



Prática Nº5 - SEL0606

Aluno:  
Vitor Alexandre Garcia Vaz

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Objetivos	3
1.2	Somador aritmético	3
<b>2</b>	<b>Materiais e métodos</b>	<b>4</b>
2.1	Obtenção das expressões booleanas	4
2.2	Implementação	5
2.3	Previsão sobre utilização dos componentes TTL da família 74LSxx	8
<b>3</b>	<b>Conclusão</b>	<b>8</b>
3.1	Círculo RTL	8
3.2	Número de células lógicas	10
3.3	Funcionamento do circuito	11

## Lista de Imagens

1	Somador RPC	3
2	Somador Completo	4
3	Tabela verdade do somador completo	4
4	Localização das chaves, botões e LEDS no DE10_lite	5
5	Código com uso do somador para reconfiguração da FPGA	6
6	Código de somador RPC	7
7	Código com uso do somador completo (para 1 bit)	8
8	Nível 1 do circuito RTL sintetizado	9
9	Nível 2 do circuito RTL sintetizado	9
10	Nível 3 do circuito RTL sintetizado	10
11	Resumo de funcionamento	10
12	Acendimento de leds para $A = 0001_2$ e $B = 0001_2$	11
13	Acendimento dos leds para $A = 0001_2$ e $B = 0011_2$	11

# 1 Introdução

## 1.1 Objetivos

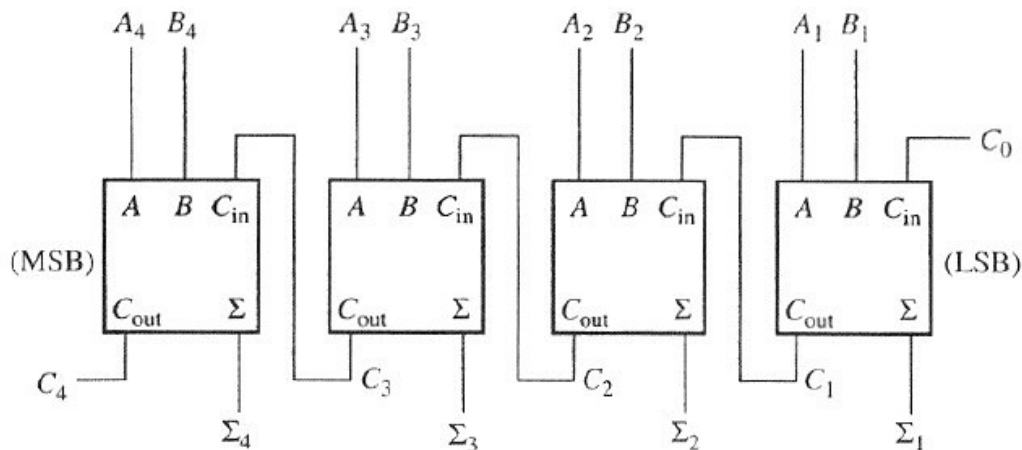
Nesta prática do Laboratório de Sistemas Digitais, tivemos como objetivo implementar um somador RPC (Ripple-Carry Chain) utilizando a linguagem VHDL, em conjunto com a ferramenta quartus, de forma que tanto o somador RPC quanto o somador completo (full-adder) fossem usados como componentes, em arquivos separados, para configurar um somador na FPGA do kit DE10-Lite.

Além disso, aplicando o código na FPGA, testamos o funcionamento do circuito visualizando a resposta das entradas nos LEDs vermelhos, de acordo com as entradas nas chaves.

## 1.2 Somador aritmético

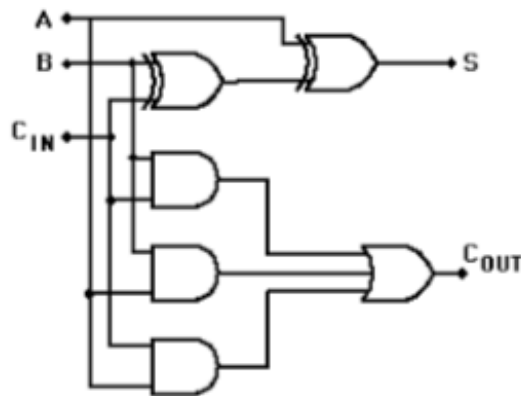
O somador aritmético é um circuito combinacional cuja função é retornar um número resultante da soma de outros dois números. Valendo ressaltar que todas as entradas e a saída possuem a mesma quantidade de bits e que, para cada bit a ser considerado na soma, é necessário o uso de um somador completo (fig.2) conectado aos demais via carry-in e/ou carry-out, formando-se um somador RPC(fig.1).

