

FPGA
(básico)

Manoel Eusebio de Lima

prototipaço / 2014

+ Agenda do curso

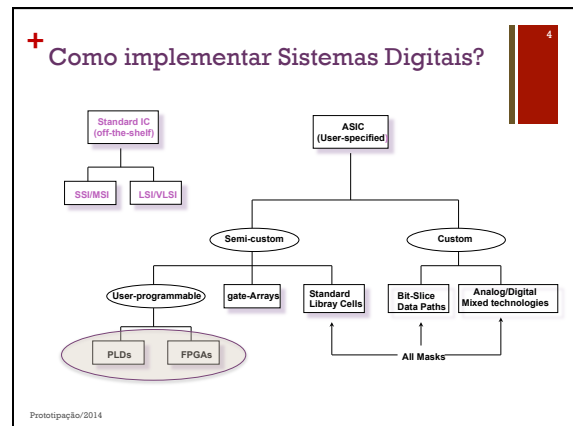
- Introdução à FPGAs
 - Histórico
 - Implementação de sistemas digitais
 - Dispositivo lógicos programáveis
 - FPGAs
 - Novo Paradigma de computação
 - Características técnicas
 - Arquitetura interna
 - Fluxo de projeto
 - Processamento de alto desempenho
 - Aplicações

prototipaço / 2014

+ Introdução

- Histórico
- Implementação de sistemas digitais
- Dispositivo lógicos programáveis
- FPGAs
 - Novo Paradigma de computação
 - Características técnicas
 - Arquitetura interna
 - Fluxo de projeto
 - Processamento de alto desempenho
 - Aplicações

Prototipaço/2014



+ PLD – Programmable Logic Device

- PLD
 - PROM - Programmable Read Only Memory
 - GAL - Generic Array Logic
 - PLA - Programmable Logic Array
 - PAL - Programmable Array Logic
 - CPLD - Complex Programmable Logic Devices
 - FPSC - Field Programmable System Chip
 - EPLD - Erasable Programmable Device
 - FPOA - Field Programmable Object Array
 - MPGA - Mask Programmable Gate Array
 - FPAA - Field Programmable Analog Array
 - FPGA – Field Programmable Gate Array

Prototipaço/2014

+ PLD

- Lançados a partir de 1970, os Dispositivo Logicos programáveis (PLDs) vieram oferecer maior flexibilidade e capacidade de implementação de circuitos lógicos digitais, até então restritos a circuitos que não possuíam nenhuma flexibilidade, tais como os circuitos TTLs e PROMs
- Estilos clássicos de circuitos PLDs
 - PLAs – Programmable Logic Array
 - PALs - Programmabel Array logic

Prototipaço/2014

+ PLA – Programmable Logic Array

- Este tipo de arquitetura oferece maior flexibilidade entre os PLDs.
- Nesta arquitetura o array de ANDs e array de ORs são programáveis.
- PLA tinham problema de desempenho (baixa velocidade) por possuir dois níveis de programação.

Prototipação/2014

+ PAL – Programmable Array Logic

- Este tipo de arquitetura é um espelho da estrutura PROM. Agora o array de ANDs é programável e o array de ORs é fixo.
- Obs:
 - PALs são mais rápidos que PLAs pois possuem apenas um nível de programação
 - PLAs permitem melhor otimização

Prototipação/2014

+ PLD - PLA

- PLA e PAL – devido a simplicidade de pouca capacidade de implementar funções lógicas estes dispositivos são normalmente denominados de SPLDs (Simple Programmable Logic Devices)
- Exemplo: Implementação de um decodificador BCD – 7 segmentos

