CMP 238 – Projeto e Teste de um Sistema VLSI

- Definição Sistemas Digitais e espaço de projeto
- Metodologia de Projeto

UFRGS

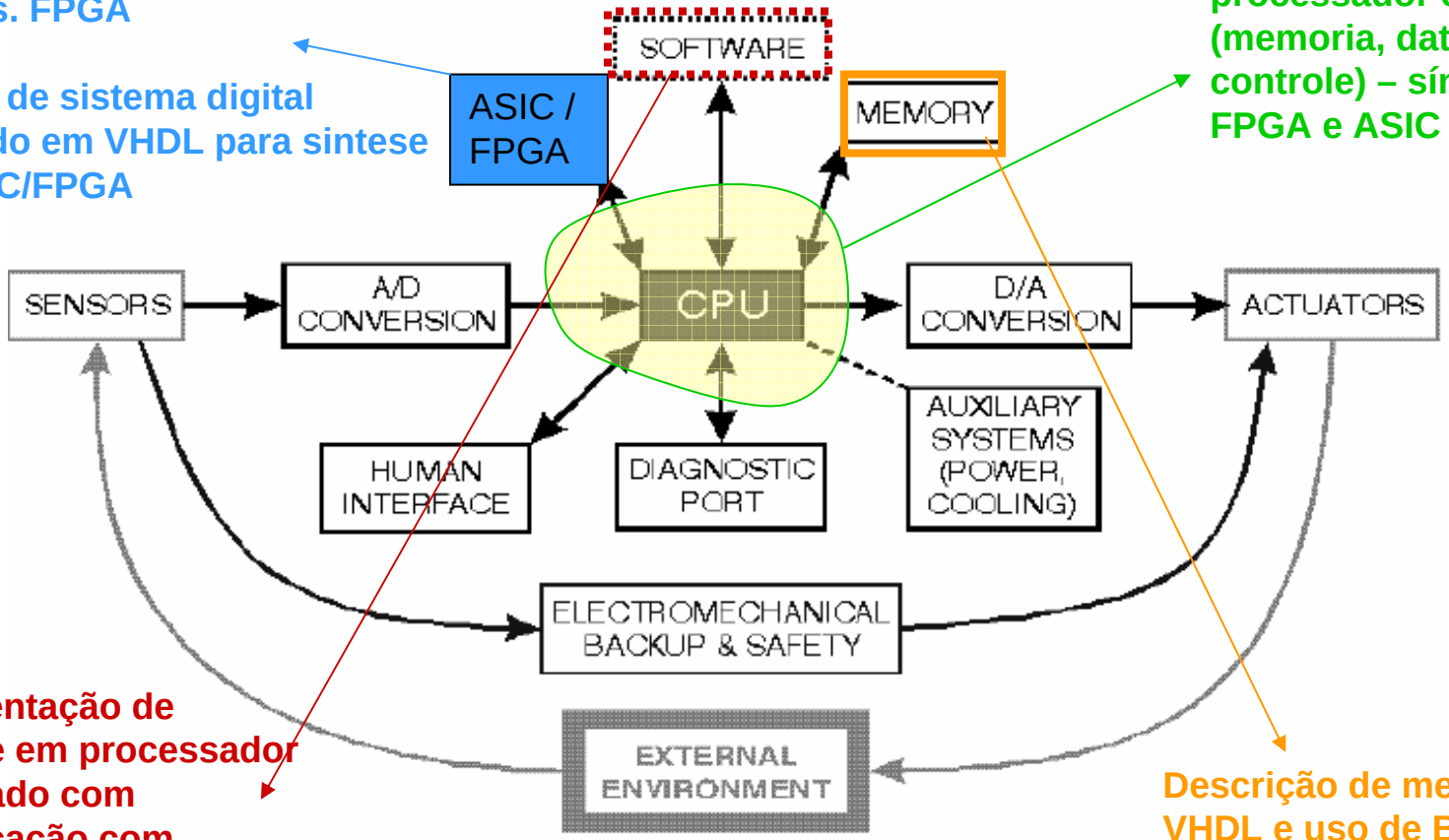
Sistemas Embarcados -> Sistemas Digitais

Definição

Estudo e comparação de projeto ASIC vs. FPGA

Projeto de sistema digital dedicado em VHDL para sintese em ASIC/FPGA

Como descrever um processador em VHDL (memoria, datapath, controle) – síntese FPGA e ASIC



Implementação de software em processador embarcado com comunicação com hardware dedicado

Descrição de memoria em VHDL e uso de BRAMs em FPGA

UFRGS

Primeira fase: Modelar o Sistema

- A modelagem é uma maneira de representar as informações importantes de um sistema.
- Um sistema pode ter diversos modelos para ele.

A modelagem é util para a verificação funcional, estimativas de custo na implementação e projeto do teste.

Motivações para a modelagem de sistemas:

1- Abstrair o comportamento de um sistema digital dando flexibilidade de implementação.

2 – Ajudar a comunicação do usuário com a funcionalidade do sistema

3 – Permitir o teste e verificação por meio de simulação

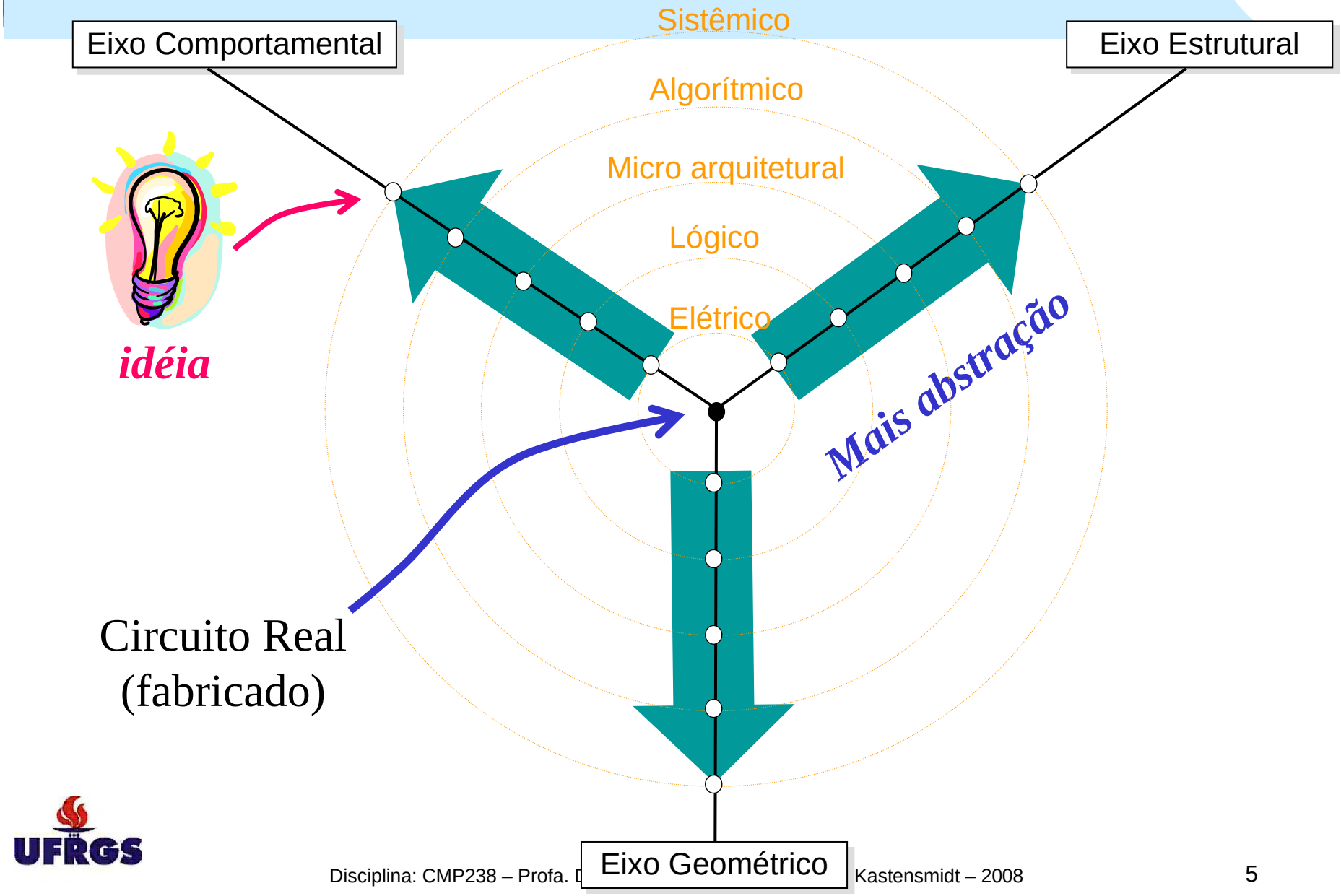
4 – Verificação formal por equações matemáticas que provem que o sistema funciona para determinadas regras de funcionamento.

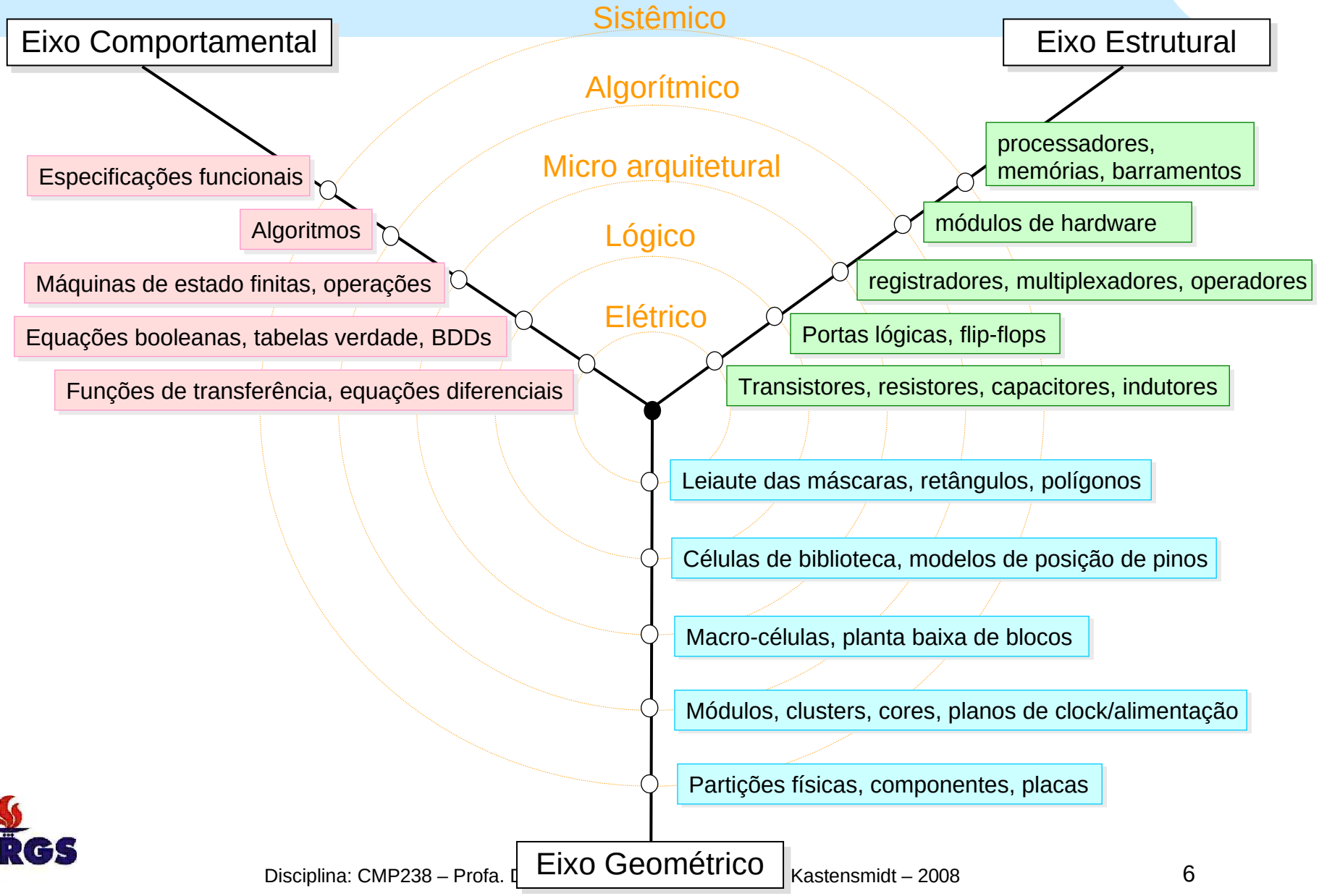
Níveis de Abstração de Sistemas VLSI

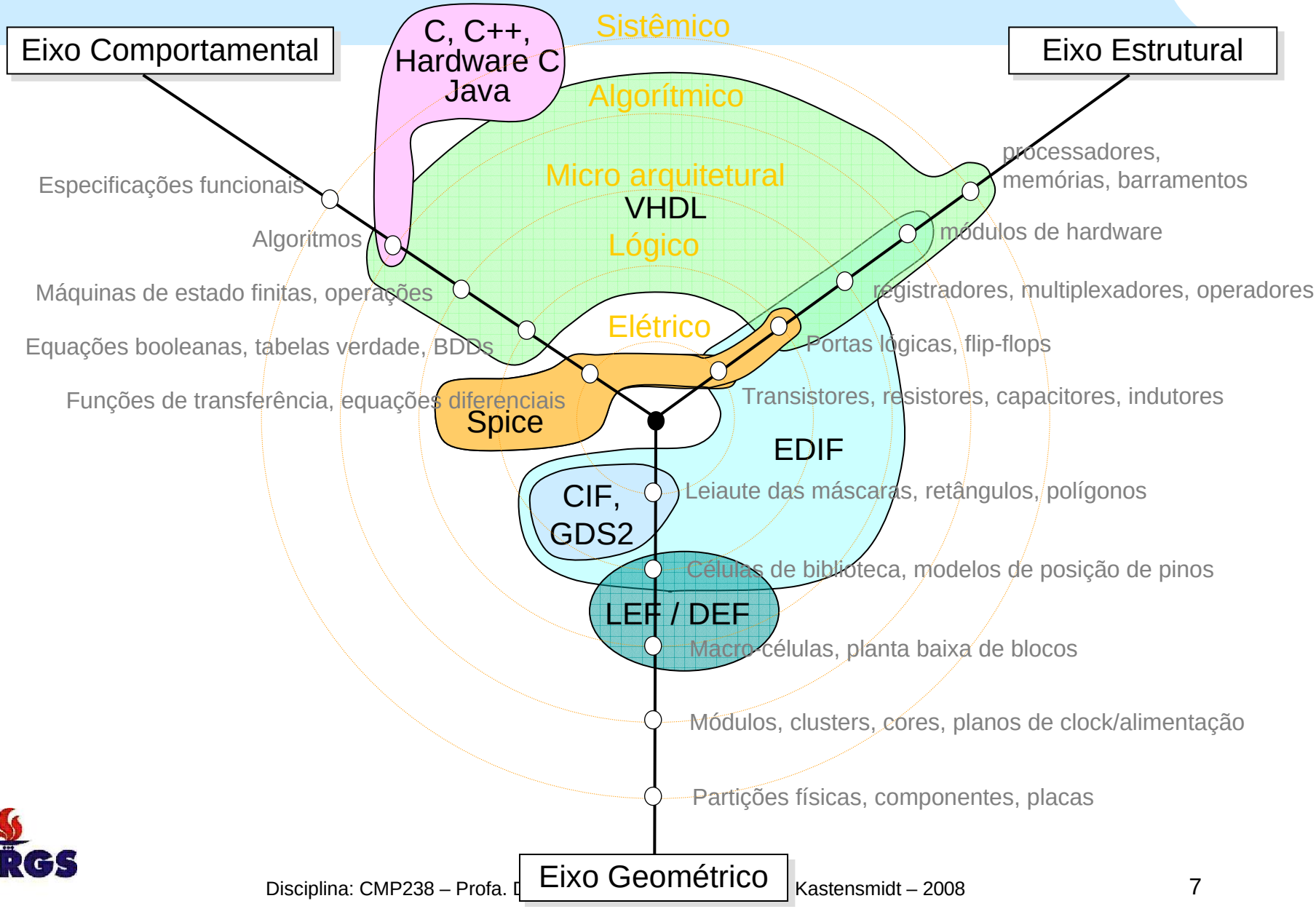
Projeto

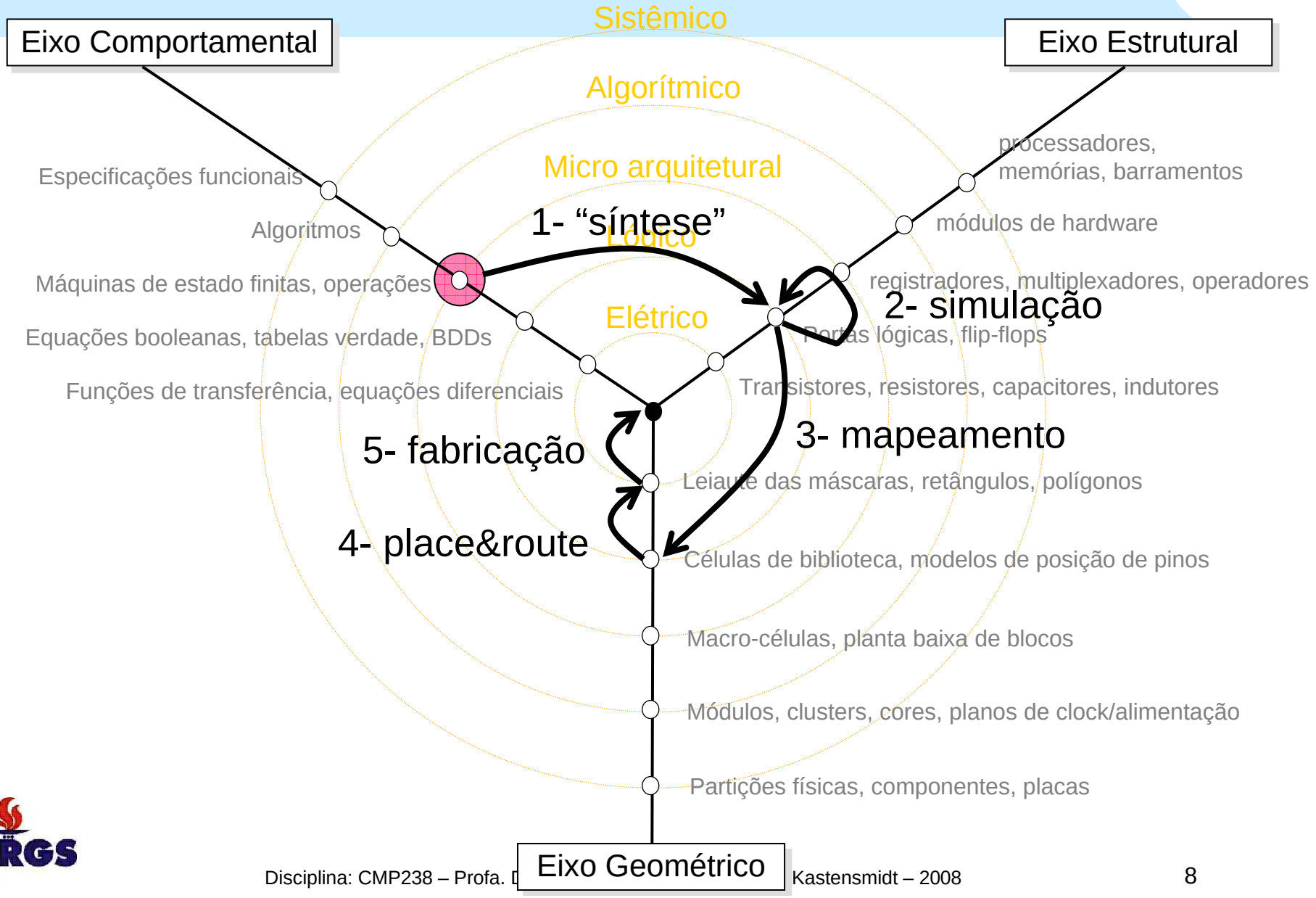
- Nível Sistêmico
- Nível Arquitetural ou Algoritmico
- Nível de transferencia entre registradores (micro-arquitetural)
- Nível lógico ou de portas
- Nivel elétrico ou de transistores
- Nivel de leiatue
- Nivel de máscaras

Verificação

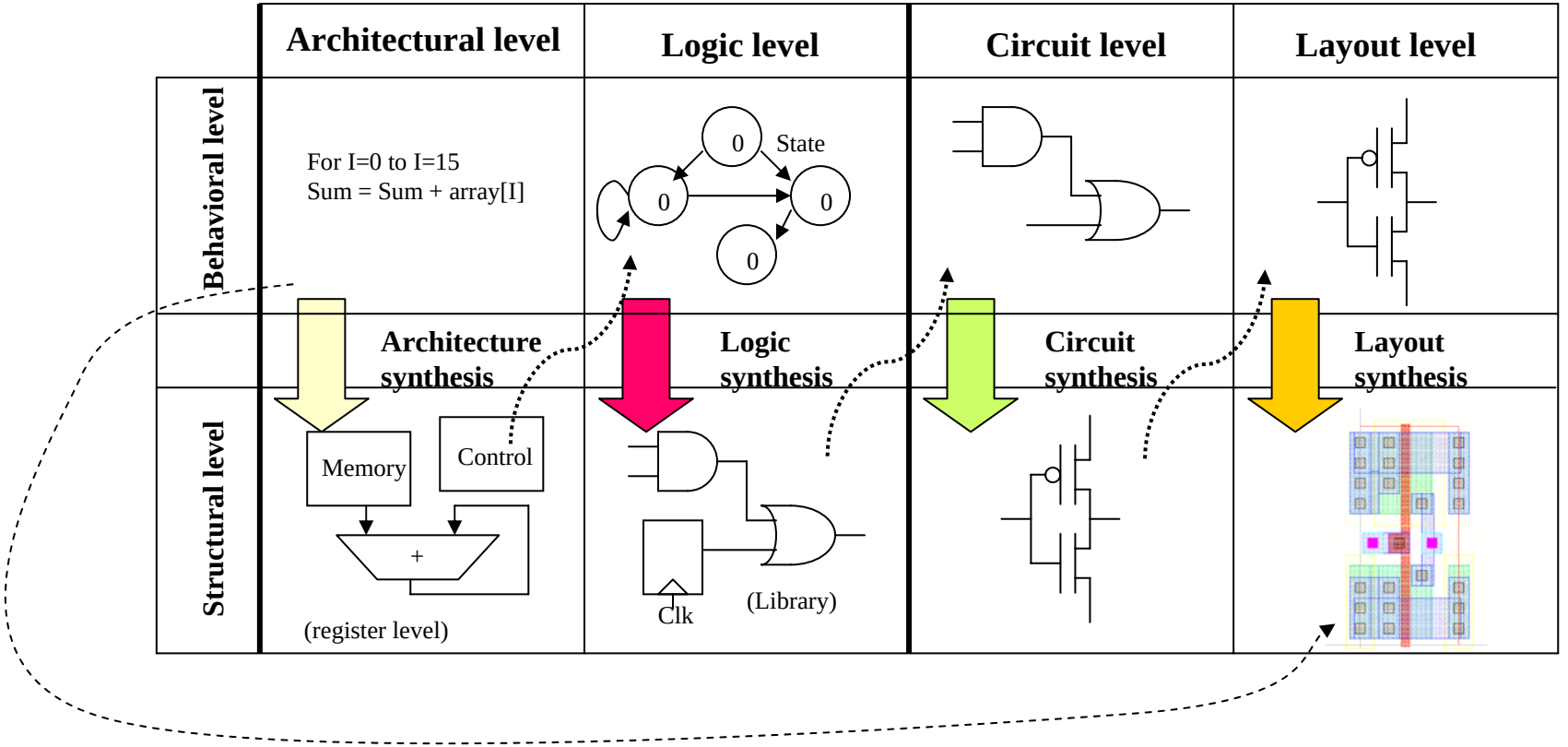

Sistêmico
Eixo Comportamental
Eixo Estrutural
Algorítmico
Micro arquitetural
Lógico
Elétrico
idéia
Mais abstração
Circuito Real
(fabricado)
Eixo Geométrico


Sistêmico
Algorítmico
Micro arquitetural
Lógico
Elétrico
Eixo Comportamental
Eixo Estrutural
Eixo Geométrico
Especificações funcionais
Algoritmos
Máquinas de estado finitas, operações
Equações booleanas, tabelas verdade, BDDs
Funções de transferência, equações diferenciais
processadores, memórias, barramentos
módulos de hardware
registradores, multiplexadores, operadores
Portas lógicas, flip-flops
Transistores, resistores, capacitores, indutores
Leiaute das máscaras, retângulos, polígonos
Células de biblioteca, modelos de posição de pinos
Macro-células, planta baixa de blocos
Módulos, clusters, cores, planos de clock/alimentação
Partições físicas, componentes, placas


Eixo Comportamental
Eixo Estrutural
Eixo Geométrico
Sistêmico
C, C++,
Hardware C
Java
Algorítmico
Micro arquitetural
VHDL
Lógico
Elétrico
Spice
EDIF
CIF,
GDS2
LEF / DEF
Especificações funcionais
Algoritmos
Máquinas de estado finitas, operações
Equações booleanas, tabelas verdade, BDDs
Funções de transferência, equações diferenciais
processadores,
memórias, barramentos
módulos de hardware
registradores, multiplexadores, operadores
Portas lógicas, flip-flops
Transistores, resistores, capacitores, indutores
Leiaute das máscaras, retângulos, polígonos
Células de biblioteca, modelos de posição de pinos
Macro-células, planta baixa de blocos
Módulos, clusters, cores, planos de clock/alimentação
Partições físicas, componentes, placas