Figure 1: Somador RPC



Fonte: [Site da internet](#)

Figure 2: Somador Completo



Fonte: [Site da internet](#)

## 2 Materiais e métodos

### 2.1 Obtenção das expressões booleanas

Através da tabela verdade da fig.3 referente ao somador completo (full-adder), construímos o componente básico para a adição de dois bits (A e B), considerando as possibilidades de recebimento ou saída de carregamentos (carry-in e carry-out).

Figure 3: Tabela verdade do somador completo

A	B	C <sub>IN</sub>	S	C <sub>OUT</sub>
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Fonte: [Site da internet](#)

Além do mais, não houve necessidade de uso de expressões booleanas no código, já que as instruções da prática indicaram que os alunos realizassem o projeto utilizando comando concorrentes ( WHEN-ELSE e WITH-SELECT ). Porém, vale ressaltar que as expressões booleanas para um somador completo, obtidas através da tabela verdade citada, foram:

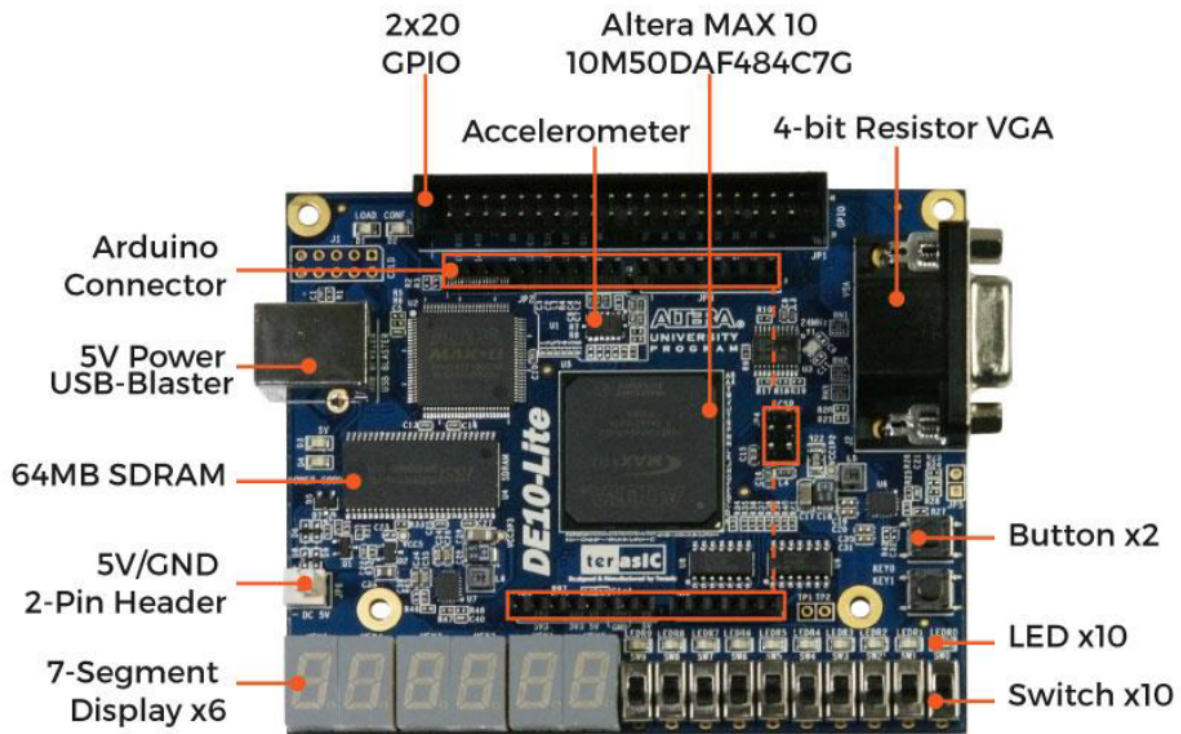
- $S = A \oplus (B \oplus C_{in})$
- $C_{out} = BC_{in} + AC_{in} + AB$

## 2.2 Implementação

Utilizando o mapeamento e barramento dos sinais descrito no manual do DE10-LITE, escrevemos o código em VHDL para o somador aritmético. Os conjuntos de chaves SW(3) a SW(0) e SW(7) a SW(4) (ambas de 4 bits) correspondem às entradas A e B do somador RPC.