Prototipação/2014

+ Decodificador 7-segmentos

Prototipação/2014

+ CPLDs - Complex Programmable Logic Devices

- Estes dispositivos foram lançados na década de 80 pela Altera, com o intuito de substituir os dispositivos tradicionais PLA e PAL na implementação de grandes circuitos.
- CPLDs são arquiteturas reconfiguráveis que usam PLAs ou PALs como suas unidades funcionais, e que têm tipicamente uma estrutura central ou hierárquica de conexão das unidades funcionais.
- Características:
 - Os CPLDs eram capazes de oferecer maior capacidade de implementar lógica complexa e circuitos maiores, com maior conexão (pinos).
 - Dispositivos PLAs e PALs possuíam poucos pinos (máximo, 32), além de um baixo desempenho (velocidade, consumo e tamanho)

Prototipação/2014

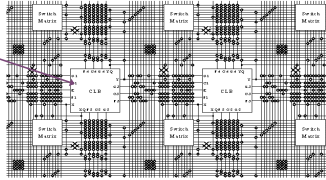
+ CPLD

- Uma coleção de PLDs em um único chip

Prototipação/2014

+ FPGA – Field programmable Gate Array

- Lançado pela Xilinx na década de 80, estes dispositivos podiam ser programados facilmente, e em geral, eram baseados em tecnologia SRAM.
- Os FPGAs não eram mais voltados apenas para implementação de lógica utilizando apenas ORs e ANDs.
- Agora blocos mais complexos estavam disponíveis nos blocos lógicos
 - LUT
 - Flip-Flop
 - MUXs



PrototipaçãO/2014

+ FPGA – Benefícios tecnológicos

- Desempenho
 - Paralelismo intrínseco em hardware, os FPGAs excedem o poder computacional de processadores tipo DSP, quebrando o paradigma de execução sequencial e realizando mais por ciclo de relógio.
- Baixa dissipação de potência
- Time to Market
 - A tecnologia FPGA oferece flexibilidade e capacidades de prototipação rápida de sistemas digitais frente ao desafio de tempo de projeto de novos produtos.
- Custo
 - No desenvolvimento de projetos, o custo de Engenharia Não Recorrente (NRE) com ASICs é bem maior que aquele realizado com FPGAs
- Confiabilidade
 - FPGAs, que não utilizam sistemas operacionais, minimizam as preocupações de confiabilidade com a execução paralela e um hardware determinista, dedicado a cada tarefa.
- Reconfigurabilidade (Manutenção de longo termo)
 - Permite longa vida ao projeto, podendo ser facilmente ajustado ao longo de sua vida útil a várias versões, atualizações do sistema digital.

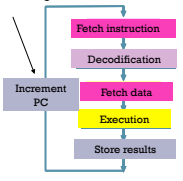
PrototipaçãO/2014

+ FPGA - Desempenho

Paradigma de software

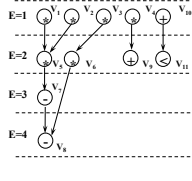
Execução Sequencial

Contador de Programa



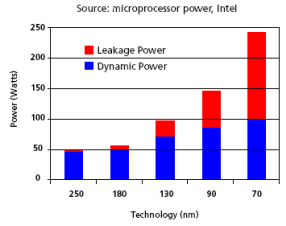
Paradigma de hardware

Concorrência (hardware)



PrototipaçãO/2014

+ FPGA – low power



Source: microprocessor power, Intel

Power (Watts)

Technology (nm)

Leakage Power

Dynamic Power

$f_{Op} > 3\text{GHz}$

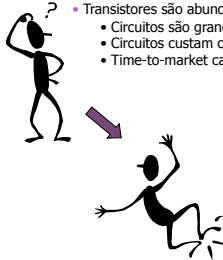
- Limite Atual de Velocidade \approx 3 GigaHertz (estagnação!!)

