Design de Computadores

Aula 1

Insper

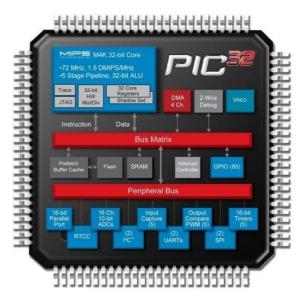
Apresentação e Introdução ao VHDL

Destino:

Projetar um processador de 32 bits.

Caminho:

- Analisar as instruções e inferir a estrutura subjacente;
- Aplicar a abstração UC + FD;
- Implementar e depurar o circuito criado;
- Propor melhorias que aumentem o desempenho.





Veículos de auxílio:

- CAD para desenvolvimento de hardware:
 - Com suporte para VHDL;
 - E gravação em FPGA.
- Simulação de circuitos;
- Análise da temporização do circuito;



Comprovação:

- Projetos:
 - Relógio / cronometro;
 - Processador de 32 bits.
- Duas provas.
- Dois quizzes.
- Quizzes semanais.
- Entregas quinzenais.



Balanço final:

- Capacidade de projetar, simular e depurar:
 - Circuitos digitais de média complexidade.
- Capacidade de análise dos requisitos da organização, de processadores, necessários para atender uma dada arquitetura;
- Capacidade de identificar os impactos no desempenho de trechos de programas em assembly.



Por quê projetar um microprocessador personalizado?

- Aplicações específicas, onde o cálculo pode ser acelerado através de instruções adequadas:
 - Equações diferenciais;
 - Criptografia.

IEEE EMBEDDED SYSTEMS LETTERS, VOL. 3, NO. 4, DECEMBER 2011

113

A Custom FPGA Processor for Physical Model Ordinary Differential Equation Solving

Chen Huang, Frank Vahid, and Tony Givargis

Abstract—Models of physical systems, such as of human physiology or of chemical reactions, are typically comprised of numerous ordinary differential equations (ODEs). Today's designers commonly consider simulating physical models utilizing field-programmable gate arrays (FPGAs). This letter introduces a resource efficient custom processor—the differential equation processing element, or DEPE—specifically designed for efficient solution of ODEs on FPGAs, and also introduces its accompanying compilation tools. We show that a single DEPE on a Xilinx Virtex6

FPGAs implement circuits, wherein thousands of connected components execute in parallel. Circuits represent an excellent match for executing physical models. A physical model typically consists of thousands of ordinary differential equations (ODEs), each ODE representing part of the physical space, each part connected with neighboring parts, all parts executing in parallel. Execution consists of solving the ODEs using known iterative techniques. Thus, ODE solvers can be mapped to cir-



No nosso caso:

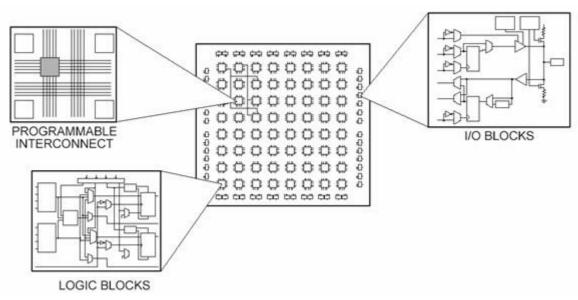
 Entender o funcionamento da CPU e seus blocos construtivos;

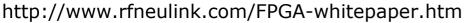
 Consequentemente, as oportunidades de melhoria de desempenho nos programas.

- O projeto de uma CPU desenvolve habilidades:
 - Que serão úteis no projeto de qualquer circuito digital.
- Entender os problemas envolvidos na criação de um conjunto de instruções.

FPGA

- Uma FPGA (field-programmable gate array):
 - É um conjunto de blocos lógicos;
 - Que podem ser interconectados (através de programação);
 - Para desempenhar funções complexas.
 - É o LEGO da eletrônica digital.









Estrutura da linguagem VHDL

- Possui dois conjuntos de construções:
 - Para a síntese:
 - Possui um conjunto menor de declarações. Usado para a conversão da descrição VHDL (ou esquema lógico) em uma lista de ligações que dependem do tipo da FPGA usada.
 - Para a simulação:
 - Usa um conjunto maior de declarações que adiciona declarações com informações de temporização.



O VHDL possui quatro elementos de projeto:

- Entidade (entity):
 - descreve a interface de uso do circuito (pinos).
- Arquitetura (architecture):
 - Descreve o funcionamento interno do circuito;
 - Pode ter partes concorrentes e partes sequenciais.
- Configurações (configuration):
 - Associa uma entidade com uma arquitetura, formando um circuito.
- Pacotes (package e package bodies):
 - Biblioteca contendo outros modelos, definições, etc...



