

FPGA

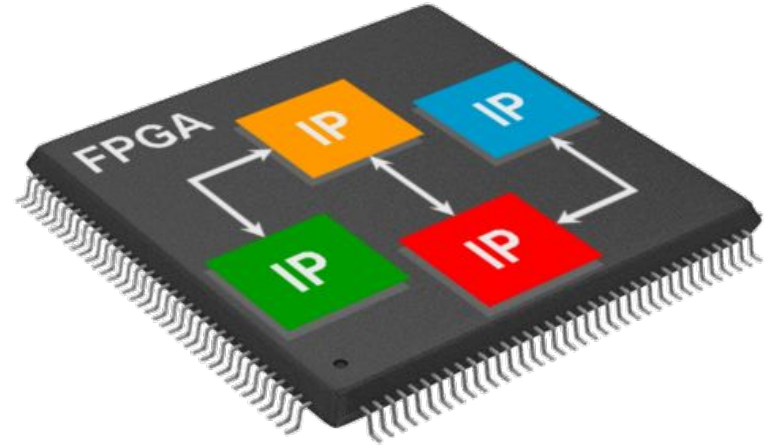
Introdução ao conceito

Bruno Rafael Aricó
Daniel Jorge Renjiffo (DJ)



Agenda

- Histórico
- O que é FPGA?
- Como funciona?
- Comparação com ASIC e CPLD
- O que são HDL's
- Aplicações
- Demonstrações



História

- Os primeiros chips FPGA surgiram em 1985 (XC2064);
- Ross Freeman e Bernard Vonderschmitt(co-Fundadores da Xilinx);
- Surgiu de uma fusão de PROM com PLD;

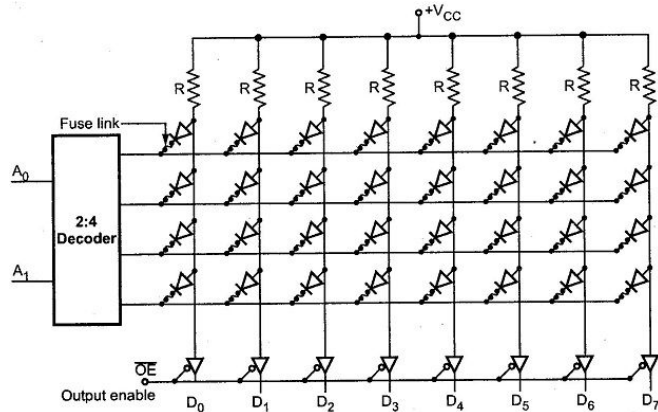
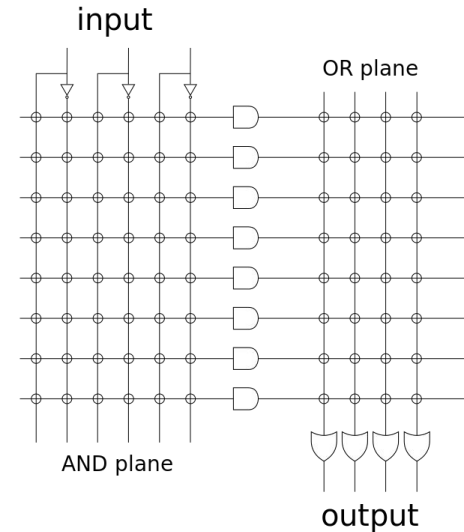


Fig. 3.71 Four byte PROM

PROM(Programmable Read Only Memory)

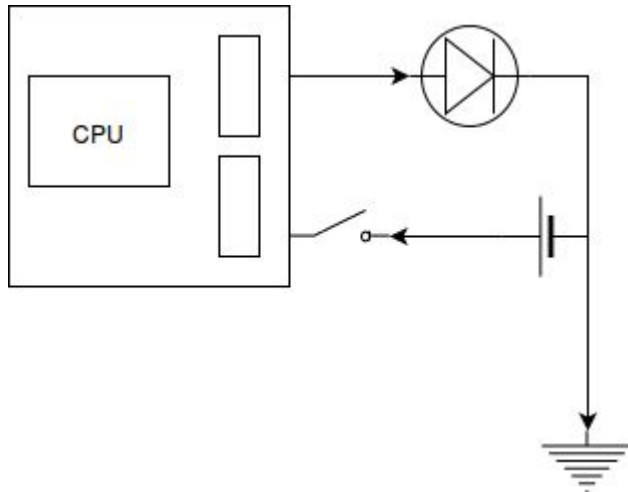


PLA (Programmable Logic Array)

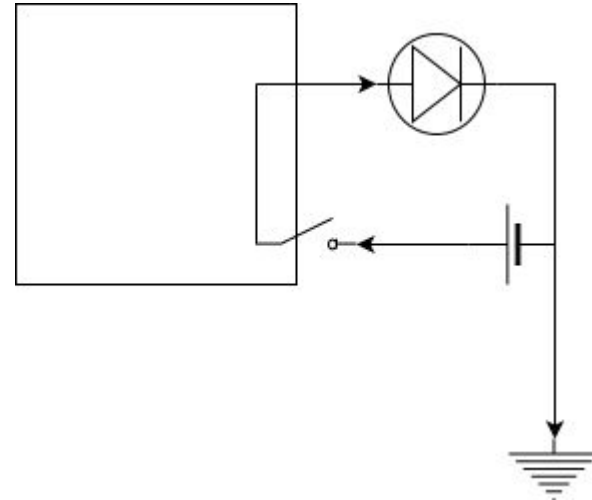
O que é FPGA?

