

# Expansão de ULA

---

## Autores

Gabriel A. F. Souza, Gustavo D. Colletta, Leonardo B. Zoccal, Odilon O. Dutra
---

Unifei
--------

## Histórico de Revisões

22 de janeiro de 2025	1.0	Primeira versão do documento.
-----------------------	-----	-------------------------------

# Tópicos

---

- ULA simples
- ULA com carry look-ahead
- Exercícios



# ULA simples

# Símbolo da ULA

---

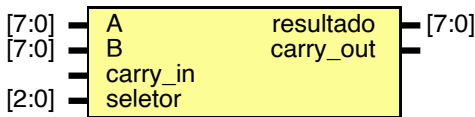


Figura 1: Símbolo da ULA.

Port name	Direction	Type	Description
A	input	[7:0]	Operando A
B	input	[7:0]	Operando B
carry_in	input		Carry-in
seletor	input	[2:0]	Seletor de operação
resultado	output	[7:0]	Resultado da operação
carry_out	output		Carry-out

# Implementação em Verilog

---

```
1 module ula_8bits (  
2     input  [7:0] A,           // Operando A  
3     input  [7:0] B,           // Operando B  
4     input  carry_in,          // Carry-in  
5     input  [2:0] seletor,      // Seletor de operação  
6     output [7:0] resultado,    // Resultado da operação  
7     output carry_out           // Carry-out  
8 );  
9     wire carry_mid; // Carry entre os dois módulos de 4 bits
```

# Implementação em Verilog

---

```
10 // Instanciando a ULA inferior (bits 3:0)
11 ula ula_lower (
12     .A(A[3:0]),
13     .B(B[3:0]),
14     .carry_in(carry_in),
15     .seletor(seletor),
16     .resultado(resultado[3:0]),
17     .carry_out(carry_mid)
18 );
```

# Implementação em Verilog

---

```
19 // Instanciando a ULA superior (bits 7:4)
20 ula ula_upper (
21     .A(A[7:4]),
22     .B(B[7:4]),
23     .carry_in(carry_mid),
24     .seletor(seletor),
25     .resultado(resultado[7:4]),
26     .carry_out(carry_out)
27 );
28 endmodule
```

# Expansão simples

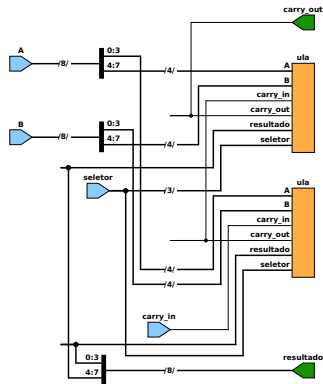


Figura 2: O *carry out* da ULA menos significativa é ligado ao *carry in* da mais significativa.

# Expansão simples

---

## ① Divisão em módulos de 4 bits:

- O operando de 8 bits é dividido em dois operandos de 4 bits (inferior e superior).

## ② Carry entre os módulos:

- O carry\_out do módulo inferior é conectado ao carry\_in do módulo superior.

## ③ Parâmetros compartilhados:

- O seletor é comum para ambas as ULAs, garantindo que elas executem a mesma operação simultaneamente.

# ULA com carry look-ahead

## Símbolo da ULA com carry look-ahead

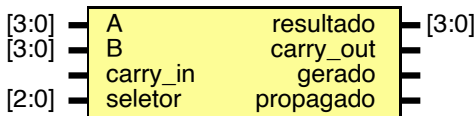


Figura 3: Símbolo da ULA.

Port name	Direction	Type	Description
A	input	[3:0]	Operando A
B	input	[3:0]	Operando B
carry_in	input		Carry-in
seletor	input	[2:0]	Seletor de operação (3 bits)
resultado	output	[3:0]	Resultado da operação
carry_out	output		Carry-out
gerado	output		Termo gerado (G) para o MSB (bit 3)
propagado	output		Termo propagado (P) para o MSB (bit 3)

## Expansão com carry look ahead

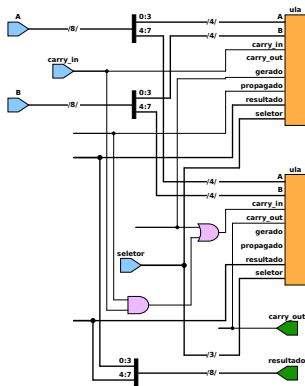


Figura 4: O cálculo do *carry in* da ULA mais significativa é realizado pelas portas lógicas.

# Expansão com carry look ahead

---

## ❶ Módulo LSB:

- Processa os bits menos significativos ( $A[3 : 0]$  e  $B[3 : 0]$ ).
- Gera os sinais `carry_out`, gerado (G), e propagado (P).

## ❷ Módulo MSB:

- Recebe como `carry_in` o valor calculado com base no gerado (G) e propagado (P) do módulo LSB:

$$C_{in_{MSB}} = G_{LSB} + (P_{LSB} \cdot C_{in})$$

- Processa os bits mais significativos ( $A[7 : 4]$  e  $B[7 : 4]$ ).
- Calcula o `carry_out` final da ULA de 8 bits.

## ❸ Sem exposição de G e P no módulo topo:

- Embora os módulos de 4 bits usem gerado e propagado internamente, o módulo topo (`ula_bits`) oculta esses sinais, simplificando a interface externa.

# Exercícios

# Exercício 1

---

- 1 Modifique a ULA fornecida para que a operação de soma seja realizada com *carry look-ahead*. Além disso, externalize os sinais P (propagado) e G (gerado) do bit mais significativo dessa ULA.
- 2 Escreva um arquivo de *testbench* para testar as alterações realizadas.

## Exercício 2

---

- 1 Utilizando dois módulos da ULA do exercício 1, realize a expansão para uma ULA de 8 bits, semelhante à apresentada na aula.
- 2 Escreva um arquivo de *testbench* para testar a ULA de 8-bits.

## Exercício 3

---

- 1 Utilizando três módulos da ULA do exercício 1, realize a expansão para uma ULA de 12-bits.
- 2 Escreva um arquivo de *testbench* para testar a ULA de 12-bits.