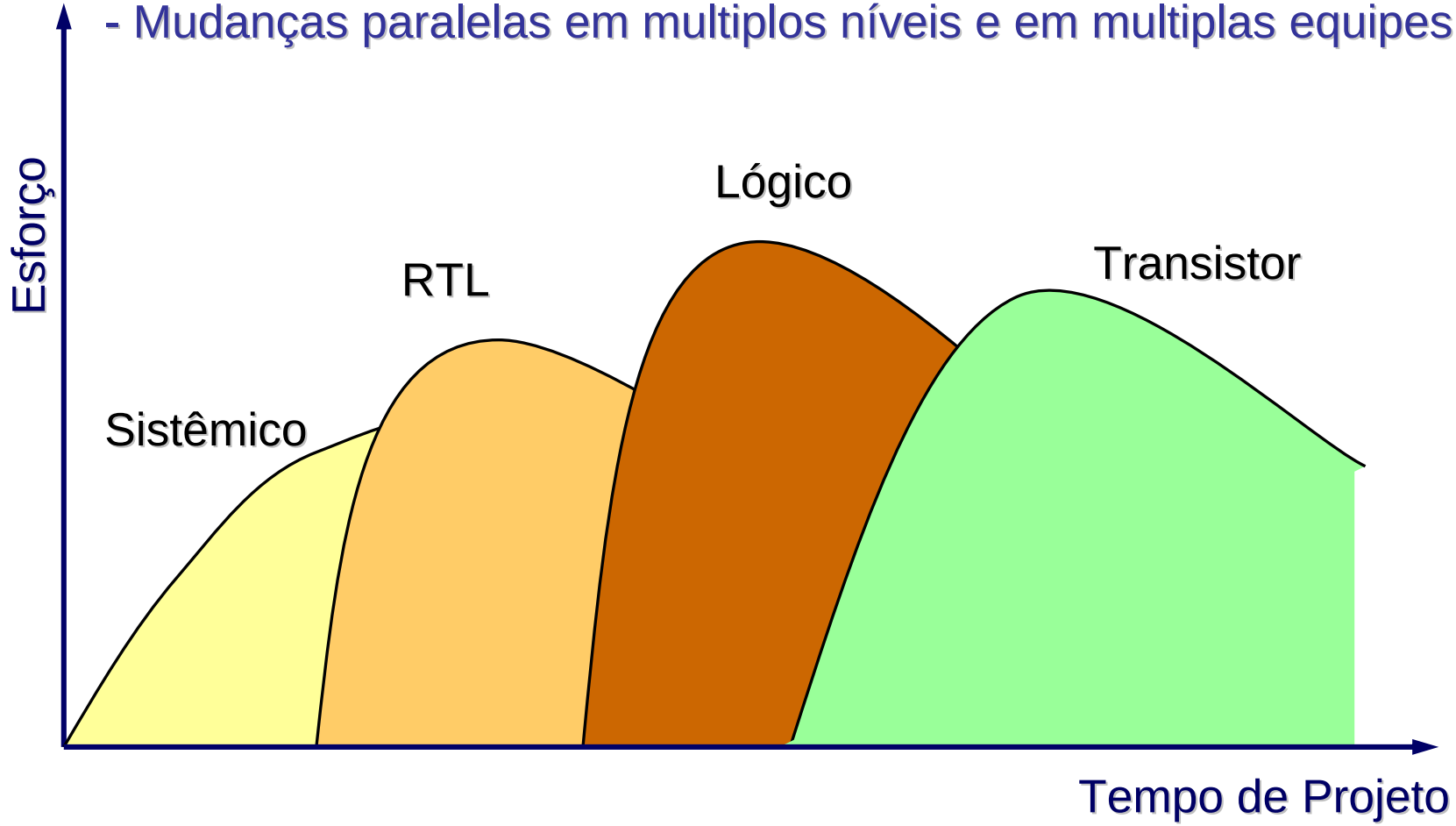

Sistêmico
Eixo Comportamental
Eixo Estrutural
Algorítmico
Micro arquitetural
Especificações funcionais
Algoritmos
Lógico
Máquinas de estado finitas, operações
Elétrico
Equações booleanas, tabelas verdade, BDDs
Funções de transferência, equações diferenciais
1- "síntese"
2- simulação
3- mapeamento
4- place&route
5- fabricação
processadores, memórias, barramentos
módulos de hardware
registradores, multiplexadores, operadores
Portas lógicas, flip-flops
Transistores, resistores, capacitores, indutores
Leiaute das máscaras, retângulos, polígonos
Células de biblioteca, modelos de posição de pinos
Macro-células, planta baixa de blocos
Módulos, clusters, cores, planos de clock/alimentação
Partições físicas, componentes, placas
Eixo Geométrico

Níveis de Abstração e Síntese



Compilação para silicio (não é um grande sucesso)

- Sobreposição de fases no projeto para atingir time-to-market
- Mudanças paralelas em multiplos níveis e em multiplas equipes

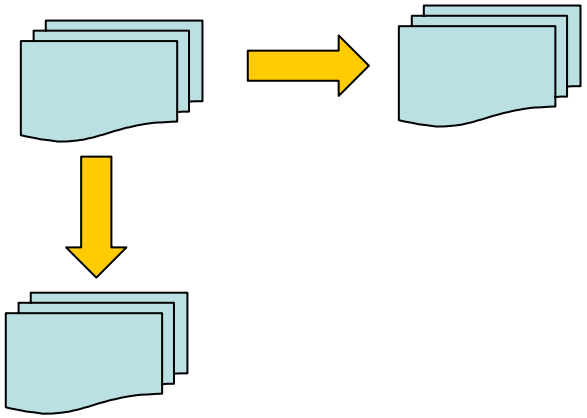


Projeto de Sistemas Digitais

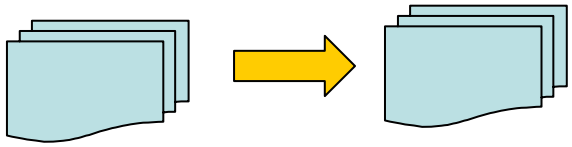
- Devido a alta complexidade dos sistemas digitais atuais, faz-se necessário a adoção de uma sistemática metodologia de projeto.
- Metodologia de projeto são sequências de transformações que partem de uma descrição ou especificação inicial até chegar a uma descrição validada desse sistema para o processo de fabricação.
- Nível de fabricação pode ser máscaras no caso de circuitos integrados de aplicação especifica ou bitstream no caso de FPGAs.

As transformações podem ser de duas naturezas:

- Transformações de síntese



- Transformações de validação

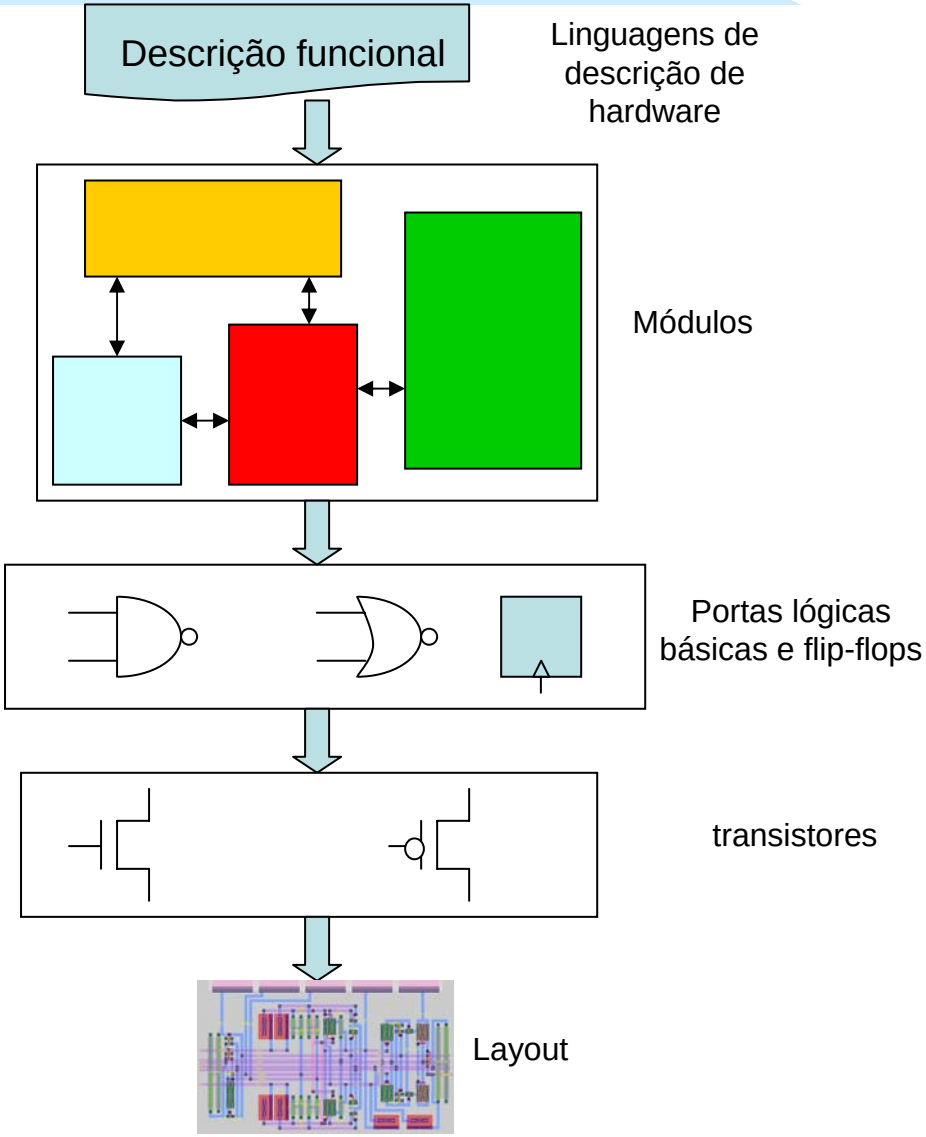


Não válida válida

Projeto

- **Aboradagem Descendente:** decompõe o sistema em subsistemas que são por sua vez decompostos em subsistemas até atingir o níve de abtração desejado.
- **Desafio:** obter a decomposição adequada para cada nível para que no final os critérios de projeto (área, desempenho, potência) sejam atingidos.

- **Abordagem Ascendente:** conecta módulos disponíveis para formar subsistemas que por sua vez são conectados para formar subsistemas até que a especificação funcional seja satisfeita.
- **Desafio:** trabalhar com um conjunto muito grande de subsistemas pequenos para compor um sistema muito complexo.



Sistemas Digitais

Projeto: **Arbodagem Descendente**

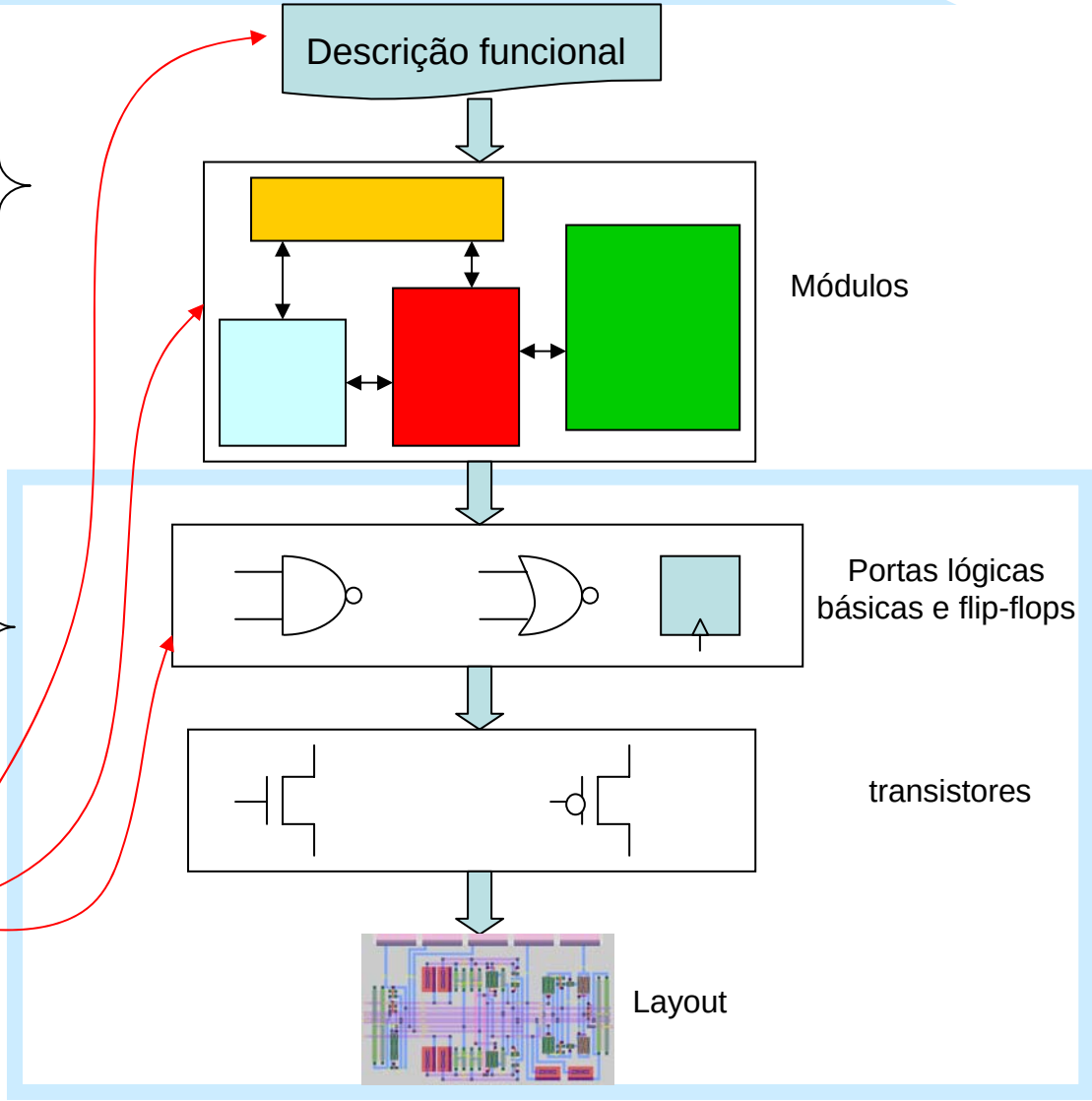
Projeto de Sistemas Digitais usando fluxogramas, grafos, máquinas de estados e diagrama de blocos

Descrever o projeto em **linguagens de descrição de hardware** como por exemplo VHDL

Usar ferramentas de síntese lógica para bibliotecas de células como o Leonardo da Mentor

Usar ferramentas de síntese lógica para plataformas programaveis como FPGAs (Xilinx – ISE, Altera – Quartus, Actel – Libero).

Verificar funcionalmente através de simulação lógica com e sem atraso.

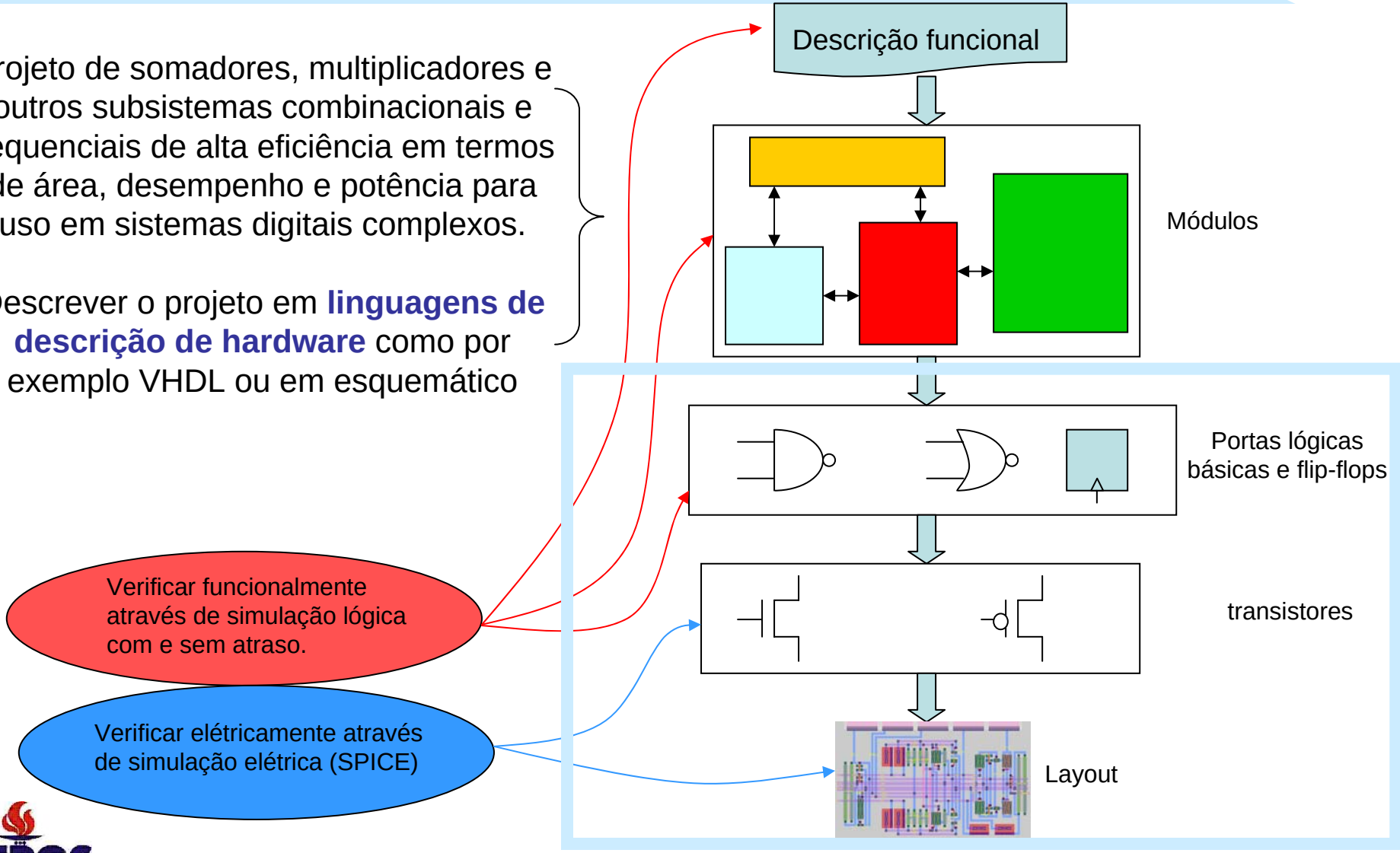


Sistemas Digitais

Projeto: **Arbodagem Ascendente**

Projeto de somadores, multiplicadores e outros subsistemas combinacionais e sequenciais de alta eficiência em termos de área, desempenho e potência para uso em sistemas digitais complexos.

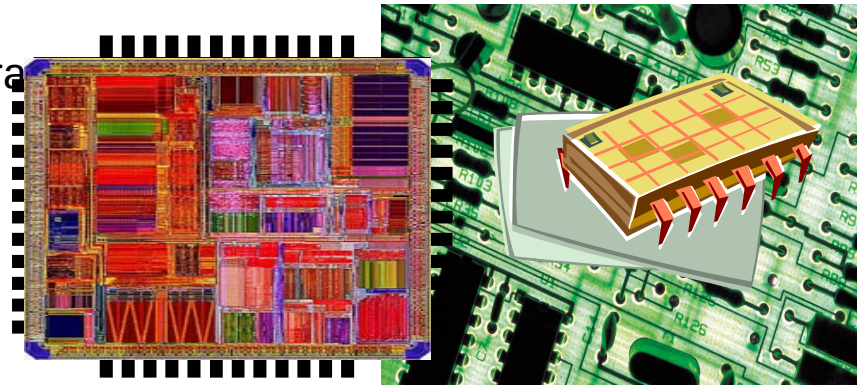
Descrever o projeto em **linguagens de descrição de hardware** como por exemplo VHDL ou em esquemático



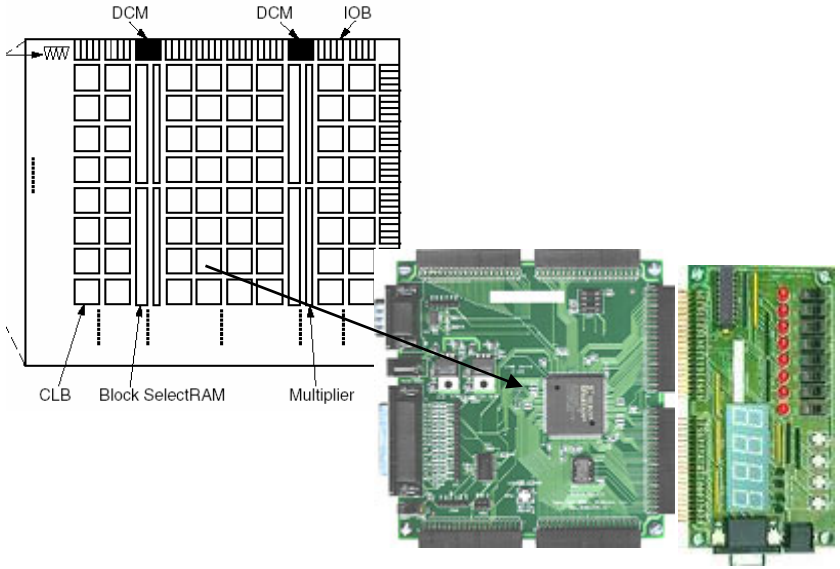
Tipos de componentes

Circuito de aplicação específica (ASIC): circuito integrado projetado especialmente para uma determinada função e sistema digital.

- Full-custom
- semi-custom
- Standard cell



Lógica programável (FPGAs): circuito que pode ser customizado e re-programado para realizar diversas funções.



Compromisso:

Custo X tempo de projeto X desempenho

1 – Sintaxe e a Semântica das descrições de entrada e saída

2 – Um conjunto de algoritmos para a tradução das descrições de entrada em descrições de saída

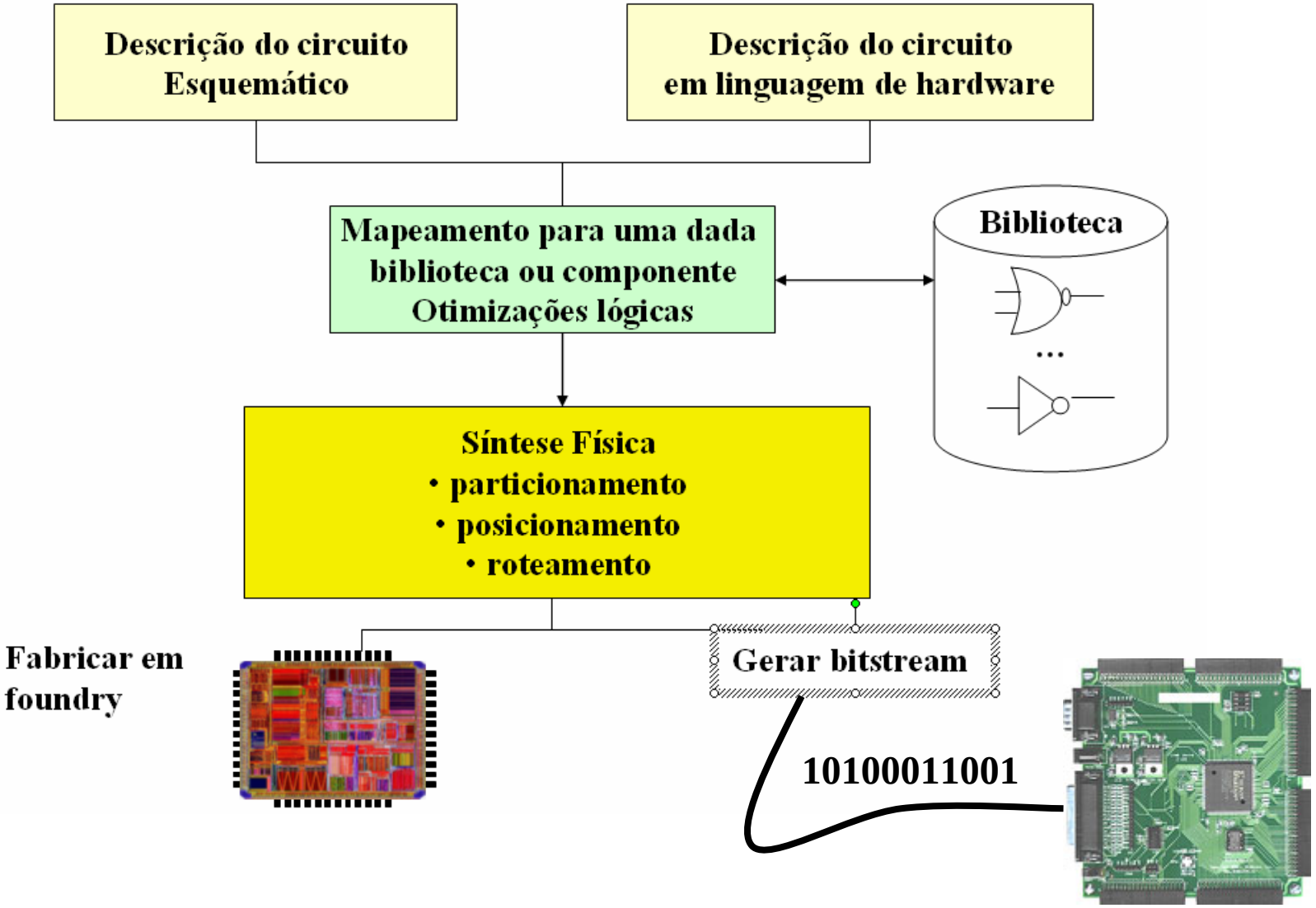
3 – Um conjunto de componentes para ser usado na implementação

4 – Definição e intervalo das restrições do projeto

5 – Os mecanismos de seleção do estilo de projeto, arquitetura, topologia e componentes.

6 – Estratégias de controle (ordem em que as tarefas são executadas).

Fluxo de Projeto (simplificado)



Parâmetros de Projeto

- Desempenho: velocidade, potência, funcionalidade e flexibilidade
- Custo de manufatura: tamanho do die (área), tecnologia a ser fabricada (ASIC), ou arquitetura programável (FPGA).
- Tempo de projeto: custo do engenheiro, agenda
- Testabilidade: geração de teste, teste on-line, off line, etc…

Desafios de Projeto

- Decisões sobre os aspectos de projeto são muito dificeis:
  - Compromisso entre desempenho, custo e time-to-market (tempo para chegar ao mercado).
  - Decisões devem ser feitas com 2 a 3 anos de antecedencia.
  - Os aspectos de projeto são difíceis de medir sem fazer o projeto realmente.
  - Ciclo do produto.

- Verificação funcional
  - A simulação ainda é o veiculo principal para a verificação funcional mas é inadequada por causa do tamanho de projeto.
  - Bugs em hardware são muito difíceis de se recuperar e muito caros (não é como em software).

Desafios de Projeto

- Principais diferenças entre os níveis de abstração:
  - Modelagem detalhada e tamanho da equipe para manter o modelo:
    - Modelos de alto-nível podem ser mantidos por 1 ou 2 pessoas.
    - Modelos detalhados devem ser particionados o que resulta no aumento em comunicação.
  - Modelagem precisa versus modelagem compacta
    - Modelos compactos omitem detalhes e mostram apenas estimações de implementação.
    - Modelos detalhados são extensos e dificil de adaptar em mudanças grandes de projeto.

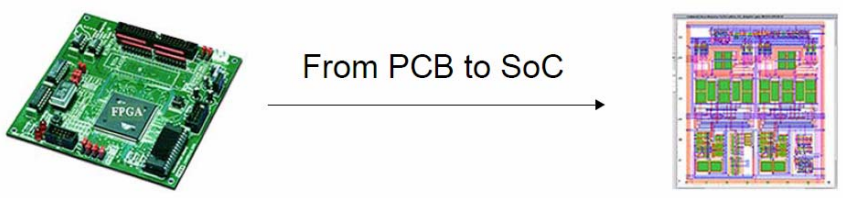
Desafios de Projeto

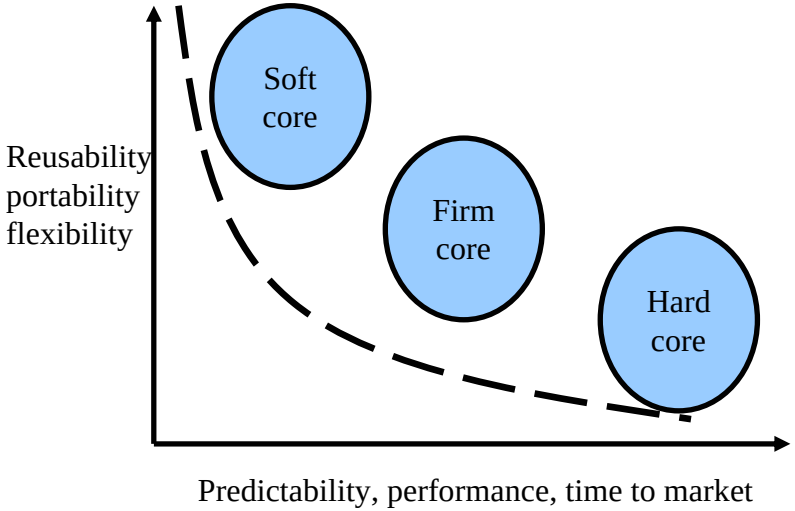
- Velocidade de simulação versus desempenho do hardware
  - Modelos de alto-nível podem ser simulados mais rapidamente mas não podem ser implementados tão facilmente automaticamente.
  - Modelos de baixo nível podem ser feitos para ter uma rápida implementação mas não podem ser simulados rapidamente.