Um exemplo simples.

Microcontrolador



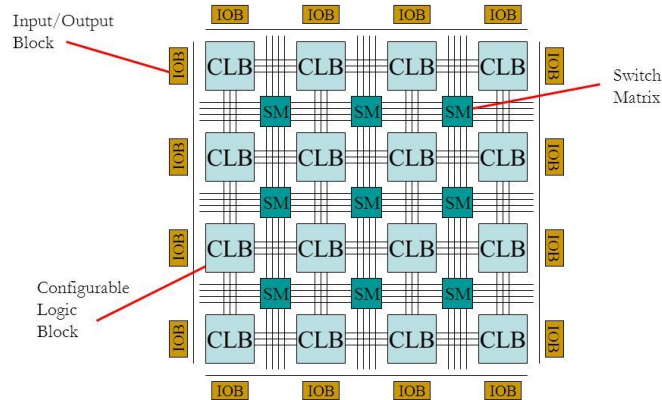
FPGA



O que é FPGA?

- Field Programmable Gate Array;
- Modo de fazer computação onde a definição de rotina é descrita a nível de hardware e não de software;
- Resumidamente uma matriz de blocos lógicos interligados de modo inteligente;

FPGA Structure



Typical Capacity : 5 million to 1 billion transistors

O que é FPGA?

- Um hardware não pré configurado (field Programmable);
- Podemos criar diversos circuitos lógicos dentro de um mesmo chip (lê-se computação paralela massiva em nível de hardware);



Fonte: Wikipedia



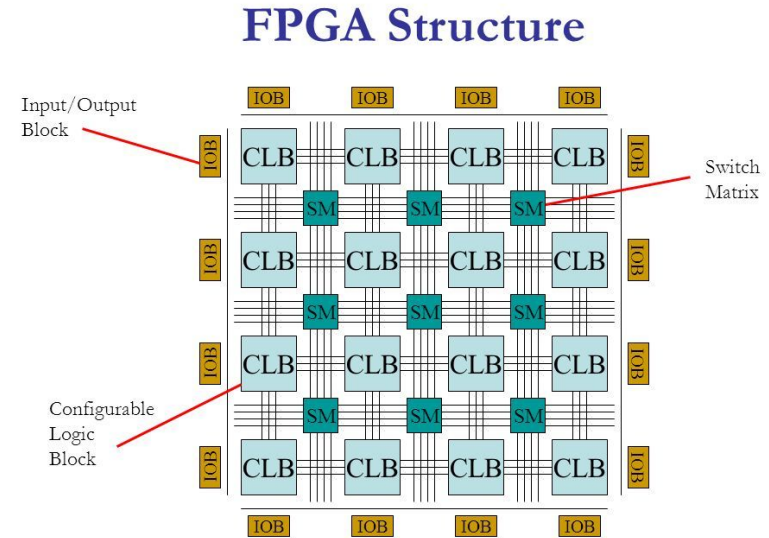
Fonte: Wikipedia



Fonte: Altera

Como funciona?

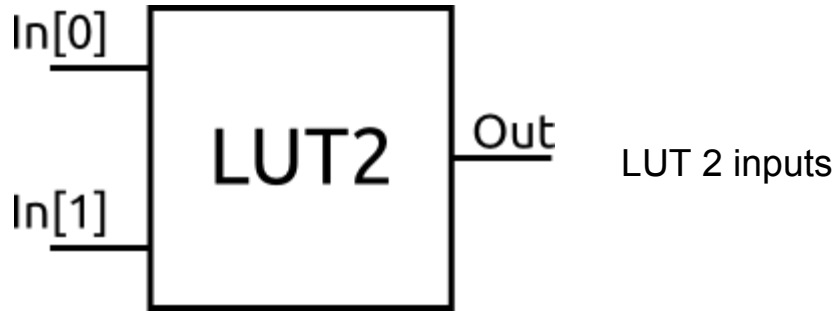
- O chip é dividido em 3 componentes principais:
 - Blocos lógicos configuráveis e de funções fixas;
 - Blocos de E/S;
 - Blocos de Comutação
- Os blocos lógicos se dividem em:
 - Grãos grandes : microprocessadores, ALU's;
 - Grãos médios: 2 ou mais LUTs' 2 ou mais flip-flops;
 - Grãos pequenos: funções lógicas de duas entradas, multiplexadores, flip flops;



Typical Capacity : 5 million to 1 billion transistors

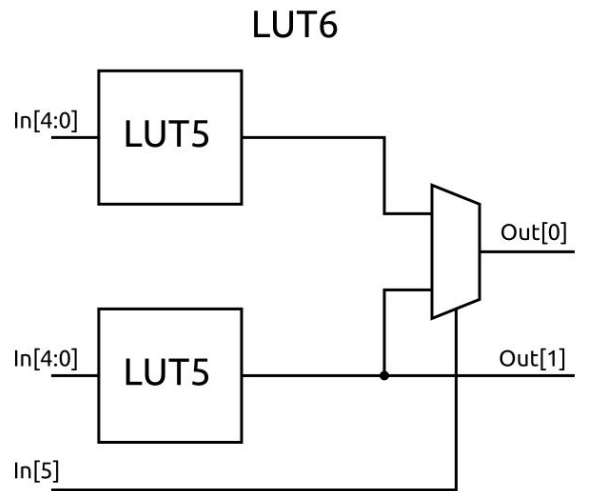
Como funciona? A LUT no Hardware

- É uma memória RAM geralmente;



