Capítulo 4

Circuitos combinacionais I

Neste capítulo, vamos construir um dos blocos de circuitos digitais mais importantes em termos de processamento de informações com as técnicas de projeto digital. Além disso, vamos começar a fazer operações matemáticas usando essas técnicas.

Antes de mais nada, é necessário relembrar que é muito comum um sistema digital processar informações recebidas. Quando falamos em processar informações, estamos nos referindo a tratar um dado coletado de um sensor, por exemplo, e executar uma ação em função do valor obtido na leitura desse dado.

Lembrando que os números são representados por bits e bytes. Podemos representar um valor de 0 a 255 usando um byte de 8 bits, ou seja, se em uma determinada aplicação precisarmos representar a leitura de um sensor usando um byte, teremos 256 combinações. Se o número que precisamos representar for maior que 255, podemos aumentar a quantidade de bits; por exemplo, se representarmos esse número com 10 bits, ele pode ser até 1023, e se utilizarmos 16 bits, o número pode ser até 65535.

Além disso, podemos representar mais de um valor usando dois conjuntos de bits, cada um representando uma determinada grandeza que está sendo medida. E, já que temos mais de um valor, podemos fazer operações matemáticas com esses valores representados usando números binários.

1 Meio somador

A primeira operação matemática que vamos abordar é a soma de binários, que deve ser feita em acordo com a matemática de Boole (IDOETA; CAPUANO, 1999). Existem regras específicas para fazer essas operações, e os detalhes de como elas são feitas serão explanados ao longo do capítulo, mas elas já devem ser de seu conhecimento.

Aqui, vamos transformar aquelas operações em circuitos eletrônicos, que podemos implementar com portas lógicas ligadas entre si adequadamente. Porém, antes de fazer o circuito, vamos conferir a menor soma que podemos realizar usando números binários. Ao somarmos números de apenas 1 bit, as combinações que podemos ter são:

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 10$

Podemos notar que, em um dos casos, temos o carry, ou seja, a soma de 1 bit pode ter como resultado 2 bits, e não apenas 1.

A partir disso, podemos montar uma tabela-verdade da soma de dois números de apenas 1 bit, sendo o primeiro número chamado de A e o segundo número, de B.

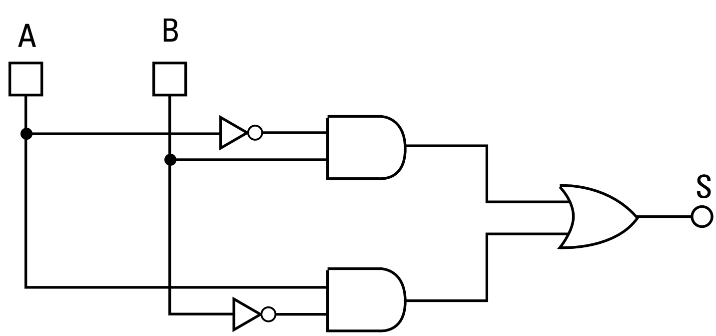
Tabela 1 — Somando dois números de 1 bit

A	B	S	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A partir da tabela-verdade apresentada, vamos criar uma equação booleana, começando pela equação que descreve a saída S:

$$S = (\overline{A} \,.\, B) + (A \,.\, \overline{B})$$

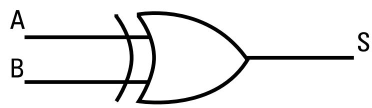
Figura 1 – Circuito gerado pela equação da saída S do meio somador



Fonte: adaptado de Idoeta e Capuano, 1999.

Porém, convém lembrar que essa combinação tem uma porta padrão já definida, que é uma porta XOR. Ou seja, S = A (+) B.

Figura 2 – S = A (+) B

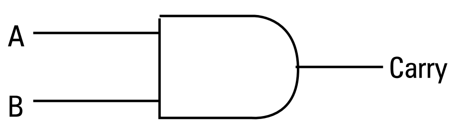


Fonte: adaptado de Idoeta e Capuano, 1999.

Agora, continuando a análise da tabela 1, podemos descrever a equação de carry como:

$$\mathrm{Carry} = \mathrm{A} \,.\, \mathrm{B}$$

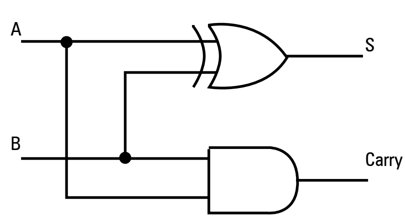
Figura 3 – Carry = A . B



Fonte: adaptado de Idoeta e Capuano, 1999.

Juntando no mesmo desenho as equações de S e de carry, temos a representação gráfica do meio somador, expressa pela figura 4.