Ferramentas de Automação de Projeto

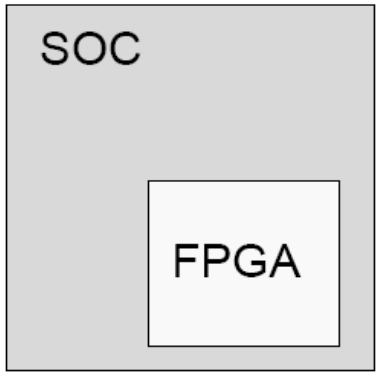
Soluções para Projeto:

- aumentar equipes (+custo e -controle)
- *Systems-on-Chip (SOCs)* = Reusabilidade

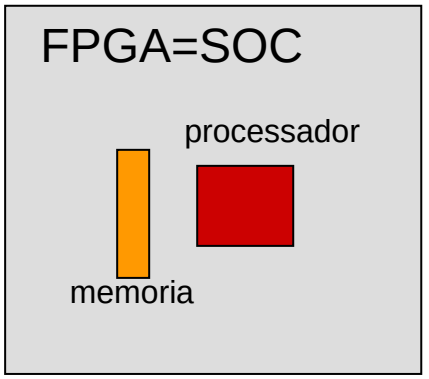




SOC e configurabilidade



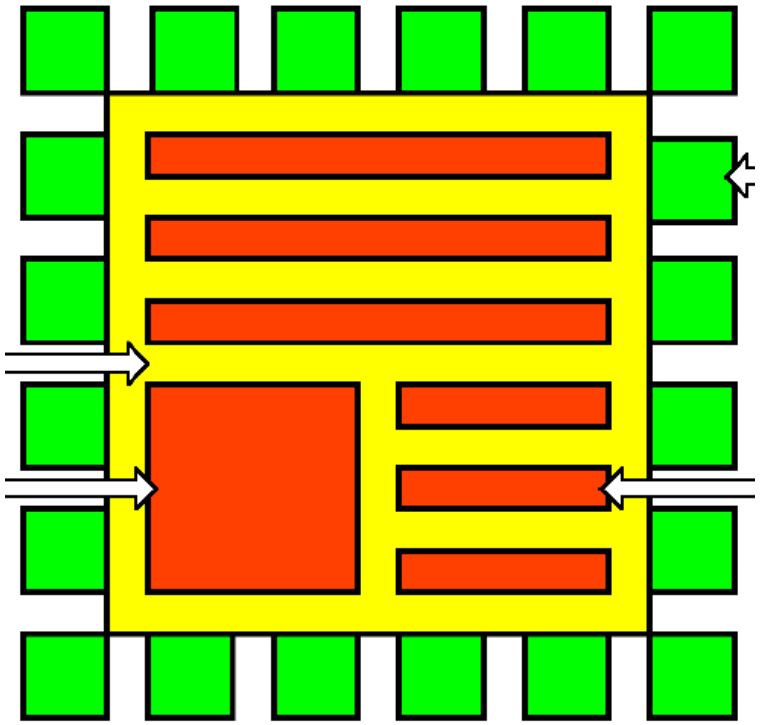
ou



Metodologias de Projeto

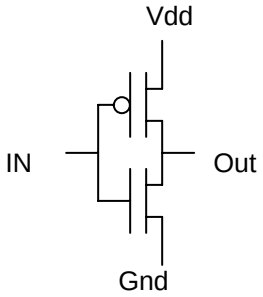
ASIC

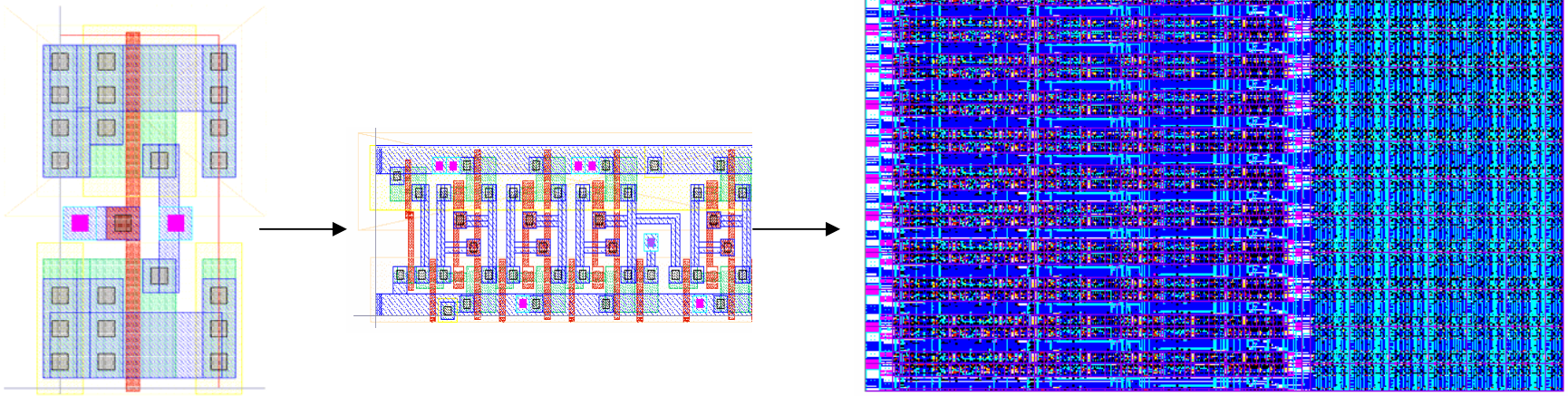
- Application Specific Integrated Circuits



Full custom

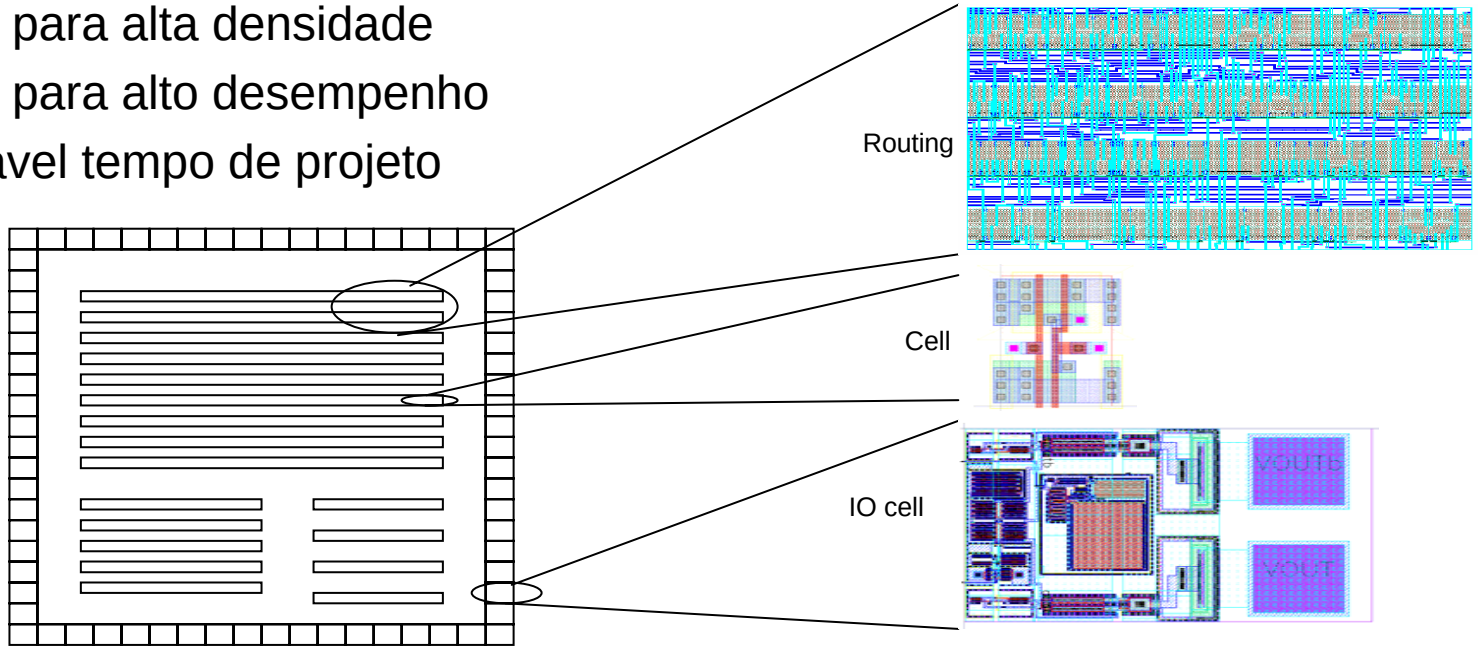
- Geometria desenhada a mão
- Todas as camadas de layout são customizadas
- Digital e analógico
- Simulação a nivel de transistor
- Alta densidade
- Alto desempenho
- Longo tempo de projeto





Standard cells

- Células padrões organizadas em linhas (and, or, flip-flops,etc.)
- Células são feitas em full custom pelo vendedor (não o usuário).
- Todos os níveis são customizaveis
- Digital com possibilidade de algumas células analógicas.
- Simulação digital a nível de portas lógicas (digital)
- Media para alta densidade
- Medio para alto desempenho
- Razoavel tempo de projeto



Níveis de Abstração de Sistemas VLSI em ASIC

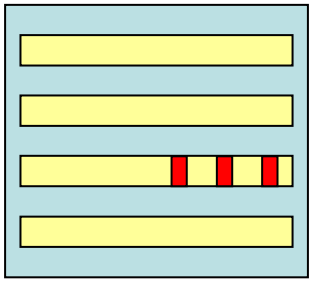
Projeto

- Nível Sistêmico
- Nível Arquitetural ou Algoritmico
- Nível de transferencia entre registradores (micro-arquitetural)
- Nível lógico ou de portas
- Nivel elétrico ou de transistores
- Nivel de leiatue
- Nivel de máscaras

Verificação

Exemplo de Projeto ASIC

- Projeto de um circuito integrado a partir de uma descrição:
  - VHDL
  - Esquemático
- Circuito integrado implementado em Standard Cell (conjunto de células lógicas de uma biblioteca)
  - Biblioteca da AMS, por exemplo
  - Biblioteca do usuário
- Passos:
  - Projeto das células da biblioteca CMOS
  - Sintese de um projeto VLSI nas biblioteca de células CMOS



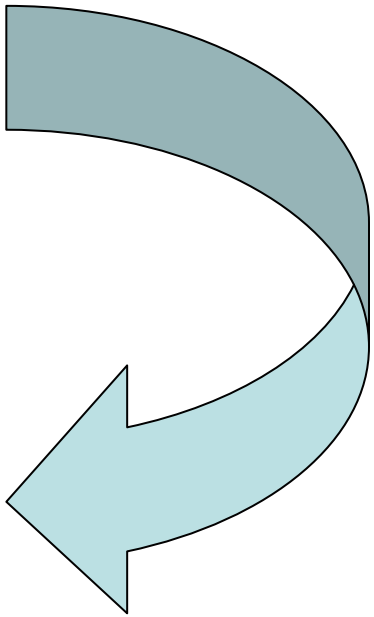
2 Exemplos de Ferramentas de Síntese

Mentor

Cadence

Synopsis

E outras ...



- Compilação
- Mapeamento
- Posicionamento
- Roteamento
- Leiaute
- Simulação e verificação

Circuitos Programáveis

- Why Programmable Logic Devices (PLDs)?
  - Low cost, low risk way of implementing digital circuits as application specific ICs (ASICs).
  - Technology of choice for low to medium volume products (say hundreds to few 10's of thousands per year).
  - Good and low cost design software.
  - Latest high density devices are over 1 million gates!

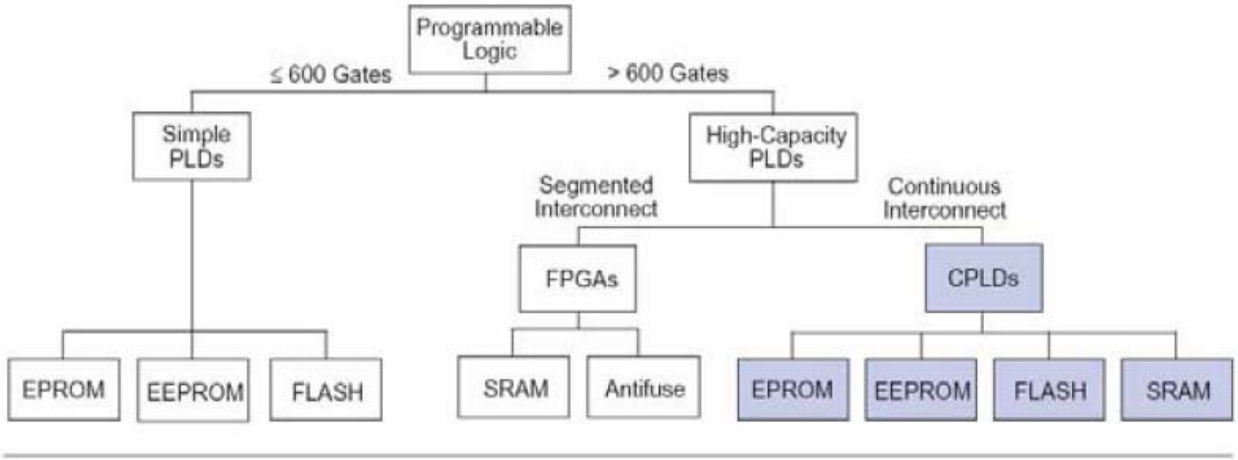
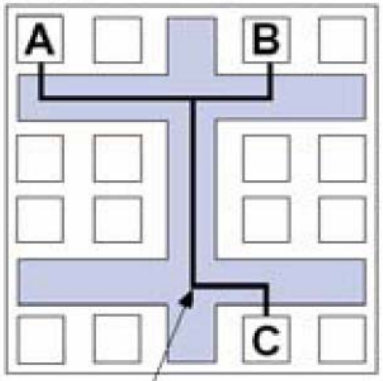


Table 1 describes CPLD and FPGA features.

Diferenças entre CPLD x FPGA

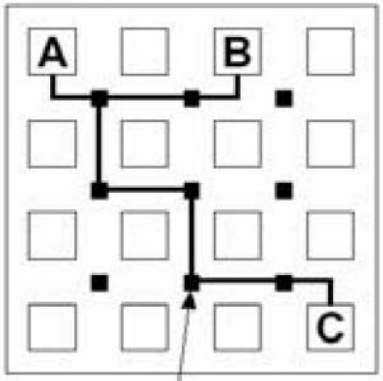
Figure 3. CPLD vs. FPGA Routing Scheme

CPLD Continuous Interconnect Structure



Fixed/Predictable Delay

FPGA Segmented Interconnect Structure



Variable/Unpredictable Delay

Interconnect structures affect the following device characteristics:

- Performance predictability
- In-system performance
- Logic utilization

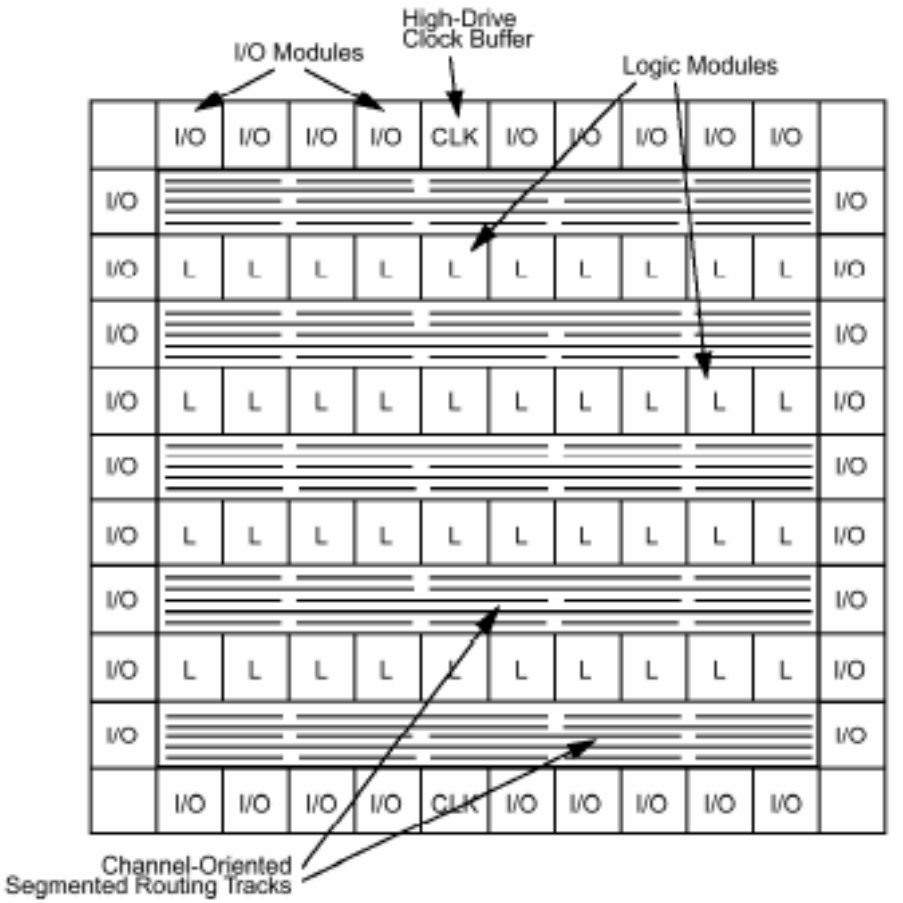
Estruturas do FPGA

- All FPGAs have the following key elements:
  - The Programming technology
  - The basic logic cells
  - The I/O logic cells
  - Programmable interconnect
  - Software to design and program the FPGA

- Currently the four main players in this field are:-
  - Actel
  - **Altera**
  - **Xilinx**
  - Atmel

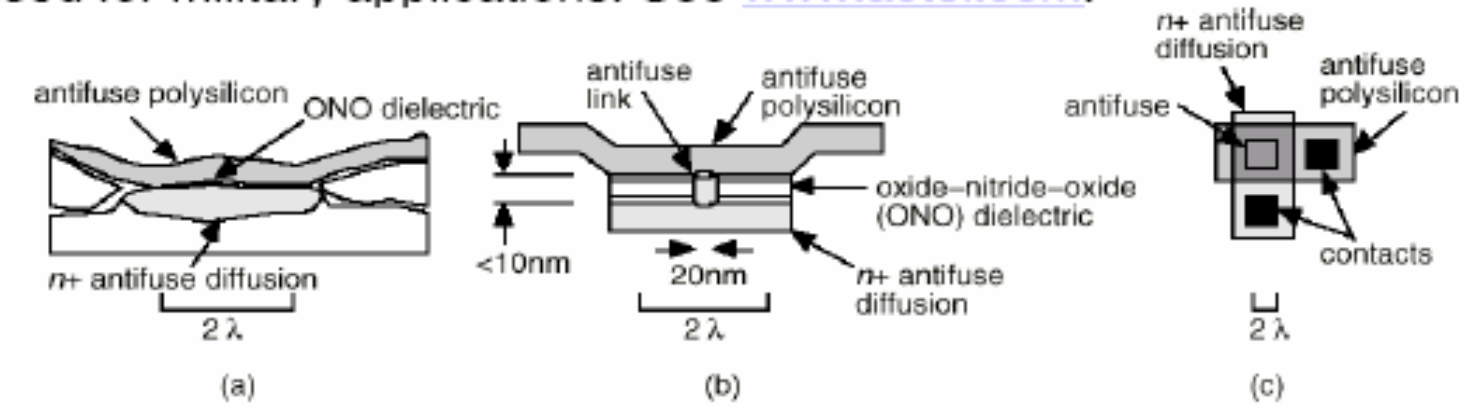
FPGA Actel

- Uses antifuse technology
- Based on **channelled gate array** architecture as shown below
- Each logic element (labelled 'L') is a combination of multiplexers which can be configured as a multi-input gate



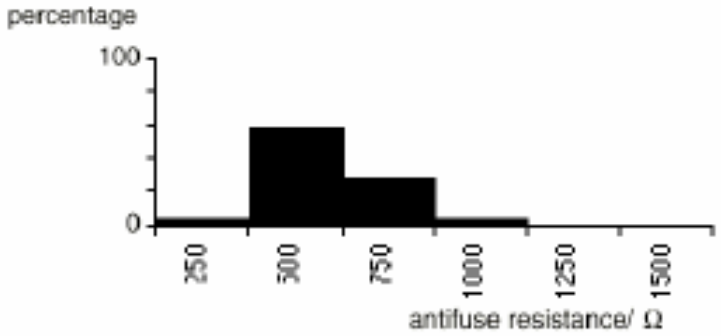
Tecnologia Anti-fusivel

- Invented at Stanford and **developed** by Actel. Currently mainly used for military applications. See www.actel.com.



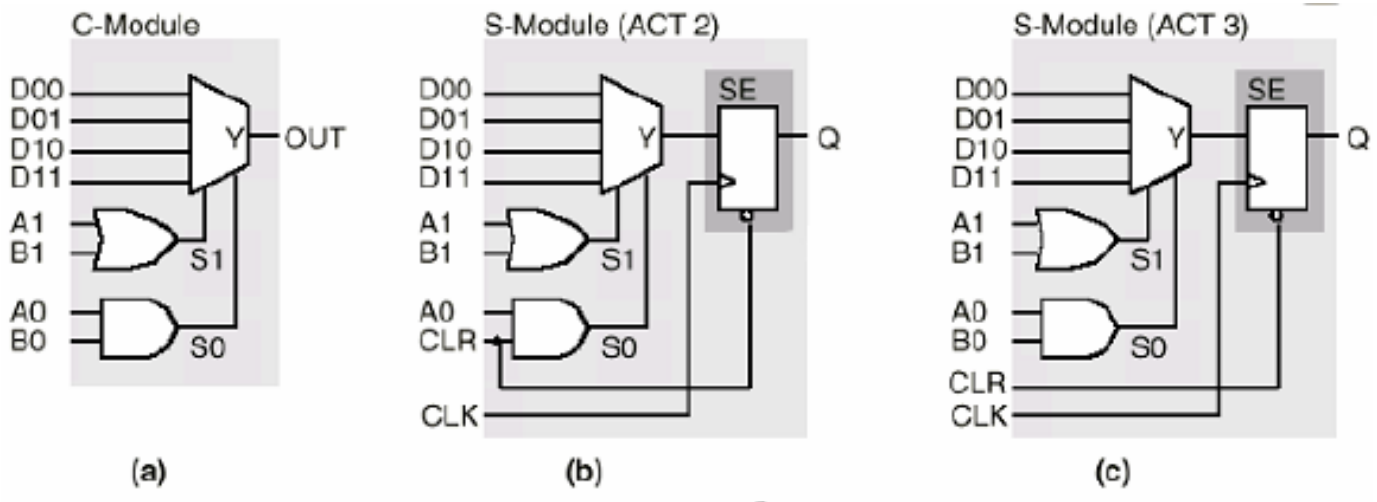
Number of antifuses on Actel FPGAs

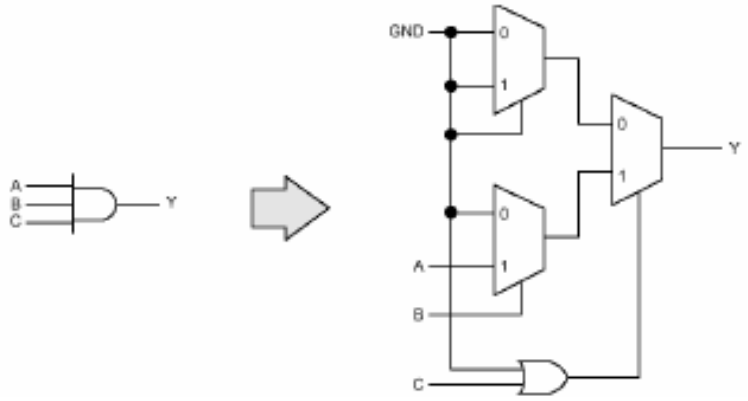
Device	Antifuses
A1010	112,000
A1020	186,000
A1225	250,000
A1240	400,000
A1280	750,000



