

Unité : CSN

# Evolution des circuits logiques programmables PLDs

heig-**vd**

HAUTE ÉCOLE  
D'INGÉNIERIE ET DE GESTION  
DU CANTON DE VAUD

[www.heig-\*\*vd\*\*.ch](http://www.heig-<b>vd</b>.ch)

Etienne Messerli

février 2021

**RED**S



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

# Contenu de la présentation ...

---

- Historique des PLDs
- Types de PLDs
- Historique des langages HDL
- Evolution des PLDs
- Quelques caractéristiques des PLDs



# Histoire des PLDs

---

- Années 70: 1<sup>er</sup> PLDs
  - PLD UV: GE (General Electric) *UV : Ultraviolet*
  - PAL: MMI (Monolithic Memories Inc.) 1978, AMD, Signetics/Intersil, National, TI, ...
  - Programmateurs: Data IO, ...
- Années 80:
  - Arrivée de nouvelles familles de PLDs
  - Début de leur utilisation pour de la *glue logic*
  - GAL: Lattice Semiconductor (1985)
  - CPLD: Altera (1984)
  - FPGA: début en 1985 (*voir dia suivant*)

# Histoire des PLDs

---

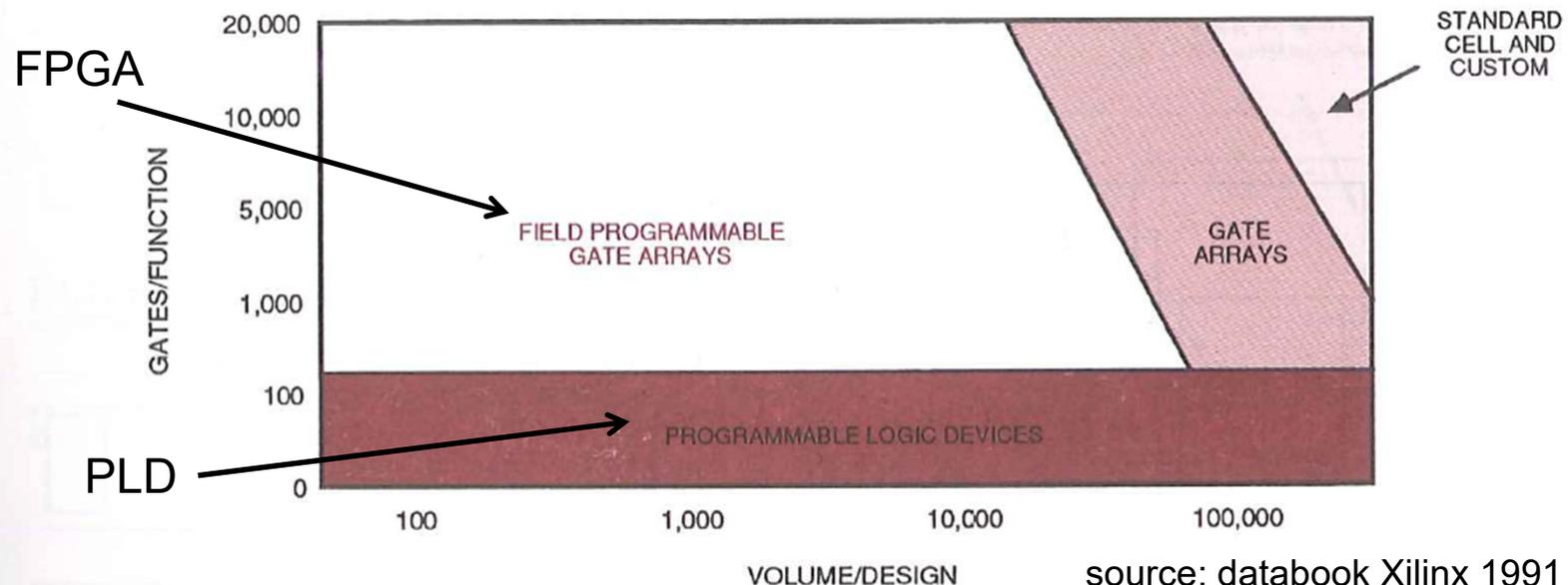
- Années 80: premiers FPGA
  - FPGA : Field-programmable gate array
    - réseau de portes programmable par l'utilisateur
  - Xilinx first commercially FPGA in 1985 : XC2064
    - 64 LEs, ~1000 gates, max 58 I/Os,
    - fichier de configuration: 12'038 bits !

# Histoire des PLDs

- Databook Xilinx 1991: FPGA alternative aux ASIC
  - Comparaison PLD/FPGA avec ASIC pour une densité max de 20'000 gates

## ASIC ALTERNATIVES

*Application Specific ICs are the best solution for most logic functions.  
The best ASIC solution depends on density requirements and production volumes.*



# Types de PLD

## "Programmable Logic Device"

---

- SPLD « Simple PLD »
  - PAL « Programmable Array Logic » †
  - GAL « Generic Array Logic » †
- CPLD  
« Complex Programmable Logic Devices »
- FPGA « Field Programmable Gate Arrays »
- HardCopy « hard PLDs », routage fixé lors fabrication
  - migration des PLDs sur un ASIC proposé par les fabricants de PLDs
- ASIC « Application Specific Integrated Circuits »

# Circuit SPLD

---

## SPLD, *Simple Programmable Logic Device*

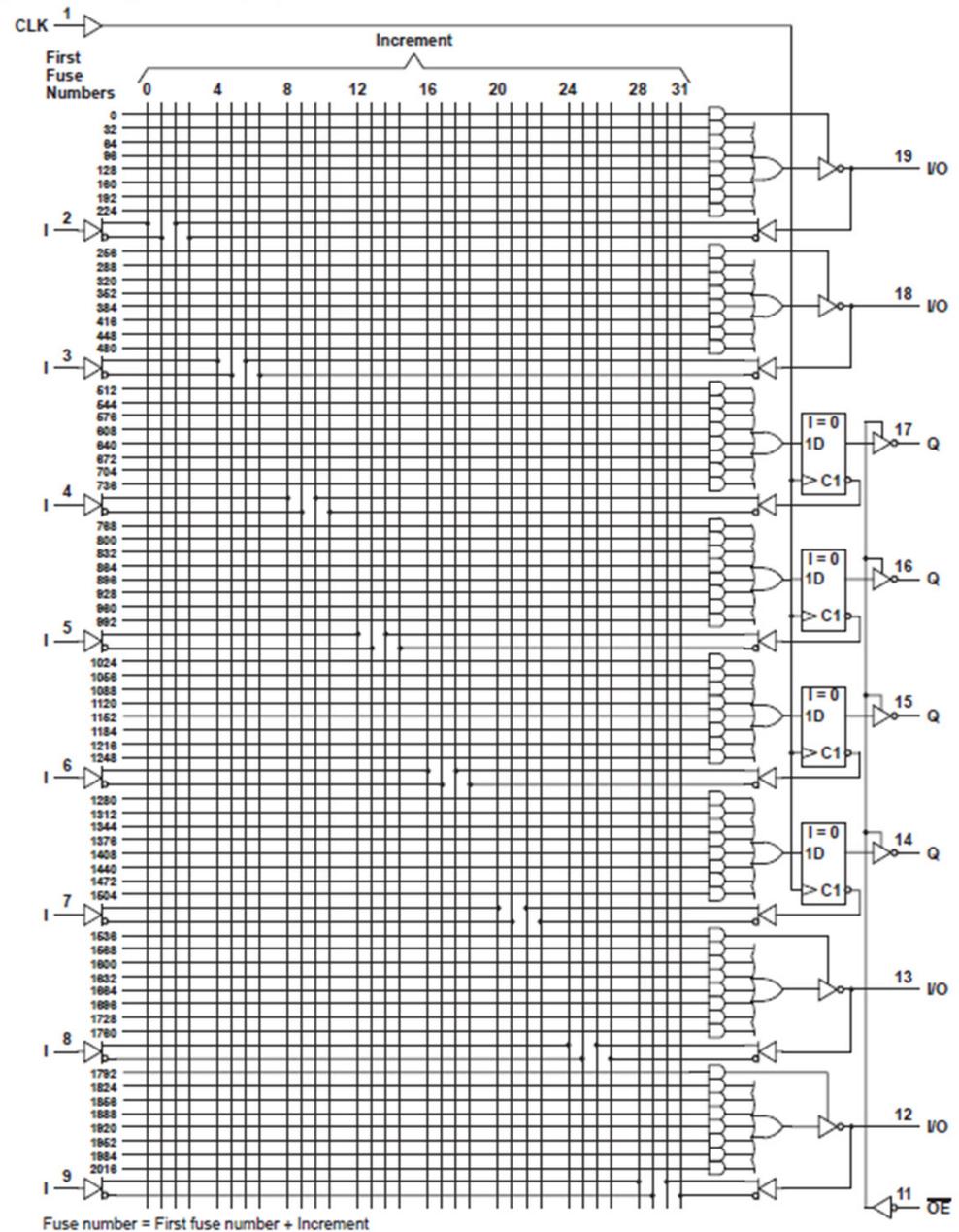
- Circuit programmable élémentaire composé d'entrée, d'une matrice ET programmable, d'une matrice OU fixe et d'un bloc de sortie comprenant souvent un registre (flip-flop) et avec I/O
- Types standards entre tous les fabricants!
- PAL: *Programmable Array Logic* †
- GAL: *Generic Array Logic* †
  - Tous les blocs de sorties sont identiques