PrototipaçãO/2014

+ Time do Market/Custo

Projetos de Sistemas Digitais cada vez maiores

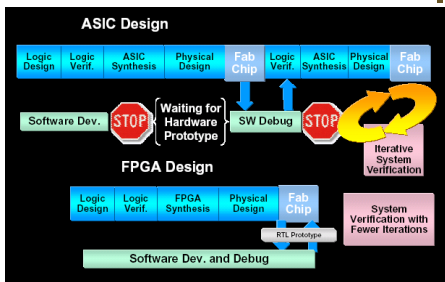
- Transistores são abundantes
- Circuitos são grandes o bastante e rápidos
- Circuitos custam caro e requerem muito tempo para fabricação
- Time-to-market cada vez menor



Comprar Circuitos integrados pré-fabricados que permitem integrar sistemas mais facilmente em plataformas

PrototipaçãO/2014

+ Time do Market/Custo



ASIC Design

Logic Design, Logic Verif., ASIC Synthesis, Physical Design, Fab Chip, Logic Verif., ASIC Synthesis, Physical Design, Fab Chip

Software Dev. [STOP] Waiting for Hardware Prototype [STOP] SW Debug [STOP] Iterative System Verification

FPGA Design

Logic Design, Logic Verif., FPGA Synthesis, Physical Design, Fab Chip

RTL Prototype, System Verification with Fewer Iterations

Software Dev. and Debug

PrototipaçãO/2014

+ Confiabilidade

- Não preciso componentes de software
- Não é necessário um S.O
- O algoritmo é implementado diretamente em hardware

Prototipação/2014

+ Reconfigurabilidade

Diferentes versões no mesmo hardware, aumentando a vida útil do produto.

Prototipação/2014

+ FPGAs - vantagens

- Velocidade - Processam informações mais rapidamente que por funções em software;
- Versatilidade - Em um Sistema Reconfigurável (RS) por exemplo, uma nova tarefa requer apenas que o usuário do sistema (PC, Workstation, etc) reconfigure o sistema desejado, reprogramando as conexões das portas lógicas, I/O, etc. em cada FPGA.
- Baixo custo. Por causa da reconfiguração de um RS, similar a um software no sistema, os custos de se criar um novo sistema (nova configuração) são baixos.
- Desenvolvimento rápido de protótipos.
- Relativamente fácil de se usar

Prototipação/2014

+ FPGAs – principais desvantagens

- Baixa velocidade em comparação a outras tecnologias como Sea-of-Gates, Standard-Cell, etc.
- Baixa densidade de portas lógicas.
- Alto custo para alta produção de chips (alto volume de produção).

Prototipação/2014

+ Estrutura de roteamento

Prototipação/2014

+ FPGA – estrutura interna

Prototipação/2014

+ FPGA - Configuração

- O FPGA deve ser visto como um dispositivo de "duas camadas" de configuração

Diagram illustrating the configuration of an FPGA. It shows a grid of logic blocks (bloques lógicos) and I/O blocks (bloques de I/O) connected to an input (Input) and output (Output). The logic blocks are connected to a logic core (Lógica do usuário) which includes a logic element (LE) and an adaptive logic module (ALM). The configuration is stored in SRAM (Memória de configuração) and controlled by SRAM control (Controle da SRAM). The diagram also shows a clock signal and a binary configuration string: 101000110010101010111010.

+ Bloco lógico

Detailed diagram of a logic block (CLB) showing its internal structure. It includes a logic element (LE) and an adaptive logic module (ALM). The CLB is connected to external signals (Sinais externos) and has a 4-bit result (4 variáveis). The diagram shows the internal components, including a look-up table (LUT) and multiplexers (MUX). The CLB is connected to a 4-bit result (4 variáveis) and a 3-bit result (3 variáveis). The diagram also shows the CLB's connection to a 4-bit result (4 variáveis) and a 3-bit result (3 variáveis).

+ LUT - Look-Up-Tables

- Multiplexadores $2^n:1$ podem implementar qualquer função de n variáveis, com $n-1$ variáveis de controle. As demais variáveis serão entrada para o mux.

$$F(A,B,C) = m_0 + m_2 + m_6 + m_7$$

$$= A'B'C' + A'B'C + ABC' + ABC$$

$$= A'B'(C') + A'B(C) + AB'(0) + AB(1)$$

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Diagram of a 4:1 MUX with inputs A, B, C and output F. The MUX is controlled by inputs S2, S1, S0. The output F is the result of the MUX operation.