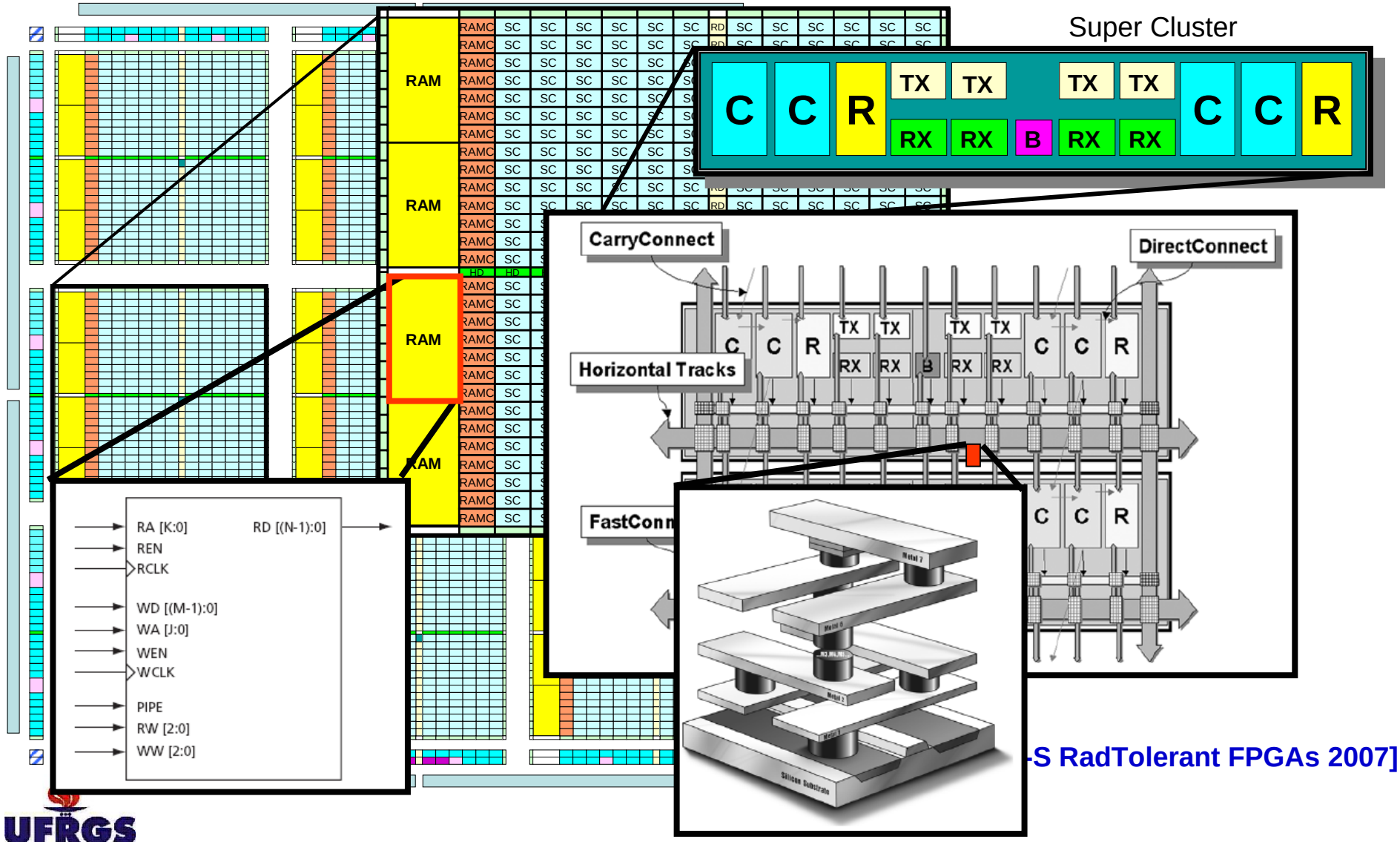
The resistance of blown Actel antifuses

Logic Elements of Actel



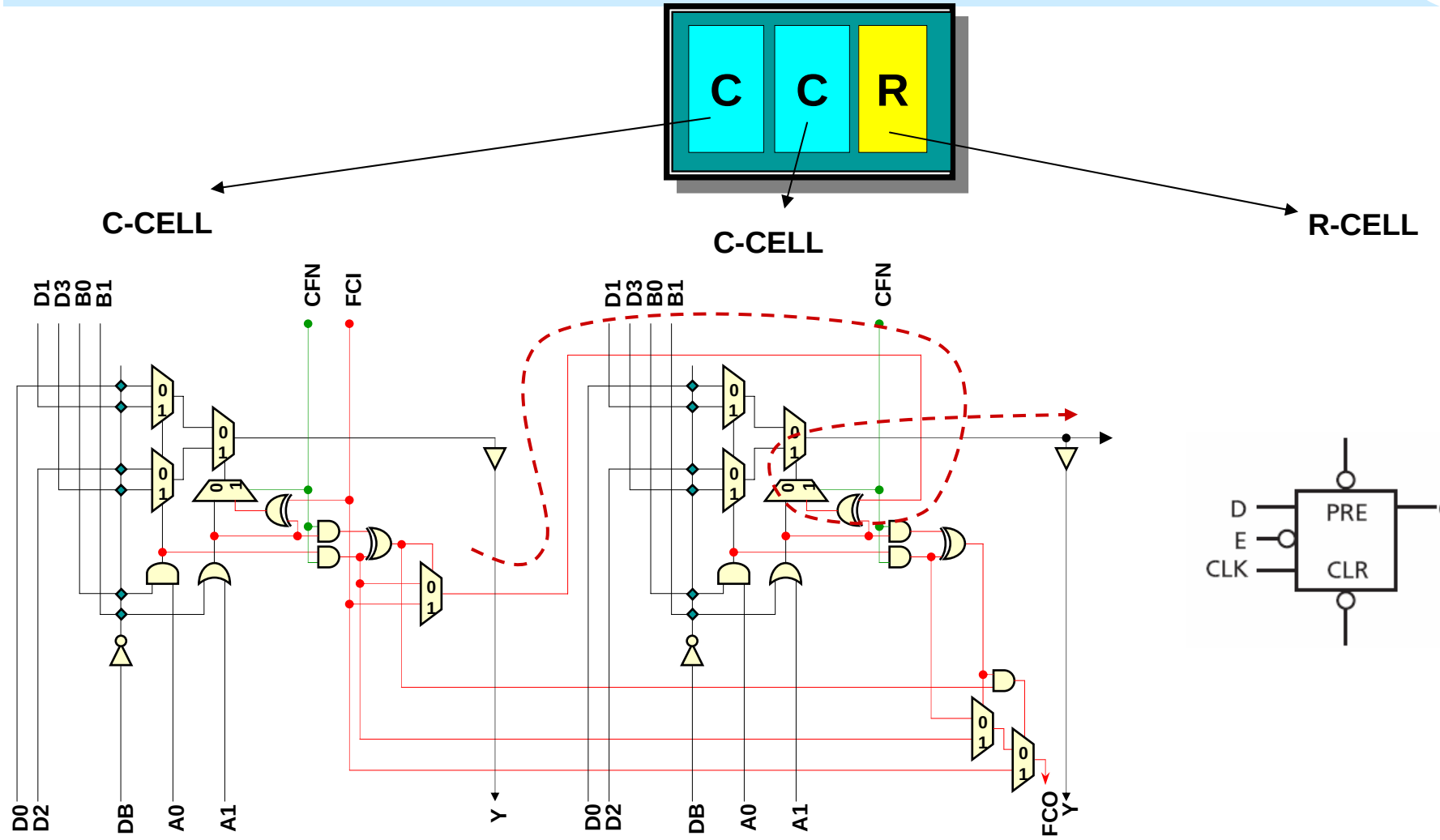


ACTEL: RTAX-S device



-S RadTolerant FPGAs 2007]

ACTEL: RTAX-S device



Address http://www.actel.com/products/tools/libero/

Google Libero Actel | Search | 53 blocked | Check | AutoLink | AutoFill | Options

Home | Company | Products & Services | Technical Documentation | Technical Support | Contact Us

Actel PRODUCTS & SERVICES products & services

Search go

Advanced Search Site Map

Software Tools: Libero Integrated Design Environment (IDE)

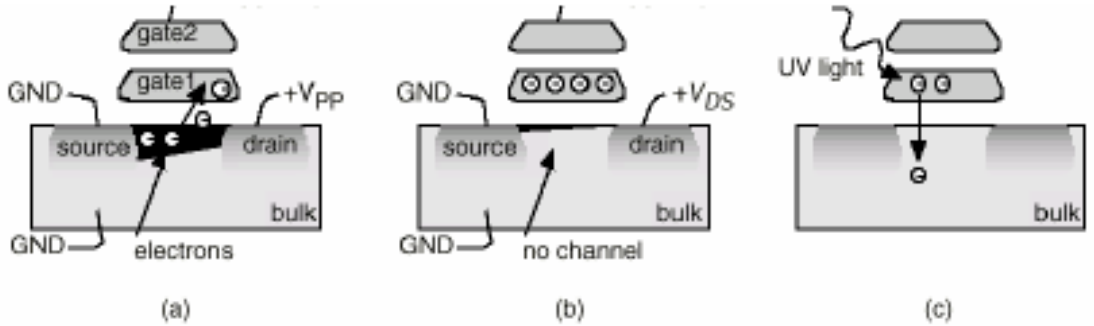
Actel's Libero IDE offers the latest and best-in-class tools from leading EDA vendors such as Magma Design Automation, Mentor Graphics, SynaptiCAD, and Synplicity. These tools, combined with custom developed tools from Actel, are integrated into a single FPGA development package. Actel truly offers a one-stop shopping solution to completely orchestrate the Libero design flow including a powerful design manager that guides you through the design process, keeps track of your design files, and seamlessly manages file exchanges between the various tools.

Libero IDE supports all currently released Actel devices including the new Fusion AFS600, ARM-enabled Fusion M7AFS600, and ARM-enabled M7A3P/E devices. Libero IDE is available in several editions to meet budget and tool needs: Libero Gold, Libero Platinum, and a Libero Platinum Evaluation version. Libero IDE includes Actel's Designer software, which offers premier back-end tools for physical implementation, including a comprehensive floorplanning capability via Actel's ChipPlanner feature. Timing constraint setting and analysis is performed with SmartTime, a new and highly advanced environment for quickly setting and meeting timing objectives. Designer is available as a standalone product for those who want to use their own design and verification tools.

- Libero IDE
  - Libero Editions
  - Product Brochure
  - User Manuals & Guides
  - Licensing & Registration
  - What's New in Libero
  - Libero Release Notes
  - Request Software
- Designer Software
- Device Support
- Power Calculators
- Tools Overview
- License Types
- Partners

Tecnologia EPROM e EEPROM

- Generally used in product-term type of PLDs.
- Non-volatile and reprogrammable.



An EPROM transistor

(a) With a high (>12V) programming voltage, V_{PP}, applied to the drain, electrons gain enough energy to "jump" onto the floating gate (gate1)

(b) Electrons stuck on gate1 raise the threshold voltage so that the transistor is always for normal operating voltages

(c) UV light provides enough energy for the electrons stuck on gate1 to "jump" back to t bulk, allowing the transistor to operate normally

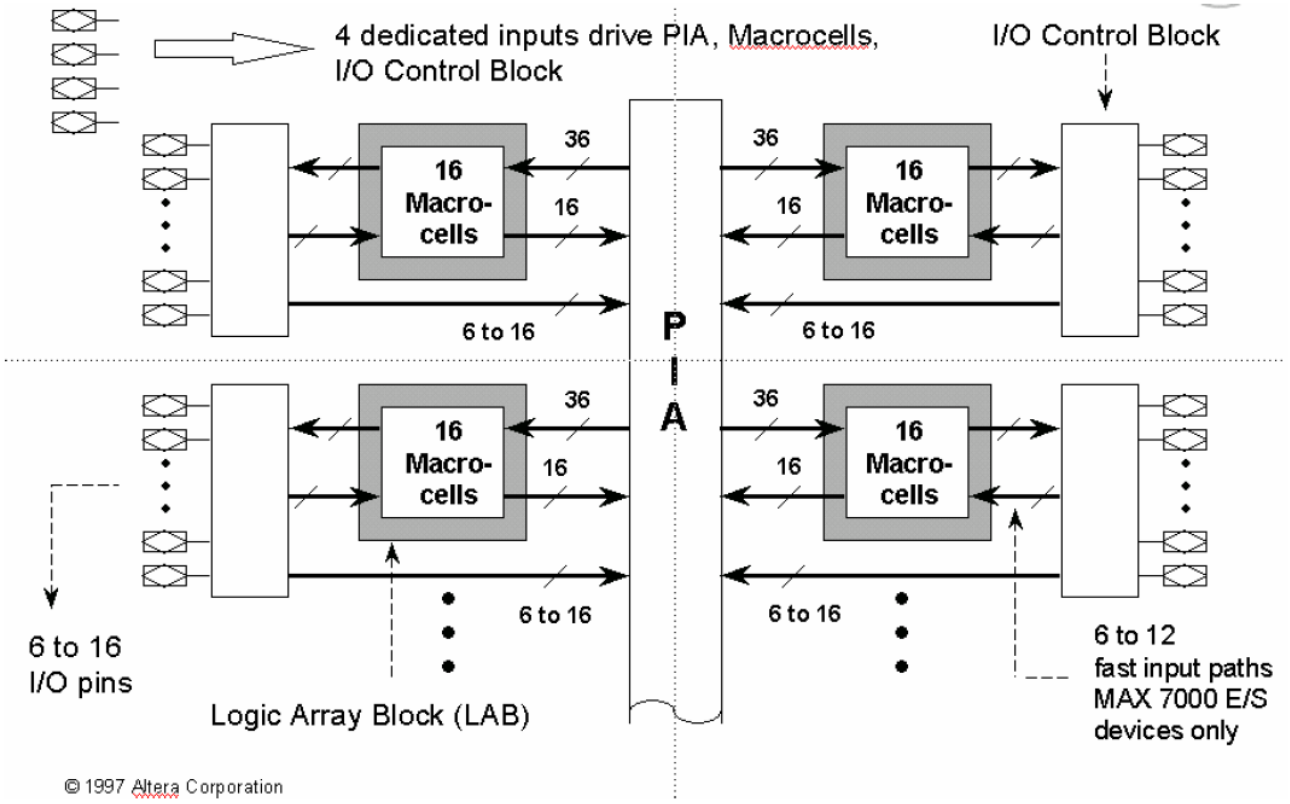
FPAG from Altera

- CPLD = Complex Programmable Logic Devices
- FPGA = Field Programming Gate Arrays
- Altera has four different PLD families:
  - MAX family – product-term based macrocells CPLDs
  - FLEX family – SRAM based lookup tables (LUTs)
  - APEX family – mixture of product-term and LUT based devices
  - Stratix family – Advanced FPGAs with embedded blocks (Stratix-2 is currently the most advanced FPGA devices)

Família	Número de Gates	Programação
MAX5000	600 a 3,7K	EEPROM
MAX7000	600 a 5K	EEPROM
MAX9000	6K a 12K	EEPROM
FLEX6000	5K a 24K	SRAM
FLEX8000	2,5K a 16K	SRAM
FLEX10K	10K a 250K	SRAM
FLEX20K	53K a 1000K	SRAM
Mercury	120k a 350k	SRAM
Apex	700k a 2M	SRAM
ApexII	1.9M a 5.2M	SRAM
Ciclone		SRAM
Stratix	10k a 40k LE	SRAM
MAX		SRAM

Bloco Lógico do PLD MAX7000

- Consists of Logic Array Blocks (LABs), each with 16 macro-cells
- PIA = Programmable Interconnect Array