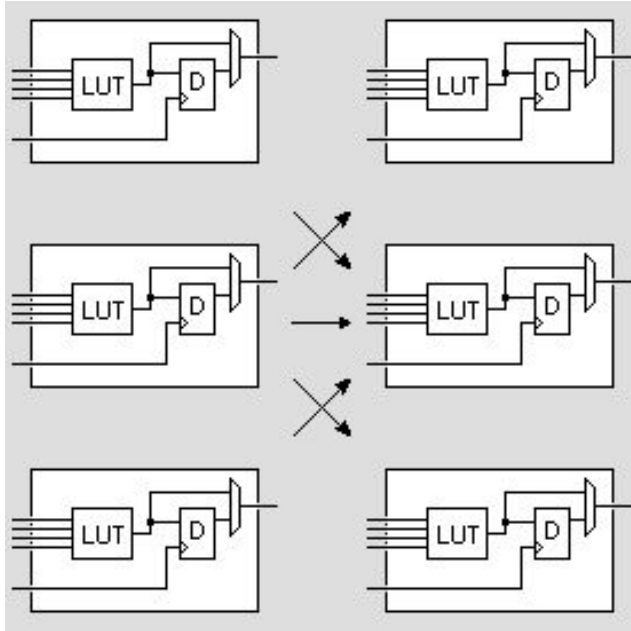
Address (In[1:0])	Value (Out)
00	0
01	0
10	0
11	1

LUT 6 ou 5 inputs

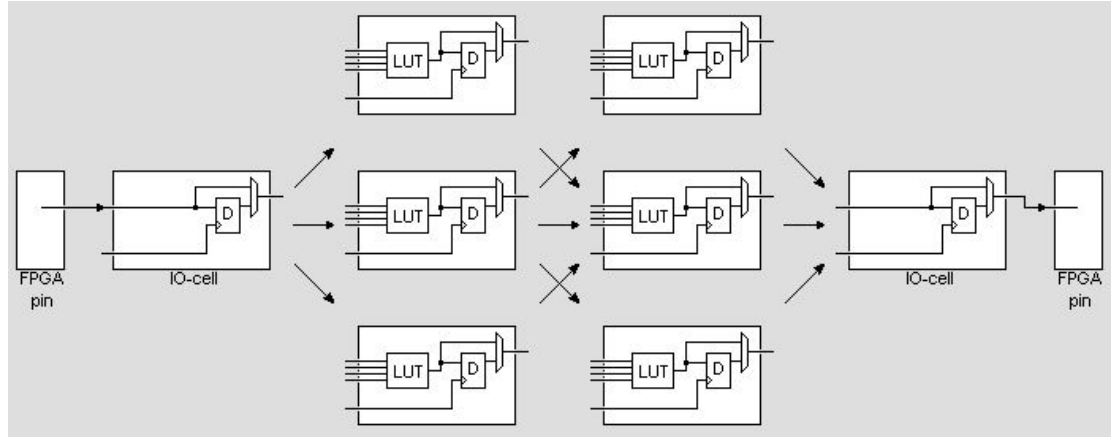


Como funciona? A LUT no Hardware

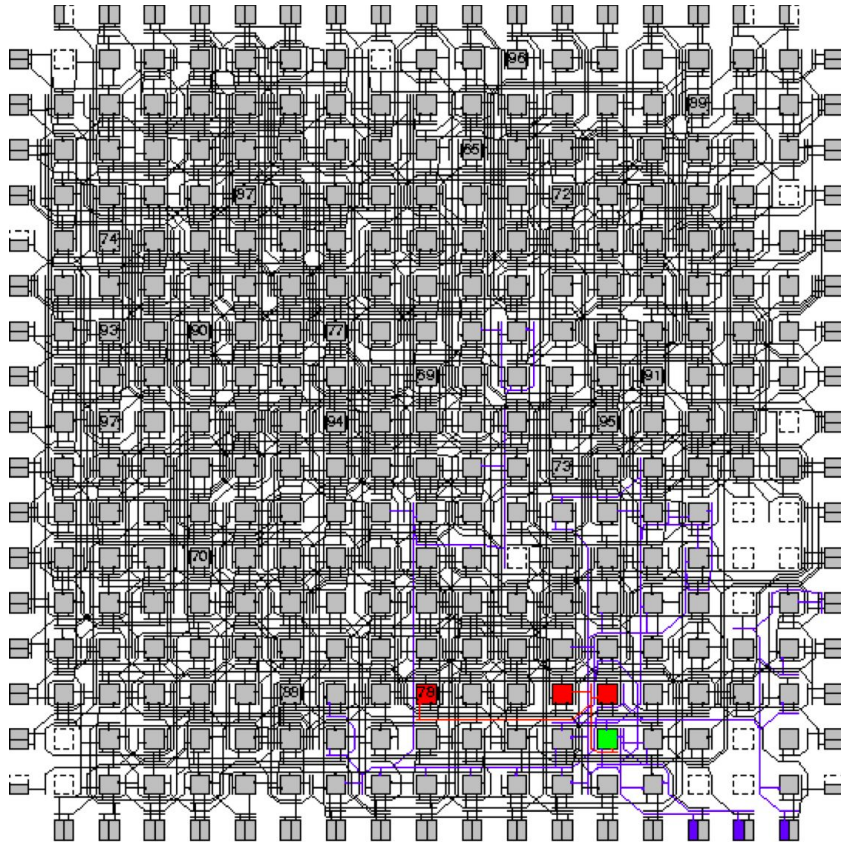
Nucleo do processamento:



