

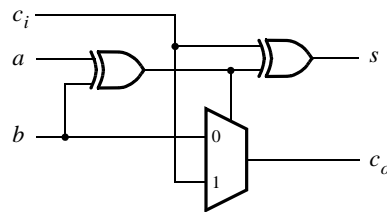
Laboratório 6

Somadores e Multiplicadores

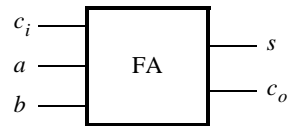
O objetivo deste laboratório é examinar circuitos aritméticos que somam, subtraem e multiplicam números.

Parte I

A Figura 1 mostra um circuito para um *somador completo*, que possui as entradas a , b , e c_i , e produz as saídas s e c_o . As partes b e c da figura mostram o símbolo do circuito e a sua tabela verdade, que produz a soma binária de 2 bits. A Figura 2d mostra como quatro instâncias deste somador completo podem ser usadas para projetar um circuito que soma dois números de 4 bits. Escreva código VHDL que implemente este circuito como descrito abaixo:



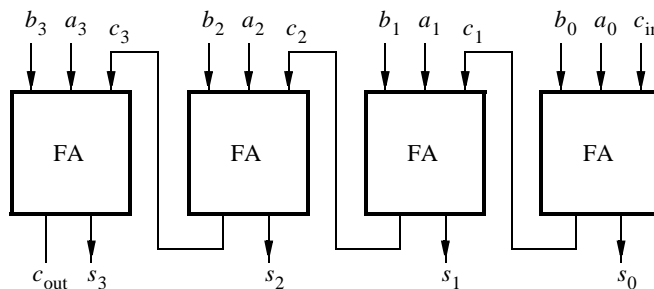
a) Full adder circuit



b) Full adder symbol

| b | a | 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 |

c) Full adder truth table



d) Four-bit ripple-carry adder circuit

Figura 1: Um circuito somador.

1. Crie um novo projeto Quartus II para esta entidade.
2. Use as chaves SW_{7-4} e SW_{3-0} para representar as entradas A e B , respectivamente. Use SW_8 para o carry-in c_{in} do somador. Conecte as chaves SW nos seus leds vermelhos correspondentes, e a saída do somador c_{out} e S nos leds verdes.
3. Compile e teste o circuito.

Parte II

Implementar o circuito mostrado na Figura 2. Para tanto:

1. Ligue a entrada A nas chaves SW_{7-0} , use KEY_0 como um reset assíncrono e KEY_1 como uma entrada de clock manual. A saída S deve ser mostrada nos leds vermelhos $LEDR_{7-0}$ e o sinal de Carry Out em $LEDG_8$. Atenção para o uso de números **sem** sinal.

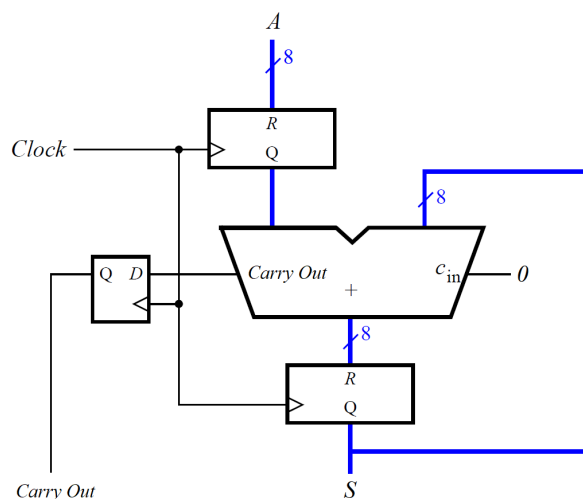


Figura 2: Um circuito acumulador de 8 bits.

Mostre esta parte para o professor em sala de aula.