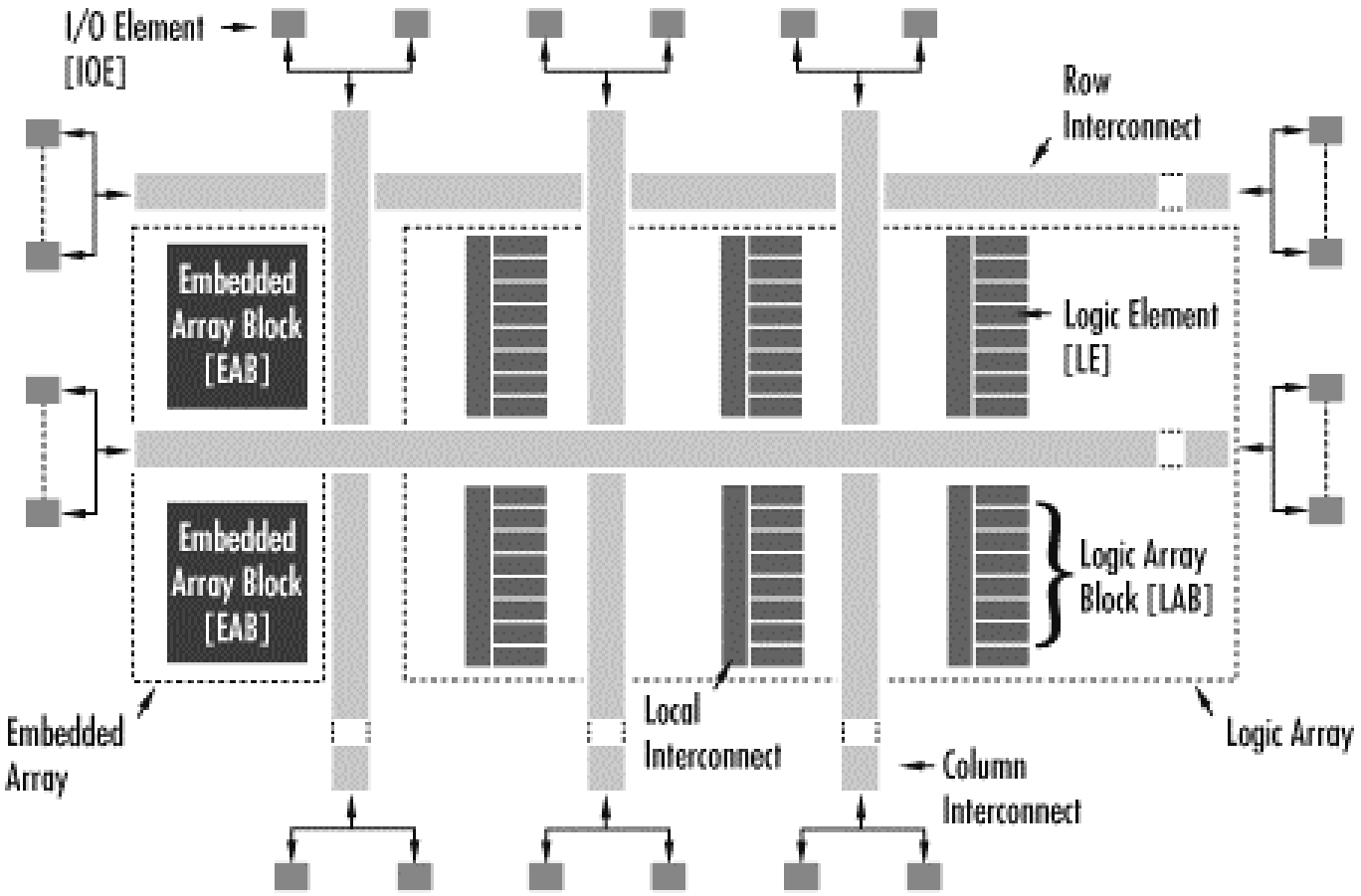


Field Programmable Gate Array

FPGAs comerciais

Altera

FLEX 10K



Field Programmable Gate Array

FPGAs comerciais

Altera

FLEX 10K

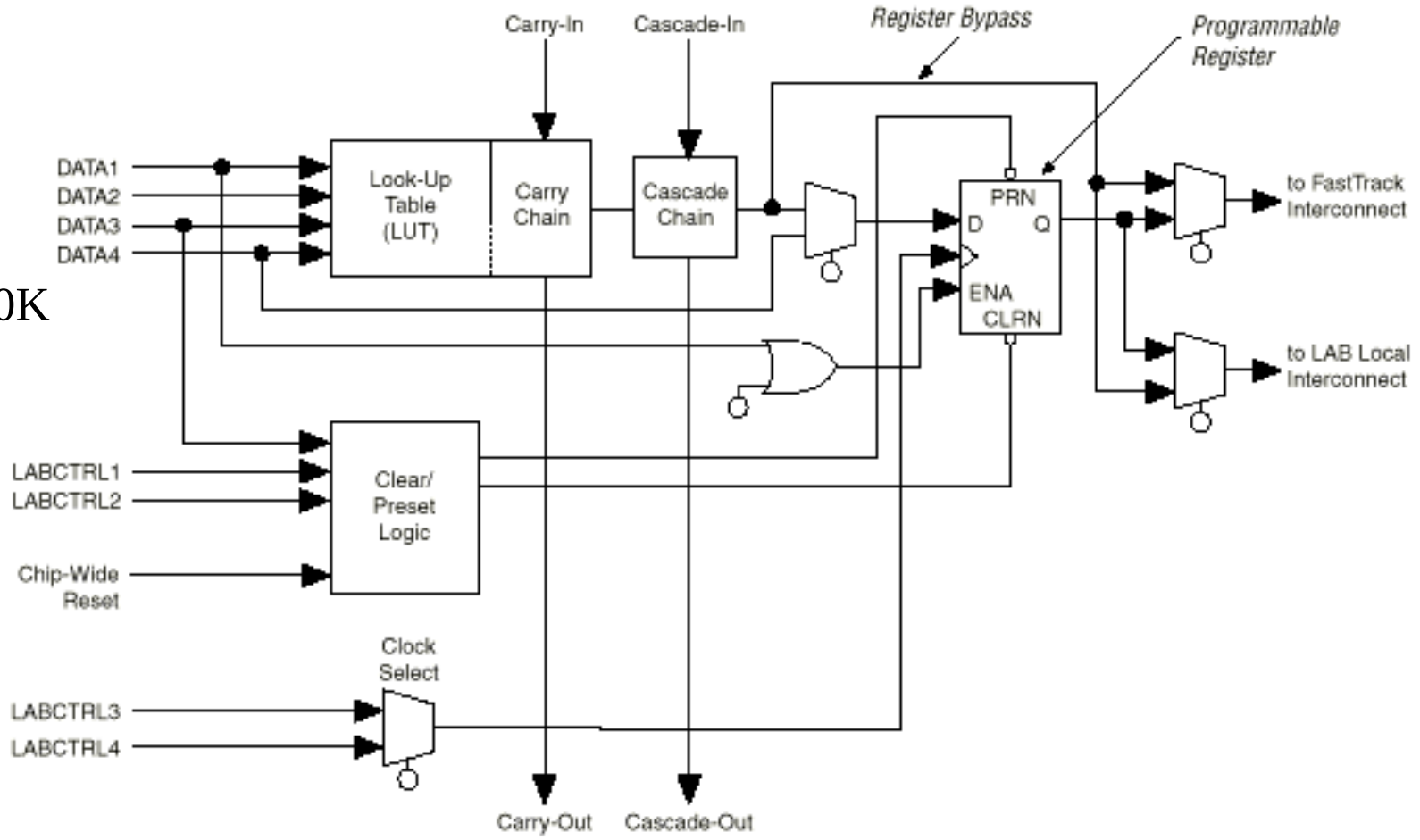


Figure 1. Mercury Architecture Block Diagram *Note (1)*

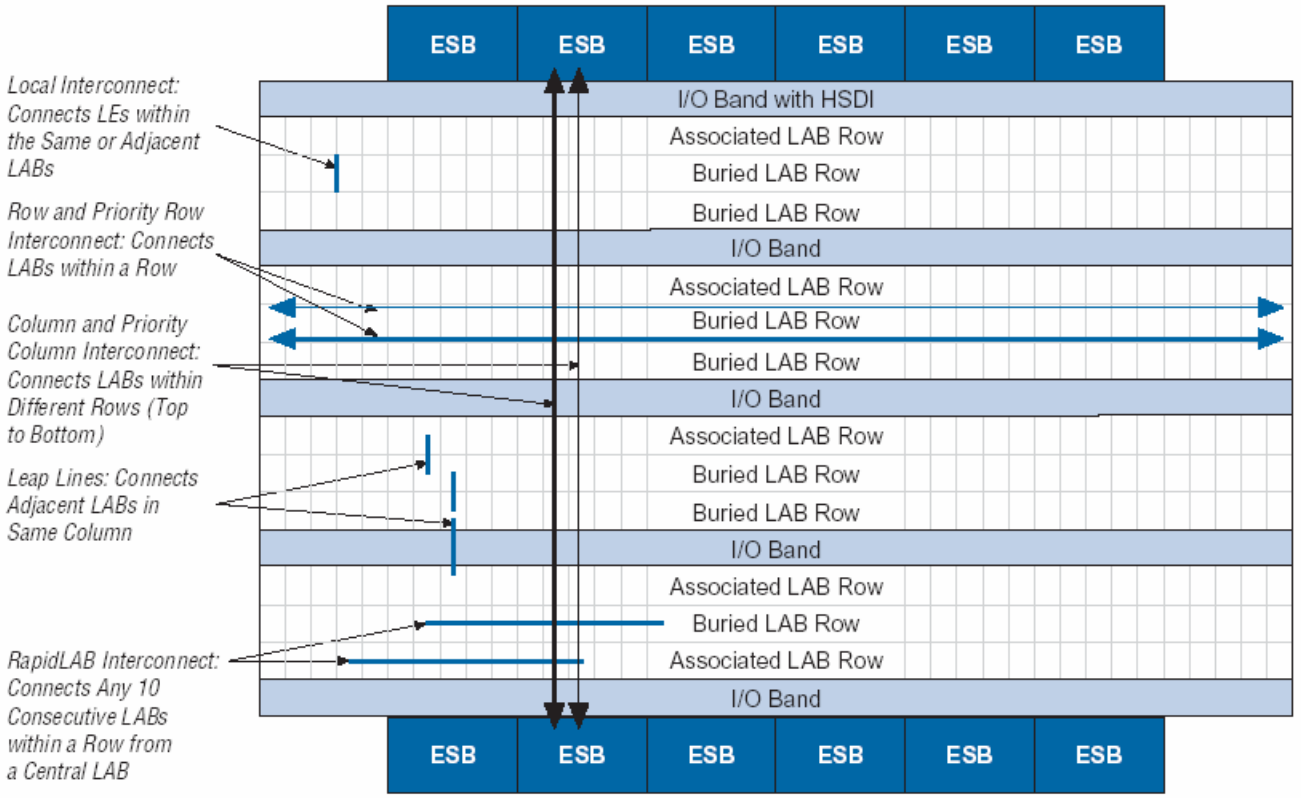


Figure 7. Mercury LE

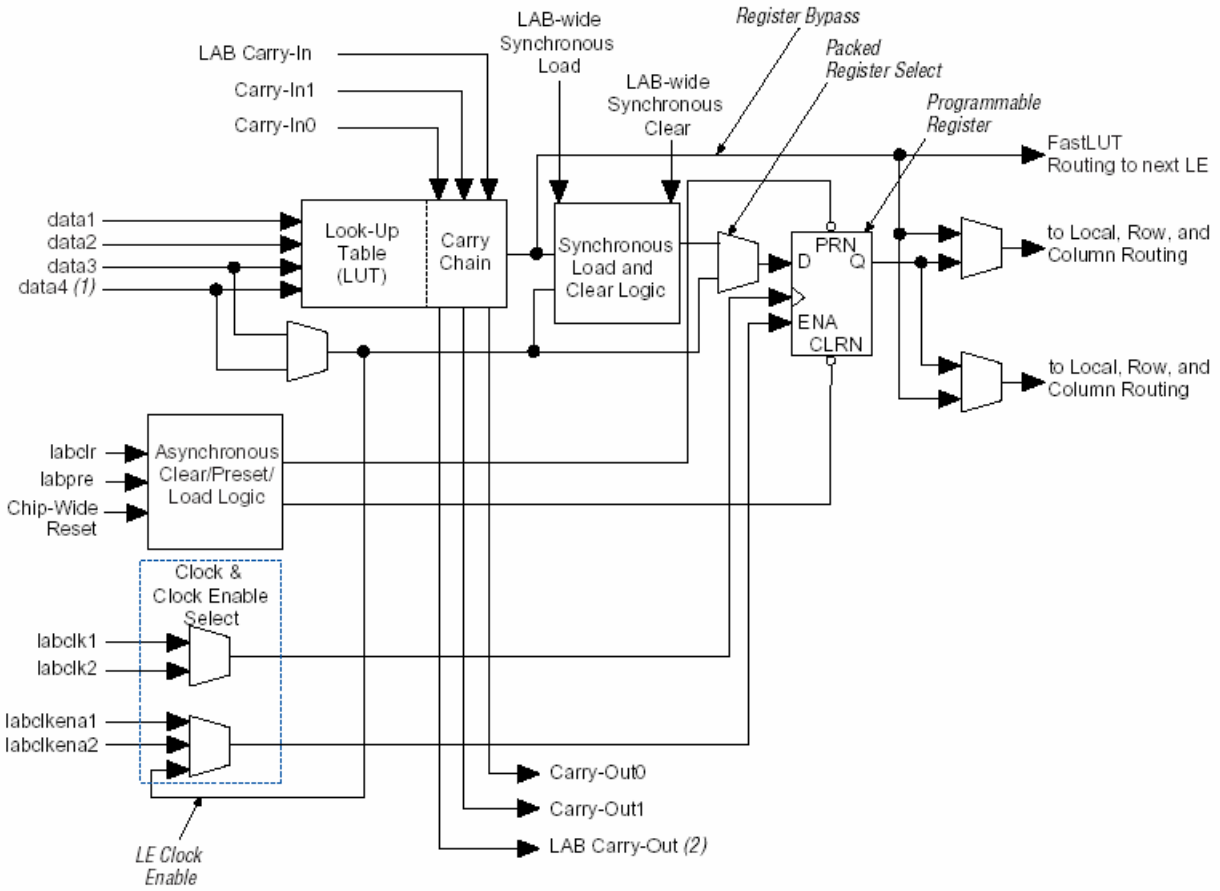


Figure 1. APEX 20K Device Block Diagram

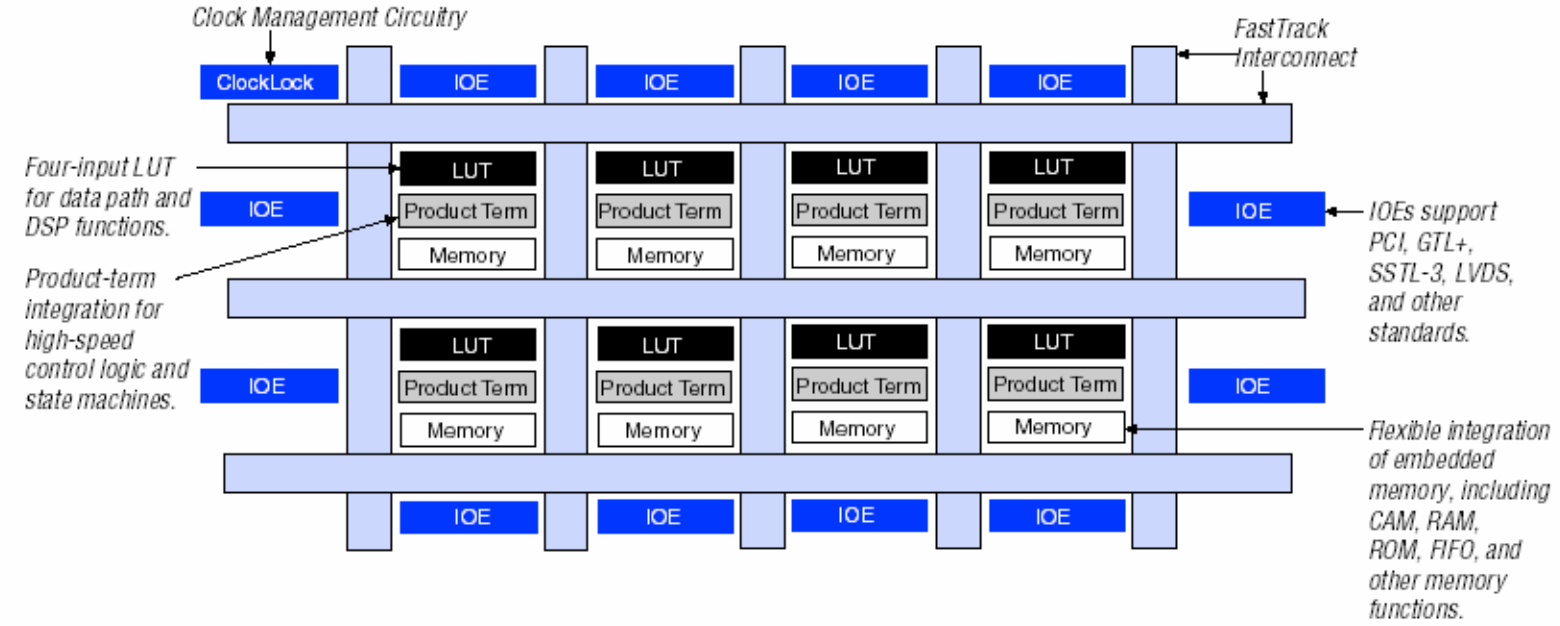
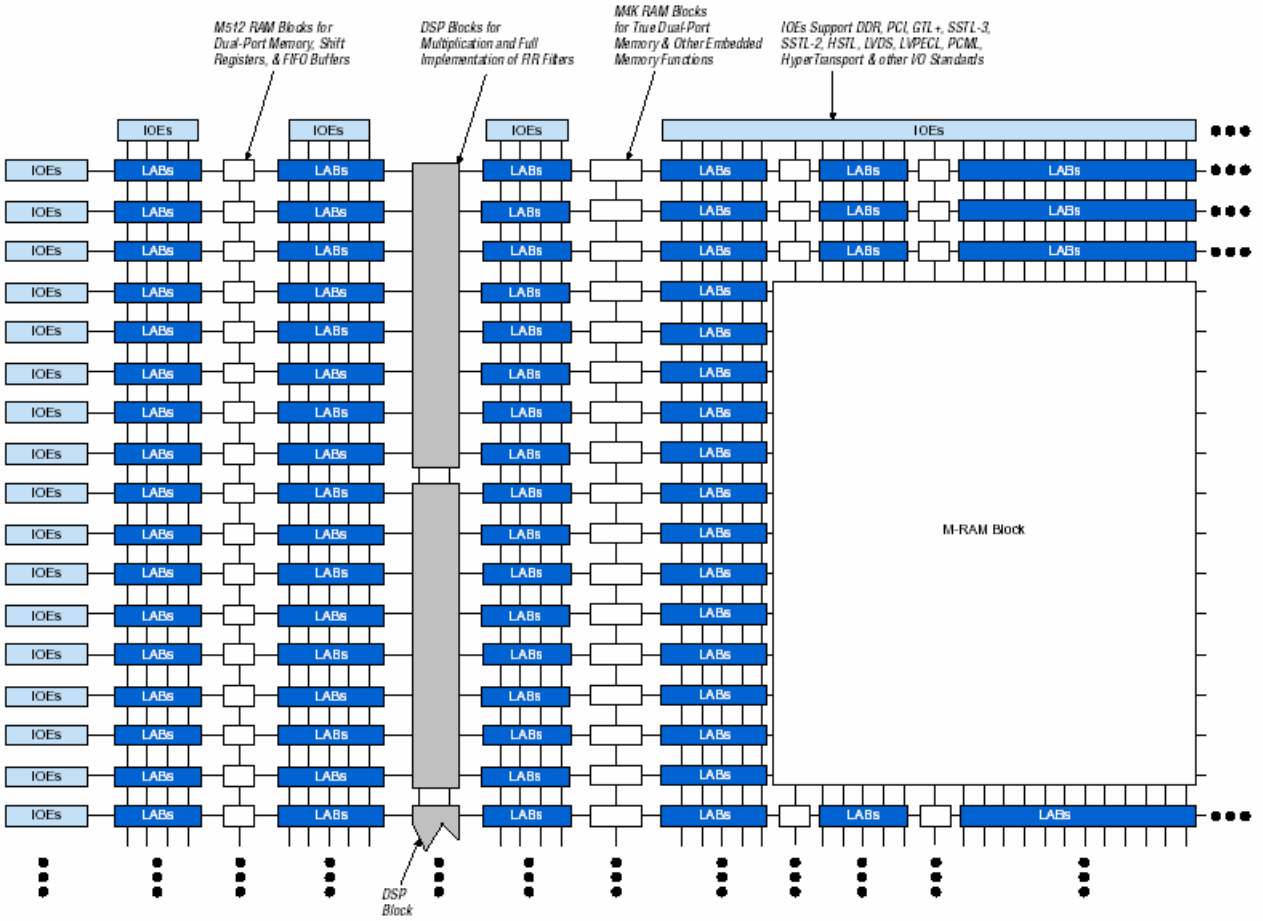
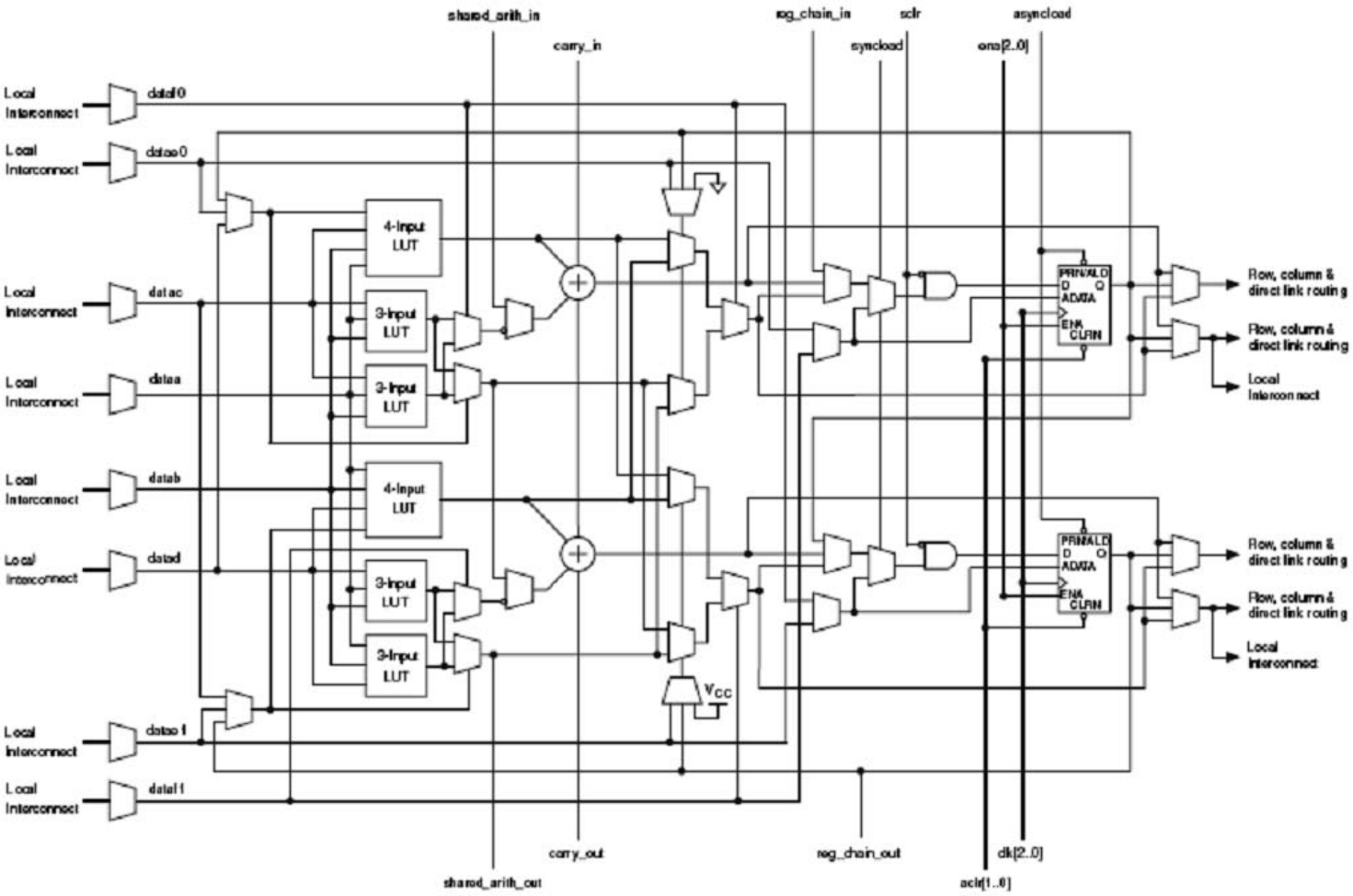
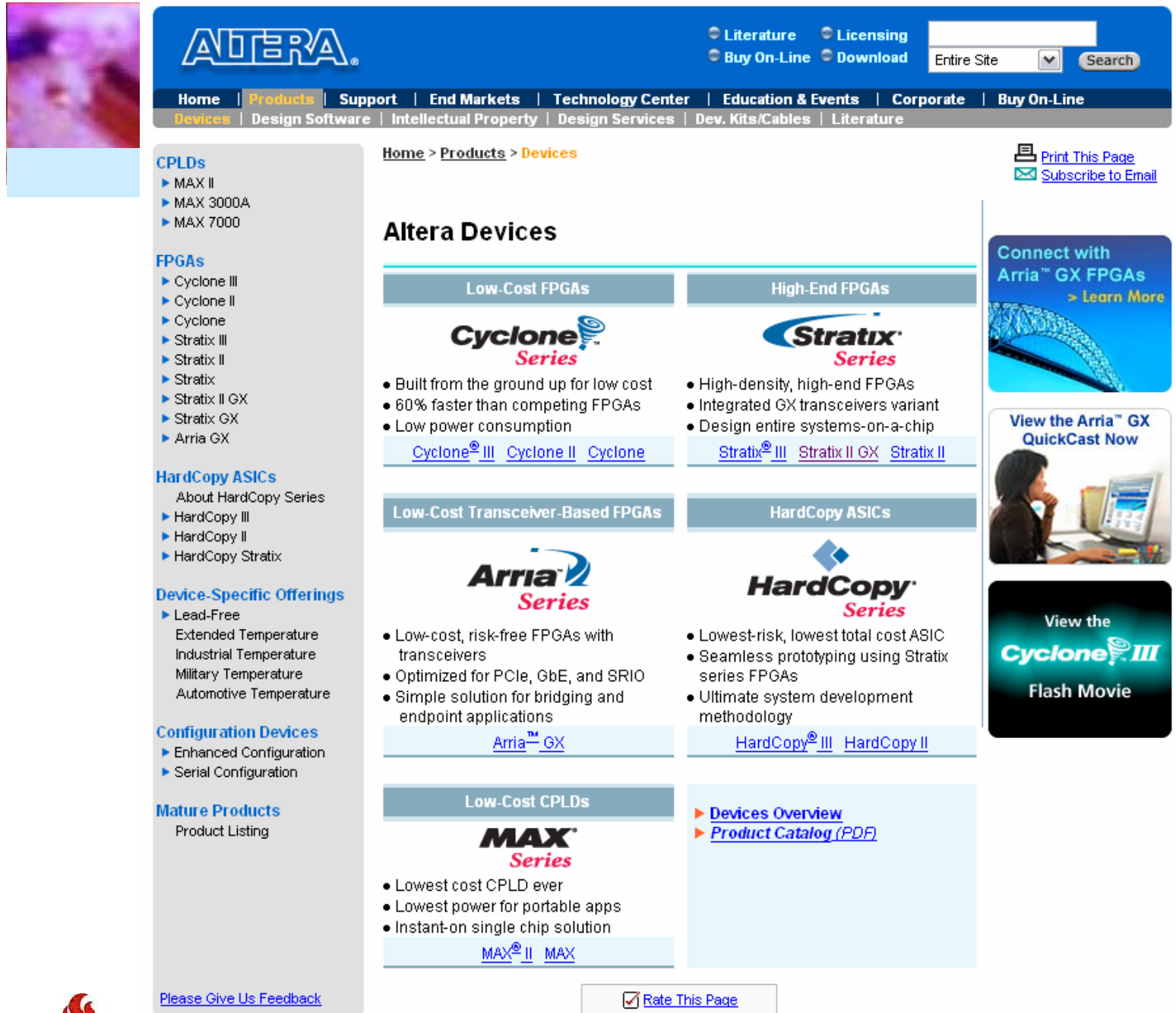


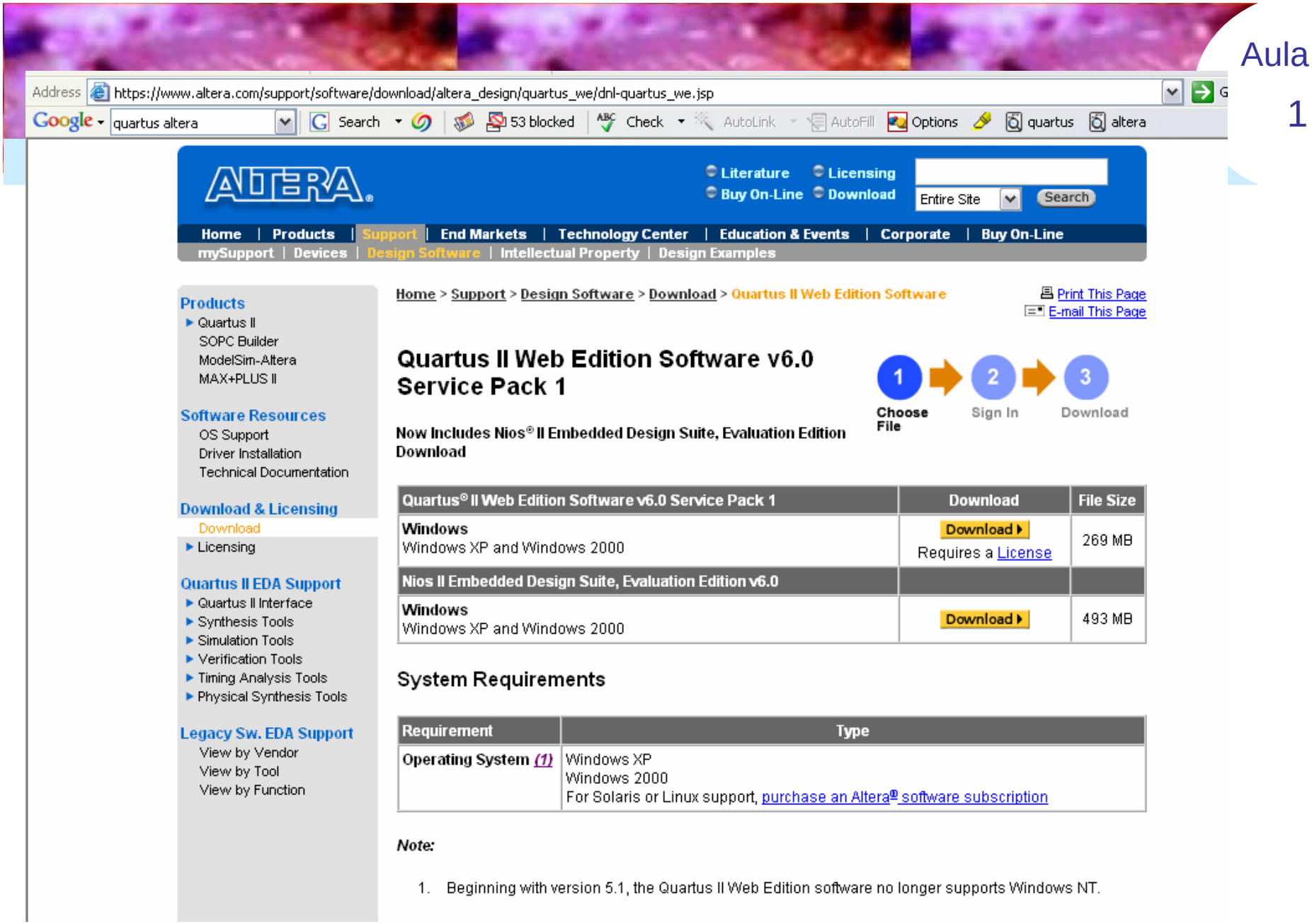
Figure 1–2. Stratix GX Block Diagram



Stradix II


shared_arith_in
carry_in
reg_chain_in
sclr
syncload
asynchload
ena[2..0]
Local Interconnect
datad0
datae0
datac
datae
datab
datad
datae1
datad1
4-Input LUT
3-Input LUT
3-Input LUT
4-Input LUT
3-Input LUT
3-Input LUT
VCC
PRN/ALD
D Q
ADATA
ENA CLRN
Row, column & direct link routing
Row, column & direct link routing
Local Interconnect
Row, column & direct link routing
Row, column & direct link routing
Local Interconnect
carry_out
shared_arith_out
reg_chain_out
clk[2..0]
aclr[1..0]


ALTERA
Literature
Licensing
Buy On-Line
Download
Entire Site
Search
Home | Products | Support | End Markets | Technology Center | Education & Events | Corporate | Buy On-Line
Devices | Design Software | Intellectual Property | Design Services | Dev. Kits/Cables | Literature
CPLDs
MAX II
MAX 3000A
MAX 7000
FPGAs
Cyclone III
Cyclone II
Cyclone
Stratix III
Stratix II
Stratix
Stratix II GX
Stratix GX
Arria GX
HardCopy ASICs
About HardCopy Series
HardCopy III
HardCopy II
HardCopy Stratix
Device-Specific Offerings
Lead-Free
Extended Temperature
Industrial Temperature
Military Temperature
Automotive Temperature
Configuration Devices
Enhanced Configuration
Serial Configuration
Mature Products
Product Listing
Please Give Us Feedback
Home > Products > Devices
Print This Page
Subscribe to Email
Altera Devices
Low-Cost FPGAs
Cyclone Series
• Built from the ground up for low cost
• 60% faster than competing FPGAs
• Low power consumption
Cyclone® III Cyclone II Cyclone
High-End FPGAs
Stratix Series
• High-density, high-end FPGAs
• Integrated GX transceivers variant
• Design entire systems-on-a-chip
Stratix® III Stratix II GX Stratix II
Low-Cost Transceiver-Based FPGAs
Arria Series
• Low-cost, risk-free FPGAs with transceivers
• Optimized for PCIe, GbE, and SRIO
• Simple solution for bridging and endpoint applications
Arria™ GX
HardCopy ASICs
HardCopy Series
• Lowest-risk, lowest total cost ASIC
• Seamless prototyping using Stratix series FPGAs
• Ultimate system development methodology
HardCopy® III HardCopy II
Low-Cost CPLDs
MAX Series
• Lowest cost CPLD ever
• Lowest power for portable apps
• Instant-on single chip solution
MAX® II MAX
Devices Overview
Product Catalog (PDF)
Rate This Page
Connect with Arria™ GX FPGAs
> Learn More
View the Arria™ GX QuickCast Now
View the Cyclone III Flash Movie



Address https://www.altera.com/support/software/download/altera_design/quartus_we/dnl-quartus_we.jsp

Google quartus altera | Search | 53 blocked | Check | AutoLink | AutoFill | Options | quartus | altera

ALTERA

Literature | Licensing | Buy On-Line | Download | Entire Site | Search

Home | Products | Support | End Markets | Technology Center | Education & Events | Corporate | Buy On-Line

mySupport | Devices | Design Software | Intellectual Property | Design Examples

Products
- Quartus II
- SOPC Builder
- ModelSim-Altera
- MAX+PLUS II

Software Resources
- OS Support
- Driver Installation
- Technical Documentation

Download & Licensing
- Download
- Licensing

Quartus II EDA Support
- Quartus II Interface
- Synthesis Tools
- Simulation Tools
- Verification Tools
- Timing Analysis Tools
- Physical Synthesis Tools

Legacy Sw. EDA Support
- View by Vendor
- View by Tool
- View by Function

Home > Support > Design Software > Download > Quartus II Web Edition Software

Print This Page
E-mail This Page

Quartus II Web Edition Software v6.0 Service Pack 1

1 Choose File → 2 Sign In → 3 Download

Now Includes Nios® II Embedded Design Suite, Evaluation Edition Download

Quartus® II Web Edition Software v6.0 Service Pack 1	Download	File Size
Windows Windows XP and Windows 2000	Download Requires a License	269 MB
Nios II Embedded Design Suite, Evaluation Edition v6.0		
Windows Windows XP and Windows 2000	Download	493 MB

System Requirements

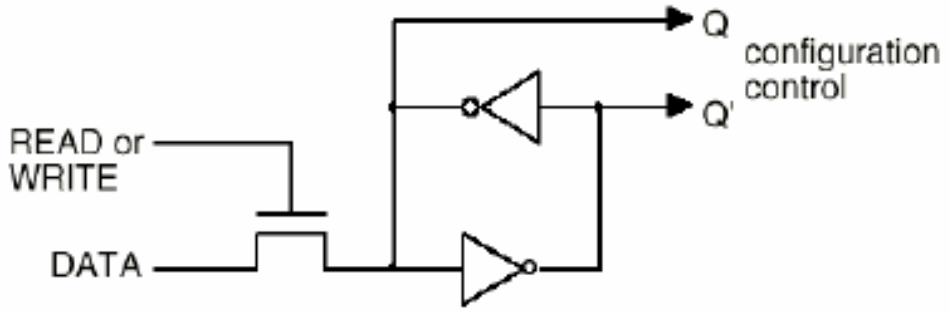
Requirement	Type
Operating System *(1)*	Windows XP Windows 2000 For Solaris or Linux support, purchase an Altera® software subscription

Note:

1. Beginning with version 5.1, the Quartus II Web Edition software no longer supports Windows NT.

Tecnologia SRAM

- Almost all **Field Programmable Gate Arrays** (FPGAs) are based on static RAMs.
- Static RAM cells are used for three purposes:
  - As lookup tables (LUTs) for implementing logic (as truth-table).
  - As embedded block RAM blocks (for buffer storage etc.).
  - As control to routing and configuration switches.
- ***Advantages:***
  - Easily changeable (even dynamic reconfiguration)
  - Good density
  - Track latest SRAM technology (moving even faster than technology for logic)
  - Flexible – no only good for FSM, also good for arithmetic circuits
- ***Disadvantages:***
  - Volatile
  - Generally high power



Field Programmable Gate Arrays

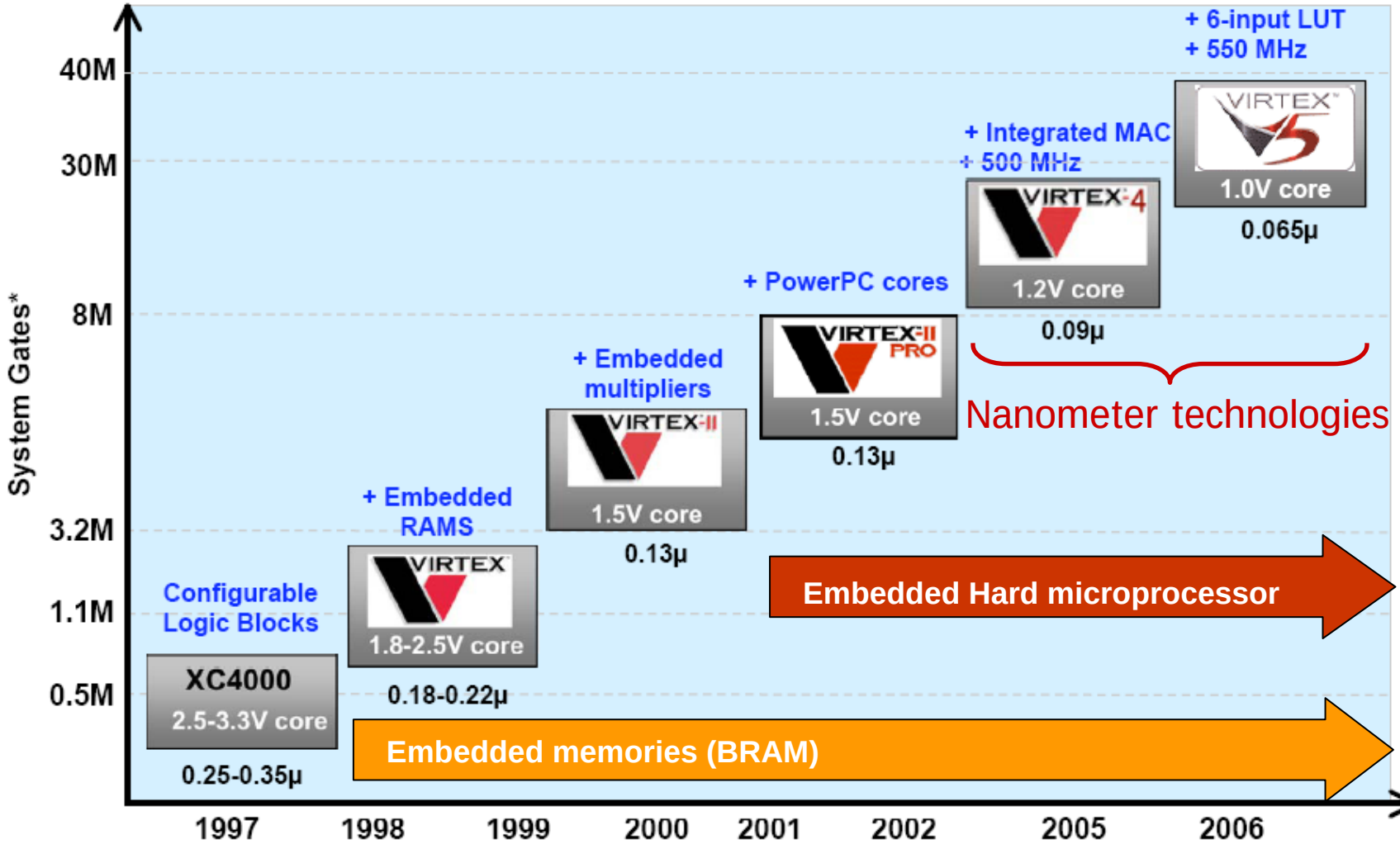
FPGAs comerciais

Xilinx

- A empresa Xilinx foi fundada em 1984 em San José (Califórnia, USA) e foi ela que introduziu o FPGA. Hoje em dia, esta empresa domina cerca de 50 % do mercado em FPGAs.