Visão desde o Input até o output:



Roteamento em um projeto de médio porte



- Existe muita pesquisa sobre algoritmos de Posicionamento e Roteamento dos blocos em circuitos FPGA, de forma a otimizar o uso da pastilha e a eficiência do circuito;

ASIC

- ASIC (Application specific Integrated Circuit): Circuito dedicado, custo de projeto elevado;
- SoC (System-on-Chip), ASSP(Application-Specific-Standart-Part) são ASIC's;



Interface USB é
um ASSP

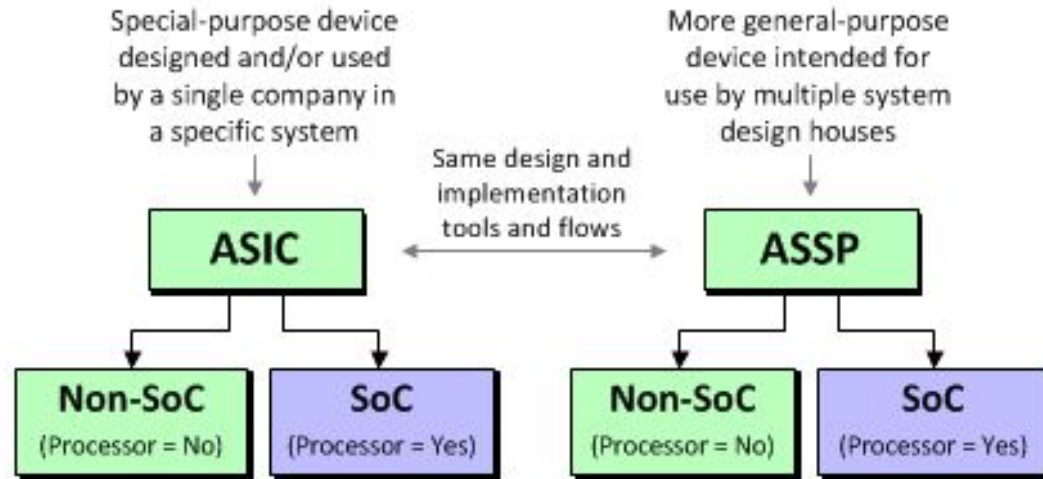
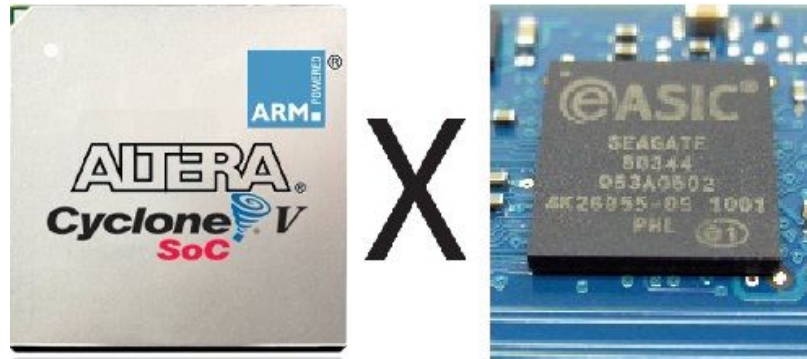


Figura: ASIC e ASSP - diferenças entre SoC e Non-Soc

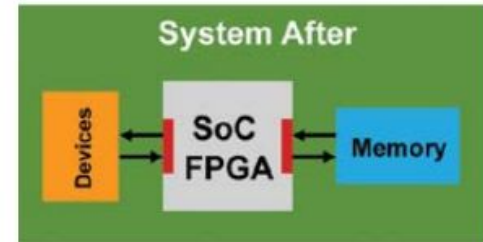
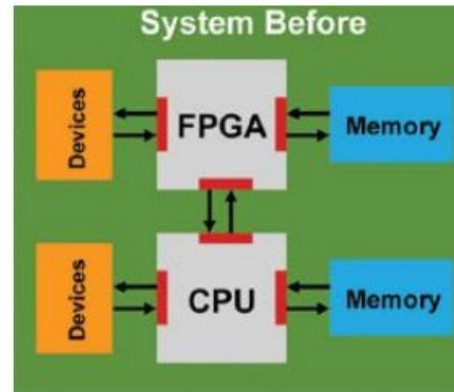
FPGA Vs ASIC

