

ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES I

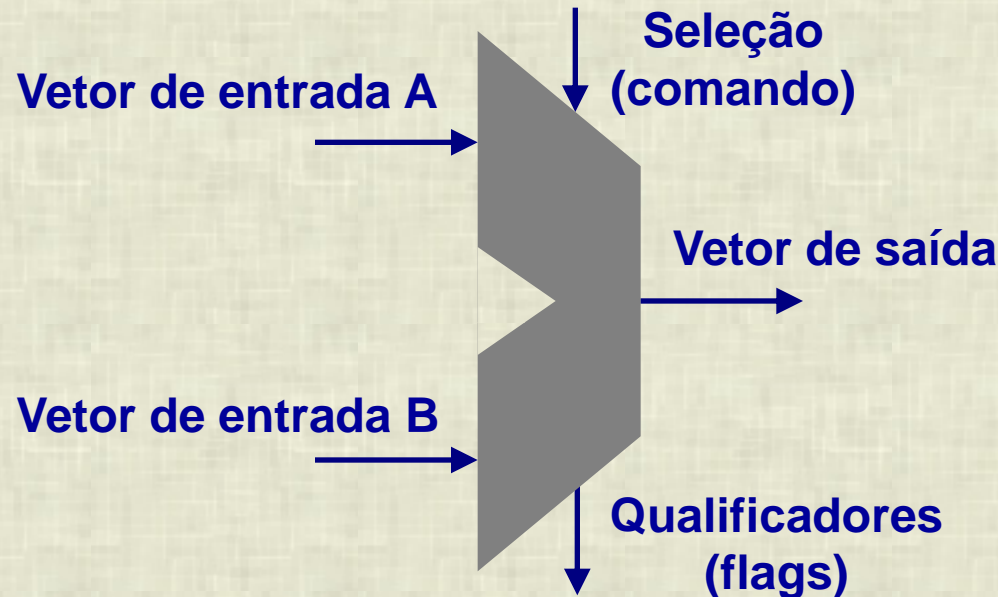
Projeto de Circuito Combinacional

Unidade Lógica e Aritmética (ULA)

prof. Dr. César Augusto M. Marcon
prof. Dr. Edson Ifarraguirre Moreno

ULA

- **Unidade Lógica e Aritmética (ULA)** é um circuito que realiza funções lógicas e aritméticas
- É um dos componentes de transformação de dados principais de um processador
- Normalmente implementado de forma combinacional
- **Representação:**



ULA - Funcionalidades Lógicas

• **Diversas são as funcionalidades lógicas. Dentre as mais comuns estão:**

- E lógico das entradas
- Ou lógico das entradas
- Ou exclusivo lógico das entradas
- Complemento de uma das entradas

• **A seleção de qual operação será realizada é obtida pela porta de comando**

- Normalmente controlada pela unidade de controle do processador onde se encontra a ULA

• **Operações lógicas usam normalmente apenas os qualificadores Z (zero) e N (negativo)**

- Qualificadores de V (overflow) e C (carry) não são considerados, pois operações lógicas não alteram o valor dos mesmos

ULA - Funcionalidades Aritméticas

- **Dentre as funcionalidades aritméticas mais comuns estão:**

- Soma das entradas
- Subtração das entradas
- Deslocamento de uma das entrada
- Rotação de uma das entradas
- E variações das funcionalidades acima utilizando a flag C

- **A seleção de qual operação será realizada é obtida pela porta de comando**

- **Operações aritméticas fazem uso dos quatro qualificadores vistos até então (Z, N, V, C)**

Especificação de uma ULA de 4 Bits

- **Projetar uma ULA em VHDL com as seguintes funcionalidades:**

- Soma
- Subtração
- Incremento
- Decremento
- E lógico
- Ou lógico
- Ou exclusivo lógico
- Complemento

- **Para dar suporte a funcionalidade completa do processador, esta ULA deve ter os seguintes qualificadores:**

- Carry
- Negativo
- Zero
- Overflow

- **A ULA deve ter as portas de entrada e saída com 4 bits**

Especificação de uma ULA de 4 Bits

- Definição da relação entre a codificação e a seleção de comandos:

Operação	Codificação
Soma	000
Subtração	001
Incremento	010
Decremento	011
E lógico	100
Ou lógico	101
Ou exclusivo lógico	110
Complemento	111

- A implementação da ULA deve ser feita utilizando uma descrição comportamental

Implementação de uma ULA de 4 Bits (Entidade)