Parte III

A Figura 3a mostra um exemplo de multiplicação $P = A \times B$ da forma que seria resolvida com papel e caneta, onde $A = 11$ e $B = 12$.

a) Decimal

$$\begin{array}{r} 11 \\ \times 12 \\ \hline 22 \\ 11 \\ \hline 132 \end{array}$$

b) Binary

$$\begin{array}{r} 1011 \\ \times 1100 \\ \hline 0000 \\ 0000 \\ 1011 \\ 1000 \\ \hline 1000100 \end{array}$$

c) Implementation

| | | x | a_3 b_3 | a_2 b_2 | a_1 b_1 | a_0 b_0 | |
|----------|----------|----------|----------------|----------------|----------------|----------------|-------|
| | | | a_3b_0 | a_2b_0 | a_1b_0 | a_0b_0 | |
| | | a_3b_1 | a_2b_1 | a_1b_1 | a_0b_1 | | |
| | a_3b_2 | a_2b_2 | a_1b_2 | a_0b_2 | | | |
| a_3b_3 | a_2b_3 | a_1b_3 | a_0b_3 | | | | |
| p_7 | p_6 | p_5 | p_4 | p_3 | p_2 | p_1 | p_0 |

Figura 3: Multiplicação de números binários.

Calculamos $P = A \times B$ como em uma adição de termos. O primeiro termo é igual a A vezes a unidade de B . O segundo termo é A vezes a dezena de B , deslocado uma posição para a esquerda. Adicionamos estes termos para obter $P = 132$.

A parte b desta mesma figura mostra o mesmo exemplo usando números binários de 4 bits. Para calcular $P = A \times B$, devemos formar os termos multiplicando A por cada dígito de B . Como cada dígito de B é 1 ou 0, os termos são versões deslocadas de A ou 0000. A Figura 3c mostra como cada termo pode ser formada usando a expressão Booleana AND de A com o bit apropriado de B .

Assim, este circuito multiplicador pode ser representado como mostrado na Figura 4.