Figura 4 — Meio somador



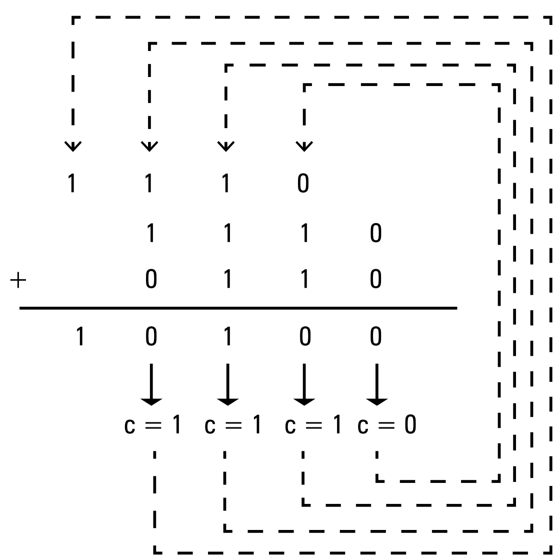
Fonte: adaptado de Idoeta e Capuano, 1999.

2 Somador completo

Conforme descrito, o meio somador possibilita fazer a soma de dois números binários de apenas um bit. Para fazermos a soma de números com múltiplos bits, precisamos de um circuito um pouco mais complexo.

Para compreender melhor o problema, vamos analisar uma soma de dois números de 4 bits, no caso, A = 1110 e B = 0110. Podemos representar essa soma de acordo com a figura 5, a seguir, em que C é o carry que descrevemos no meio somador.

Figura 5 – Somando dois números binários de 4 bits



Fonte: adaptado de Idoeta e Capuano, 1999.

Assim, para implementar um somador que considere o carry do bit anterior, temos a tabela-verdade a seguir.

Tabela 2 – Tabela-verdade do somador completo de 1 bit

A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fonte: adaptado de Idoeta e Capuano, 1999.

Na tabela 2, descrevemos a operação de soma de apenas um dos bits somados na figura 5. Ou seja, essa tabela-verdade descreve cada uma das colunas (ou bits) da soma da figura 5. Nessa tabela, Ci é o carry de entrada, que é obtido do bit anterior – ou menos significativo em relação ao bit cuja a saída S estamos gerando –, e Co é o carry de saída, que será o carry de entrada do próximo bit cuja saída vamos gerar.

Explicado isso, vamos gerar as equações que descrevem S e Co:

$$S = (\overline{A} . \overline{B} . Ci) + (\overline{A} . B . \overline{Ci}) + (A . \overline{B} . \overline{Ci}) + (A . B . Ci)$$

$$Co = (\overline{A} . B . Ci) + (A . \overline{B} . Ci) + (A . B . \overline{Ci}) + (A . B . Ci)$$

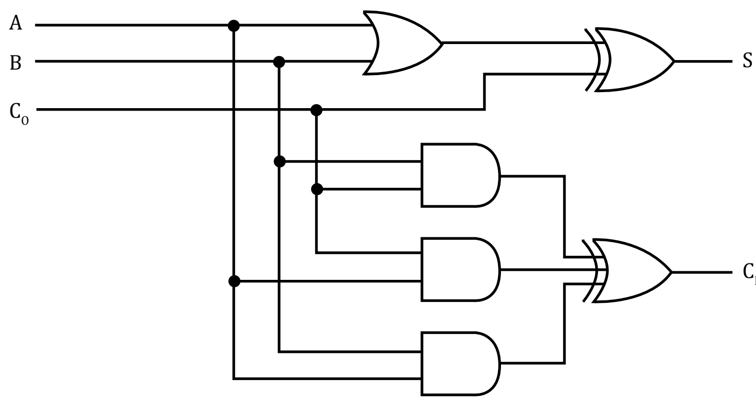
Simplificando essas equações, obtemos:

$$S = A (+) B (+) Ci$$

$$Co = (A . Ci) + (B . Ci) + (A . B)$$

Implementando as equações, obtemos o circuito a seguir (figura 6), que corresponde à implementação de qualquer uma das colunas do somador da figura 5 (p. 61).

Figura 6 – Somador completo



Fonte: adaptado de Idoeta e Capuano, 1999.

O somador completo que desenhamos até agora gera o resultado de apenas um bit, mas podemos usá-lo para somar quantos bits forem necessários.

