

# *PCS3115 - Sistemas Digitais I - Trabalho 2*

*por Bruno de Carvalho Albertini*

*12/10/2021*

Neste trabalho você desenvolverá dois dos principais componentes de um computador: a ULA (Unidade Lógica e Aritmética) e o registrador. Ambos os componentes serão usados (com modificações apropriadas) em trabalhos futuros.

ALU: *Arithmetic and Logic Unit*

## *Introdução*

Um processador pode ser visto como um projeto que utiliza o paradigma de unidade de controle e fluxo de dados. A unidade de controle é responsável pelo ciclo de busca, decodificação, execução e gravação dos dados, ciclo este que rege o funcionamento de um computador de uso geral pela sua característica programável. Já o fluxo de dados é composto por elementos de memória (e.g. banco de registradores), componentes de controle de fluxo (quase sempre combinatórios) e **unidades funcionais**.

O cálculo computacional é realizado nas unidades funcionais, portanto não é exagero afirmar que são os componentes principais de um processador. É possível construir máquinas com um ou até mesmo sem registradores, mas uma máquina sem uma unidade funcional simplesmente não realiza computação alguma.

Como as unidades funcionais são muito comuns, é costume juntar as principais funções computacionais em uma única unidade, que chamamos de ULA. Como o próprio nome diz, uma ULA reúne em um único componente operações lógicas (e.g. AND, OR, XOR, etc.) e aritméticas (adição, subtração, etc.), além de operações de comparação (menor, maior, igual, etc.). Para se ter uma ideia da sua importância, cerca de 40% das instruções em um processador moderno utilizam a ULA.

Outro componente muito importante em um processador são os elementos de memória dentro do processador, chamados de registradores, que também estão dentro do fluxo de dados. Os registradores são usados no computador para armazenar valores (e.g. variáveis dos programas ou endereço de retorno da função chamada).

## Atividades

A ULA que você deve construir, como várias outras implementações, agrupa algumas unidades funcionais. Esta ULA em particular é capaz de realizar as operações aritméticas adição e subtração, as lógicas AND, OR e NOT, além de três especiais: duas que copiam uma das entradas para a saída uma que e inverte uma saída. A Tabela 1 sumariza as operações capazes de serem realizadas nesta ULA.

S	Unidade	Descrição
000	nenhuma	copia $A$ para a saída
001	adição	$A + B$
010	AND	$A \& B$ , bit a bit
011	OR	$A   B$ , bit a bit
100	subtração	$A - B$
101	NOT	$A$ , bit a bit
110	inversão	inverte a ordem dos bits de $A$
111	nenhuma	copia $B$ para a saída

Tabela 1: Operações que podem ser realizadas pela ULA. O sinal  $S$  seleciona qual o operação a ULA deve realizar.

As operações de soma e subtração sempre consideram os operandos e o resultado em complemento de 2. As operações lógicas AND, OR e NOT são bit a bit. A operação de inversão inverte os bits menos significativos e mais significativos do operando  $A$  (e.g. se  $A = 01101$  a operação resulta em  $10110$ ).

```
entity alu is
  generic (
    size : natural := 8
  );
  port (
    A, B : in  bit_vector(size-1 downto 0); -- inputs
    F : out bit_vector(size-1 downto 0); -- output
    S : in  bit_vector(2 downto 0); -- op selection
    Z : out bit; -- zero flag
    Ov : out bit; -- overflow flag
    Co : out bit -- carry out
  );
end entity alu;
```

Listagem 1: Entidade para a ULA

Acima podemos ver a entidade VHDL e o diagrama da ULA. No diagrama, a entrada de controle “ALU operation” corresponde ao sinal  $S$  da entidade ou da tabela de operações. Os *flags* produzidos por esta ULA são  $Z$ , que é alto quando o resultado é zero,  $Ov$ , que é alto quando o resultado produz um *overflow*, e  $Co$ , que é alto quando o resultado produz um *carry-out*. O *flag*  $Z$  deve ser honrado em todas as operações, mas os demais só são honrados na soma e na subtração.