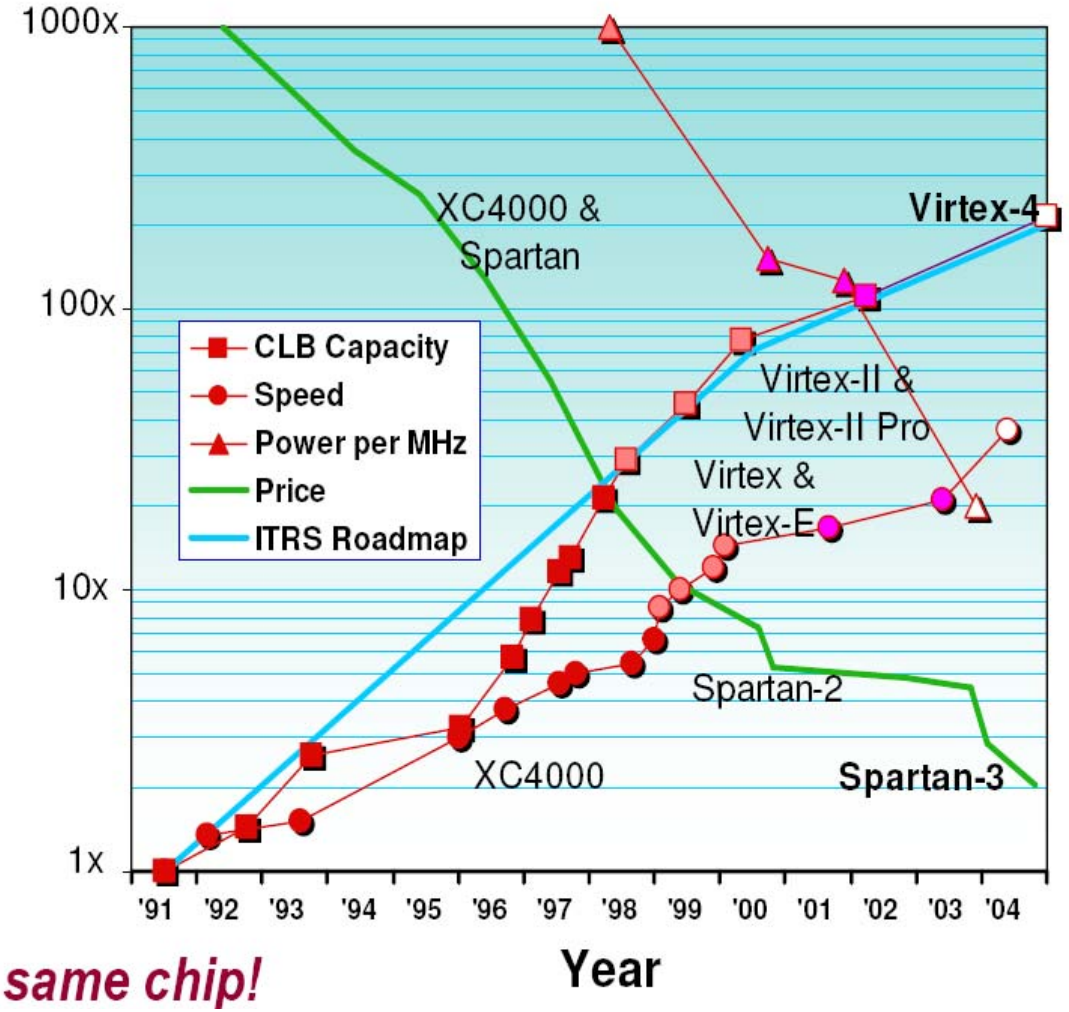
Família	Número de Portas	Característica	
XC2000	1,2K a 1,8K	-	
XC3000	2K a 9K	Low-power	
XC4000E	2K a 20K	Low-power	
XC4000XL/XLA	10K a 200K	High-density	
XC4000XV	75K a 500K	High-density	
XC5200	3 K a 23K	Low-power	
SPARTAN/XL	2K a 40K	Low-power	
VIRTEX	50K a 1M	High-density	220nm
SPARTAN-2	**1k a 15k**	**LOW COST**	
SPARTAN-3	**2k a 33k**	**LOW COST**	
VIRTEXII	**40k a 8 M**	**High density**	130nm
VIRTEXII-PRO	Power-PC inside		
VIRTEX4	13K a 200K	Low-power	90nm
VIRTEX4-FX	Power-PC inside		
VIRTEX5			65 nm

Technology Scaling in Xilinx FPGAs



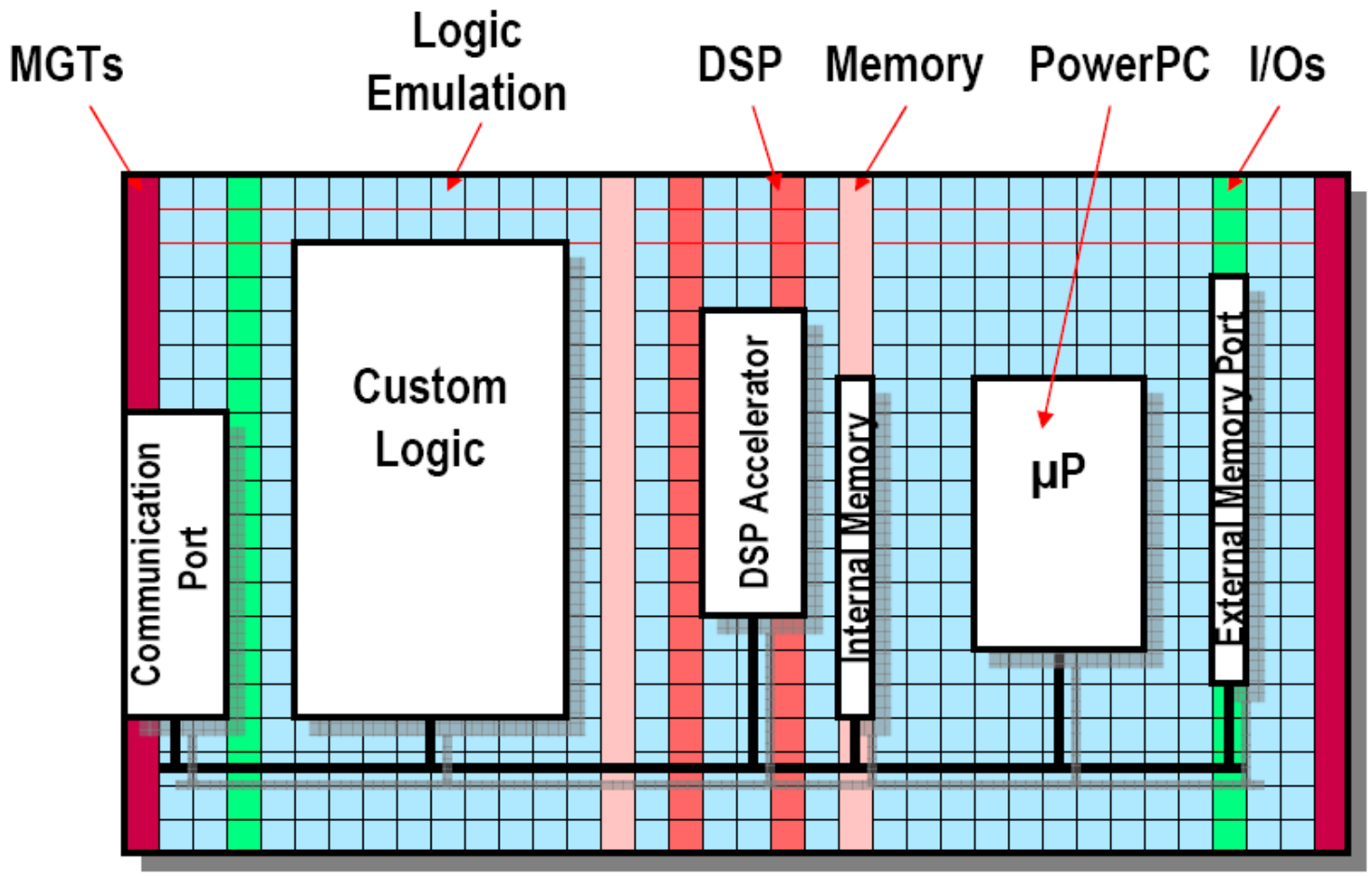
FPGAs and Moore's Law

- 200x More Logic
  - Plus Embedded IP
    - Memory
    - Microprocessor
    - DSP
    - Gigabit Serial I/O
- 40x Faster
- 50x Lower Power
- 500x Lower Cost

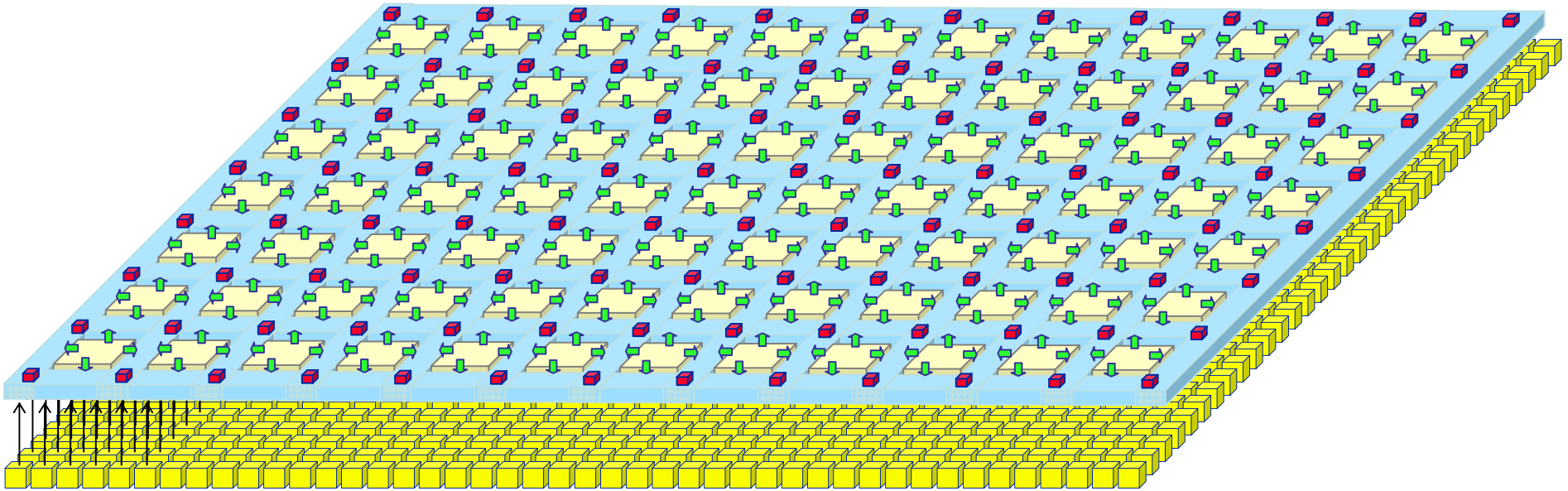


These are not all the same chip!

Targeting an "Age of Accumulation" FPGA

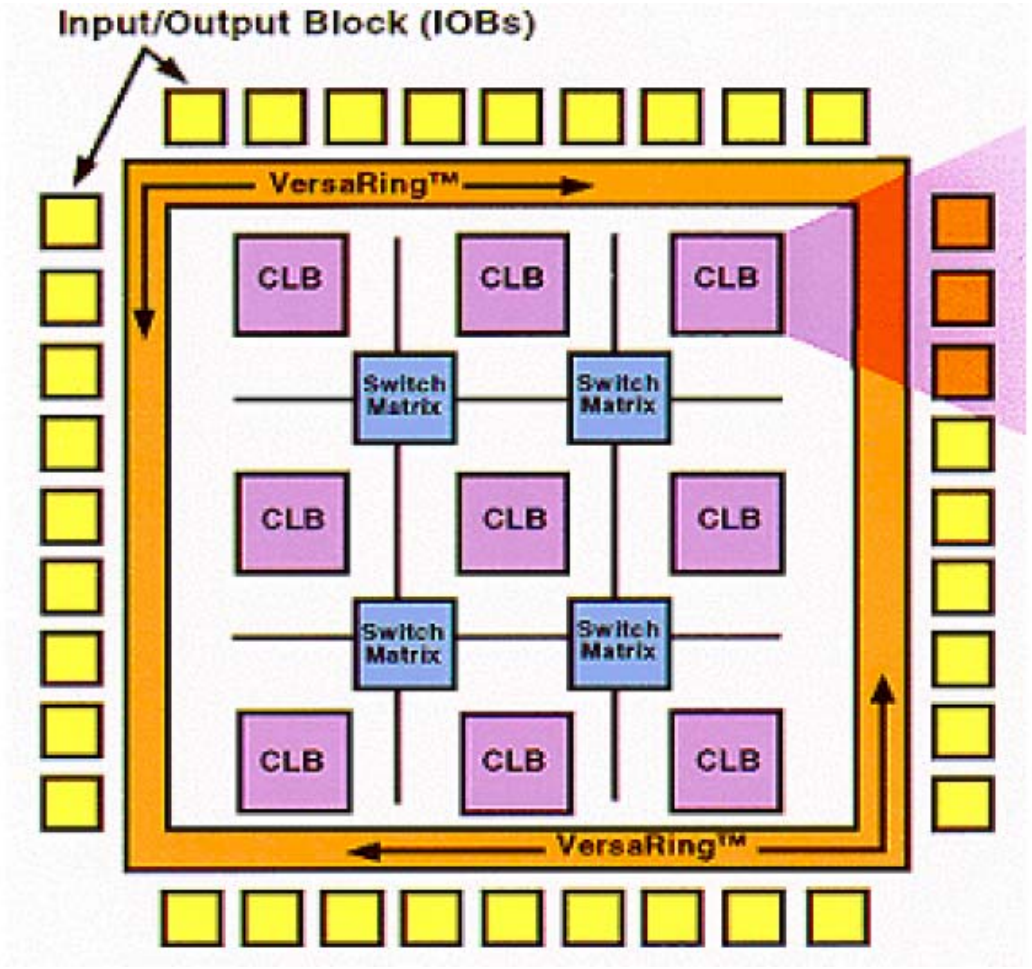


A informação é customizada por um vetor de bits chamado de BITSTREAM (set of SRAM bits)

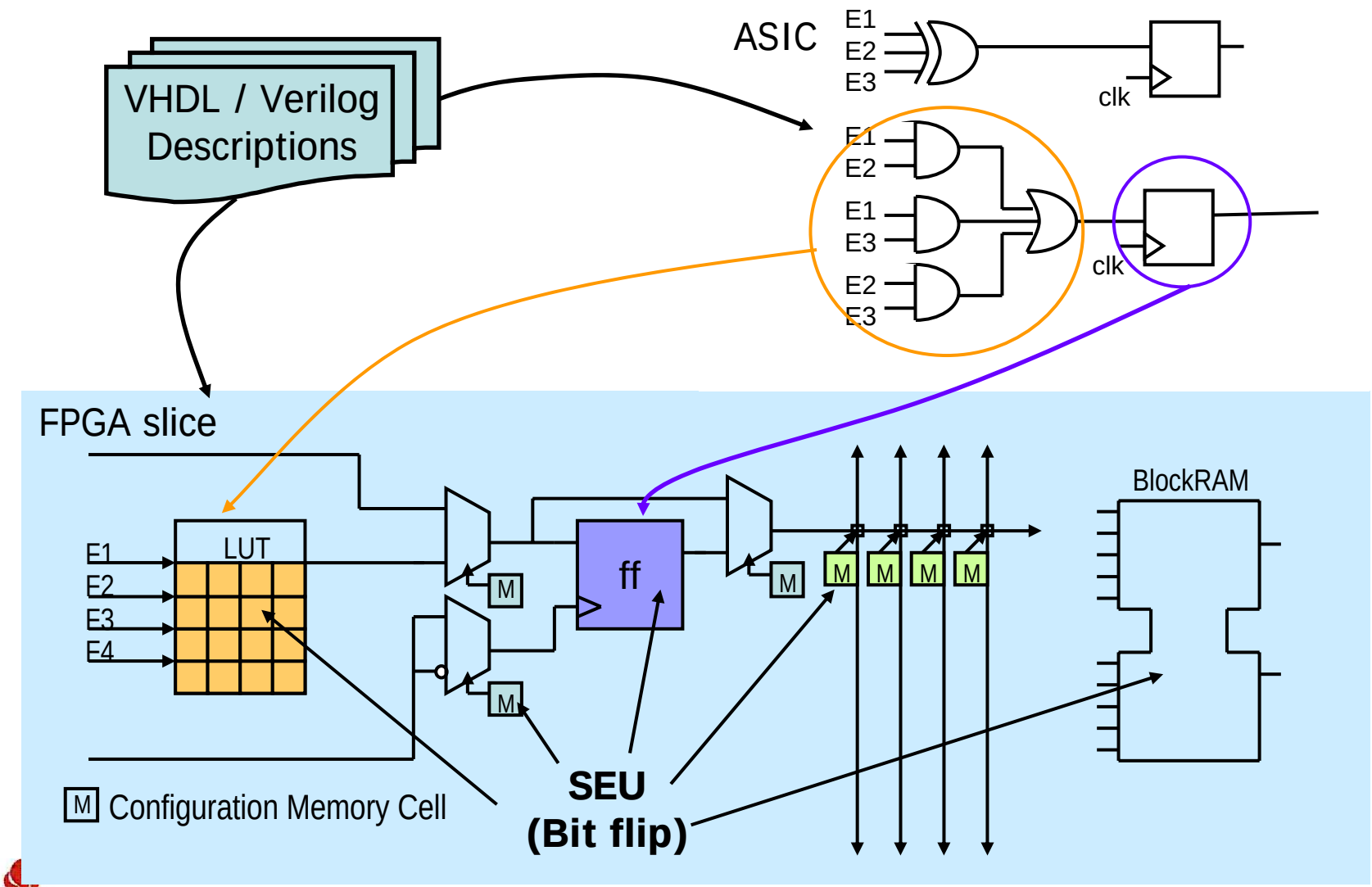


Xilinx XC4000

- Xilinx – first to introduce SRAM based FPGA using **Lookup Tables** (LUTs)
- Xilinx 4000 series contains four main building blocks:
- **Configurable Logic Block** (CLB)
- Switch Matrix
- VersaRing
- Input/Output Block



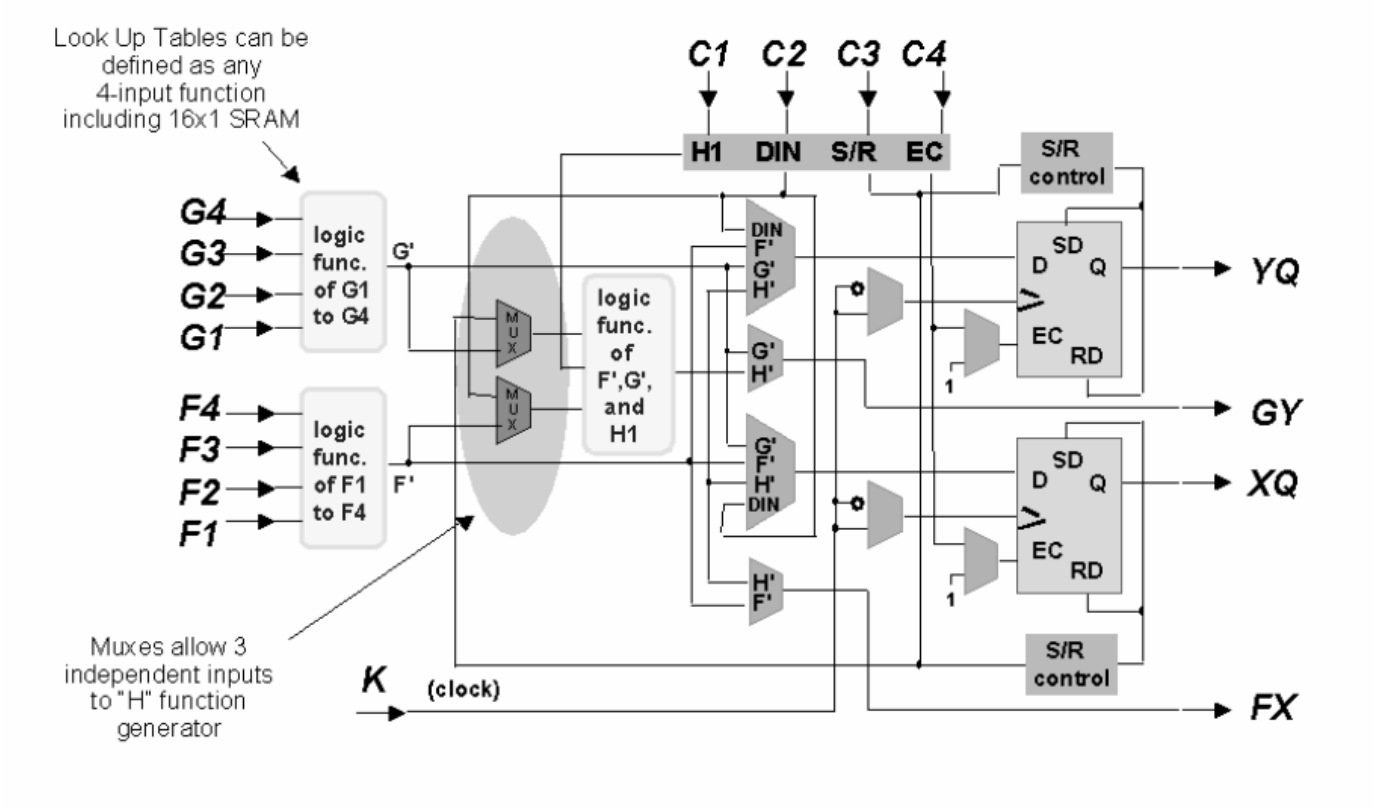
Logic Mapping in SRAM-based FPGAs



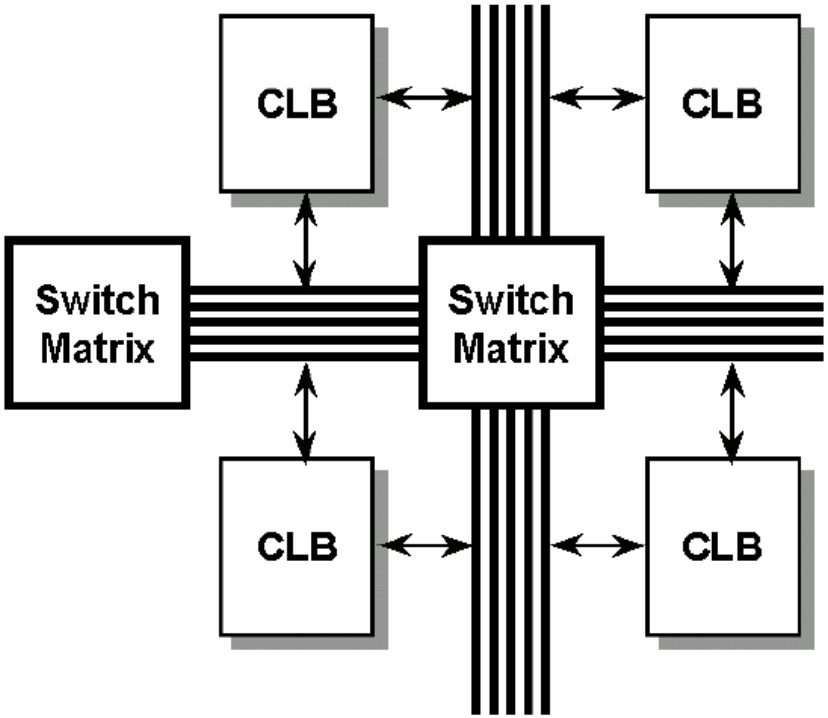
Xilinx XC4000

- Each CLB has two 4-input **Lookup Tables** (LUTs) and two reigsters.
- The two LUTs implement two independent logic functions F and G.
- The outputs F' and G' from the two LUTs inside each CLB can be combined to form a more complex function H.
- CLBs are linked together to form **carry** and **cascade chain** circuits (not shown in diagram)
- For the 4000E familiess, each CLB can be configured as synchronous RAM. Write address, data, and control are synchronized to write clock. This is called **distributed RAM**.
- Possible configurations are:
  - Two independent 16 x 1 RAMS
  - One 32 x 1 or 16 x 2 RAM
  - One 16 x 1 dual-port RAM (second port is read-only)

Logic Element: Basic XC4000



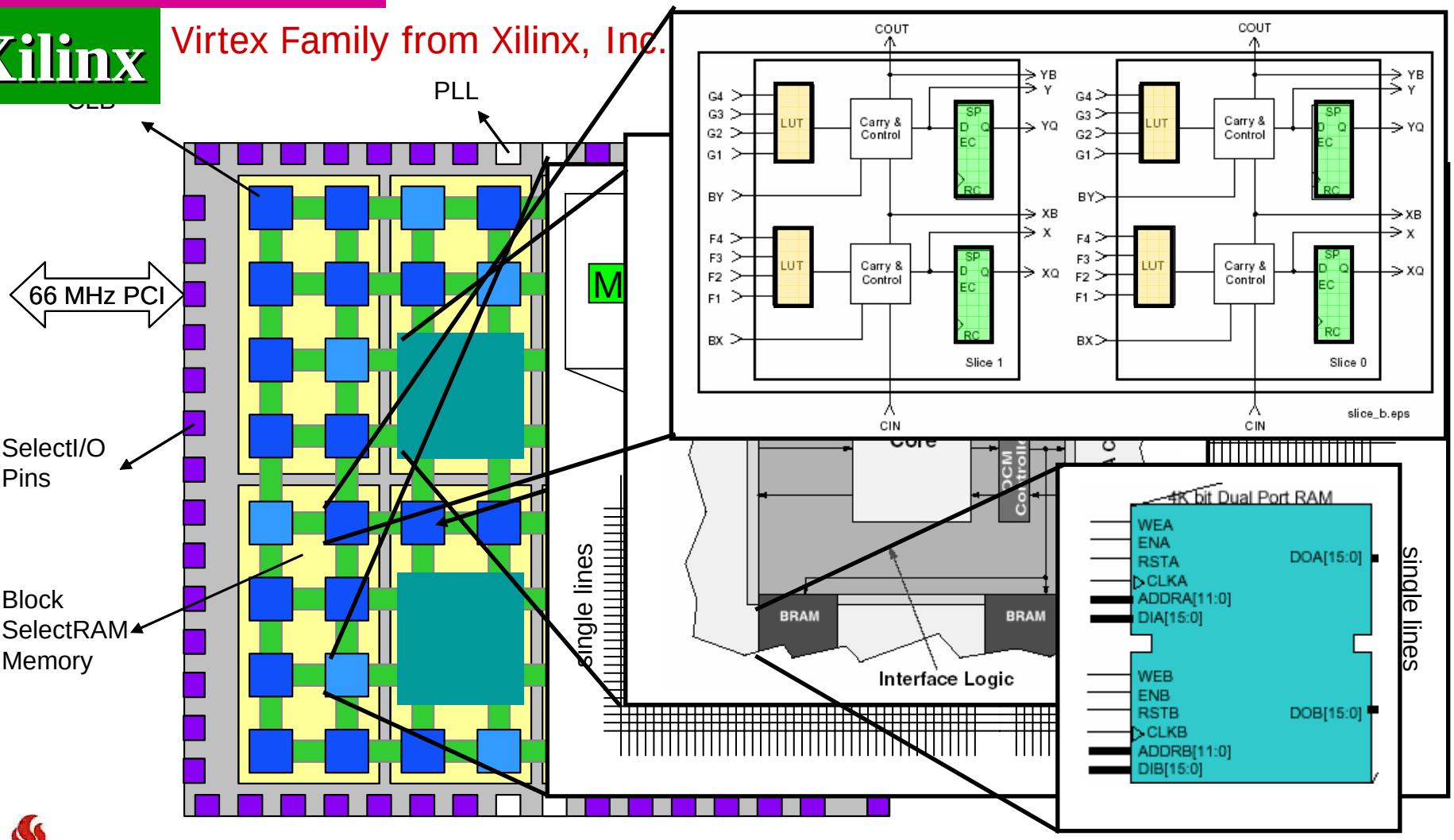
General Routing Matrix (GRM)



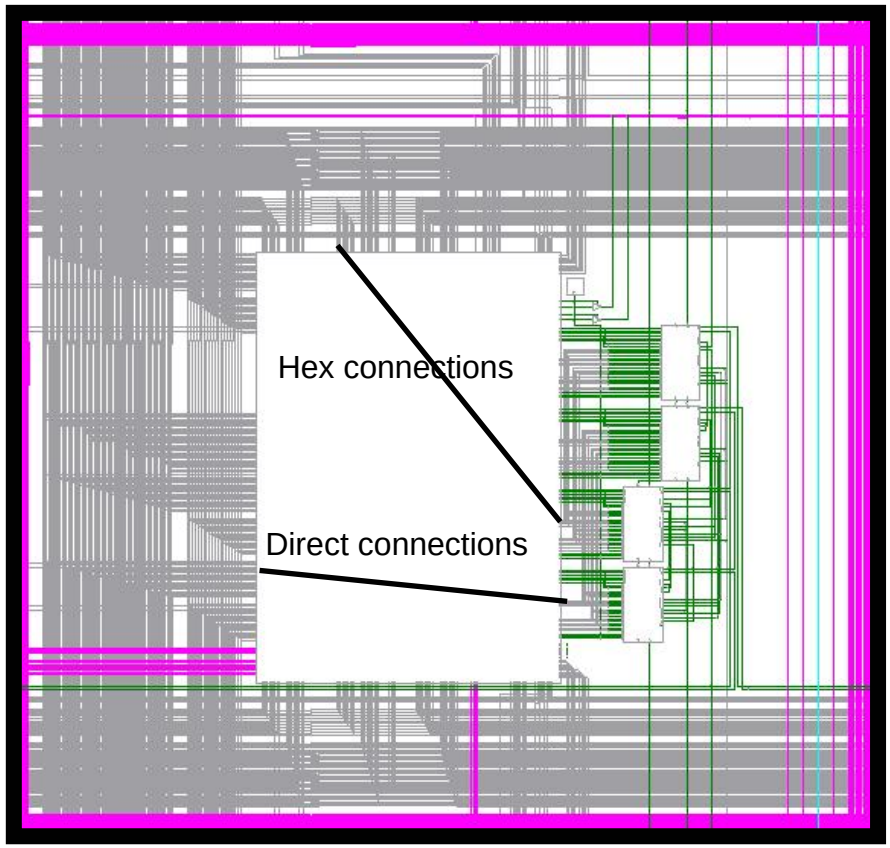
Field Programmable Gate Arrays

FPGAs comerciais

Xilinx Virtex Family from Xilinx, Inc.