- Um projeto com FPGA leva semanas para ser produzido, com ASIC meses;
- Ferramentas para desenvolver FPGA são baratas, ASIC caros;
- FPGA é menos eficiente que os ASIC;
- O projeto com FPGA é muito mais barato que ASIC, dependendo do grau de escalabilidade desejado;
- FPGA já está pronto para utilizar no projeto, ASIC não;



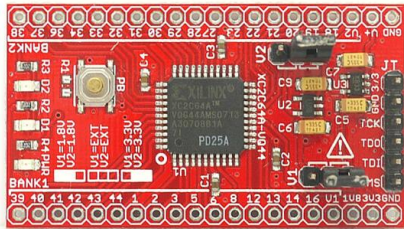
Fusão

- Antigamente:
 - Portas lógicas, RAM's
- Hoje em dia:
 - DSP (Digital Signal Processor), multiplicadores, somador, etc...
- São os SoC FPGA's;



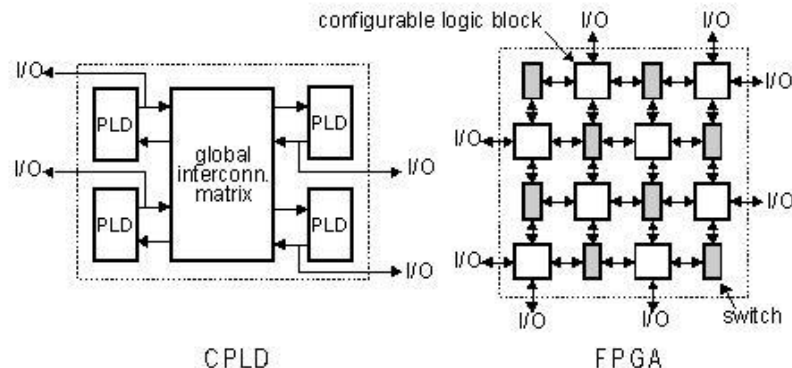
CPLD

- CPLD(Complex Programmable Logic Device), dispositivo lógico complexo programavel
- Circuitos com centenas ou até milhares de blocos lógicos programáveis



FPGA Vs CPLD

- CPLD's são tão rápidos quanto FPGA's ;
- CPLD's possuem CLB's na ordem de 1000 enquanto FPGA 1.000.000;
- CPLD's não precisam de EEPROM externa para armazenar ligações;
- CPLD's custo bem menor que FPGA (chips por 1 dólar);

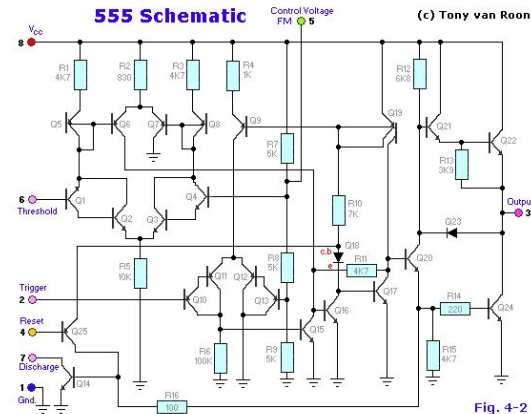
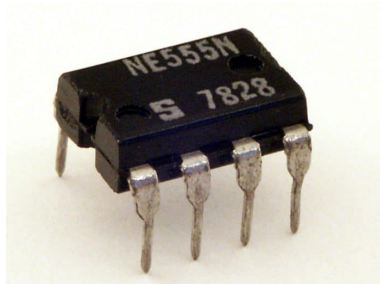


CPLD and FPGA Architectures



Ok, mas como “programar” um Hardware?

- Utiliza-se HDL's (Hardware Description Language);
- Criada em meados de 1980 pela DARPA para documentar ASIC's;
- De acordo com a Lei de Moore, descrição alto nível do Hardware,
- Independente da tecnologia(preservando DataFlow e o Timing);
- Antes das HDL's a melhor representação era o esquemático;



HDL - Hardware Description Languages

- DARPA criou o VHDL (VHSIC Hardware Description Language)[Very High Speed Integrated Circuit];
- Sintaxe baseada na linguagem ADA;
- É utilizada na documentação, descrição, síntese, simulação, teste e verificação formal do Hardware;
- 1987 a IEEE a tornou padrão;
- Linguagens de descrição de hardware: Verilog, SystemC, Lola, Lava, MyHDL,...

HDL São linguagens de Descrição de Hardware e não de Programação de Hardware!

VHDL: Exemplos

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_NUMERIC_STD.ALL;

entity not1 is
    port(a:in STD_LOGIC;
          b:out STD_logic);
end not1;

architecture behavioral of not1 is
begin
    b <= not a;
end behavioral;
```

Uma porta NOT

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity mux4x1 is
4  Port ( din : in  STD_LOGIC_VECTOR (3 downto 0);
5        en : in  STD_LOGIC;
6        sel: in  std_logic_vector(1 downto 0);
7        dout : out STD_LOGIC);
8  end mux4x1;
9  architecture mux4x1_arch of mux4x1 is
10 signal iy:std_logic;
11 begin
12     with sel select
13         iy<=din(0) when "00",
14             din(1) when "01",
15             din(2) when "10",
16             din(3) when "11",
17             '0' when others;
18     dout<=iy when en='1' else '0';
19 end mux4x1_arch;
```