+ Look-up table

- Tabela baseada em endereços de memória que implementa uma função lógica:
- Exemplo: somador de dois bits (half adder)

Bits a serem somados

Address	Data
A B C _{out} Σ	
0 0 0 0	0
0 1 0 1	1
1 0 0 1	0
1 1 1 0	0

Diagram of a 4x2 ROM with inputs A, B and outputs D₀, D₁, Σ, C_{out}. The ROM is used to implement a half adder function.

- Ao contrário do circuito do half-adder construído das portas lógicas, este dispositivo pode ser ajustado para executar qualquer função lógica com as duas entradas e as duas saídas, não apenas a função do half-adder.
- Para mudar a função da lógica, tudo que nós precisamos fazer é escrever uma tabela diferente dos dados na memória.

+ Look-up table

- A essência de LUTs é que existe um dispositivo que guarda (memoriza) as saídas de uma dada função lógica (saídas definidas) para cada entrada específica. Ou seja, o dispositivo de memória "look-up" (verifica) qual saída deveria ser mostrada para uma dada combinação de entrada.
- Exemplo: Somador de bits
- 256 endereços x 4 bit

Diagram of a 256 x 4 ROM with inputs A₀ through A₇ and outputs D₀ through D₃. The ROM is used to implement a 4-bit result (4 bit result).

+ LUT - Look-Up-Tables

Diagram illustrating a LUT implementing combinational logic. The LUT has four inputs (A, B, C, D) and one output (Z). The LUT is connected to a 4-bit result (4 bit result). The diagram shows the LUT's connection to a 4-bit result (4 bit result).

DCBA	Z
0000	0
0001	0
0010	0
0011	0
0100	1
0101	0
0110	0
0111	0
1000	1
1001	0
1010	0
1011	1
1100	1
1101	1
1110	1
1111	1

+ Carry Chain logic

31

Mapeamento lógico

posicionamento

Roteamento

Protótipo/2014

+ Bloco lógico de I/O

32

- Os blocos de I/O incluem, entre outros:
 - Registadores de entrada
 - Registadores de saída
 - Multiplexadores
 - Sinais de relógio
 - Controle de atraso
 - Tri-state (ou não)
 - Pull-up or Pull-down
 - Impedância de terminação
 - ...
- Padrões de sinais de I/O clock
 - Single-ended
 - PCI, LVTTTL
 - Diferencial
 - LVDS, LVPECL

Protótipo/2014

+ Manipulação de relógio

33

- Métodos para manipulação de relógio (clock)
 - Phase-locked loop (PLL)
 - Delay-locked loop (DLL)
- Phase-Locked Loop (PLL)
 - PLLs geram a fase desejada ou frequência de saída através de um circuito oscilador Voltage-Controlled Oscillator (VCO)
- Delay-Locked Loop (DLL)
 - DLLs acessam sinais de um circuito calibrado com controle de atraso interno no FPGA, para produzir a fase do relógio desejado ou frequência

Protótipo/2014

+ PLL - Stratix IV

34

Protótipo/2014

+ FPGA - Estrutura interna

35

linhas de tamanho simples linha de duplo tamanho linhas longas

Protótipo/2014

Xilinx - XC4000

+ Matriz de chaveamento - roteamento

36

PSM - Programmable Switch Matrix

Ponto de interconexão

6 transistores de passagem

Transistor de passagem programável

0,1 SRAM

Características Funcionais:

- Permitem a passagem de dados em ambos os sentidos
- Um simples bit de controle é suficiente para controlar cada transistor de passagem

Protótipo/2014

+ FPGA – Matriz de chaveamento

37

Características de atraso

Figure 3-15. Xilinx interconnect. Figure courtesy of Xilinx, Inc. © Xilinx, Inc. 1991. All rights reserved.