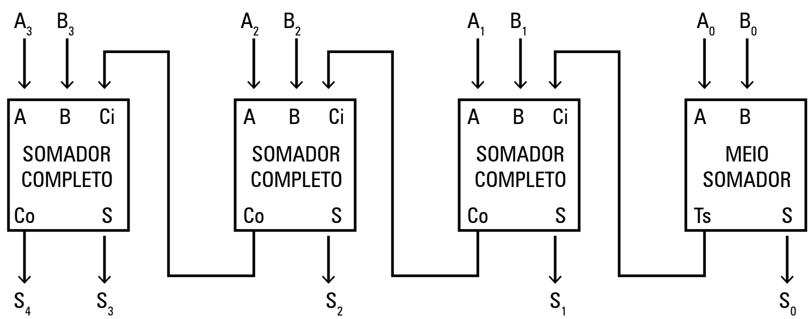
Para ilustrar isso, vamos implementar um somador de dois números de 4 bits, que vai gerar uma saída de 5 bits, sendo o primeiro número, A, desmembrado nos bits A_3, A_2, A_1 e A_0, e o segundo número, B, desmembrado em B_3, B_2, B_1 e B_0. A saída S, por sua vez, é desmembrada em S_4, S_3, S_2, S_1 e S_0. Esquematizando a soma, vamos ter o expresso pela figura 7, a seguir.

Figura 7 – Montando a soma de binários

$$\begin{array}{ccccc} & A_3 & A_2 & A_1 & A_0 \\ + & B_3 & B_2 & B_1 & B_0 \\ \hline S_4 & S_3 & S_2 & S_1 & S_0 \end{array}$$

Usando o somador que descrevemos anteriormente, podemos implementar a soma descrita ligando a saída Co de um bit à entrada Ci do bit imediatamente mais significativo, à exceção dos bits A_0, B_0 e S_0, que são os menos significativos. Nesse caso, podemos usar o meio somador para implementar essa etapa do circuito, como mostrado na figura 8.

Figura 8 – Implementando um somador de 4 bits



Fonte: adaptado de Idoeta e Capuano, 1999.

Podemos generalizar esse circuito para somar números de quantos bits forem necessários para a nossa aplicação. Ou seja, esse circuito funciona como um bloco para a construção de somadores binários.

3 Outra forma de construir um somador completo

Outra forma de descrever um somador completo é utilizando dois meios somadores.

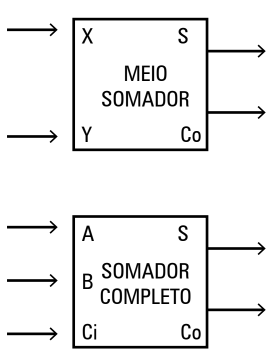
IMPORTANTE

A vantagem da técnica de dois meios somadores é que ela diminui o bloco que deve ser repetido diversas vezes no circuito em construção, bem como diminui a diversidade de blocos de que precisamos no nosso projeto.

Quando usamos a técnica de projeto digital, percebemos que quanto mais conseguimos dividir os problemas que estamos resolvendo, mais fácil o projeto como um todo se torna. Outro ponto importante é que, quando temos circuitos que se repetem muito, podemos gastar mais energia na sua otimização, melhorando a relação entre o esforço de projeto e o benefício que se obtém em termos de desempenho e redução de custos.

Para fazer a conversão de um somador completo na junção de dois meios somadores, vamos começar analisando as expressões que definem o comportamento de ambos os blocos, conforme apresentado na figura 9.

Figura 9 – Comparação entre as expressões lógicas do meio somador e do somador completo



$$S = X \oplus Y$$

$$Co = XY$$

$$S = A \oplus B \oplus Ci$$

$$Co = \bar{A}BCi + A\bar{B}Ci + AB\overline{Ci} + ABCi$$

Fonte: adaptado de Idoeta e Capuano, 1999.

Se reescrevermos a expressão de Co do somador completo, teremos:

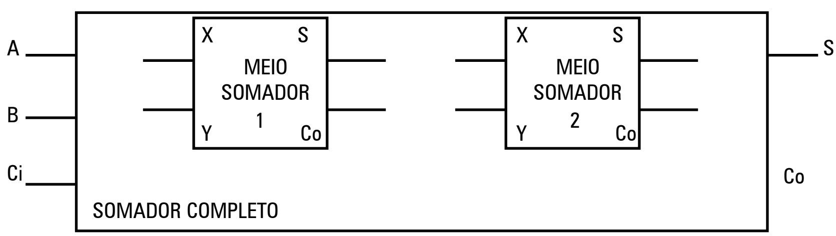
$$\mathrm{Co} = \mathrm{Ci}\,.\,(\overline{\mathrm{A}}\,.\,\mathrm{B} + \mathrm{A}\,.\,\overline{\mathrm{B}}) + \mathrm{A}\,.\,\mathrm{B}\,.\,(\overline{\mathrm{Ci}} + \mathrm{Ci})$$

Podemos simplificá-la da seguinte forma:

$$\mathrm{Co} = \mathrm{Ci}\,.\,(\,\mathrm{A}\,(+)\,\mathrm{B}) + \mathrm{A}\,.\,\mathrm{B} \rightarrow \text{Expressão carry out}$$

Lembremos que a ideia é utilizar dois meio somadores para criar um somador completo. No início, vamos ter dois meio somadores: o meio somador 1 e o meio somador 2, como mostrado na figura 10, a seguir.