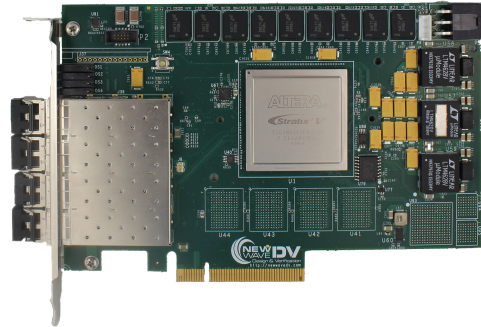
MUX 4x1

WorkFlow do projeto com FPGA

- Especificação (Esquemático/HDL);
- Síntese Lógica
 - Otimização lógica (Mapas de Karnaugh) para redução de portas e delays;
 - Mapeamento nas células;
- Posicionamento e Roteamento: Quais blocos serão usados e quais trilhas, de forma a otimizar o uso do silício;
- Verificação e teste;
- Programação do FPGA: Carregar tabelas nas respectivas LUT's e configurar os blocos de Comutação;

Aplicações: Telecomunicações

- Processamento de sinais em tempo real: Modulação de sinais feitas em hardware onde diversos algoritmos podem compartilhar a mesma pastilha de silício e ser reconfigurado on-the-fly;
- Switches de grande porte: FPGA em switches desafia um pouco a carga de servidores, processando SSL (Security Socket Layer), decodificando vídeo, entre outras tarefas de peso, processamento de regras de Firewall;



Aplicações: Telecomunicações

Implementação do paradigma FTT (Flexible Time-Triggered):
Reconfigurável em tempo real, dependendo da política de escalonamento de tráfego, com alto desempenho.

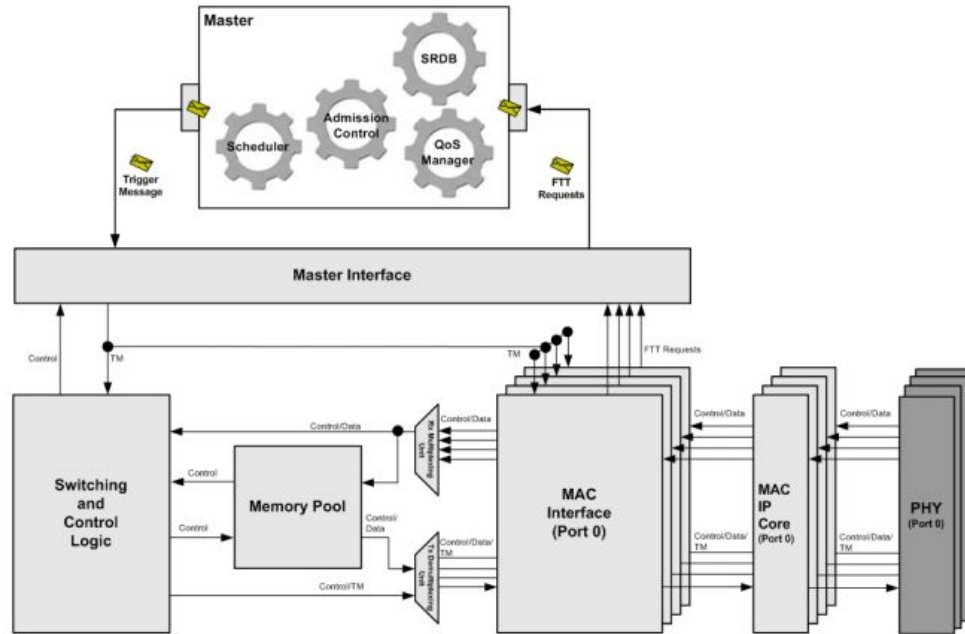


Figure 4. FTT-enabled switch - hardware implementation (top level).

Aplicações: Criptografia

- Criptografia usando um (GPP) General Purpose Processor:
 - Muitos ciclos de máquina (Não possui em hardware funções para descrição de frequência)
- Com FPGA o projetista pode criar as funções que deseja e executar a rotina de (De)cifragem em menos ciclos de clock.
- DES(Data Encryption Standard) algoritmo criado na IBM por volta de 1970
chaves com (56 bits +8 parity bits)
- Em 2008 RIVYERA máquina criada com FPGA's (\$ 10.000) quebra criptografia DES em menos de um dia

Aplicações: Criptografia

Tabela 1. Temporização do algoritmo DES implementado em C.