PrototipaçãO/2014

+ SRAM – conexão

38

- SRAM conectando dois blocos lógicos através de “pass transistors” (transistores de passagem)
- SRAM controla também (exemplo) a saída de um multiplexador para implementar a lógica do circuito.

ERAD-SP / 2011

+ FPGA

39

- Mais flexibilidade** (Mais flexibilidade)
- Mais desempenho** (Mais desempenho)

- CPU de propósito geral
- Uso geral
- Bom desempenho (aplicações sequenciais)
- Alto desempenho para aplicações paralelas
- Flexibilidade
- Tempo de vida maior devido a reconfigurabilidade do dispositivo
- ASIC
- Aplicação específica
- Muito bom desempenho

PrototipaçãO/2014

+ Fatores de reconfigurabilidade

40

Datapath Arithmetic, Logic, Memory, Register, Multiplexer	Device Memory 64 KB X 64 PE	Function 8x8 MAC (Processing Element)	Precision 24-bit Multiply (Processing Element)
Interface RC Device, BRAM Memory Controller, BRAM	Mode PE1 Prog-A, PE2 Prog-A, PE3 Prog-A, PE4 Prog-A	Power PE, PE, PE, PE, PE, PE Performance ↓ Power ↓	Interconnect PE, PE, PE, PE, PE, PE, PE, PE

PrototipaçãO/2014

+ FPGA atual – Estrutura interna

41

PrototipaçãO/2014

+ Recursos disponíveis em FPGAs

42

- Processadores em hardware
 - PowerPC (300 – 500 MHz) (Xilinx)
 - Módulos DSP (Altera)
- Sistemas operacionais embarcados
 - Linux
- Transceivers gigabit
- Blocos serializadores / deserializadores para receber dados em altas taxas de transmissão
 - Virtex-4 é capaz de receber e transmitir dados em frequências de 3.2 Gbps

PrototipaçãO/2014

+ FPGA - arquitetura

Lattice ECP2M Block Diagram

Embedded 3.125 Gbps SERDES – support PCI express, Ethernet.

Programmable Function Unit (PFU) – perform logic, arithmetic, Distributed RAM & ROM functions.

Flexible Sys I/O Buffers – support LVCMOS, LVDS, etc.

Sys Clock – PLLs & DLLs for clock management.

On-Chip Oscillator

Advanced Configuration Logic supports dual boot, exception, and interrupt IC.

sysMEM Embedded Block RAM (EBR) provides 128K dual port RAM.

Sys DSP Blocks – implement multipliers, adders, subtractors, accumulators.

Configuration Port – supports SPI, serial and parallel configuration.

<http://www.latticesemi.com/products/fpga/ecp2/optimizedfpgaarchitecture.cfm>

Prototipação/2011 Kenneth R. Laker, University of Pennsylvania, updated 25Mar13

+ System on Programmable Circuits

- Milhões de gates em um único chip
- Operação acima dos 300 MHz
- Grande variedade de cores
- Lógica mista/IP/memória
- Verificação x Criação
- SoPC

Field Programmable System-on-Chip

Design Reuse

Programmable Logic

IP/CPU

Memory

Datapath

I/O

FPGA Vendor/3rd Party

FPGA Vendor

Prototipação/2014

+ FPGA - Altera

Aplicações de alto desempenho

Feature	Stratix V E FPGA	Stratix V GS FPGA	Stratix V GX FPGA	Stratix V GT FPGA
High-performance adaptive logic modules (ALMs)	397,000	265,000	358,500	234,750
Variable-precision DSP blocks (18x18)	704	4,096	798	512
M20K memory blocks	2,640	2,688	2,660	2,560
External memory interface	✓	✓	✓	✓
Partial reconfiguration	✓	✓	✓	✓
PLL	✓	✓	✓	✓
Design security	✓	✓	✓	✓
SEU mitigation	✓	✓	✓	✓
PCI Express Gen3, Gen2, Gen1 hard IP blocks	-	Up to 2	Up to 4	1
Embedded HardCopy Blocks and hard IP	-	✓	✓	✓
Transceivers (1)	-	14.1 Gbps / 48	14.1 Gbps / 66	28G / 4 12.5 Gbps / 32