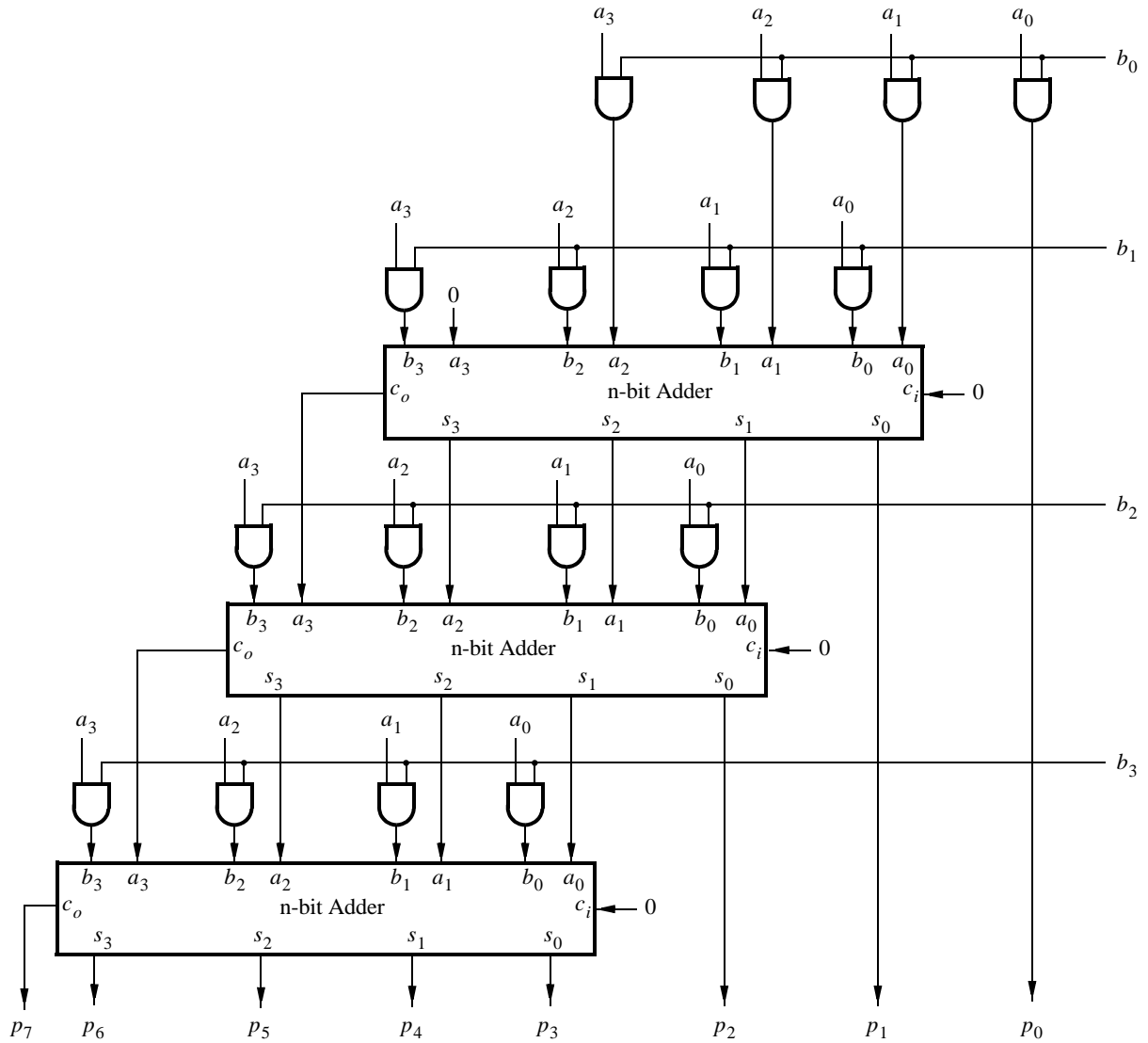


Figura 4: Circuito multiplicador.

Cada somador de n -bits adiciona uma versão shiftada de A para dada linha da soma parcial da linha acima. Com esta abstração é possível construir multiplicadores para qualquer tamanho de entrada. Use esta técnica para implementar um multiplicador 8x8 com entradas e saídas registradas, como mostrado na Figura 5.

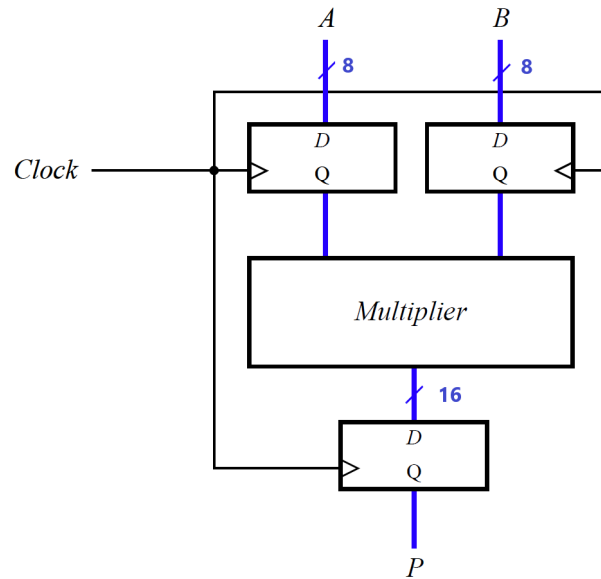


Figura 5: Um circuito multiplicador registrado.

Importante: Não use o operador '*' definido na biblioteca. O objetivo do laboratório é treinar a construção do circuito.

Execute os seguintes passos:

1. Use as chaves SW_{15-8} para representar o número *A* e as chaves SW_{7-0} para representar *B*. Use KEY_0 como um reset assíncrono e KEY_1 como uma entrada de clock manual. O valor hexadecimal de *A* e *B* devem ser mostrados nos displays de 7-segmentos *HEX7-6* e *HEX5-4*, respectivamente. O resultado $P = A \times B$ deve ser mostrado em *HEX3-0*. Utilize representação numérica **sem** sinal.
2. Envie esta parte para o Ensino Aberto.