# PAL16R4

## Caractéristiques:

- 8 In
- 4 In/Out combinatoires
- 4 Out avec registres
- 8 ou 7 produits avec 16 signaux
- signal OE (externe)
- 5 [V]
- 25 MHz
- DIL-20 pins

## logic diagram (positive logic)

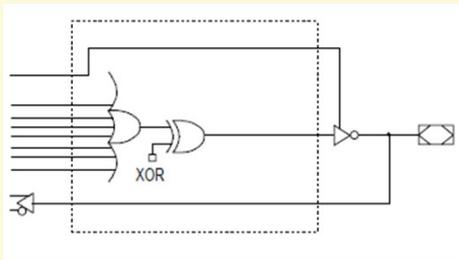


# GAL16V8

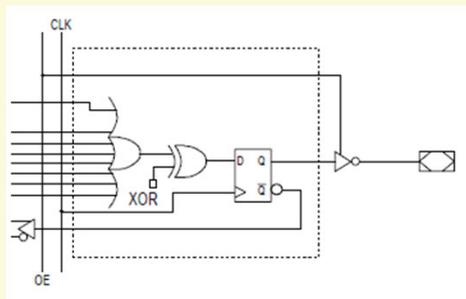
GAL16V8 Lattice: 1996

- 8 In, 8 IO
- OLMC configurable!
- 50 MHz, 5 [V]

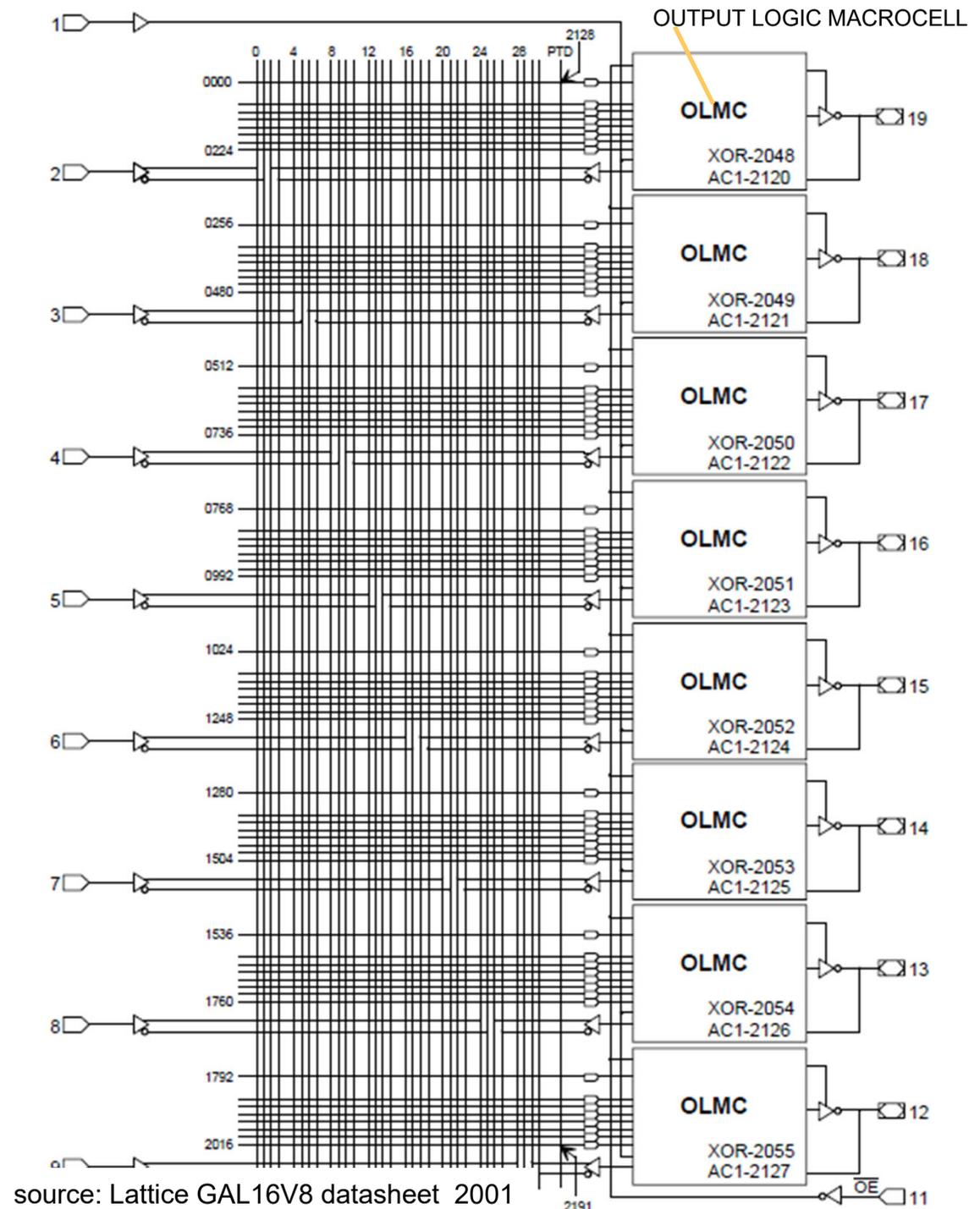
Bloc logique mode combinatoire:



Bloc logique mode registre:



E. Messerli (HES-SO / HEIG-VD / REDS), 2021



source: Lattice GAL16V8 datasheet 2001

# Circuit CPLD ...

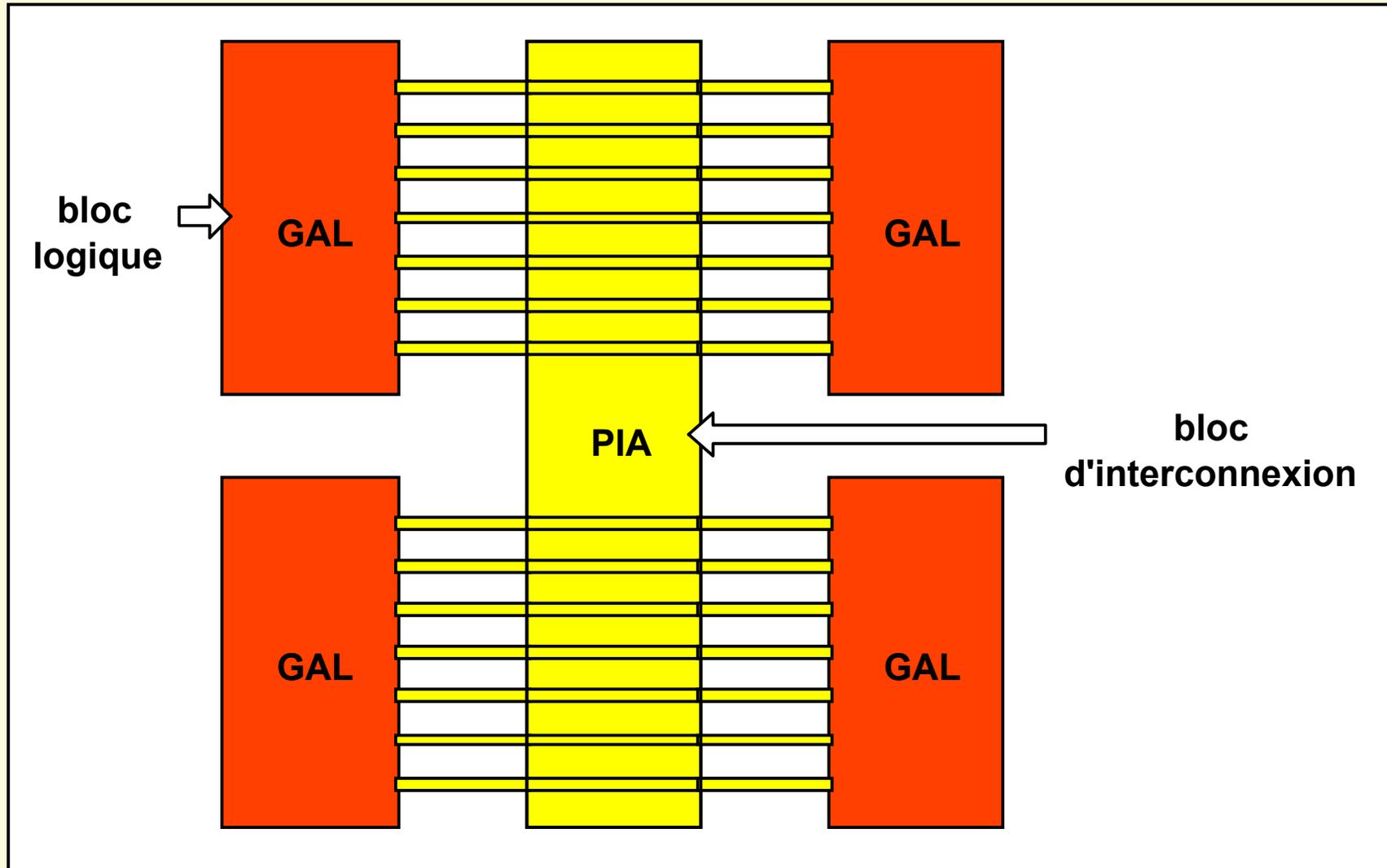
---

Définition :

- CPLD, *Complex Programmable Logic Device*
  - circuit programmable hiérarchique regroupant un ensemble de circuits programmables élémentaires (SPLD) et un réseau d'interconnexion programmable
  - chacun des blocs logiques élémentaires est équivalent à un circuit GAL
  - réseau d'interconnexion interne PIA (*Programmable Interconnect Array*)