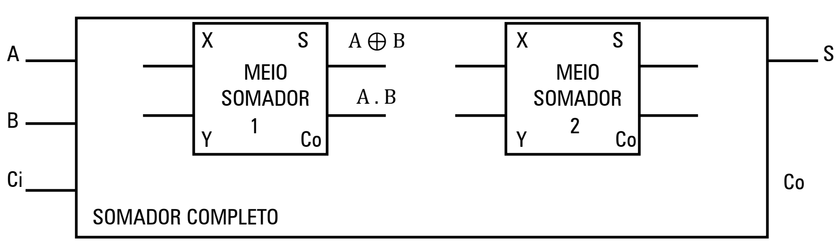
Figura 10 – Somador completo usando dois meios somadores



Fonte: adaptado de Idoeta e Capuano, 1999.

Se ligarmos, na entrada do meio somador 1, as entradas A e B a X e Y, respectivamente, teremos a situação expressa pela figura 11, a seguir.

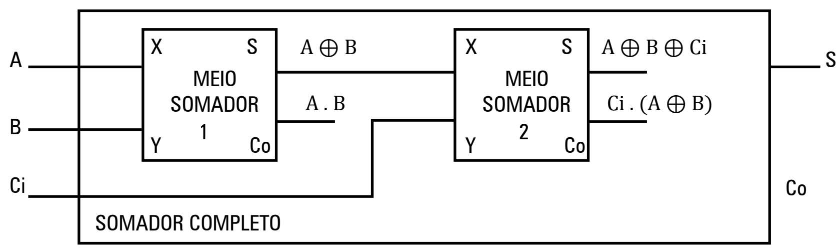
Figura 11 – Ligando as entradas A e B do somador completo no meio somador 1



Fonte: adaptado de Idoeta e Capuano, 1999.

Se ligarmos a saída S do meio somador 1 à entrada X do meio somador 2, e a entrada Ci do somador completo à entrada Y do meio somador 2, teremos, na saída S do meio somador 2, a exata expressão que descreve a saída do somador completo. Ou seja, já teremos metade do nosso somador completo definido.

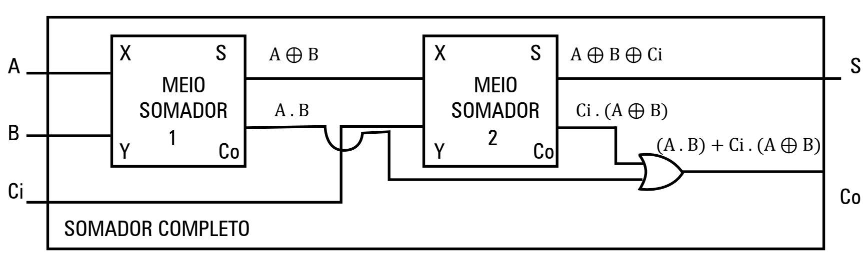
Figura 12 – Ligando o meio somador 2



Fonte: adaptado de Idoeta e Capuano, 1999.

Neste ponto, falta apenas definir a saída Co do somador completo; porém, ao observarmos a equação carry out definida anteriormente, podemos concluir que os dois termos da equação já estão presentes no circuito, nas saídas Co do meio somador 1 e na saída Co do meio somador 2. Para obtermos a equação carry out do somador completo, basta realizarmos a operação OR entre as saídas Co dos dois meio somadores, e teremos o nosso somador completo construído a partir de dois meios somadores, como podemos conferir na figura 13.

Figura 13 – Somador completo totalmente montado a partir de dois meios somadores



Fonte: adaptado de Idoeta e Capuano, 1999.

Considerações finais

Neste capítulo, pudemos usar a técnica digital para processar informações – no caso, fazer a soma de dois números binários. Essa técnica certamente tem muitas aplicações, seja na forma de um circuito discreto em uma aplicação muito específica, seja dentro de um processador, que precisa ter um circuito eletrônico para implementar a operação de soma de dois números binários.

Outro ponto importante trabalhado neste capítulo foi a exploração de várias formas de implementar a mesma função lógica, e como isso pode ser utilizado para que se tenha um melhor aproveitamento dos recursos físicos de projeto.

É importante compreender que essa é apenas uma das possibilidades de implementar uma soma binária e que existem diversas formas de obter o mesmo resultado, com diferenças entre área e velocidade de cálculo. Essas outras possibilidades não foram exploradas aqui, mas, caso haja interesse, é possível buscá-las em artigos científicos na área de projeto de circuitos digitais.

Referências

IDOETA, Ivan V.; CAPUANO, Francisco G. **Elementos de eletrônica digital**. São Paulo: Érica, 1999.