

Introdução ao projeto de sistemas digitais

Prof. Ilan Sousa Correa

Universidade Federal do Pará (UFPA)

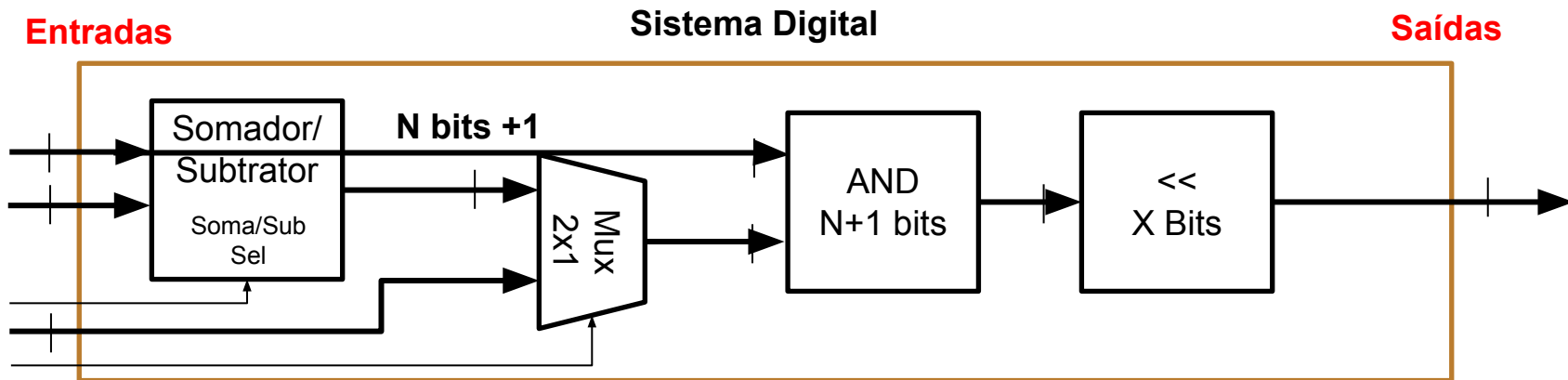
Instituto de Tecnologia (ITEC)

Faculdade de Eng. da Computação e Telecomunicações (FCT)

Simulação com testbench com dados a partir de arquivos

Revisitando o exemplo da aula 1.3

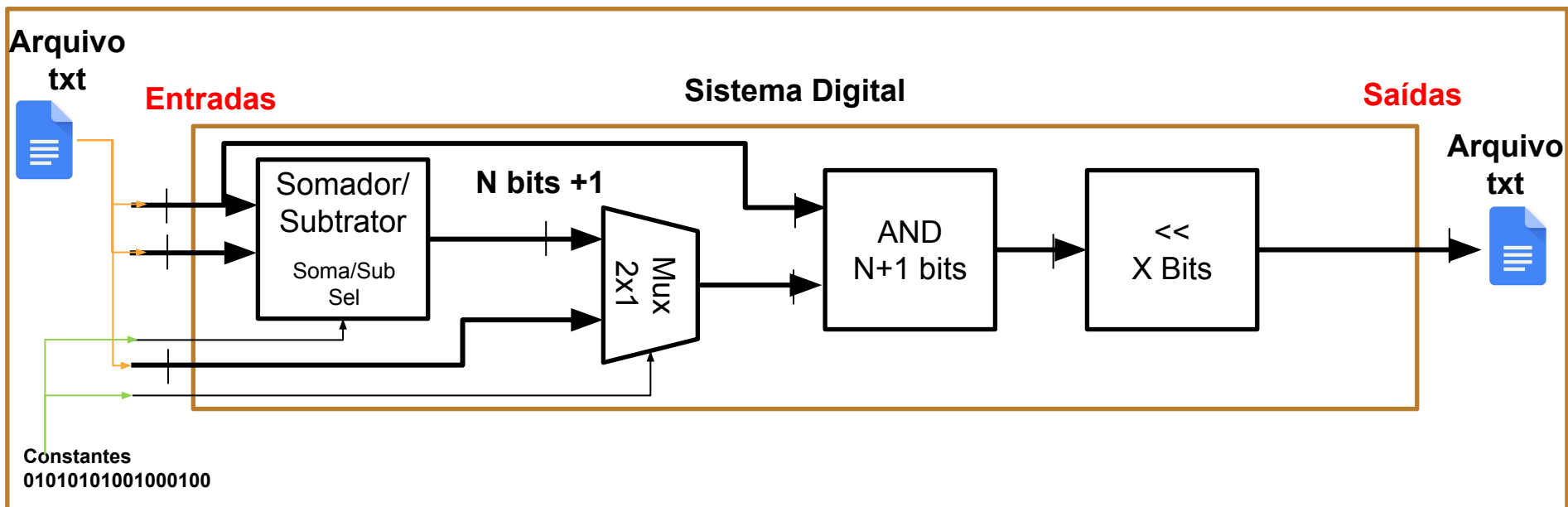
- Carregando dados a partir de arquivos e salvando resultados em arquivos
- Geração de entradas a partir de constantes no código VHDL