# ... circuit CPLD

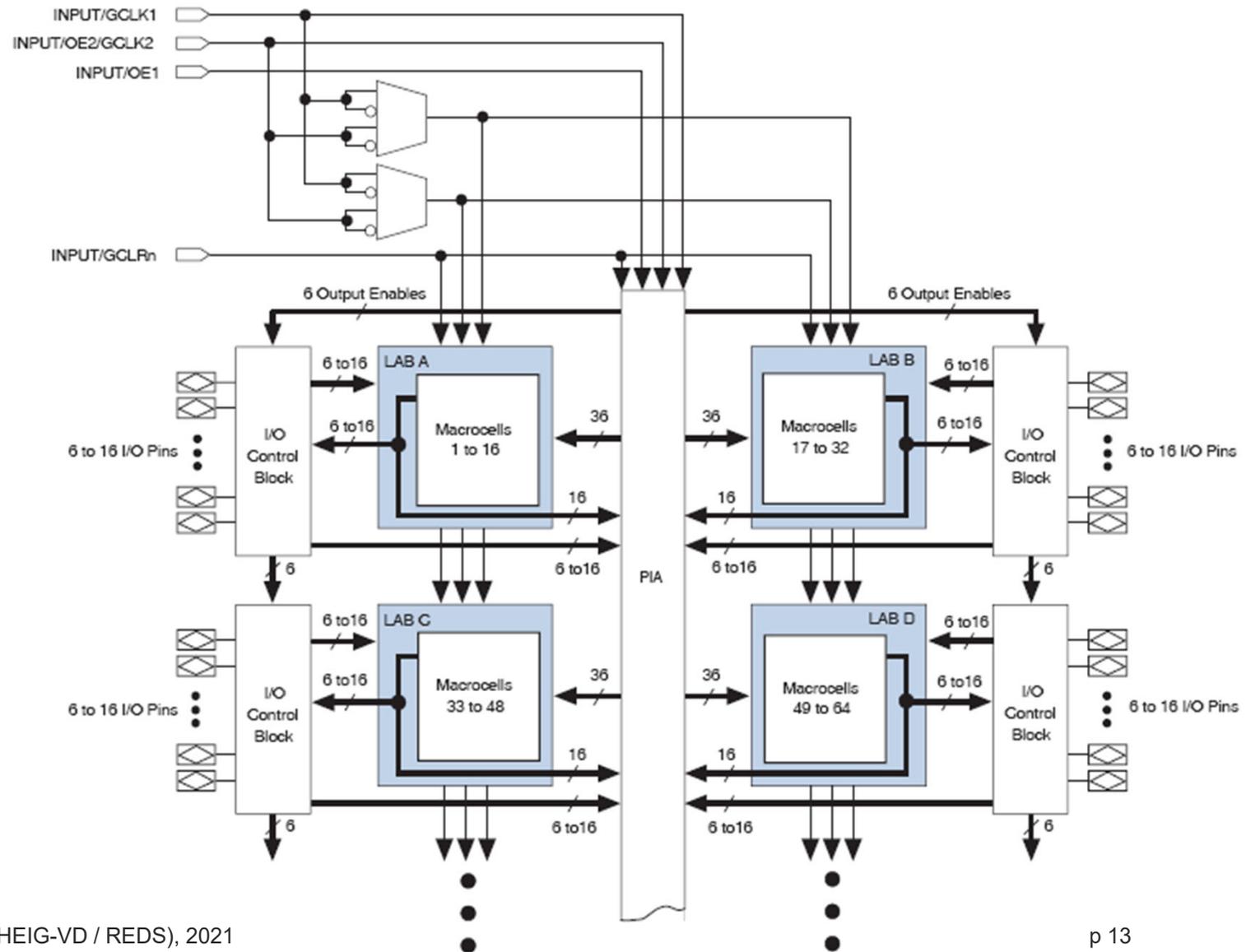
## Architecture générale



# CPLD Altera EPM7000S

Figure 2. MAX 7000E & MAX 7000S Device Block Diagram

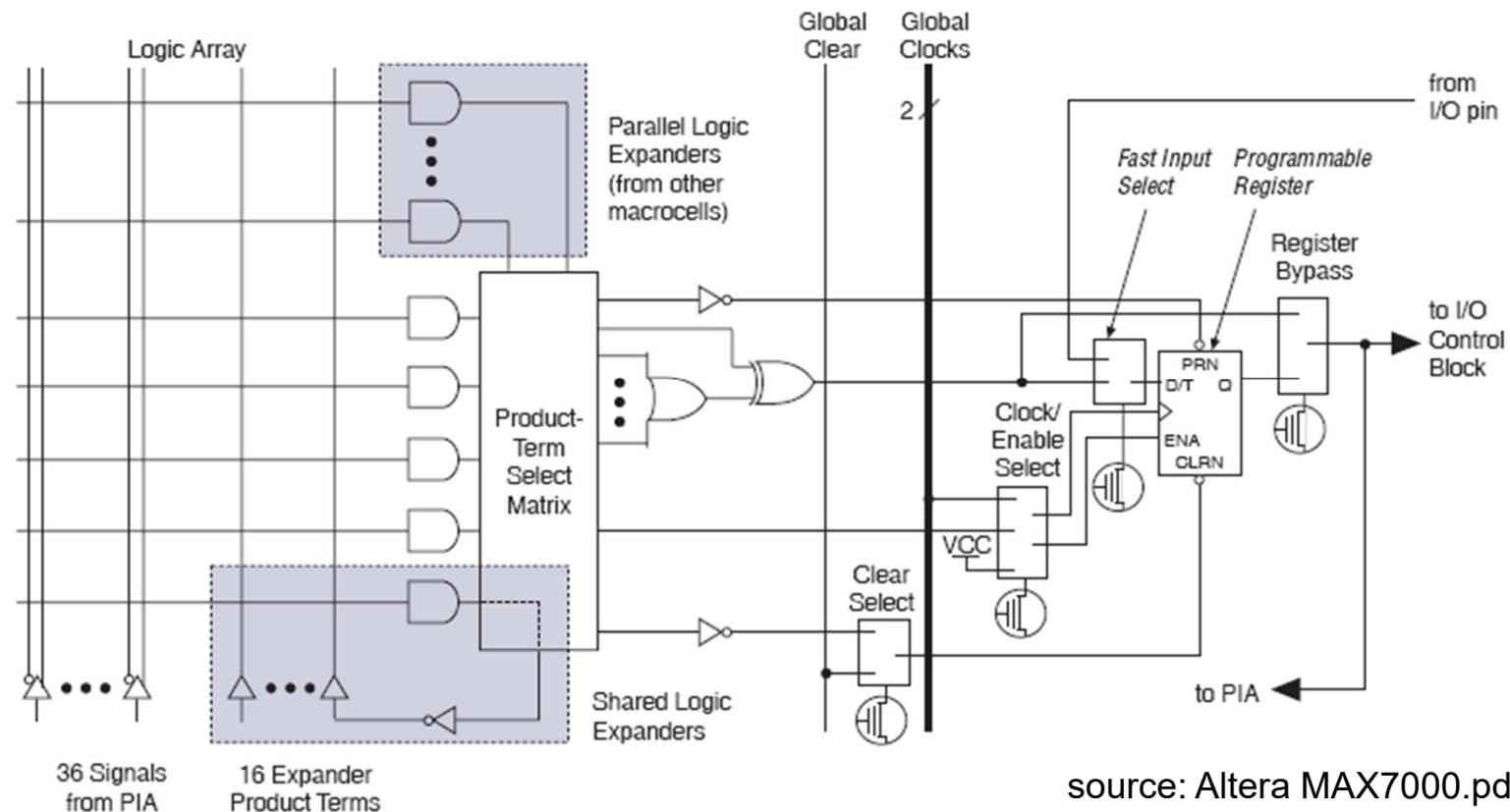
source: Altera MAX7000.pdf



# CPLD Altera EPM7000S

- Structure cellule logique: MacroCell (MC)

Figure 4. MAX 7000E & MAX 7000S Device Macrocell



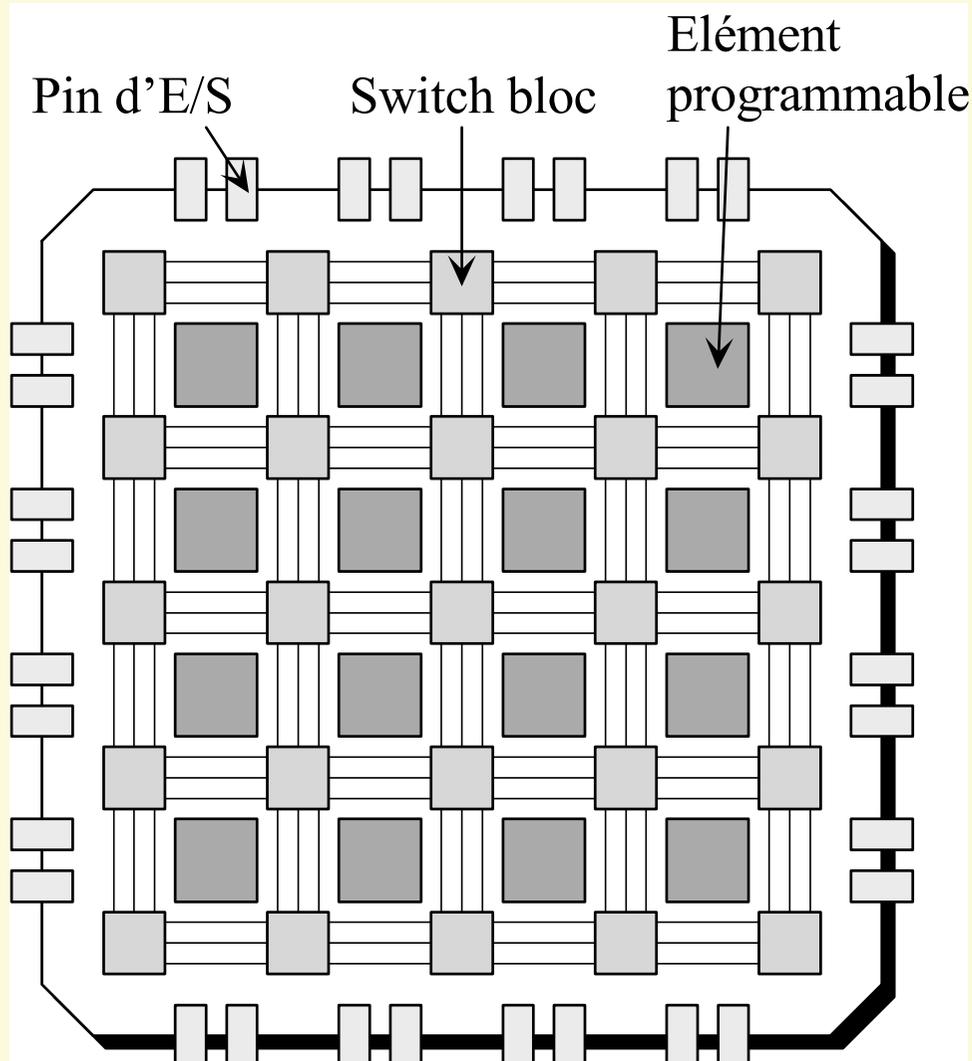
# Circuit FPGA

---

Définition :

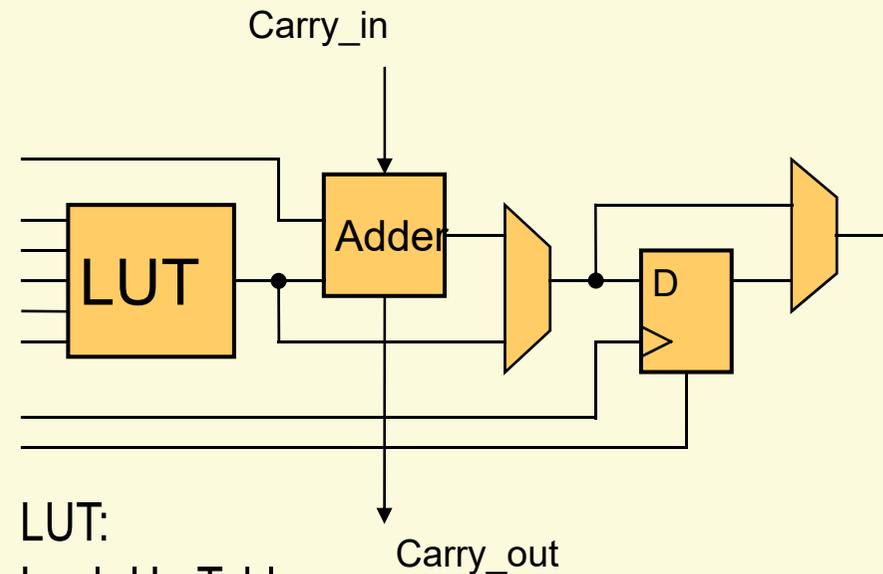
- **FPGA, *Field Programmable Gate Array***
  - circuit programmable composé d'un réseau de petits blocs logiques, de cellules d'entrée-sortie et de ressources d'interconnexion totalement flexibles
  - la granularité des cellules logiques des FPGA est spécifiée comme fine, car :
    - nombre d'entrées d'une cellule : 4 à 6
- **Cas des CPLD : granularité grossière**
  - une cellule permet de réaliser une fonction avec 5 à 7 termes comprenant plus de 32 signaux