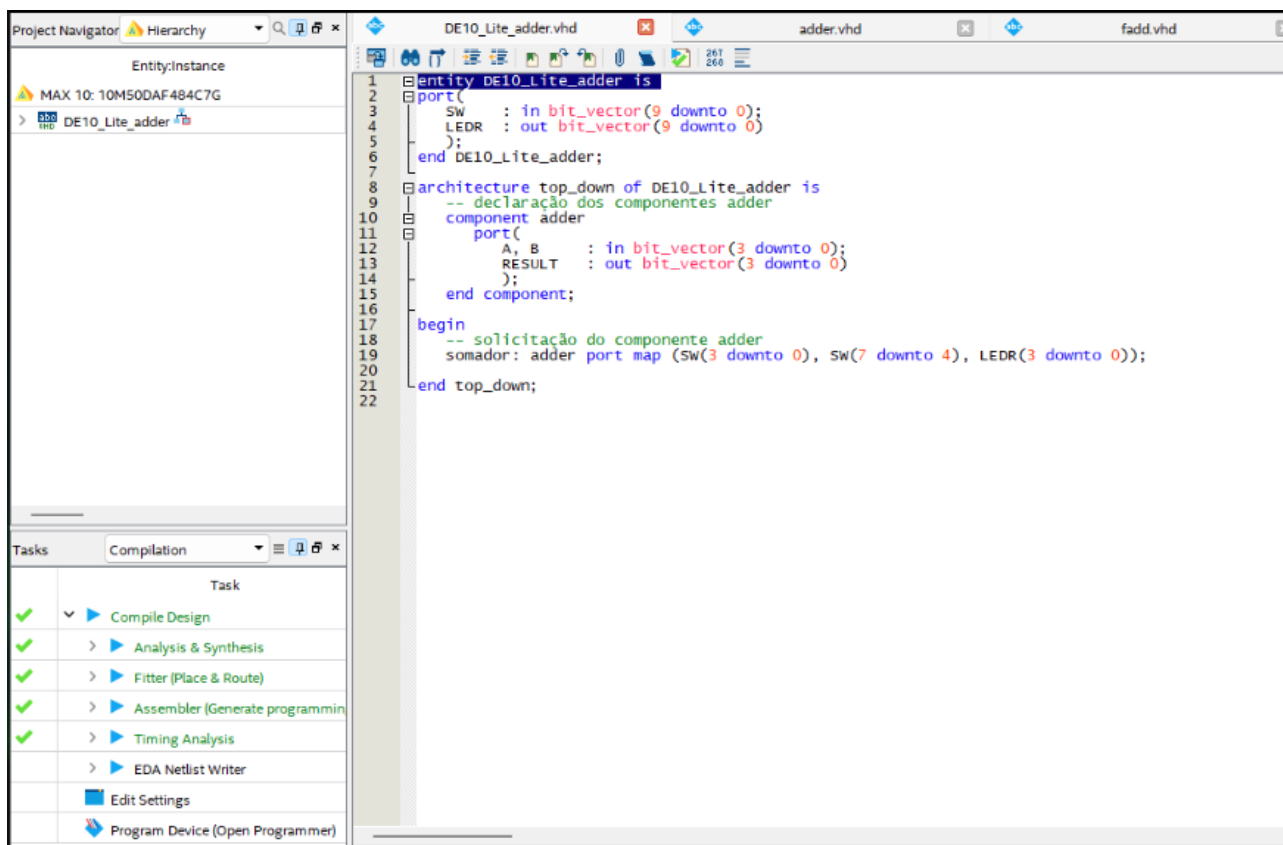
A saída (também de 4 bits), por sua vez, foi transmitida para LEDR(3) a LEDR(0), que recebem o valor binário resultante da soma aritmética das entradas A e B.