Simulação com testbench com dados a partir de arquivos

Simulação a ser mostrada na aula

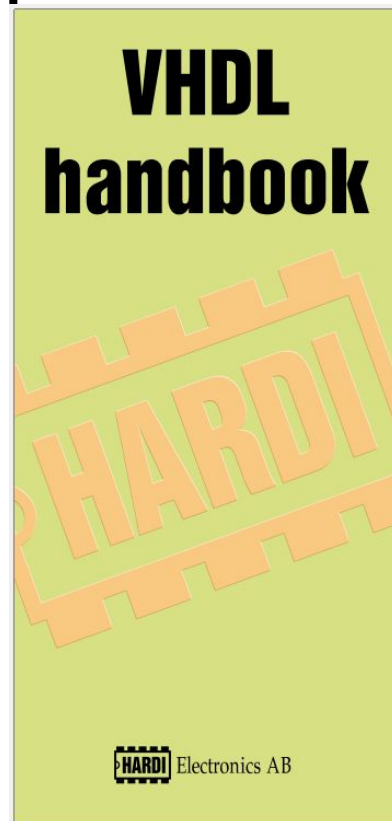
Testbench



Simulação com testbench com dados a partir de arquivos

Material de referência

- VHDL Handbook □ livro gratuito
- <https://redirect.cs.umbc.edu/portal/help/VHDL/VHDL-Handbook.pdf>
- Documento para consulta rápida sobre VHDL
 - Bibliotecas e funções
 - Estruturas e palavras reservadas da linguagem
 - Etc
- Usaremos esse material par consultar sobrecarga de funções e funções disponíveis em cada biblioteca.