# FPGA Xilinx (1985)



E. Messerli (HES-SO / HEIG-VD / REDS), 2021

Elément programmable:  
élément logique de base d'un FPGA  
LE : Logic Element



LUT:  
Look-Up Table  
de 4 à 6 entrées

# Xilinx FPGA

- Famille XC4000 en 1991
  - série la plus dense de l'époque!
  - densité max: 20'000 gates
  - nombre max IOs: 240

## *XC4000 Family Members*

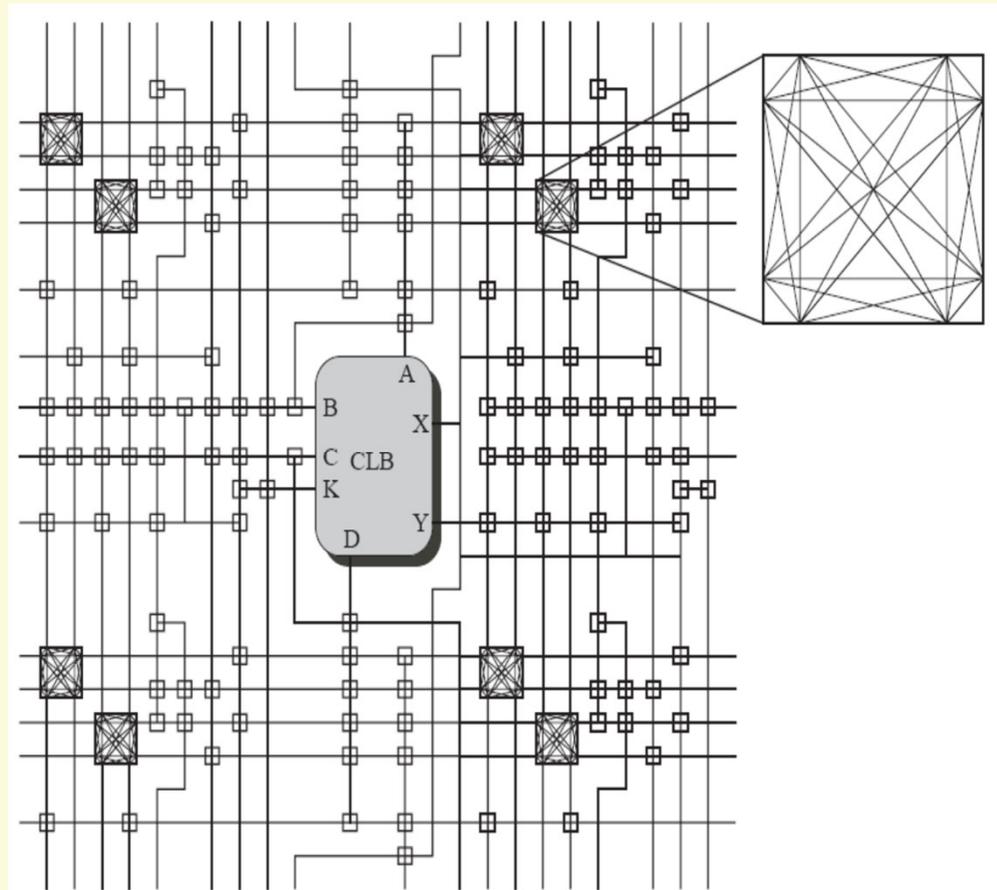
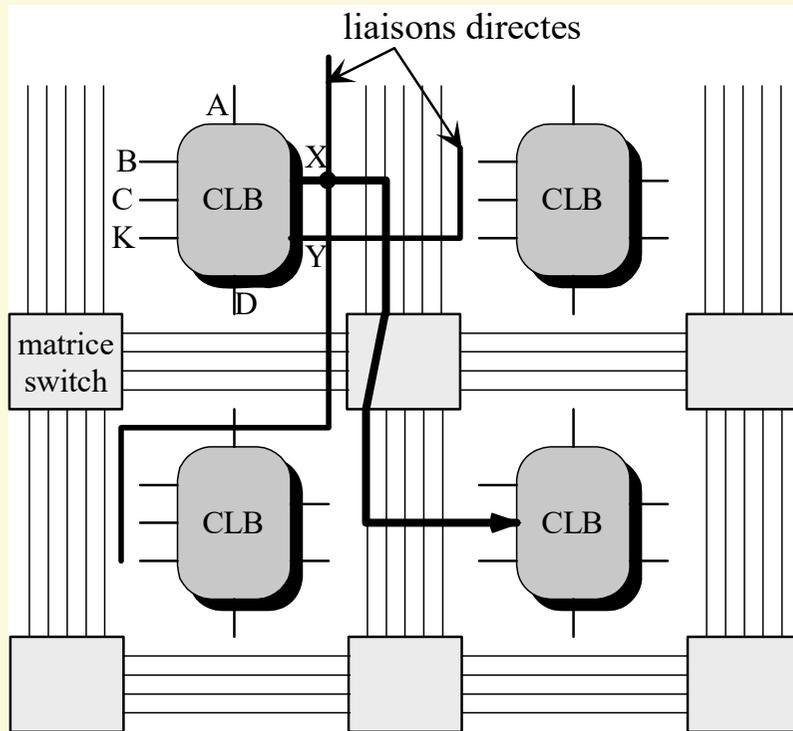
	XC4002	4003	4004	4005	4006	4008	4010	4013	4016	4020
Appr. Gate Count	2,000	3,000	4,000	5,000	6,000	8,000	10,000	13,000	16,000	20,000
CLB Matrix	8 x 8	10 x 10	12 x 12	14 x 14	16 x 16	18 x 18	20 x 20	24 x 24	26 x 26	30 x 30
Configurable Logic Blocks	64	100	144	196	256	324	400	576	676	900
Max RAM Bits	2,048	3,200	4,608	6,272	8,192	10,368	12,800	18,432	21,632	28,800
Input/Outputs	64	80	96	112	128	144	160	192	208	240

source: Xilinx Databook 1991

# FPGA: réseau d'interconnexions

## FPGA Xilinx: XC2000

Vue de détail du datasheet



# Comparaison CPLD-FPGA

---

- Délai constant dans CPLD
  - un seul réseau d'interconnexion
  - cellule de taille importante, fréquence moyenne
- Délai variable dans FPGA
  - délai dépend fortement du routage
    - Actuellement **tp connexion > tp logique!**
  - fréquence d'un LE est élevée
  - en pratique: une fonction utilise plusieurs LEs
    - dès lors la fréquence réelle  $\ll F_{LE}$

# Histoire des PLD - SoC

---

- Premier CPU + FPGA en 1998
  - Motorola: famille Core+
    - processeur ColdFire 5206 (32bits, 54MHz)
    - FPGA de 22'000 gates