Figure 4: Localização das chaves, botões e LEDS no DE10\_lite



Fonte: DE10\_lite User Manual

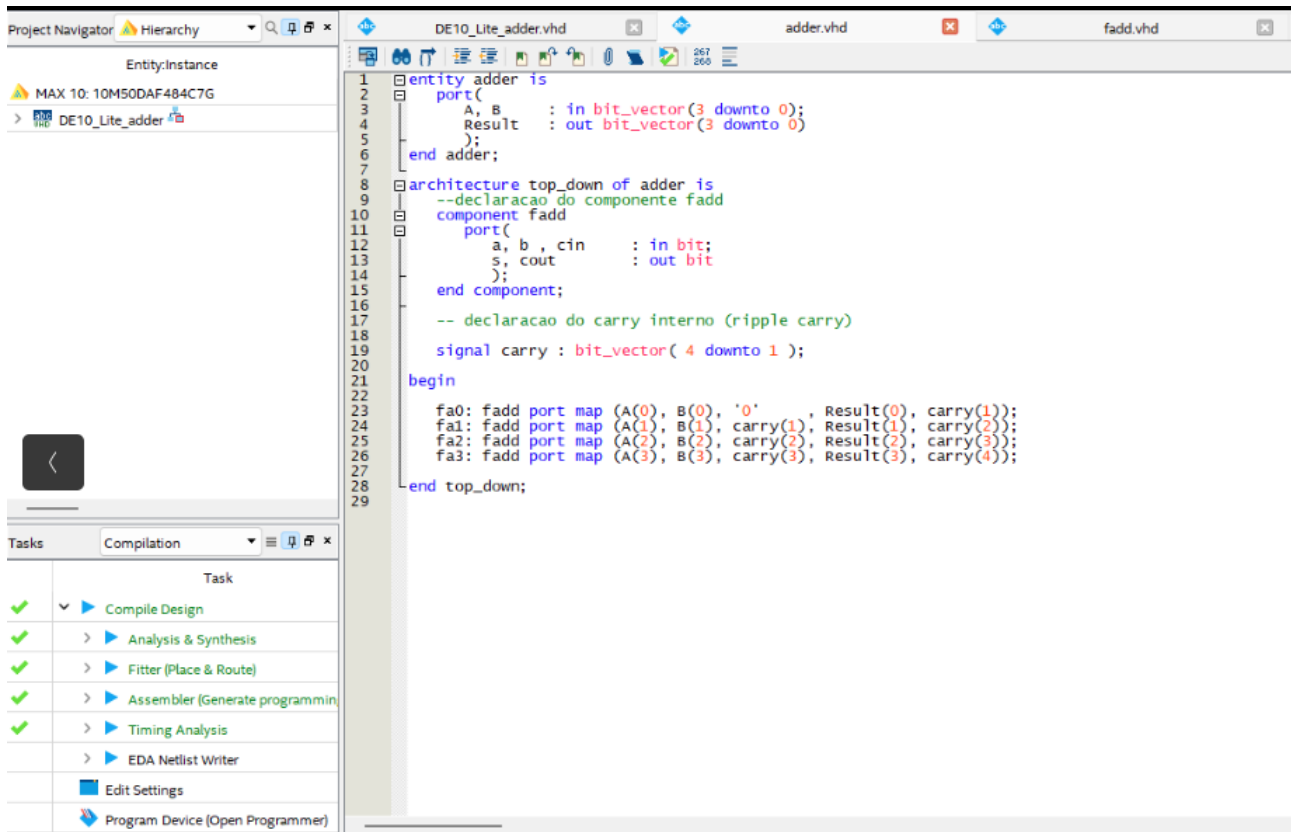
Figure 5: Código com uso do somador para reconfiguração da FPGA