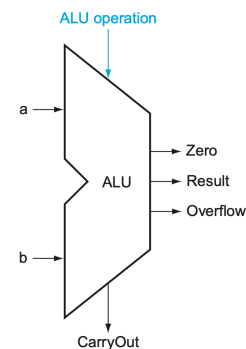


Figura 1: Diagrama da ULA.

## Atividades

**T2A1** Projete e faça uma descrição em VHDL de uma ULA conforme a entidade da Listagem 1. Sua entidade deve seguir exatamente a entidade da listagem e a sua arquitetura deve ser combinatória (não é permitido usar processo). As funções da sua ULA devem seguir a Tabela 1 e sua ULA deve honrar o parâmetro (i.e. deve funcionar para qualquer tamanho instanciado).

Trabalho 2, Atividade 1, 10 envios, maior nota, 6 pontos

**T2A2** Projete e descreva em VHDL um registrador seguindo a entidade da Listagem 2.

Trabalho 2, Atividade 2, 10 envios, maior nota, 4 pontos

```
entity d_register is
    generic(
        width: natural := 4;
        reset_value: natural := 0
    );
    port(
        clock, reset, load: in bit;
        d: in bit_vector(width-1 downto 0);
        q: out bit_vector(width-1 downto 0)
    );
end d_register;
```

Listagem 2: Entidade para o registrador

O registrador é do tipo D simples síncrono, sensível à borda de subida e com *reset* assíncrono ativo alto. O tamanho do registrador é parametrizável através do parâmetro *width* e seu projeto deve funcionar para qualquer tamanho. O valor da saída após o *reset* também é parametrizável através do parâmetro *reset\_value*, ou seja, após um *reset* o valor da saída deve ser o mesmo de *reset\_value*. O sinal de *load*, quando em alto, faz com que o registrador amostra a entrada *d* para a saída *q* na borda de subida do *clock*. Em qualquer outra situação o valor armazenado pelo registrador deve permanecer inalterado. Para esta atividade, é permitido o uso de processos.

### *Instruções para Entrega*

Para este trabalho está permitida apenas a biblioteca `numeric_bit` do pacote `ieee`. Na atividade A1 você não pode usar o `process` mas na atividade A2 você pode. As funções estão proibidas em ambas atividades. A violação destas restrições acarreta nota zero automaticamente, sem direito a revisão.

Restrições, preste atenção!

Para cada atividade deste trabalho, há um *link* específico no e-Disciplinas. Acesse-o somente quando estiver confortável para enviar sua solução. Em cada atividade, você pode enviar apenas um único arquivo com sua descrição VHDL em UTF-8. O nome do arquivo não importa, mas sim a descrição que está dentro. As entidades devem ser exatamente como as especificadas ou o juiz te atribuirá nota zero.

Veja as datas limites no e-Disciplinas!

Quando acessar o *link* no e-Disciplinas, o navegador abrirá uma janela para envio do arquivo. Selecione-o e envie para o juiz. Jamais recarregue a página de submissão pois seu navegador pode enviar o arquivo novamente, o que vai ser considerado pelo juiz como um novo envio e pode prejudicar sua nota final. Caso desista do envio, simplesmente feche a janela antes do envio.

Depois do envio, a página carregará automaticamente o resultado do juiz, quando você poderá fechar a janela. Se não quiser esperar o resultado, feche a janela após o envio e verifique sua nota no e-Disciplinas posteriormente. A nota dada pelo juiz é somente para a submissão que acabou de fazer. Sua nota na atividade poderá ser vista no e-Disciplinas e pode diferir da nota dada pelo juiz dependendo da estratégia de atribuição de notas utilizada pelo professor que montou o problema.

Pode demorar alguns segundos até o juiz processar seu arquivo.