Revue:  
Electronique no 82  
Juin 1998

## PROCESSEUR RISC ET FPGA SUR LE MEME SILICIUM

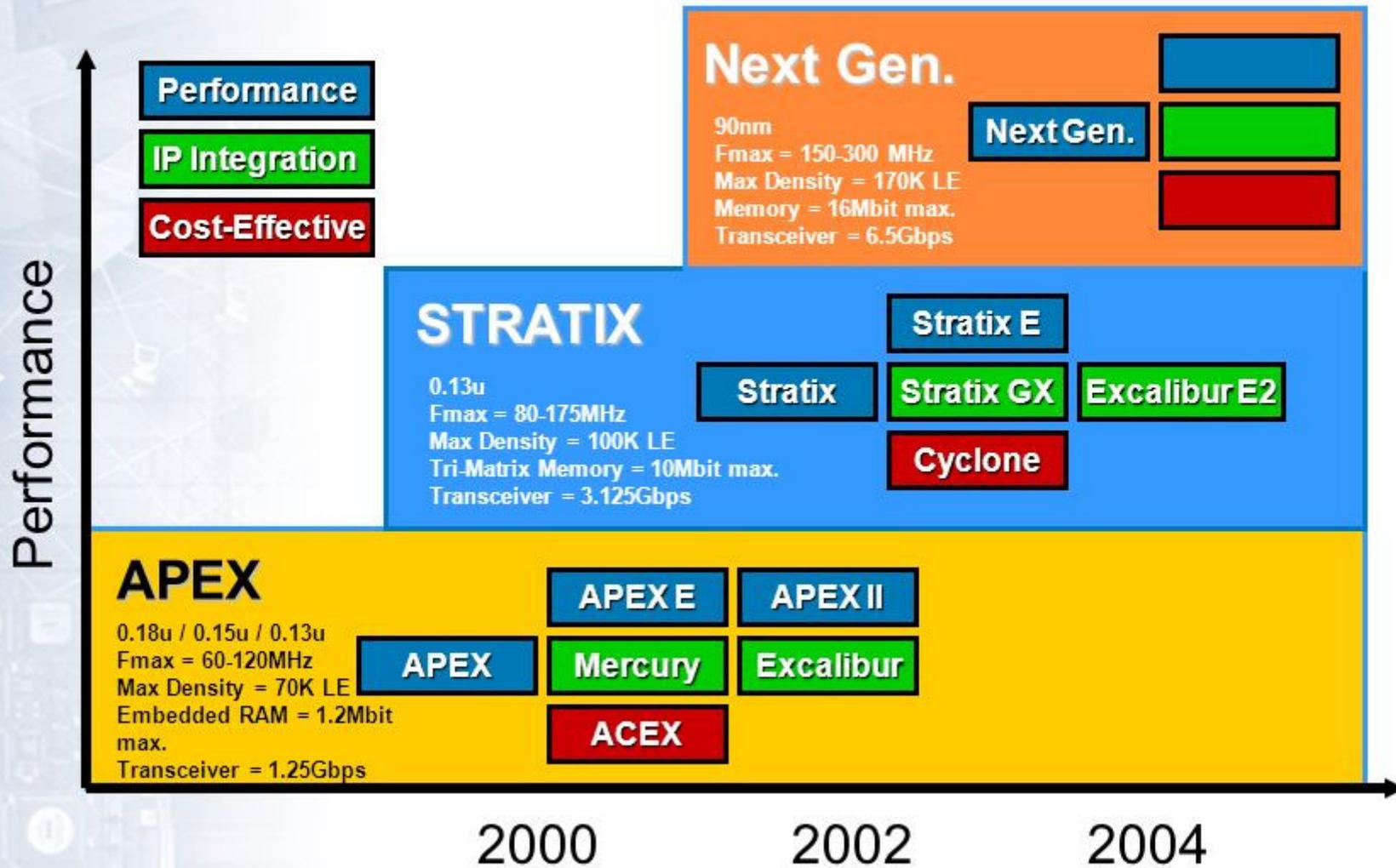
■  
La famille Core+ de Motorola, associant Risc et FPGA en un circuit, constitue une alternative séduisante aux approches multipuces ou exclusivement FPGA.

# Histoire des PLD - SoC

---

- Années 2000: émergence des SoC
  - Altera Excalibur 2001
    - hard core ARM922T (32bits, 200MHz)
    - soft core Nios (16bits), Nios-II (32bits)
  - Xilinx Virtex-II 2001/2002
    - hard core PowerPC (32bits, 300MHz), jusqu'à 4
    - soft core MicroBlaze (32bits)
  - Lattice
    - LatticeMico32 (2006)

# Altera FPGA Roadmap



# Histoire des PLD - SoC

---

- Année 2010: SoC-FPGA ARM
  - Altera SoC-FPGA 2012
    - dual core ARM Cortex-A9
  - Xilinx Zynq 2011
    - dual core ARM Cortex-A9
- Outils performant pour développement d'application SoC-FPGA
- Présentation de cette technologie dans :  
**Evolution des circuits SoC-FPGA**

# Langages de description HDL

---

- Pour les SPLD et CPLD:
  - PALASM, PAL assembler
    - premier langage de description pour PLD
  - CUPL, 1983:
    - Compiler for Universal Programmable Logic
  - ABEL, 1983:
    - Advanced Boolean Expression Language
    - Data IO
  - Dès 1990: utilisation de VHDL et Verilog
    - VHDL stable dès la norme VHDL-1993
    - Déployée dans l'industrie dès la fin des années 1990

# PALASM

- Description d'un compteur 4 bits avec Clear synchrone
  - Clear actif haut!
- OE actif bas:
  - câblé fixe
- Etat compteur inversé (notQ)

Source:

[http://commons.wikimedia.org/wiki/](http://commons.wikimedia.org/wiki/File:PALASM_VD/REFS)

E. Messerli (File:PALASM\_Design.jpg) 2021

```
PAL16R4 PAL                PAL DESIGN SPECIFICATION
CNT4SC
4 bit counter with synchronous clear
Michael Holley and Dave Pellerin
Clk Clear NC NC NC NC NC NC NC GND
OE NC NC /Q3 /Q2 /Q1 /Q0 NC NC VCC
```

```
Q3 := Clear
    + /Q3 * /Q2 * /Q1 * /Q0
    + Q3 * Q0
    + Q3 * Q1
    + Q3 * Q2
```

```
Q2 := Clear
    + /Q2 * /Q1 * /Q0
    + Q2 * Q0
    + Q2 * Q1
```

```
Q1 := Clear
    + /Q1 * /Q0
    + Q1 * Q0
```

```
Q0 := Clear
    + /Q0
```

## FUNCTION TABLE

OE	Clear	Clk	/Q0	/Q1	/Q2	/Q3
L	H	C	L	L	L	L
L	L	C	H	L	L	L
L	L	C	L	H	L	L
L	L	C	H	H	L	L
L	L	C	L	L	H	L
L	H	C	L	L	L	L

# CUPL

- Exemple d'un décodeur Hex => 7 segments

```
/** Inputs **/  
  
Pin 1 = W; /* Msb of HEX input */  
Pin 2 = X; /* */  
Pin 3 = Y; /* */  
Pin 4 = Z; /* Lsb of HEX input */  
  
/** Outputs **/  
  
Pin 19 = a; /* Output Segments for 7-segment Display */  
Pin 18 = b; /* */  
Pin 17 = c; /* */  
Pin 16 = d; /* */  
Pin 15 = e; /* */  
Pin 14 = f; /* */  
Pin 13 = g; /* */
```

```
/** Logic Equations **/  
FIELD INPUT = [W,X,Y,Z]; /* Defines input array */  
  
FIELD OUTPUT = [a,b,c,d,e,f,g]; /* Defines output array */  
  
TABLE INPUT => OUTPUT {  
  'b'0000 => 'b'0000001;  
  'b'0001 => 'b'1001111;  
  'b'0010 => 'b'0010010;  
  'b'0011 => 'b'0000110;  
  'b'0100 => 'b'1001100;  
  'b'0101 => 'b'0100100;  
  'b'0110 => 'b'0100000;  
  'b'0111 => 'b'0001111;  
  'b'1000 => 'b'0000000;  
  'b'1001 => 'b'0001100;  
  'b'1010 => 'b'0001000;  
  'b'1011 => 'b'1100000;  
  'b'1100 => 'b'0110001;  
  'b'1101 => 'b'1000010;  
  'b'1110 => 'b'0110000;  
  'b'1111 => 'b'0111000;  
}
```

Description à l'aide  
d'une table de vérité

# ABEL

## Description d'un compteur 4 bits:

- Clear synchrone
  - actif bas !
- Load synchrone
  - actif bas !