Simulação com testbench com dados a partir de arquivos

Material de referência

- Exemplo de sobrecarga

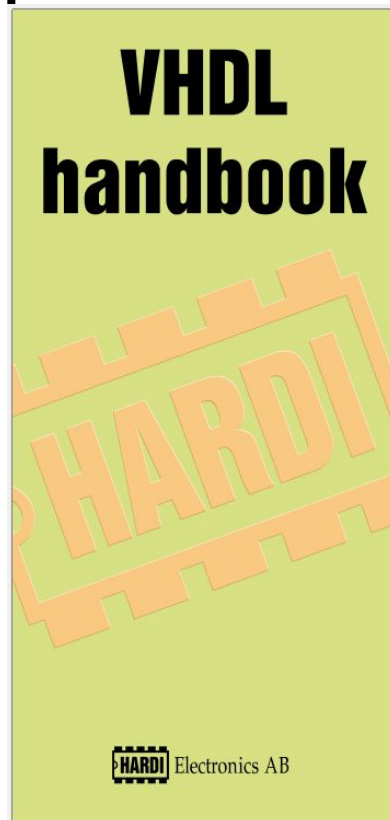
NUMERIC_BIT

Types and subprograms for designing with arrays of BIT. Is precompiled into the library "IEEE" (accessed via `USE IEEE.NUMERIC_BIT.ALL`).

```
package NUMERIC_BIT is
  type UNSIGNED is array (NATURAL range <>) of BIT;
  type SIGNED is array (NATURAL range <>) of BIT;

  function "abs" (ARG:SIGNED) return SIGNED;
  function "-" (ARG:SIGNED) return SIGNED;
  function "+" (L,R:UNSIGNED) return UNSIGNED;
  function "+" (L,R:SIGNED) return SIGNED;
  function "+" (L:UNSIGNED; R:NATURAL) return UNSIGNED;
  function "+" (L:NATURAL; R:UNSIGNED) return UNSIGNED;
  function "+" (L:INTEGER; R:SIGNED) return SIGNED;
  function "+" (L:SIGNED; R:INTEGER) return SIGNED;
  function "-" (L,R:UNSIGNED) return UNSIGNED;
  function "-" (L,R:SIGNED) return SIGNED;
  function "-" (L:UNSIGNED; R:NATURAL) return UNSIGNED;
  function "-" (L:NATURAL; R:UNSIGNED) return UNSIGNED;
  function "-" (L:SIGNED; R:INTEGER) return SIGNED;
  function "-" (L:INTEGER; R:SIGNED) return SIGNED;
  function "*" (L,R:UNSIGNED) return UNSIGNED;
  function "*" (L,R:SIGNED) return SIGNED;
  function "*" (L:UNSIGNED; R:NATURAL) return UNSIGNED;
  function "*" (L:NATURAL; R:UNSIGNED) return UNSIGNED;
  function "*" (L:SIGNED; R:INTEGER) return SIGNED;
  function "*" (L:INTEGER; R:SIGNED) return SIGNED;
  function "/" (L,R:UNSIGNED) return UNSIGNED;
  function "/" (L,R:SIGNED) return SIGNED;
  function "/" (L:UNSIGNED; R:NATURAL) return UNSIGNED;
```

Sobrecarga do operador +
Possibilidades de operandos



Simulação com testbench com dados a partir de arquivos

Material de referência

- Exemplo de sobrecarga

TEXTIO

Types and subprograms to handle text files. Is precompiled into the library "STD" (accessed via `USE STD.TEXTIO.ALL`).

```
package TEXTIO is
  type LINE is access STRING;
  type TEXT is file of STRING;
  type SIDE is (RIGHT, LEFT);
  subtype WIDTH is natural;

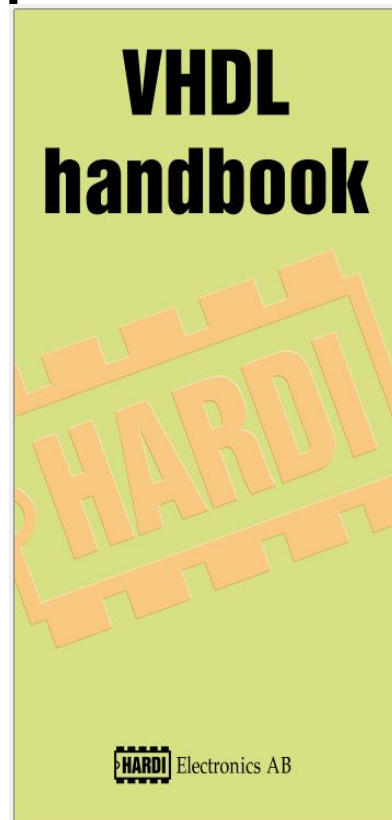
  file INPUT : TEXT open READ_MODE is "STD_INPUT";
  file OUTPUT : TEXT open WRITE_MODE is "STD_OUTPUT";

  procedure READLINE (file F:TEXT; L:out LINE);
  procedure READ (L:inout LINE; VALUE:out BIT;
    GOOD:out BOOLEAN);
  procedure READ (L:inout LINE; VALUE:out BIT);
  procedure READ (L:inout LINE; VALUE:out BIT_VECTOR;
    GOOD:out BOOLEAN);
  procedure READ (L:inout LINE; VALUE:out BIT_VECTOR);
  procedure READ (L:inout LINE; VALUE:out BOOLEAN;
    GOOD:out BOOLEAN);
  procedure READ (L:inout LINE; VALUE:out BOOLEAN);
```