Classes de objetos:

- Sinal (signal):
 - É a representação de um fio, usado para conectar módulos do projeto.
 - Possuem atributo de temporização (simulação).
- Constante (constant):
 - Possuem um único valor em todo o código.
 - Utilizadas para melhorar a legibilidade do código.
- Variável (variable):
 - Armazena valores intermediários entre expressões, com atribuição imediata.





Vetores de Objetos

A declaração do vetor define a ordem dos bits:

signal S1: bit_vector (4 DOWNTO 0); -- little endian (LSB à direita)



signal S2: std_logic_vector (0 TO 4); -- big endian (MSB à direita)

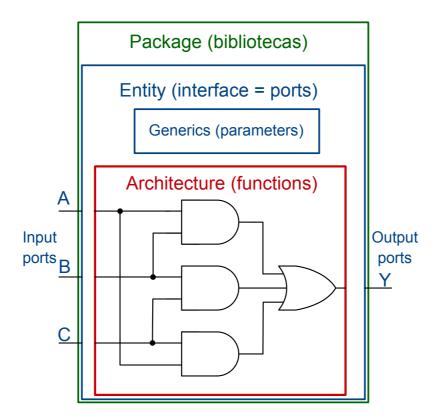


- O acesso aos elementos pode ser feito:
 - Pelo índice do elemento: S1(2).
 - Por um intervalo de elementos: S1(2 dowto 0).

Insper

Estrutura Mínima

- A estrutura mínima de uma descrição possui:
 - Package (biblioteca);
 - Entity (interface e parâmetros);
 - Architecture (funcionamento).





Exemplo da declaração de biblioteca:

library ieee;

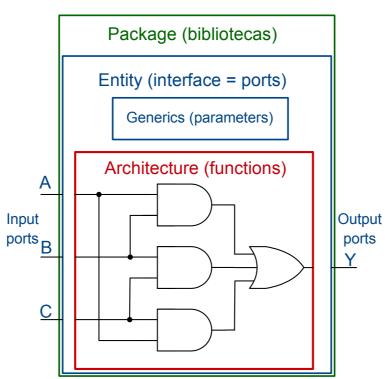
→ Definição da biblioteca.

use ieee.std_logic_1164.all; → Definição do pacote utilizado.

→ Definição do pacote

use ieee.numeric_std.all;

utilizado.





Exemplo da declaração da entidade:

```
entity meuCircuito is

Generic (DATA_WIDTH : natural := 8); → Parâmetros (opcional)

Port ( → Interface = entradas e saídas

Sinais (ou vetores de sinais) de entrada:

A : in std_logic;

B : in std_logic;

C : in std_logic;

Sinais (ou vetores de sinais) de saída:

Y : out std_logic );

end entity; → Fim da descrição da entidade
```

- A declaração de "port" possui o objeto "signal" implícito.
- Os objetos podem ser inicializados no momento da declaração.

Insper

 Exemplo de declaração da arquitetura para o código anterior:

architecture fluxoDados of meuCircuito is

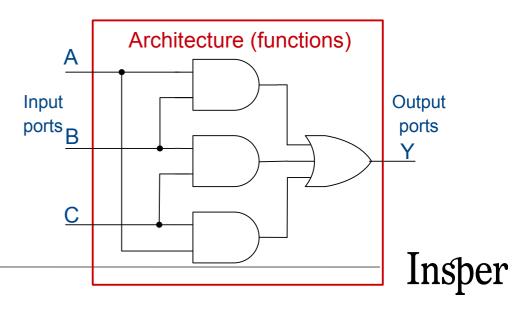
-- Área reservada para declarações.

begin

-- Descrição do funcionamento do circuito.

 $Y \le (A AND B) OR (B AND C) OR (A AND C);$

end architecture;



- A arquitetura pode ser descrita de três formas:
 - Comportamental:
 - Um conjunto de comandos concorrentes que representam o comportamento da entidade;
 - Um conjunto de comandos sequenciais que representam o comportamento da entidade.
 - Estrutural:
 - Um conjunto de componentes interconectados que representam a estrutura da entidade.
 - Fluxo de dados ou físico:
 - Um conjunto de comandos, concorrentes e/ou expressões lógicas, que representam o fluxo dos dados pelo circuito.

Lógica combinacional no Quartus.

Atividade disponível no Blackboard.

Pré requisito: instalar o Quartus Prime.

Insper

www.insper.edu.br