## Déclaration d'un signal registre:

- istype 'reg\_D, pos, feed\_reg'

**Langage intéressant, mais pas hiérarchique !**

```
module SN74162 flag '-r3';
title 'Synchronous 4-bit decade counter with look ahead carry
Michael Holley    30 Jan 1989'
```

```
sn74162 device 'P22V10';
```

```
Clk, Load, Clear, EnaP, EnaT    pin 1,2,3,4,5;
DataA, DataB, DataC, DataD      pin 6,7,8,9;
QA, QB, QC, QD                 pin 15,16,17,18;
Carry                           pin 19;
```

```
QA, QB, QC, QD
Carry
```

```
Count    = [QD    , QC    , QB    , QA    ];
Data     = [DataD, DataC, DataB, DataA];
```

```
C, H, L = .C., 1, 0;
```

```
Ena = EnaP & EnaT;
```

```
equations
```

```
Count := (Count + 1) & Clear & Load & Ena & (Count < 9)
# (Count    ) & Clear & Load & !Ena
# Data      & Clear & !Load;
```

```
Carry = (Count == 9) & EnaT;
```

```
test_vectors
```

```
([Clk, Clear, Load, EnaP, EnaT, Data] -> [Count, Carry])
[ C , 0 , 1 , 0 , 0 , 0 ] -> [ 0 , 0 ];
[ C , 1 , 1 , 0 , 0 , 0 ] -> [ 0 , 0 ];
[ C , 1 , 1 , 1 , 1 , 0 ] -> [ 1 , 0 ];
[ C , 1 , 1 , 1 , 1 , 0 ] -> [ 2 , 0 ];
[ C , 1 , 1 , 1 , 1 , 0 ] -> [ 3 , 0 ];
[ C , 1 , 1 , 1 , 0 , 0 ] -> [ 3 , 0 ];
```

istype 'reg\_D, pos, feed\_reg';  
istype 'com, pos';

type d'élément

opérateur d'addition

opérateur de comparaison

Possible de saisir des vecteurs de test

# VHDL

## Description compteur 4 bits:

- Décomposition selon la structure d'un système séquentiel simple (méthodologie REDS)

VHDL plus formaliste qu'ABEL

VHDL permet plus d'abstraction

## Le VHDL offre:

- Un standard IEEE
- Hiérarchique
- Notion de temps
- Instructions concurrentes
- Instruction process
- Paquetages
- Génériques
- ...

```
library ieee;
  use ieee.Std_Logic_1164.all;
  use ieee.numeric_std.all;

entity cpt4_163 is
  port(clock_i : in std_logic;
        nClr_i  : in std_logic;
        nLoad_i : in std_logic;
        enp_i   : in std_logic;
        ent_i   : in std_logic;
        val_i   : in std_logic_vector(3 downto 0);
        -- val_i = D & C & B & A
        cpt_o   : out std_logic_vector(3 downto 0);
        rco_o   : out std_logic );
end cpt4_163;

architecture comport of cpt4_163 is
  signal cpt_fut, cpt_pres : unsigned(3 downto 0);
  signal clear_s, load_s   : std_logic;
  signal en_s              : std_logic;
begin
  --Adaptation polarite
  clear_s <= not nCLR_i;
  load_s  <= not nLoad_i;
  en_s    <= enp_i and ent_i;

  --Decodeur d'etats futurs
  cpt_fut <= "0000" when clear_s = '1' else --init à 0
             unsigned(val_i) when load_s = '1' else --chargement
             cpt_pres + 1 when en_s = '1' else --incrémente
             cpt_pres; -- maintien

  --Registe (4 flip-flop)
  Mem: process (clock_i)
  begin
    if Rising_Edge(clock_i) then
      cpt_pres <= cpt_fut;
    end if;
  end process;

  --Affectation des sorties
  rco_o <= '1' when (cpt_pres = 15) and (ent_i = '1') else
           '0';
  cpt_o <= std_logic_vector(cpt_pres);
end Comport;
```

# Design and verification languages

---

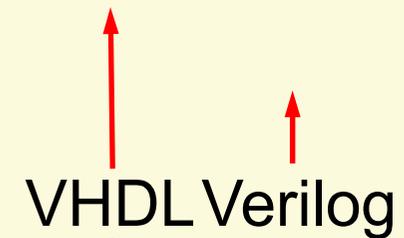
Requirements

Algorithm

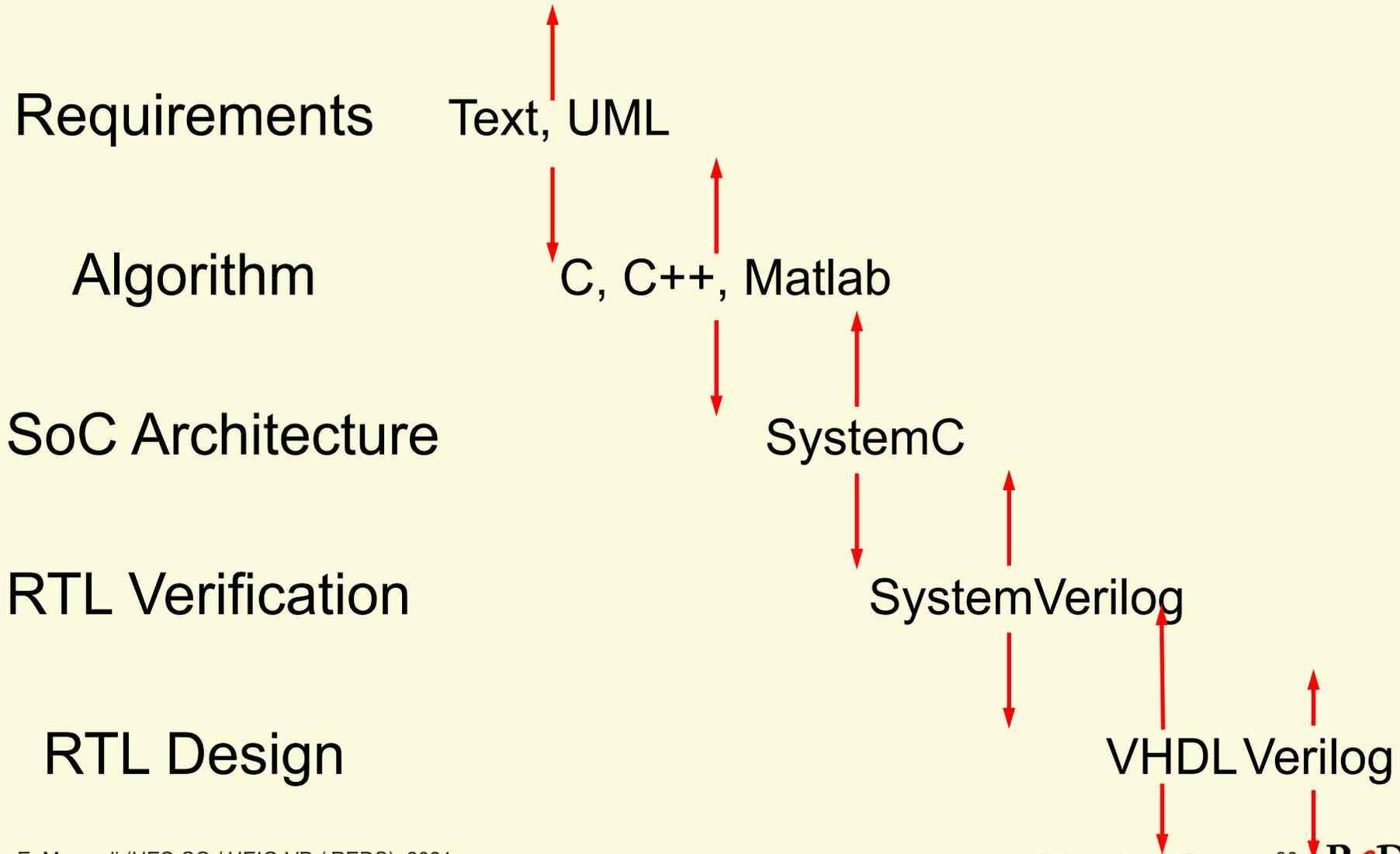
SoC Architecture

RTL Verification

RTL Design



# Design and verification languages



# Evolution "Circuits logiques programmables"

Depuis 1995, formidable évolution des circuits logiques programmables

Caractéristiques	Année 2000	Année 2019
Technologie	<b>150 nm</b>	<b>10 nm</b>
Densité	90K LEs, 90K DFF RAM jusqu'à 3Mbits	5'500K LEs, 5'500K DFF, RAM jusqu'à 500Mbits
Hard Core :	Transceivers 1.25 Gbps	Transceiver up to 57.8 Gbps PCI express, Ethernet MAC, ...
Fréquence	jusqu'à 350 MHz	jusqu'à 1.1 GHz
Prix (Fr/gate)	2000 $\approx$ 0,25 ct/gate	2008 $\approx$ 0,0001ct/gate 2015 $\approx$ tend vers zéro!
Boitier	1 puce	Multi puces !