Tamanho do arquivo criptografado	Tempo (segundos) gasto com arquivo armazenado em disco	Tempo (segundos) gasto com arquivo armazenado na memória	% de melhora do desempenho em disco e em memória
1 MB	1,25s	0,94s	25,39 %
5 MB	5,82 s	3,90 s	32,96 %
10 MB	10,27 s	7,08 s	30,96 %

Tabela 2. Desempenho do algoritmo DES implementado em VHDL

Tempo de Propagação 19,55ns	Frequência Máxima 51,146 MHz		
Taxa de ocupação do FPGA			
	CLBs	FLIP-FLOPs	LUTs (Look-Up Table)
Nº de componentes utilizados	381	68	755
Nº de componentes disponíveis	9.48	18.816	18.816
Porcentagem de Ocupação	4,04%	0,30%	4,01%
Consumo de tempo para cifragem			
Tamanho do texto claro (MB)	Tempo (segundos)		
25	1		

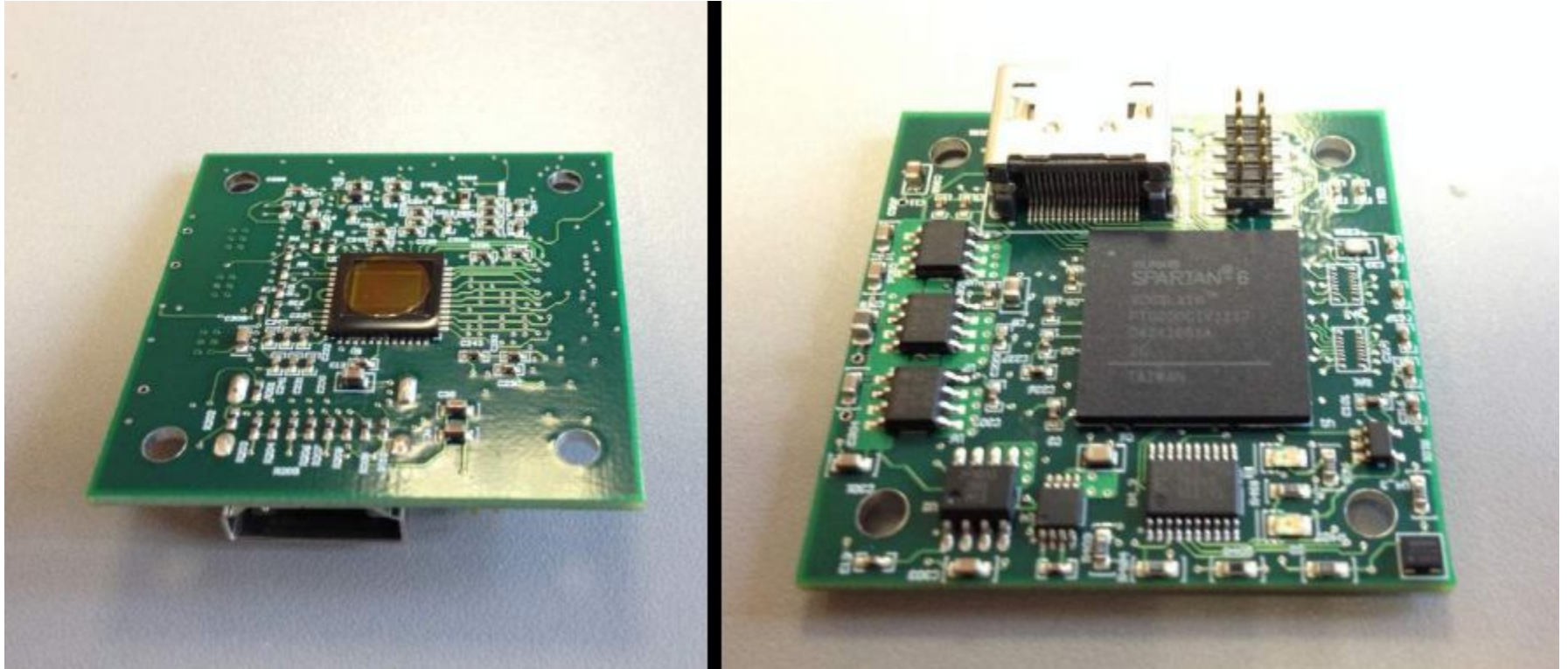
Aplicações: Visão Computacional

- Paralelismo é fundamental em visão computação com processamento em tempo real;
- Conceito de smart-cameras, processamento da imagem diretamente na câmera;
- CUDA x FPGA;

	FPGA	CUDA
Advantages	• Low power • High performance • Small size, possible to migrate to VLSI	• Easy to program • Speed up • Floating point
Inconvenient	• Complex to implement • Long to learn • Architecture complexity vs speedup	• Reformulate in parallel: core + memory use • Power consumption

Fonte: Performance Computing Lab INAOE – Puebla, Mexico

Aplicações: Visão computacional



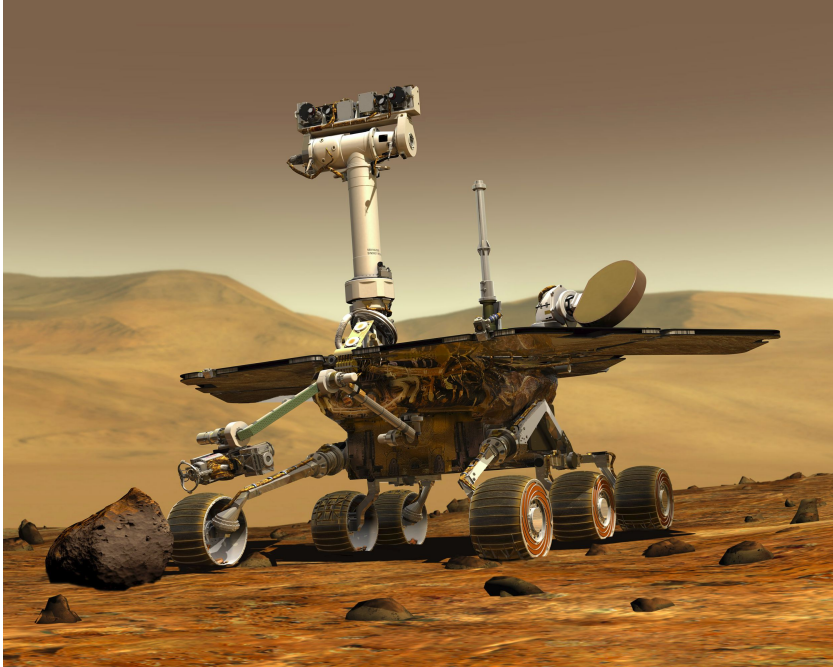
Fonte: Performance Computing Lab INAOE – Puebla, Mexico