Prototipação/2014

+ FPGA - Xilinx

Aplicações de alto desempenho

MAXIMUM CAPABILITY	ARTIX7 FPGAs	KINEX7 FPGAs	VIRTEX7 FPGAs
Logic Cells	848K	478K	1,950K
Block RAM	19Mb	34Mb	68Mb
DSP Slices	1,040	1,820	3,600
Peak DSP Performance (symmetric FIR)	1,129 GMACS	2,450 GMACS	5,112 GMACS
Transceiver Count	16	32	96
Peak Transceiver Speed	6.6Gbps	12.5Gbps	28.05Gbps
Peak Serial Bandwidth (full duplex)	211Gbps	800Gbps	2,784Gbps
PCI Express® Interface	Gen2 v4	Gen2 v6	Gen3 v6*
Memory Interface	1,066Mbps	1,866Mbps	1,866Mbps
I/O Pins	600	500	1,200
I/O Voltage	1.2V, 1.35V, 1.5V, 1.8V, 2.5V, 3.3V	1.2V, 1.35V, 1.5V, 1.8V, 2.5V, 3.3V	1.2V, 1.35V, 1.5V, 1.8V, 2.5V, 3.3V*
Packaging Options	Low-cost wire bond	Low-cost BGA flip-chip and high-performance flip-chip	Highest performance flip-chip

Prototipação/2014

+ Projeto

Altera Quartus+II

Entradas:
-Esquemática
- Verilog

Entrada

simulação

Prototipação/
Implementação

Prototipação/2014

+ FPGA - fluxo de projeto

SystemVerilog
Verilog
VHDL
Esquemático

Especificação

Test bench

Simulação funcional e verificação

Síntese lógica

Análise de potência

Depuração do hardware

Floorplanning

Layout

Placement & Routing

Análise temporal (timing)

Ajuste temporal

Simulação (gates/timing)

Programação

Prototipação/2014

+ HDLs x SDL

HDL's

- HardwareC
- Verilog
- AHDL
- VHDL
- Verilog
- SystemVerilog

SDL'

- C
- Pascal
- ADA
- C++
- SystemC
- OpenCL

Prototipaço/2014

+ Exemplo – contador VHDL

Entrada VHDL (independente da tecnologia)

```

entity refctr is
  port (COUNT: in std_ulogic_vector(5 downto 0);
        CLR: in std_ulogic;
        RESET: out std_ulogic);
  ...
architecture refctr_rtl of refctr is
  signal s_ref_ctr_out : std_ulogic;
  signal s_load : std_ulogic;
  s_next_ctr_val : std_ulogic_vector(5 downto 0);
  s_counter_input : std_ulogic_vector(5 downto 0);
  s_counter_output : std_ulogic_vector(5 downto 0);
  s_reset : std_ulogic_vector(5 downto 0);
begin
  s_reset(0) <= RESET;
  ...
  process (CLK)
  begin
    if (CLK = '1' and not CLR' stable) then
      s_counter_output <= s_counter_input and not s_reset;
      s_ref_ctr_out <= s_load;
    end if;
  end process;
  ...
end refctr_rtl;
    
```

Prototipaço/2014

+ Síntese lógica

■ Gera netlist

```

entity bombom01 is
  port( reset, clk, c, d : in bit;
  ...
architecture arc_bomboms of bombom01 is
  ...
begin
  process (clk)
  begin
    if (reset = '1') then
      estado_atual <= zero_zero;
      libera_bombom <= '0';
    elsif (clk = '1' and clk'event) then
    ...
    end if;
  end process;
  ...
end arc_bomboms;
            
```