Fonte: os autores

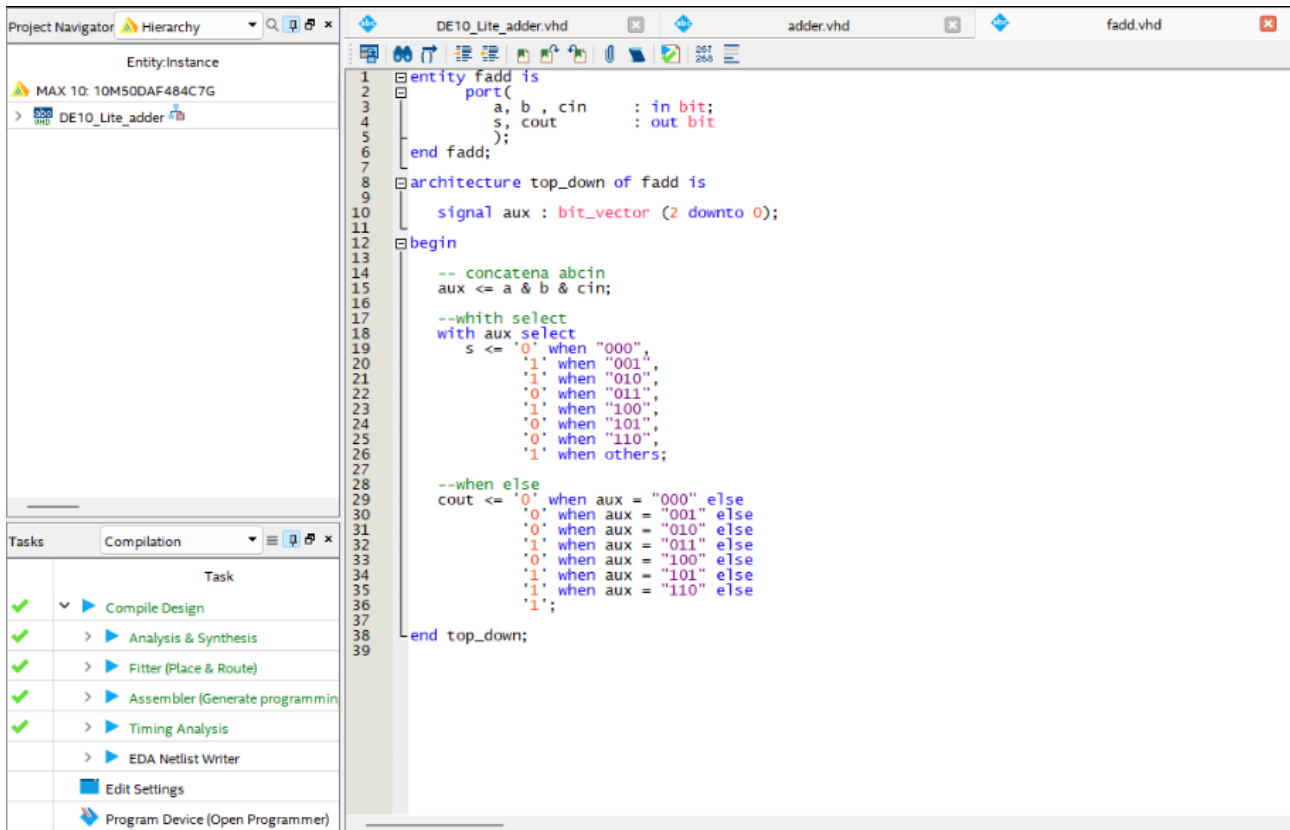
A elaboração do somador RPC (fig.6) valeu-se do uso do somador completo, como componente, há medida que compôs 4 somadores de 1 bit ligados em cascata, de forma que a saída *Cout* de um full-adder conectasse com a entrada *Cin* do somador correspondente ao próximo bit e a saída *S* correspondesse a um bit do resultado da soma. Valendo ressaltar que o full-adder correspondente ao bit menos significativo recebe sempre '0' na entrada *Cin*.

Figure 6: Código de somador RPC



A implementação do somador completo (fig.7), componente para 1 bit, valeu-se do uso das estruturas lógicas "WITH SELECT" e "WHEN-ELSE" pertencentes à linguagem VHDL, para as saídas *Cout* e *S*, respectivamente. Por fim, todo o código, compilação e reprogramação da FPGA foram feitos através do software Quartus, com a versão gratuita.

Figure 7: Código com uso do somador completo (para 1 bit)



Fonte: os autores

## 2.3 Previsão sobre utilização dos componentes TTL da família 74LSxx

Para essa previsão, foi considerado que seriam utilizados apenas circuitos integrados TTL da família 7408 (que possui 4 portas "and" de duas entradas) e da família 7486 (que possui 4 portas "xor" de duas entradas).

Assim, é possível concluir que, tomando como base a implementação do circuito da fig.2, cada somador completo, ligado em cascata com os demais, utiliza quatro portas "and" e duas portas "xor". Com isso, um somador RPC - possuindo 4 somadores completos ligados em cascata - necessitaria do uso de 4 componentes TTL 7408 e 2 componentes 7486, totalizando 16 portas "and" e 8 portas "xor", respectivamente.