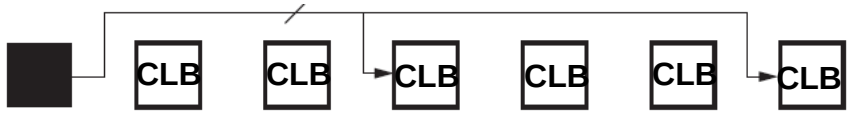


General Routing Matrix (GRM) - VirtexII

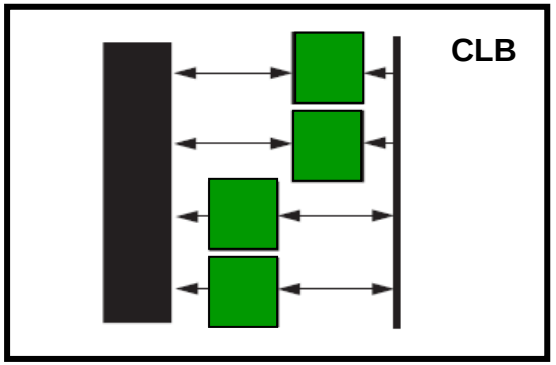


Long lines

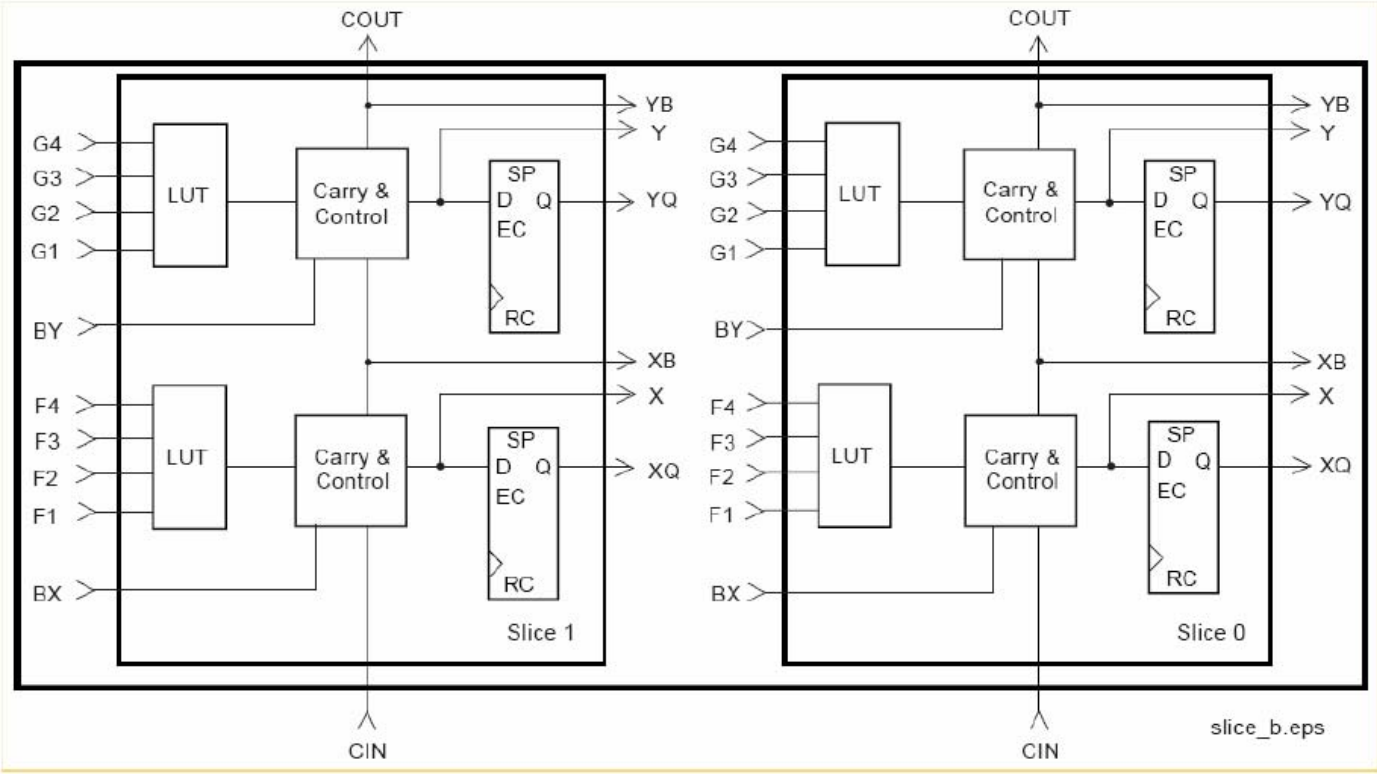
Hex lines



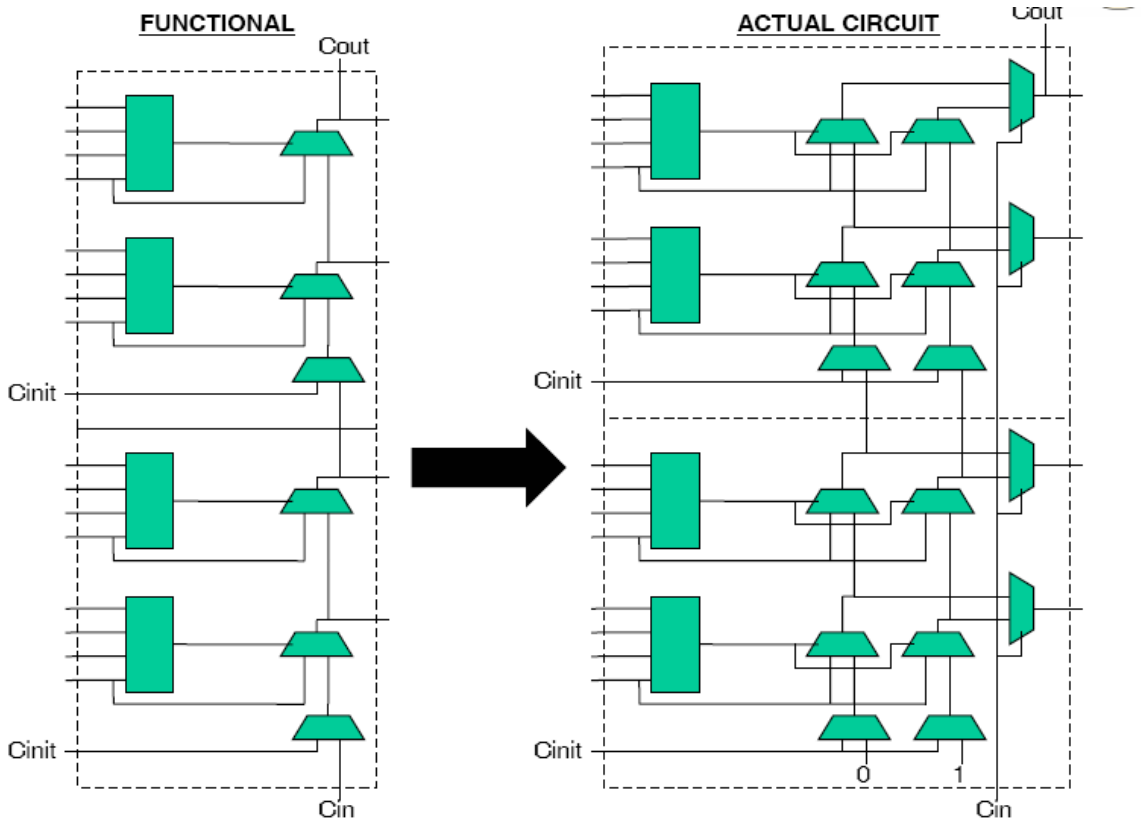
Fast connect



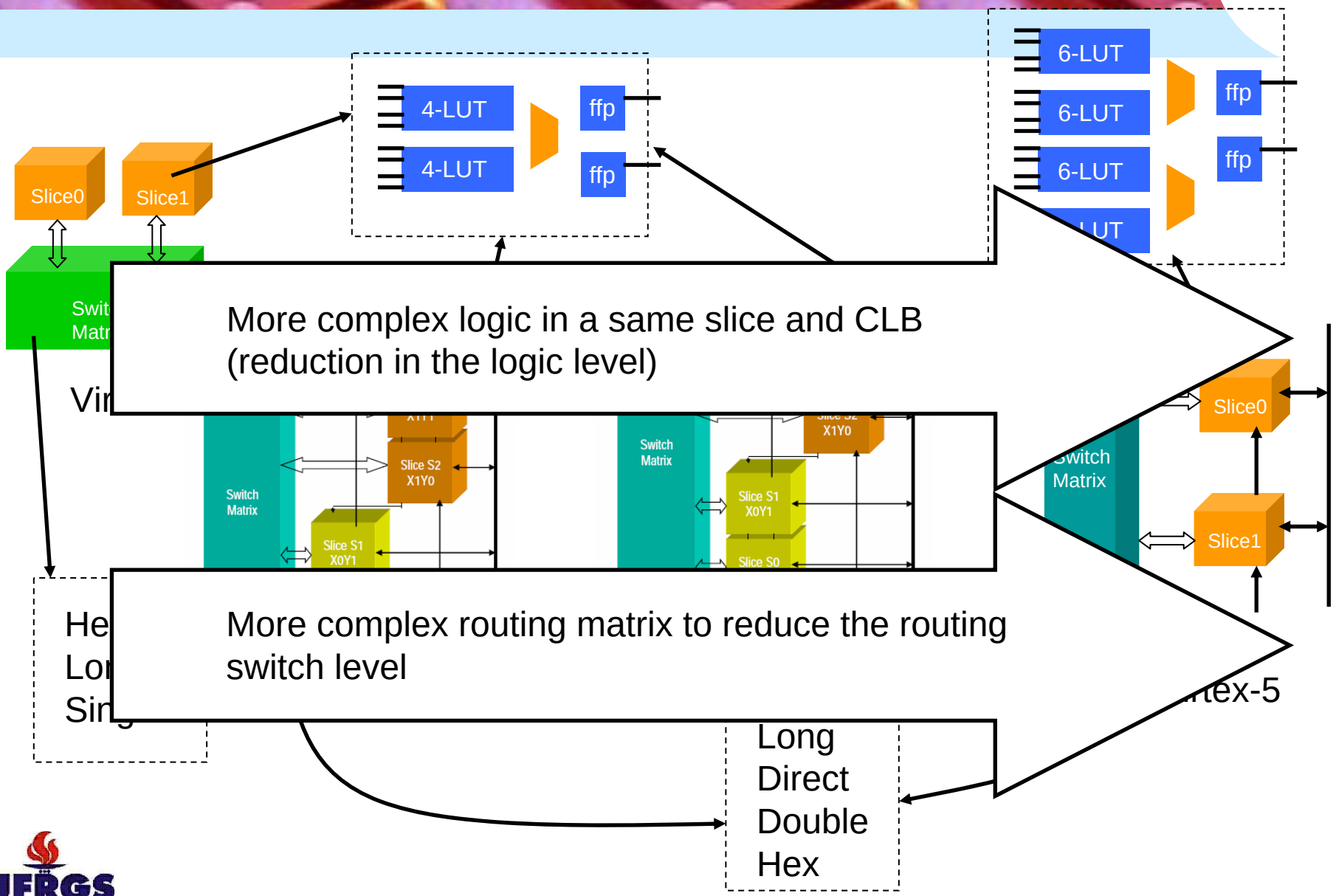
Xilinx VirtexII - CLB



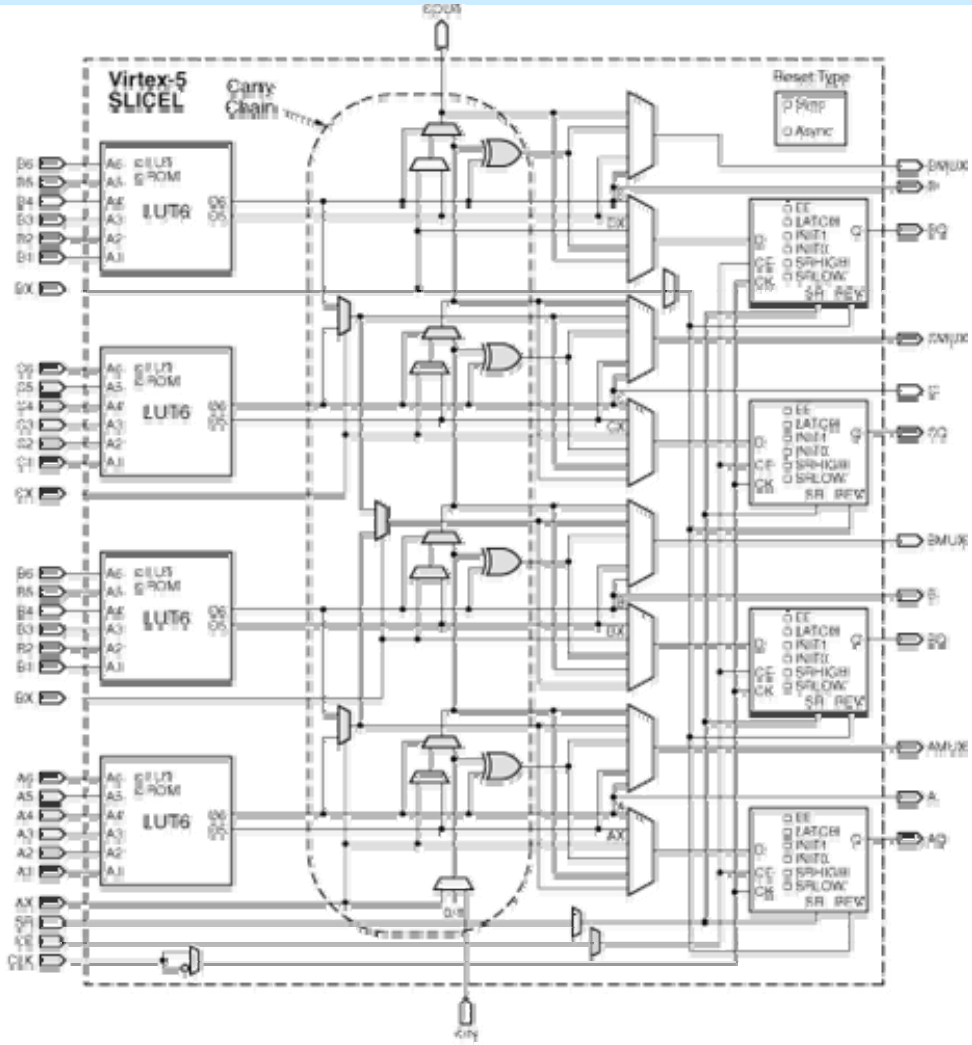
Virtex Carry Select



CLB Evolution

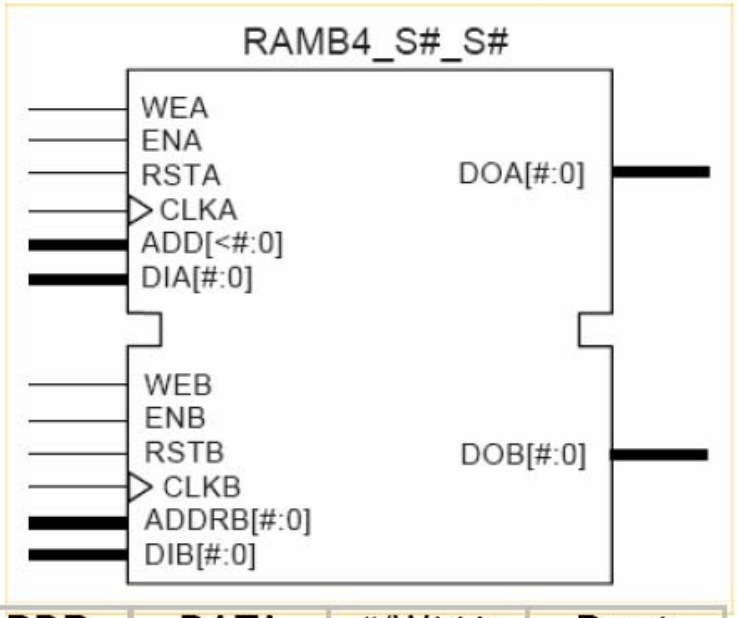


Virtex 5: CLB



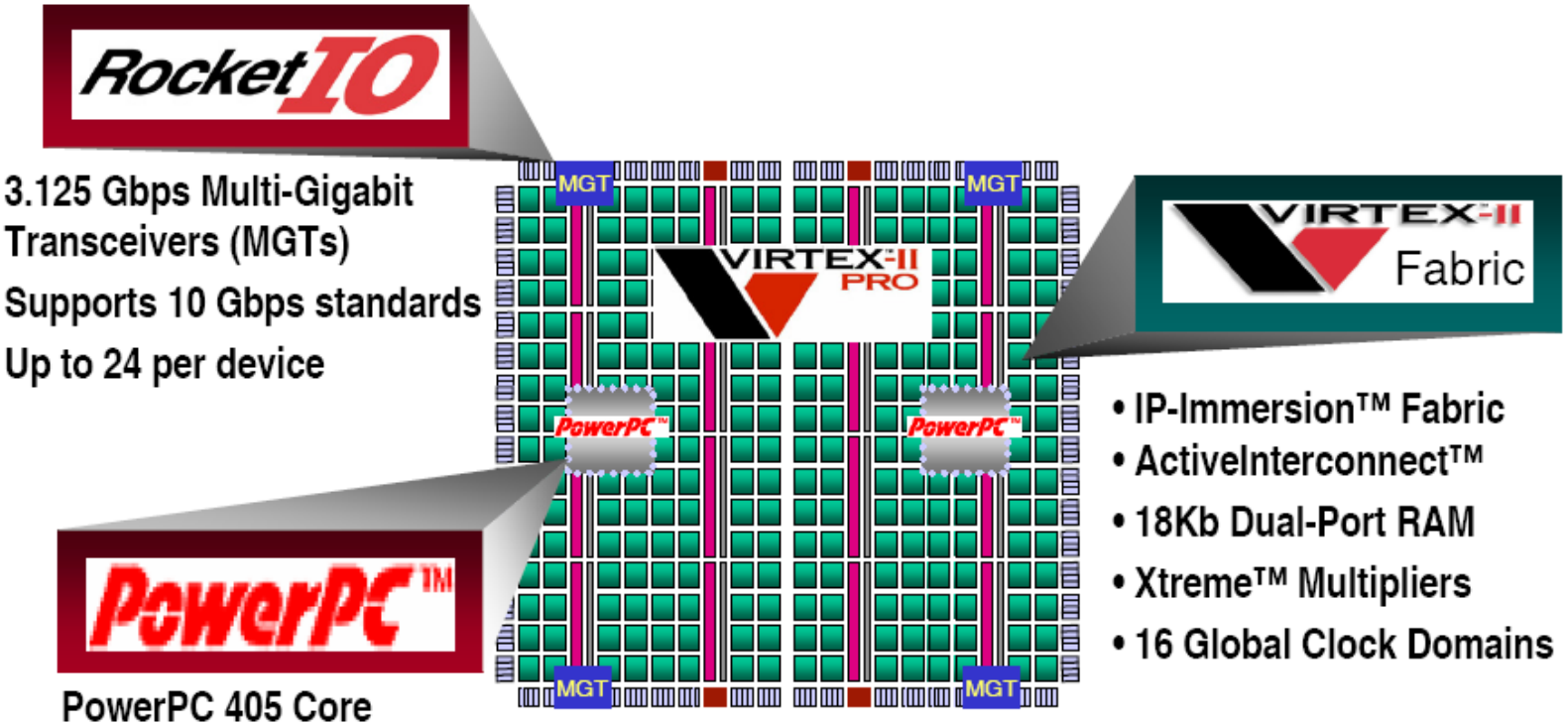
BRAM

- 2 Columns of Blocks on left and right
- 1 Block per 4 CLB rows.
- 4K bits of data
- Full Synchronous operation
  - No Asynchronous Read
- Ports can be configured to different widths
- Synchronous reset for Finite State Machine

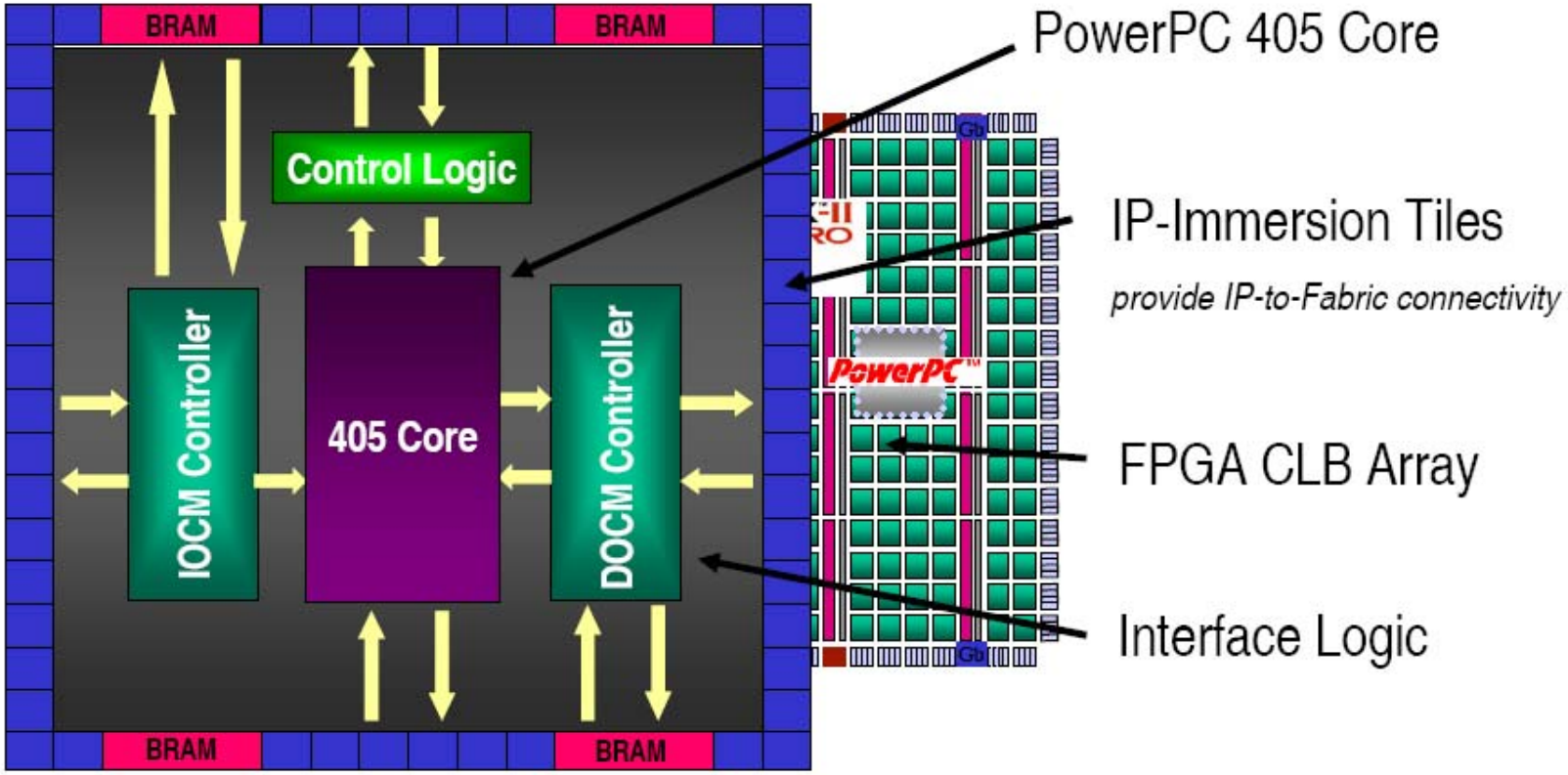


ADDR	DATA	#/Width	Depth
(11:0)	(0:0)	1	4096
(10:0)	(1:0)	2	2048
(9:0)	(3:0)	4	1024
(8:0)	(7:0)	8	512
(7:0)	(15:0)	16	256