Síntese Lógica

- netlist schematic view
- netlist in VHDL
- netlist in Verilog
- netlist in EDIF

Prototipaço/2014

+ Netlist – visão esquemática

Prototipaço/2014

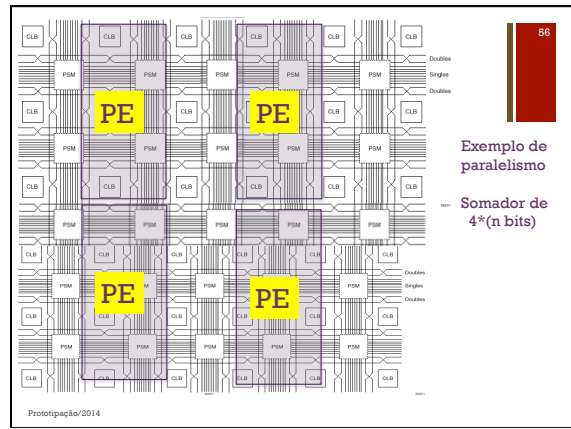
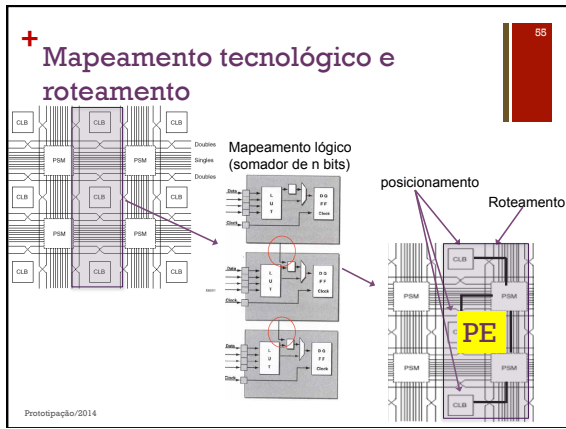
+ Netlist VHDL, Verilog

VHDL	Verilog
<pre> entity refctr is ... architecture rtl of refctr is ... component INVERT_A port(i : in std_ulogic; out : out std_ulogic); end component; component AND2_8 port(i : in std_ulogic; j : in std_ulogic; out : out std_ulogic); end component; component NOR3_4 port(i : in std_ulogic; j : in std_ulogic; k : in std_ulogic; out : out std_ulogic); end component; component D_F_LPH0001_4 port(i : in std_ulogic; j : in std_ulogic; out : out std_ulogic); end component; begin ... U68 : INVERT_A port map (i => s_load, o => sload); U87 : AND2_8 port map (i => n275, j => n276, o => n277); U88 : NOR3_4 port map (i => n275, j => n276, k => n277, o => n277); U89 : AND2_8 port map (i => n275, j => n276, o => n277); D_F_LPH0001_4 : D.F. LPH0001_4 port map (i => n275, j => n276, o => n277); ... end rtl; </pre>	<pre> module refctr (COUNT, CLR, RESET, sload); ... AND2_8 U87 (.A(sload), .B(COUNT[0]), .Y(sload)); NOR3_4 U88 (.A(COUNT[1]), .B(COUNT[2]), .C(COUNT[3]), .Y(sload)); AND2_8 U89 (.A(sload), .B(COUNT[4]), .Y(sload)); D_F_LPH0001_4 U89001 (.A(sload), .B(COUNT[5]), .Y(sload)); ... endmodule; </pre>

Prototipaço/2014

+ Simulação funcional

Prototipaço/2014



+ Aplicações

67

- Indústria automobilística
- Medicina
- Comunicação
- Indústria bélica

Prototipação/2014

+ Computação de alto desempenho

68

- CRAY
- Silicon Graphics
- Convey
- Maxeler
- DRC
- Alpha Data
- Gidel
- Novo-G
- Nallatech
- Dini Group
- Hunt engineering
- Pico Computing
- SRC Computers
- Alpha Data
- XtremeData

Prototipação/2014