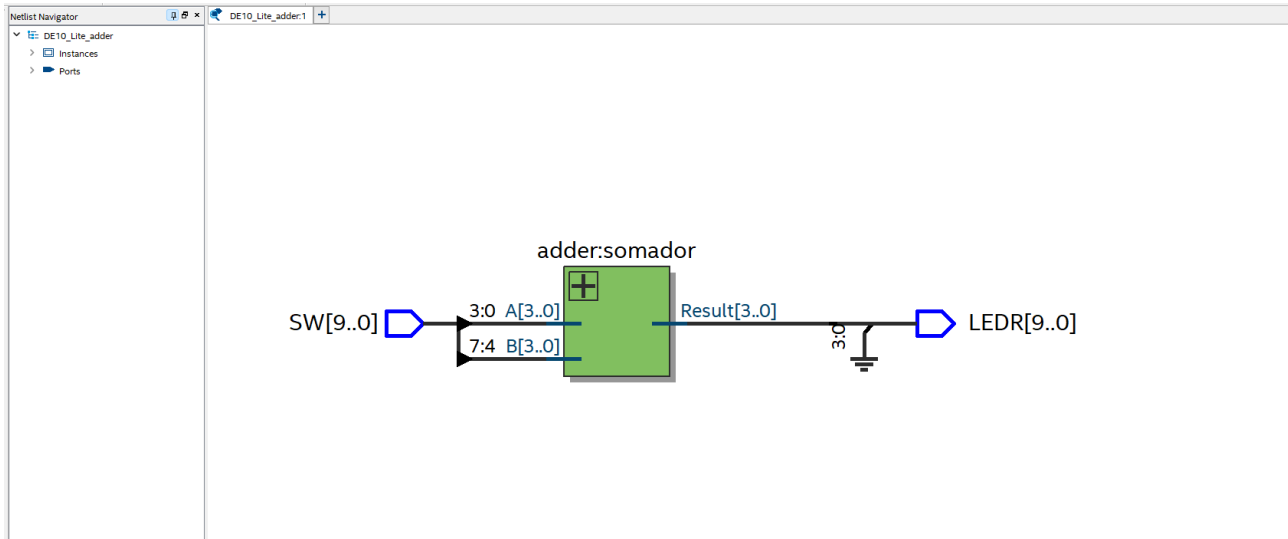
## 3 Conclusão

### 3.1 Circuito RTL

A partir do código em VHDL feito em laboratório, e usando a ferramenta de síntese disponibilizada pelo quartus, os seguintes circuitos foram obtidos:

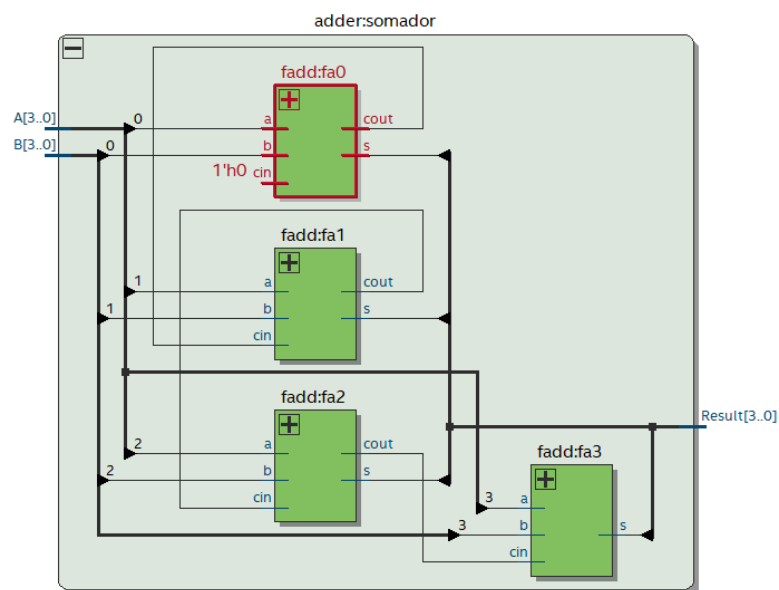


Figure 8: Nível 1 do circuito RTL sintetizado



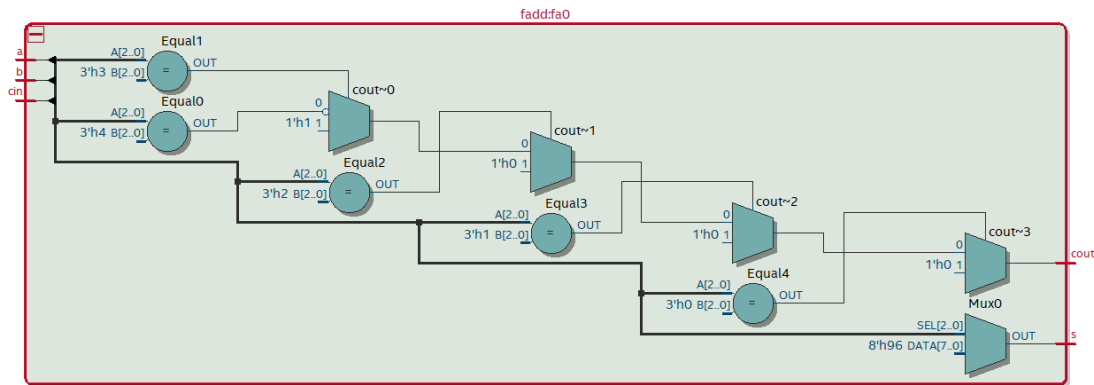
Fonte: os autores

Figure 9: Nível 2 do circuito RTL sintetizado



Fonte: os autores

Figure 10: Nível 3 do circuito RTL sintetizado

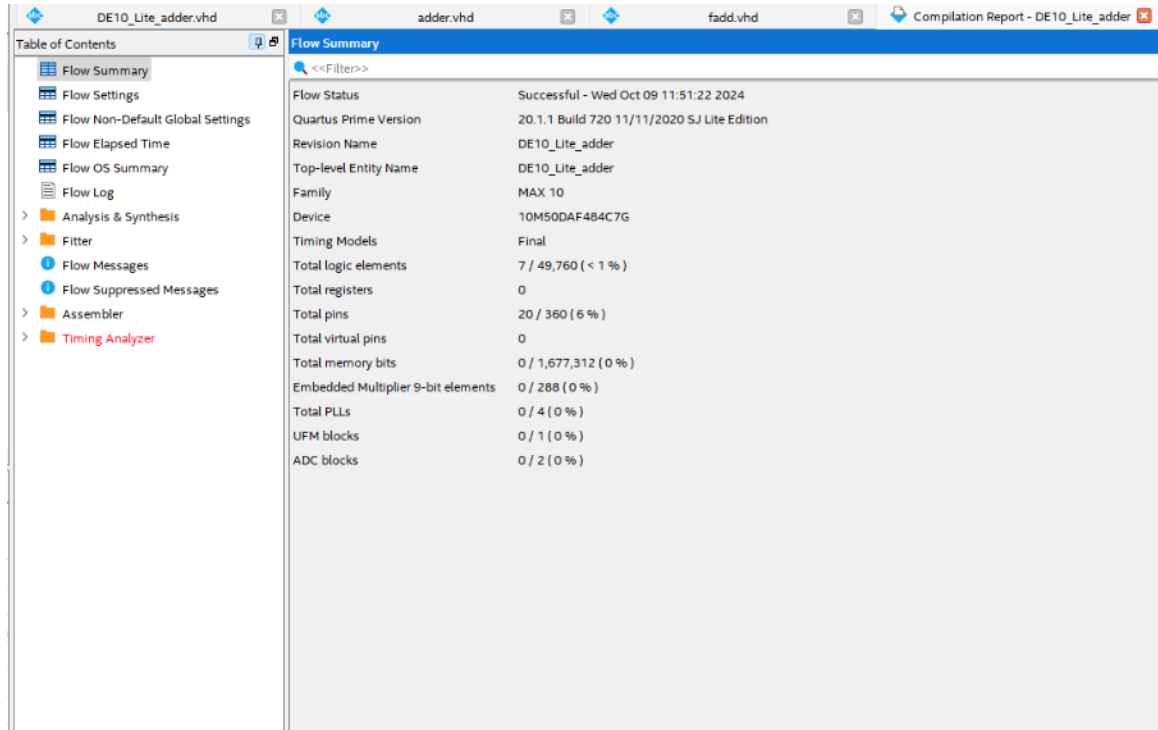


Fonte: os autores

### 3.2 Número de células lógicas

Ademais, por meio do resumo do funcionamento e constituição do circuito descrito em VHDL no software (fig11), chegamos à conclusão de que o circuito sintetizado apresentou um uso de 7 células lógicas e 20 pinos do dispositivo reconfigurável (FPGA).

