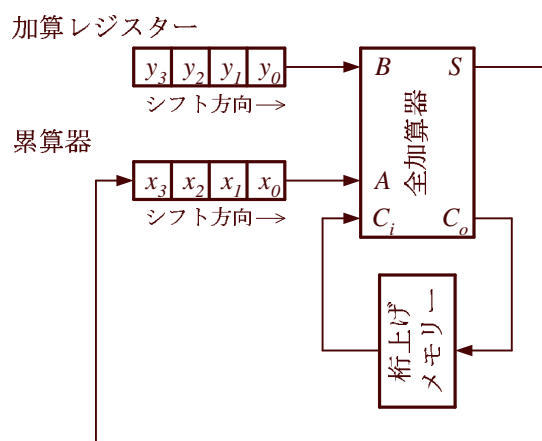


図 11: 4 桁の直列加算器

5 付録

5.1 排他的論理和

排他的論理和の真理値表と MIL 記号を以下に示す。

表 3: 回路の応答を表現

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



表 4: XOR 素子