LRM

§ 14.3

Sobrecarga da função READ
Possibilidades de entradas



Primeira atividade avaliativa

Projeto de sistemas digitais combinacionais em VHDL – Projeto 1

- Deseja-se projetar uma **Unidade Lógica Aritmética (ALU)** que realiza as operações de soma, subtração, AND, OR, NOT, deslocamento de 1 bit para esquerda, deslocamento de 1 bit para direita, incremento de 1, decremento de 1;
- Esta ALU tem duas entradas de dados A e B;
- Uma saída S.
- O resultado será atribuído à S de acordo com uma das operações selecionadas, para isso há uma entrada auxiliar, denominada SEL que indica que operação deverá ser realizada.
- Deve-se projetar um arquivo VHDL para cada operação, que deverão ser conectados em outro arquivo VHDL (top-level). Deve-se usar a biblioteca numeric_std (operandos +,-,and, ...)

Primeira atividade avaliativa

Projeto de sistemas digitais combinacionais em VHDL – Projeto 2

- Deseja-se projetar um **conversor de número inteiro binário** para mostrar o caractere correspondente em um display de 7 segmentos. Conversor da tabela ASCII para representação em 7 segmentos. Projetar um arquivo VHDL para esta conversão;
- Criar em um arquivo top-level duas instâncias do conversor e conectar para converter dois caracteres por vez.
- O arquivo top-level deve ter duas entradas A e B e duas saídas S1 e S2;
- O arquivo top-level deve ter ainda uma entrada auxiliar para permitir intercambiar as entradas convertidas, ou seja,
 - A ☐ Conversão ☐ S1 e B ☐ Conversão ☐ S2
 - A ☐ Conversão ☐ S2 e B ☐ Conversão ☐ S1

Primeira atividade avaliativa

Projeto de sistemas digitais combinacionais em VHDL – Projeto 3

- Deseja-se projetar um somador completo de 1 bit **a partir de portas lógicas** usando funções lógicas da biblioteca `std_logic_1164`;
- A partir do somador de 1 bit, deve-se, então, construir um arquivo top-level que implemente um somador completo com número de bits configurável a partir do somador de 1 bit;
- O somador de múltiplos bits (arquivo top-level) deve ainda poder ser configurado para permitir selecionar soma ou subtração;
 - Uma subtração pode ser implementada como uma soma do operando A com o complemento a 2 do operando B
 - Deve-se construir uma arquivo VHDL auxiliar que gera o complemento a 2 do operando B

Primeira atividade avaliativa

Projeto de sistemas digitais combinacionais em VHDL – Informações gerais

- Desenvolvimento de um testbench para cada um dos três projetos Este repositório concentra todo o material e pesquisa para auxiliar futuras implementações.
 - Deve-se criar um testbench para avaliar a funcionalidade desenvolvida
 - Os dados de entrada devem ser lidos a partir de um arquivo de texto e os resultados devem ser salvos em um arquivo de texto
- Equipes de até 3 alunos
 - Apresentação da solução no dia 27/09
 - 15 minutos de apresentação + 5 minutos de discussão
 - Incluir na apresentação: diagrama de blocos do sistema projetado, capturas de telas do simulador, comparação/comentário do resultado gera nos arquivos, dados de entrada e saída, etc.
 - O projetos serão sorteados entre as equipes.

Dúvidas?