Aplicações: Aeroespaciais

- Condições Adversas:
 - Radiação, pressão, temperatura, etc)
 - Exige um hardware que seja adaptável;
- Em tecnologias ASIC:
 - Dano no chip, sofre perda total com danos de ionização, devido alto custo de projeto espacial
- Reprogramação de novos usos do equipamento a longa distância:
 - Satélites: hoje em dia duram mais de 10 anos



Aplicações: Aeroespaciais



- Xilinx e Atmel atualmente produzem FPGA's em uma linha específica para aplicações espaciais, sendo considerados o Estados da Arte da tecnologia.

Utiliza um Xilinx XQVR4000XL tolerante a radiação

Ferramentas

- Fabricantes de FPGA's e ferramentas: Altera, Xilinx, Lattice;

ALTERA



Model*Sim*

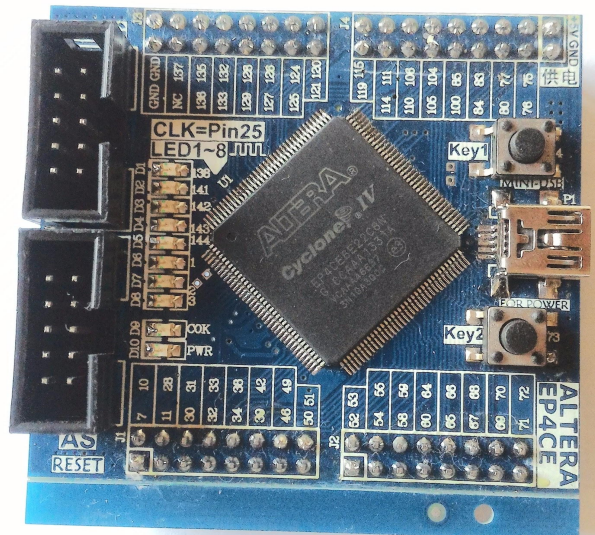


XILINX®

LATTICE
SEMICONDUCTOR

Demonstração de dois exemplos simples

- Blink (“Hello World” para embarcados)
- Contador binário;



```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY Blink IS
  PORT(
    clk_in : IN STD_LOGIC;
    led_out : OUT STD_LOGIC_VECTOR(7 downto 0)
  );
END ENTITY;

ARCHITECTURE behavior OF Blink IS
  SIGNAL led_off : STD_LOGIC_VECTOR (7 downto 0);
BEGIN
  PROCESS(clk_in)
    VARIABLE counter : INTEGER RANGE 0 to 25000000;
  BEGIN
    IF(RISING_EDGE(clk_in)) THEN
      IF(counter = 25000000) THEN
        counter := 0;
        led_off <= NOT led_off;
      ELSE
        counter := counter + 1;
      END IF;
    END IF;
  END PROCESS;

  led_out <= led_off;
END ARCHITECTURE;
```

Interessou ?

Encontros toda **quarta 14:00 até 18:00** (Auditório CCSL)

- Faça parte da lista de emails:
hardwarelivreusp@googlegroups.com
- Entre na comunidade no Facebook:
<https://www.facebook.com/Hardwarelivreusp/>
- hardwarelivreusp.org

HardwareLivreUSP

Fontes e referências:

<https://embeddedmicro.com/tutorials/mojo-fpga-beginners-guide/how-does-an-fpga-work>

<http://www.ni.com/white-paper/6983/pt/>

<http://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/logica-programavel.pdf>

<https://www.embarcados.com.br/asic-assp-soc-fpga/>

https://www.altera.com/en_US/pdfs/literature/ab/ab1_soc_fpga.pdf

https://www.microsemi.com/document-portal/doc_view/130056-telecom-digsig-an

Fontes e referências:

<http://www.decom.ufop.br/imobilis/fpga-o-prodigio-de-flexibilidade/>

http://www.enacomp.com.br/2010/cd/artigos/resumidos/enacomp2010_24.pdf

<https://pdfs.semanticscholar.org/e9d6/4526cc400e6a339965dfa979bafe0118ce77.pdf>

<http://hartes.av.it.pt/files/papers/en-2009-rec-switch.pdf>

http://eunevis.org/wasc2014/images/slides/wasc2014_estrada.pdf

<https://wiki.sj.ifsc.edu.br/wiki/images/c/c6/DLP29006-AE1-Tema4-2016-1.pdf>

<http://www.macnicadhw.com.br/newsletter/fpga-versus-asic>