VirtexII-Pro Platform



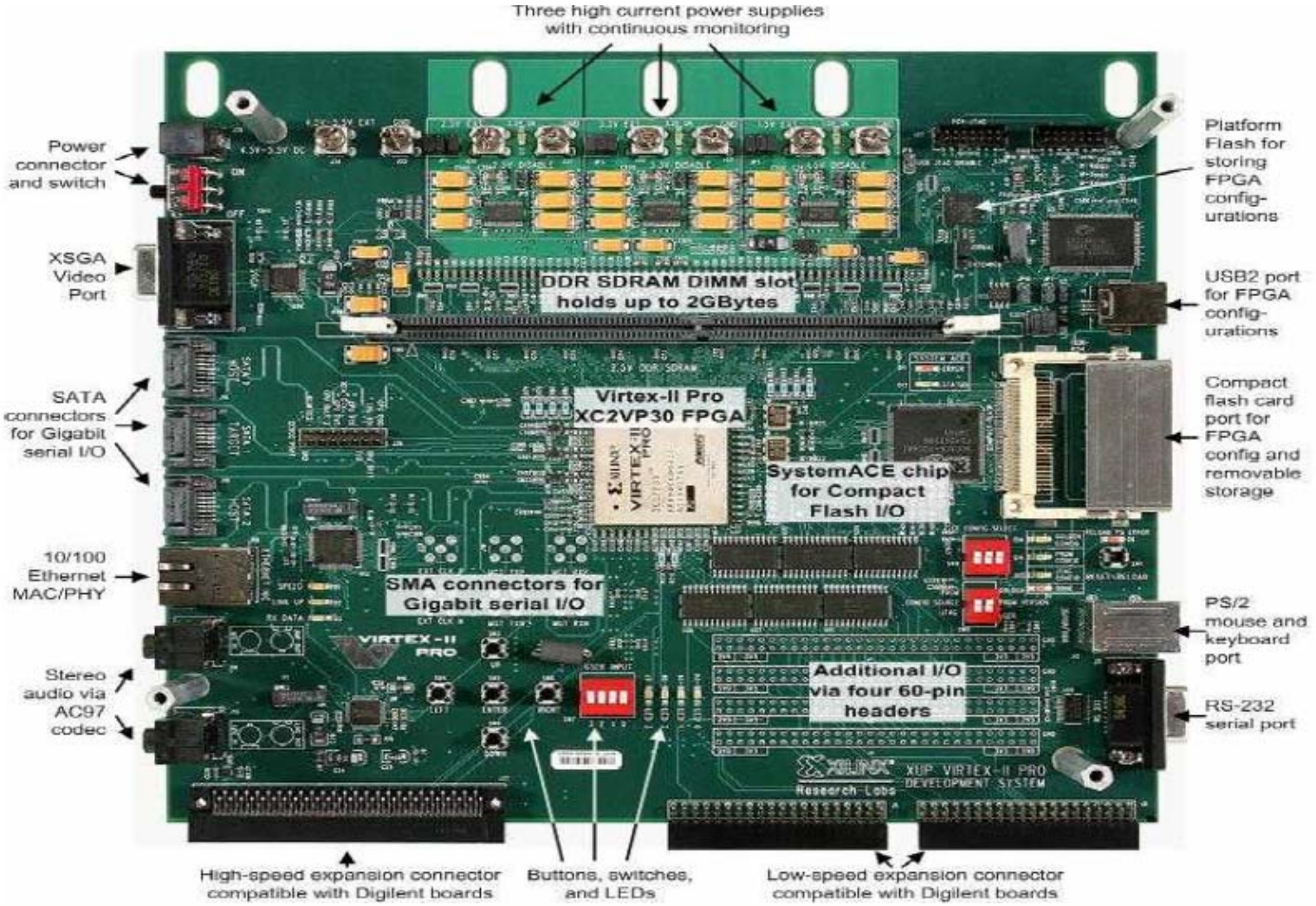
Embedded Processor



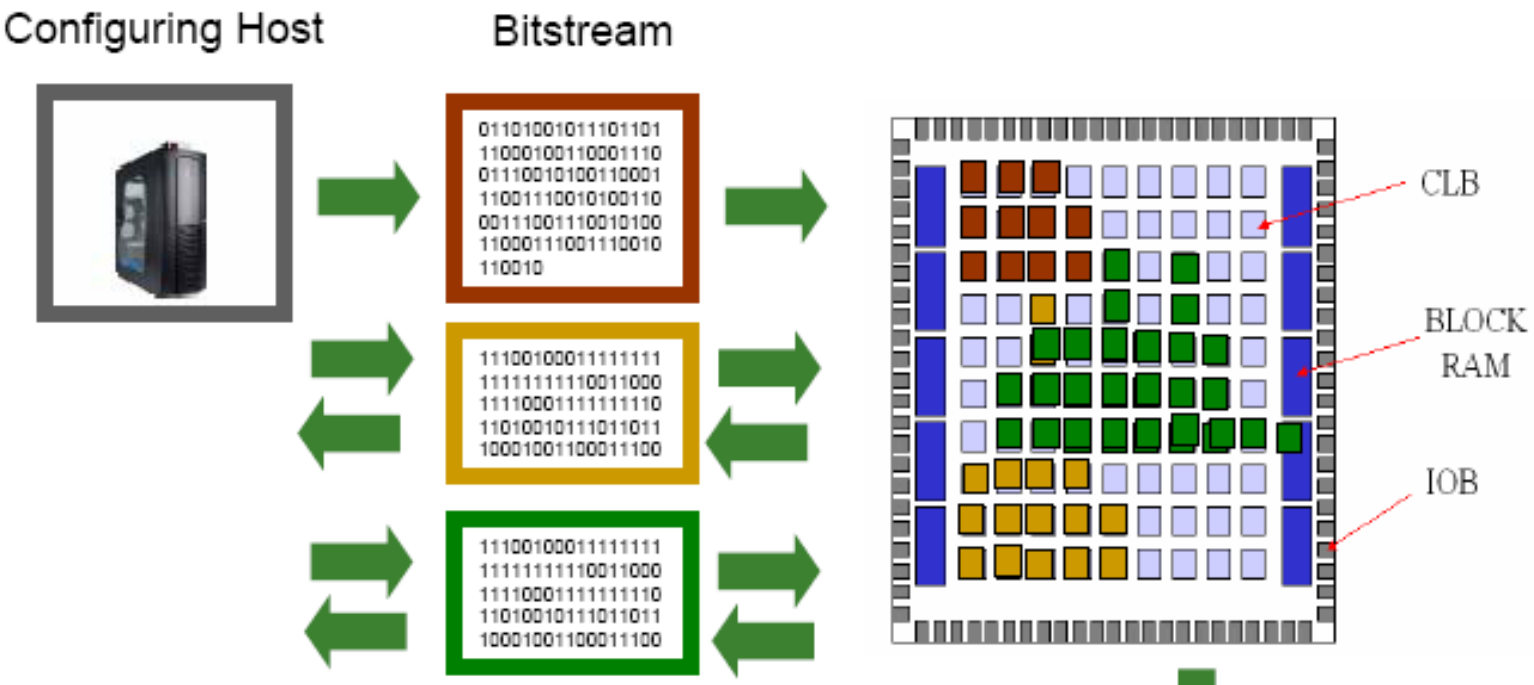
Microblaze (soft core)

- RISC
  - 32-bit ALU, 32-bit data bus, 32-bit instruction word, 32 x 32 General Purpose Register file
- Harvard architecture (i.e. separate program and data memory space)
- 3 stage pipeline (IF, OF, EX)
- Proprietary instruction set has been created for MicroBlaze.

VirtexII-Pro Board



Reconfigurable Computing

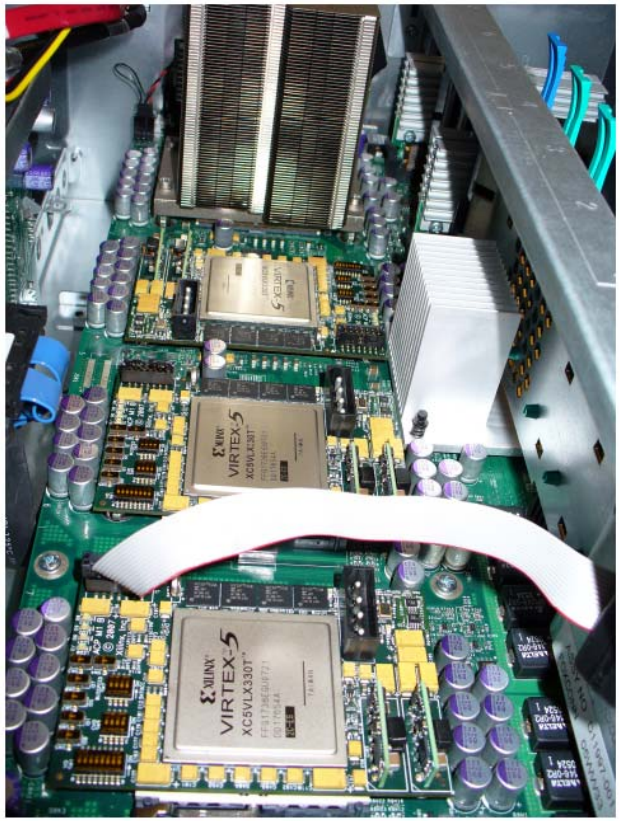


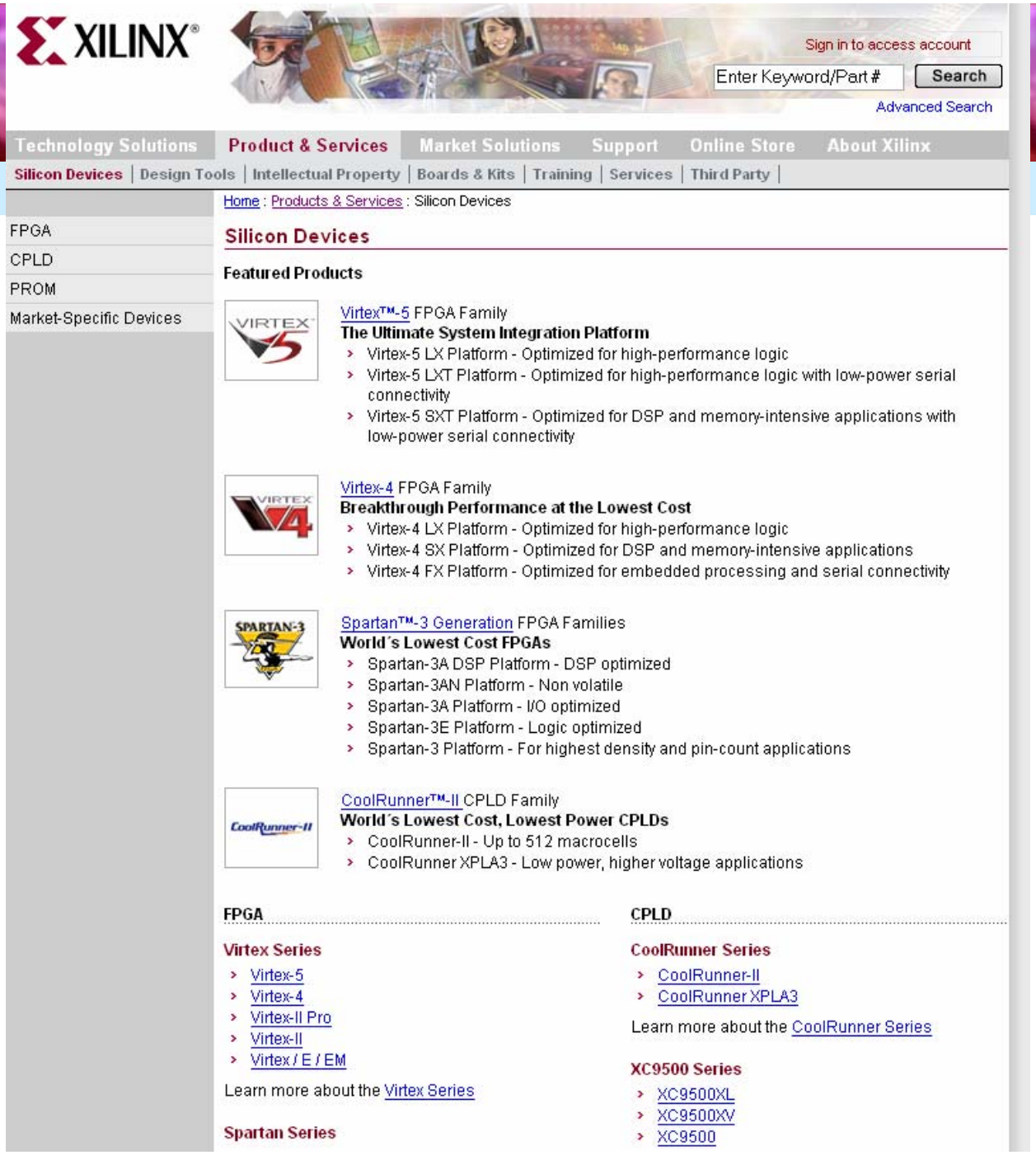
- We could also use a placement algorithm to possibly fit all requested HB into the FPGA

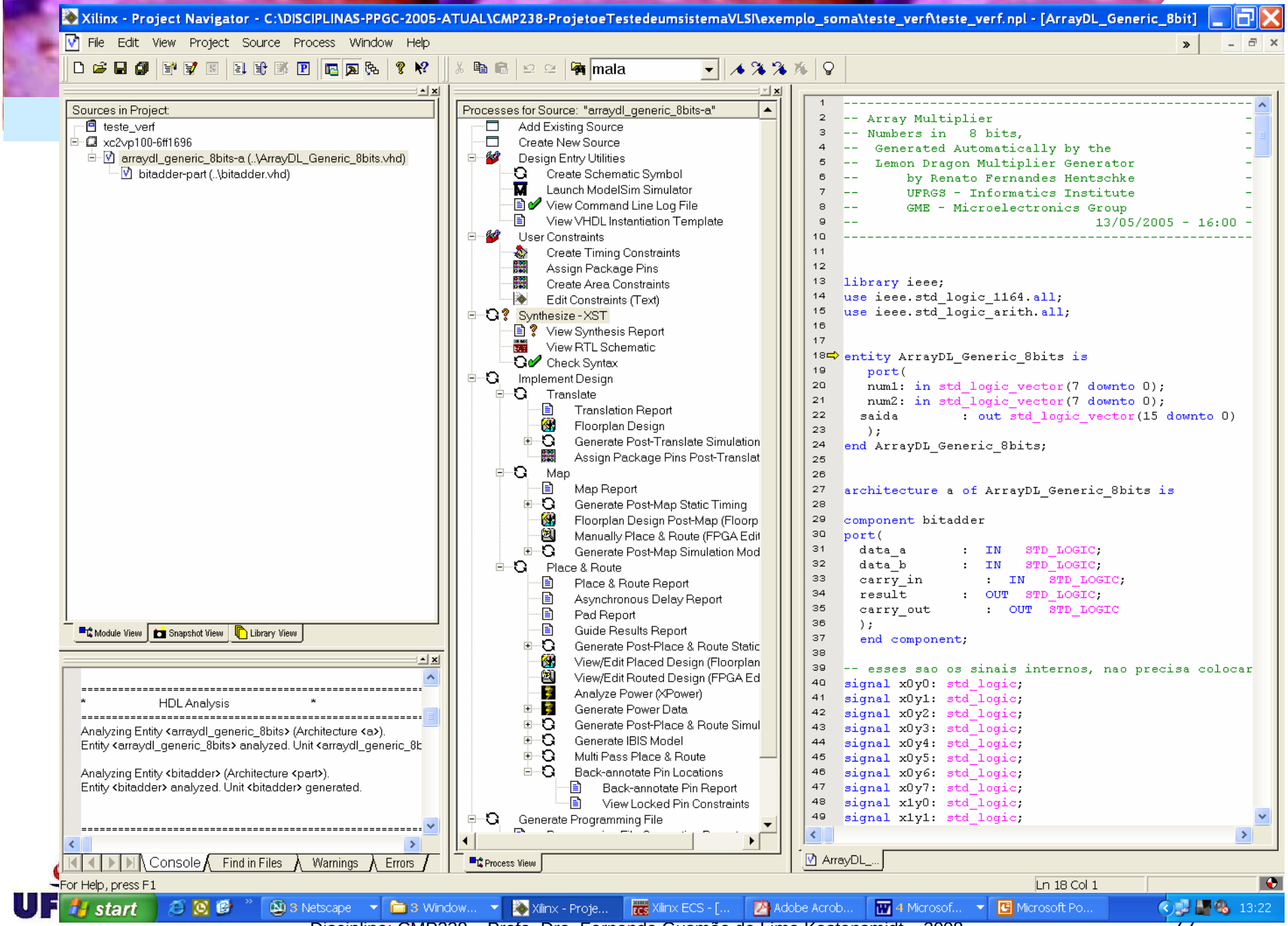
Presented by Trimberger (FPL, 2007)

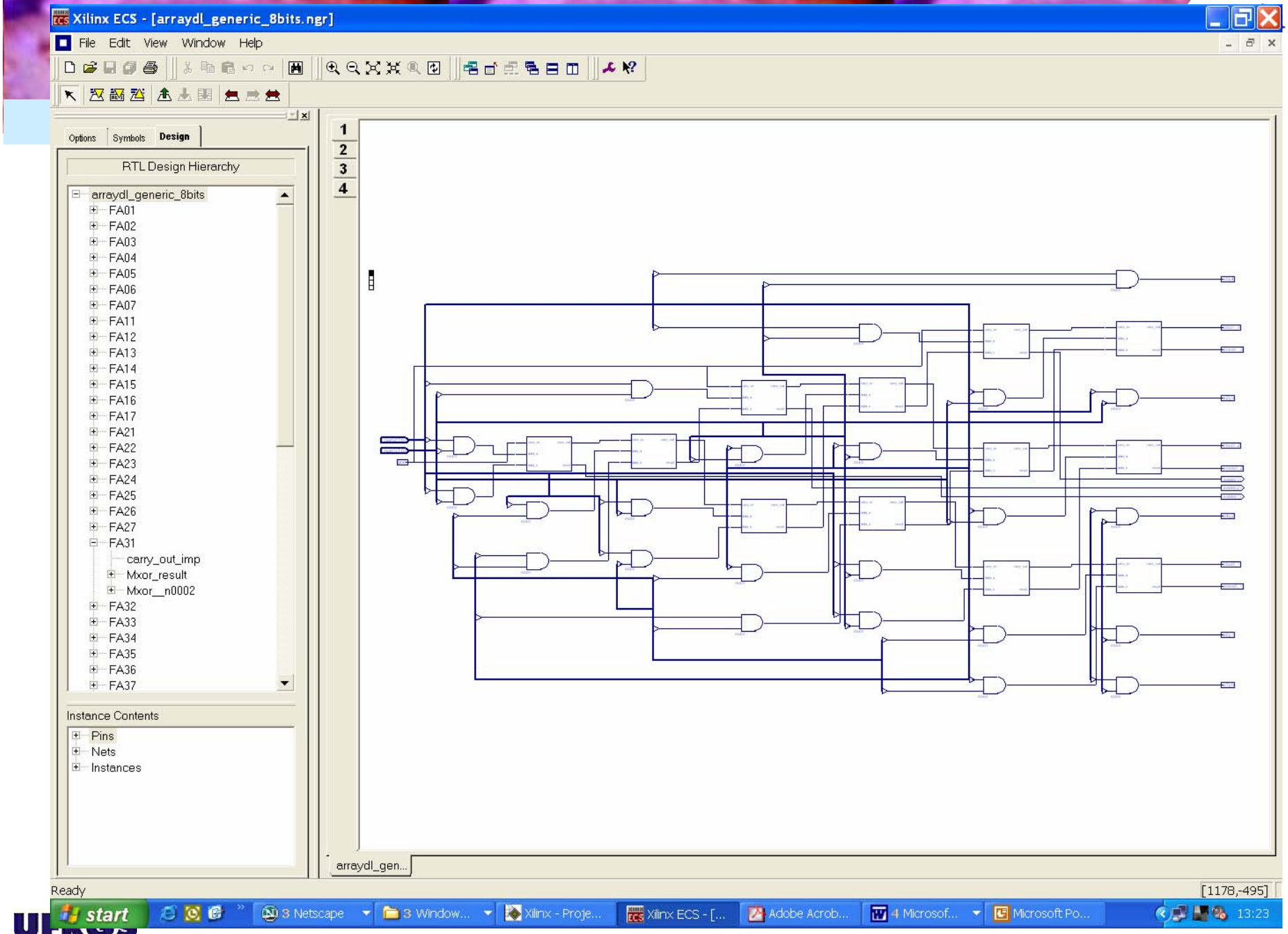
FPGA Computing

- Xilinx Virtex-5 FPGA in standard Intel Xeon server platform socket
  - Intel Front-Side Bus (FSB) connection
  - Increased performance in the available power budget
- FSB: An excellent interface for accelerated computing
  - Move to more BW (PCIe x8: 2.0 GB/s, FSB1066: 8.5 GB/s)
  - Much lower latency
  - Coherent system protocol




XILINX
Sign in to access account
Enter Keyword/Part #
Search
Advanced Search
Technology Solutions
Product & Services
Market Solutions
Support
Online Store
About Xilinx
Silicon Devices | Design Tools | Intellectual Property | Boards & Kits | Training | Services | Third Party |
Home : Products & Services : Silicon Devices
FPGA
CPLD
PROM
Market-Specific Devices
Silicon Devices
Featured Products
Virtex™-5 FPGA Family
The Ultimate System Integration Platform
Virtex-5 LX Platform - Optimized for high-performance logic
Virtex-5 LXT Platform - Optimized for high-performance logic with low-power serial connectivity
Virtex-5 SXT Platform - Optimized for DSP and memory-intensive applications with low-power serial connectivity
Virtex-4 FPGA Family
Breakthrough Performance at the Lowest Cost
Virtex-4 LX Platform - Optimized for high-performance logic
Virtex-4 SX Platform - Optimized for DSP and memory-intensive applications
Virtex-4 FX Platform - Optimized for embedded processing and serial connectivity
Spartan™-3 Generation FPGA Families
World's Lowest Cost FPGAs
Spartan-3A DSP Platform - DSP optimized
Spartan-3AN Platform - Non volatile
Spartan-3A Platform - I/O optimized
Spartan-3E Platform - Logic optimized
Spartan-3 Platform - For highest density and pin-count applications
CoolRunner™-II CPLD Family
World's Lowest Cost, Lowest Power CPLDs
CoolRunner-II - Up to 512 macrocells
CoolRunner XPLA3 - Low power, higher voltage applications
FPGA
Virtex Series
Virtex-5
Virtex-4
Virtex-II Pro
Virtex-II
Virtex / E / EM
Learn more about the Virtex Series
Spartan Series
CPLD
CoolRunner Series
CoolRunner-II
CoolRunner XPLA3
Learn more about the CoolRunner Series
XC9500 Series
XC9500XL
XC9500XV
XC9500


Xilinx - Project Navigator - C:\DISCIPLINAS-PPGC-2005-ATUAL\CMP238-ProjetoeTestedeumsistemaVLSI\exemplo_soma\teste_verf\teste_verf.npl - [ArrayDL_Generic_8bit]
File Edit View Project Source Process Window Help
mala
Sources in Project:
teste_verf
xc2vp100-6ff1696
arraydl_generic_8bits-a (..\ArrayDL_Generic_8bits.vhd)
bitadder-part (..\bitadder.vhd)
Module View
Snapshot View
Library View
Processes for Source: "arraydl_generic_8bits-a"
Add Existing Source
Create New Source
Design Entry Utilities
Create Schematic Symbol
Launch ModelSim Simulator
View Command Line Log File
View VHDL Instantiation Template
User Constraints
Create Timing Constraints
Assign Package Pins
Create Area Constraints
Edit Constraints (Text)
Synthesize - XST
View Synthesis Report
View RTL Schematic
Check Syntax
Implement Design
Translate
Translation Report
Floorplan Design
Generate Post-Translate Simulation
Assign Package Pins Post-Translat
Map
Map Report
Generate Post-Map Static Timing
Floorplan Design Post-Map (Floorp
Manually Place & Route (FPGA Edit
Generate Post-Map Simulation Mod
Place & Route
Place & Route Report
Asynchronous Delay Report
Pad Report
Guide Results Report
Generate Post-Place & Route Static
View/Edit Placed Design (Floorplan
View/Edit Routed Design (FPGA Ed
Analyze Power (XPower)
Generate Power Data
Generate Post-Place & Route Simul
Generate IBIS Model
Multi Pass Place & Route
Back-annotate Pin Locations
Back-annotate Pin Report
View Locked Pin Constraints
Generate Programming File
Process View
-- Array Multiplier
-- Numbers in 8 bits,
-- Generated Automatically by the
-- Lemon Dragon Multiplier Generator
-- by Renato Fernandes Hentschke
-- UFRGS - Informatics Institute
-- GME - Microelectronics Group
-- 13/05/2005 - 16:00
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity ArrayDL_Generic_8bits is
port(
num1: in std_logic_vector(7 downto 0);
num2: in std_logic_vector(7 downto 0);
saida : out std_logic_vector(15 downto 0)
);
end ArrayDL_Generic_8bits;
architecture a of ArrayDL_Generic_8bits is
component bitadder
port(
data_a : IN STD_LOGIC;
data_b : IN STD_LOGIC;
carry_in : IN STD_LOGIC;
result : OUT STD_LOGIC;
carry_out : OUT STD_LOGIC
);
end component;
-- esses sao os sinais internos, nao precisa colocar
signal x0y0: std_logic;
signal x0y1: std_logic;
signal x0y2: std_logic;
signal x0y3: std_logic;
signal x0y4: std_logic;
signal x0y5: std_logic;
signal x0y6: std_logic;
signal x0y7: std_logic;
signal x1y0: std_logic;
signal x1y1: std_logic;
ArrayDL_...
HDL Analysis
Analyzing Entity <arraydl_generic_8bits> (Architecture <a>).
Entity <arraydl_generic_8bits> analyzed. Unit <arraydl_generic_8b
Analyzing Entity <bitadder> (Architecture <part>).
Entity <bitadder> analyzed. Unit <bitadder> generated.
Console
Find in Files
Warnings
Errors
For Help, press F1
Ln 18 Col 1
start
3 Netscape
3 Window...
Xilinx - Proje...
Xilinx ECS - [...
Adobe Acrob...
4 Microsof...
Microsoft Po...
13:22


Xilinx ECS - [arraydl_generic_8bits.ngr]
File Edit View Window Help
Options Symbols Design
RTL Design Hierarchy
arraydl_generic_8bits
FA01
FA02
FA03
FA04
FA05
FA06
FA07
FA11
FA12
FA13
FA14
FA15
FA16
FA17
FA21
FA22
FA23
FA24
FA25
FA26
FA27
FA31
carry_out_imp
Mxor_result
Mxor__n0002
FA32
FA33
FA34
FA35
FA36
FA37
Instance Contents
Pins
Nets
Instances
1
2
3
4
arraydl_gen...
Ready
[1178,-495]
start
3 Netscape
3 Window...
Xilinx - Proje...
Xilinx ECS - [...
Adobe Acrob...
4 Microsof...
Microsoft Po...
13:23