Figure 11: Resumo de funcionamento

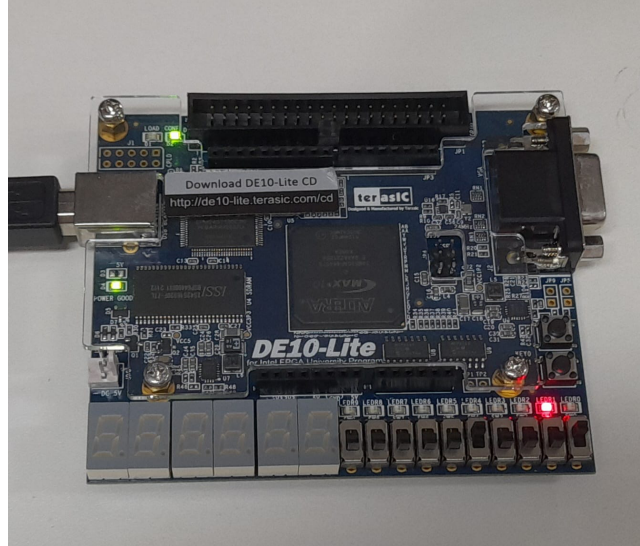


Fonte: os autores

### 3.3 Funcionamento do circuito

Nesse sentido, o circuito funcionou da forma esperada, já que, ao ligarmos as chaves  $Sw(3 : 0)$  e  $Sw(4 : 7)$ , ambamente, nas posição correspondentes ao valor binário  $0001_2$ , visualizamos o acendimento dos leds LEDR(3:0) correspondentemente ao valor binário  $0010_2$  conforme às imagens fig.12, ou seja, a saída do circuito (visualizada por meio do acendimento dos leds) foi igual à soma dos valores de entrada recebidos via ligamento das chaves.

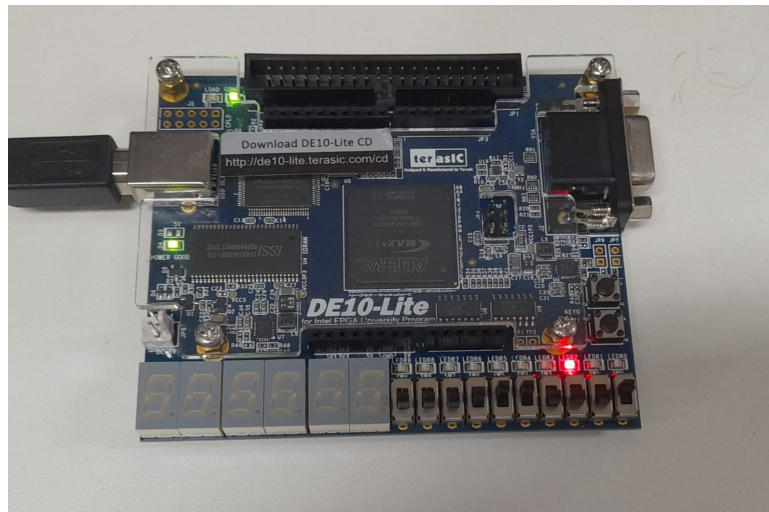
Figure 12: Acendimento de leds para  $A = 0001_2$  e  $B = 0001_2$



Fonte: os autores

Ademais, o mesmo comportamento foi observado ao ligarmos as chaves  $Sw(3 : 0)$  e  $Sw(4 : 7)$  nas posições correspondentes aos valores binários  $0001_2$  e  $0011_2$ , respectivamente, o que resultou na visualização do valor binário  $0100_2$  por meio do acendimento dos leds, conforme a fig.13.

Figure 13: Acendimento dos leds apra  $A = 0001_2$  e  $B = 0011_2$



Fonte: os autores

Por fim, tais resultados mostraram a efetividade do somador aritmético de 4 bits implementado em laboratório, o qual foi elaborado com o uso da separação de componentes em diferentes arquivos e o uso de comandos concorrentes da linguagem VHDL para a obtenção das saídas necessárias.