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity Ula4bits is
    port
    (
        A, B: in STD_LOGIC_VECTOR(3 downto 0);
        oper: in STD_LOGIC_VECTOR(2 downto 0);
        N, Z, C, V: out STD_LOGIC;
        S: out STD_LOGIC_VECTOR(3 downto 0)
    );
end Ula4bits;
```


Implementação de uma ULA de 4 Bits (Arquitetura)

```

architecture ula4bits of ula4bits is
    signal iA, iB, i_S: STD_LOGIC_VECTOR(4 downto 0);
    signal SS: STD_LOGIC_VECTOR(3 downto 0);
begin

    iA <= A(3) & A; -- Extensão de sinal
    iB <= B(3) & B;
    SS <= i_S(3 downto 0);
    S <= SS;

    with op select
        i_S <= iA + iB when "000",
              iA - iB when "001",
              iA + 1 when "010",
              iA - 1 when "011",
              iA and iB when "100",
              iA or iB when "101",
              iA xor iB when "110",
              not iA when others;

    N <= '1' when SS < 0 else '0';
    Z <= '1' when SS = 0 else '0';
    C <= i_S(4);
    V <= '1' when
        ((op="000") and ((iA>0 and iB>0 and SS<0) or (iA < 0 and iB < 0 and SS > 0))) or
        ((op="010" or op="011") and ((iA > 0 and SS < 0) or (iA < 0 and SS > 0))) or
        ((op="001") and ((iA>0 and iB<0 and SS<0) or (iA < 0 and iB > 0 and SS > 0)))
    else '0';

end ula4bits;

```


Implementação de uma ULA de 4 Bits

•Exercício:

- Faça alguns exemplos de vetores e teste se os quatro qualificadores estão implementados corretamente
- Pergunta:

•A descrição comportamental apresentada poderia ser mais clara?

- Sim. Bastaria usar uma definição do que é cada operação. Isto pode ser obtido com uma **definição de um tipo** em um **pacote**
- Exemplo:
 - Definição do tipo OP_ULA no pacote ULA, conforme a especificação

```
package ULA is
  type OP_ULA is
  (
    soma,
    subtracao,
    incr,
    decr,
    op_and,
    op_or,
    op_xor,
    op_not
  );
end ULA;
```

Implementação de uma ULA de 4 Bits

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
use work.ULA.all; -- Inclusão do pacote ULA
entity Ula4bits is
    port(
        A, B: in STD_LOGIC_VECTOR(3 downto 0);
        oper: in OP_ULA;
        N, Z, C, V: out STD_LOGIC;
        S: out STD_LOGIC_VECTOR(3 downto 0)
    );
end Ula4bits;
```

Implementação de uma ULA de 4 Bits (Arquitetura)

```

architecture ula4bits of ula4bits is
    signal iA, iB, i_S: STD_LOGIC_VECTOR(4 downto 0);
    signal SS: STD_LOGIC_VECTOR(3 downto 0);
begin

    iA <= A(3) & A;
    iB <= B(3) & B;
    SS <= i_S(3 downto 0);
    S <= SS;

    with op select
        i_S <= iA + iB when soma,
              iA - iB when subtracao,
              iA + 1 when inc,
              iA - 1 when dec,
              iA and iB when op_and,
              iA or iB when op_or,
              iA xor iB when op_xor,
              not iA when others;

    N <= '1' when SS < 0 else '0';
    Z <= '1' when SS = 0 else '0';
    C <= i_S(4);
    V <= '1' when
        ((op=soma) and ((iA>0 and iB>0 and SS<0) or (iA < 0 and iB < 0 and SS > 0))) or
        ((op=inc or op=dec) and ((iA > 0 and SS < 0) or (iA < 0 and SS > 0))) or
        ((op=subtracao) and ((iA>0 and iB<0 and SS<0) or (iA < 0 and iB > 0 and SS > 0)))
    else '0';

end ula4bits;

```

Exercício

- **Projete a mesma especificação de ULA, mas agora de forma estrutural. Para tanto, utilize as descrições de circuitos aritméticos descritos nas aulas anteriores e complemente os mesmos com as novas funcionalidades**
- **Complemente o projeto da ULA inserindo instruções de multiplicação e divisão. Para tanto, refaça a ULA de forma a esta ter uma saída com o dobro de bits. Organize estes bits para ter funcionalidade adequada para ponto flutuante na divisão e ter um número inteiro maior na multiplicação. Faça, também, uma codificação adequada para as novas instruções**