# Caractéristiques PLDs 2019 ...

- technologie 45 à 10 nm en production (2000 : 150 nm)
- alimentation du cœur: 0.90 à 0.70 V/ 40 A !
- multiples arbres d'horloges et PLL, jusqu'à 100
- fréquence jusqu'à 1.1 GHz (bloc DSP 900 MHz)
- nombre de *Logic Elements* ~5'500K LEs
  - LUT à 6 entrées au lieu de 4!  
ALM (2 LEs) à 8 entrées => LUT à 7 entrées
- nombre de flip-flops jusqu'à 7'400K
- mémoire RAM : jusqu'à 500 Mbits
- integrated 3D HBM : jusqu'à 16GB **new !**

# ... caractéristiques PLDs 2019

---

- blocs pré-câblés (DSP, SERDES, ...)
  - transceivers high speed: max 128, max 57.8 GigaBit/s
  - jusqu'à 3'500 DSP bloc avec multiplicateurs
- Hard Core (PCI express, Ethernet MAC, PCS, ...)
- multiples standards I/O:
  - LVTTTL, LVCMOS, SSTL et différentiel SSTL, LVDS, LVECL, ...
- Nbr I/O jusqu'à 1'600
  - boitier FBGA2597 pins => 1'600 I/O

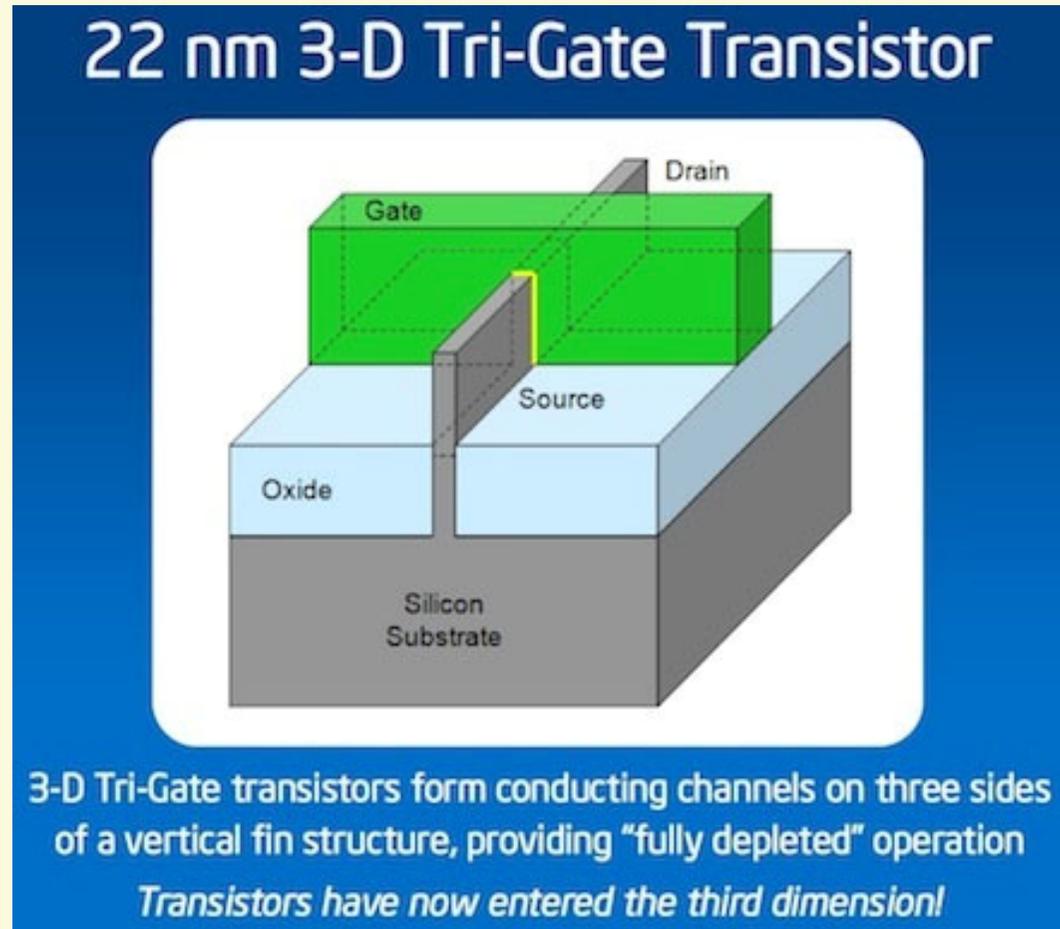


# Nouvelle technologie 3D ...

Augmentation de la section du canal dans les 3 dimensions !

## Avantages:

- Forte augmentation du gain à basse tension
- Amélioration des caractéristiques de commutation
- Courant important
- Coûts supplémentaires de seulement 2-3% !



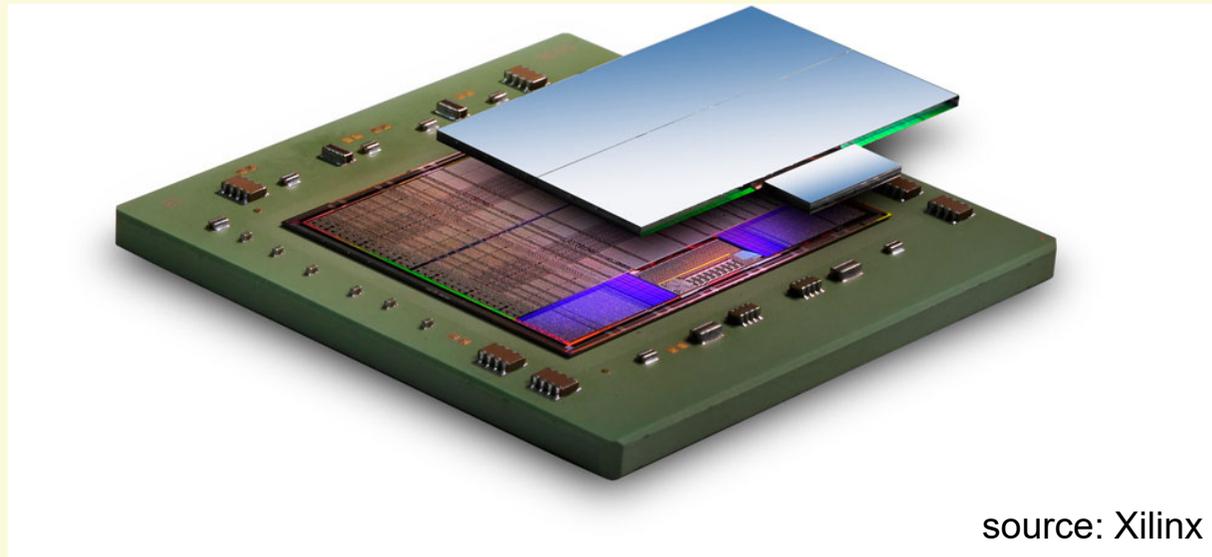
source: Intel

# Nouvelle technologie: Virtex-7 HT ...

Site Xilinx: mars 2013

## Xilinx's Stacked Silicon Interconnect (SSI)

- Combines enhanced Super Logic Region (SLR) FPGA die slices and 25-28.05 Gb/s transceivers on a passive silicon interposer to create a three dimensional (3D) die stack



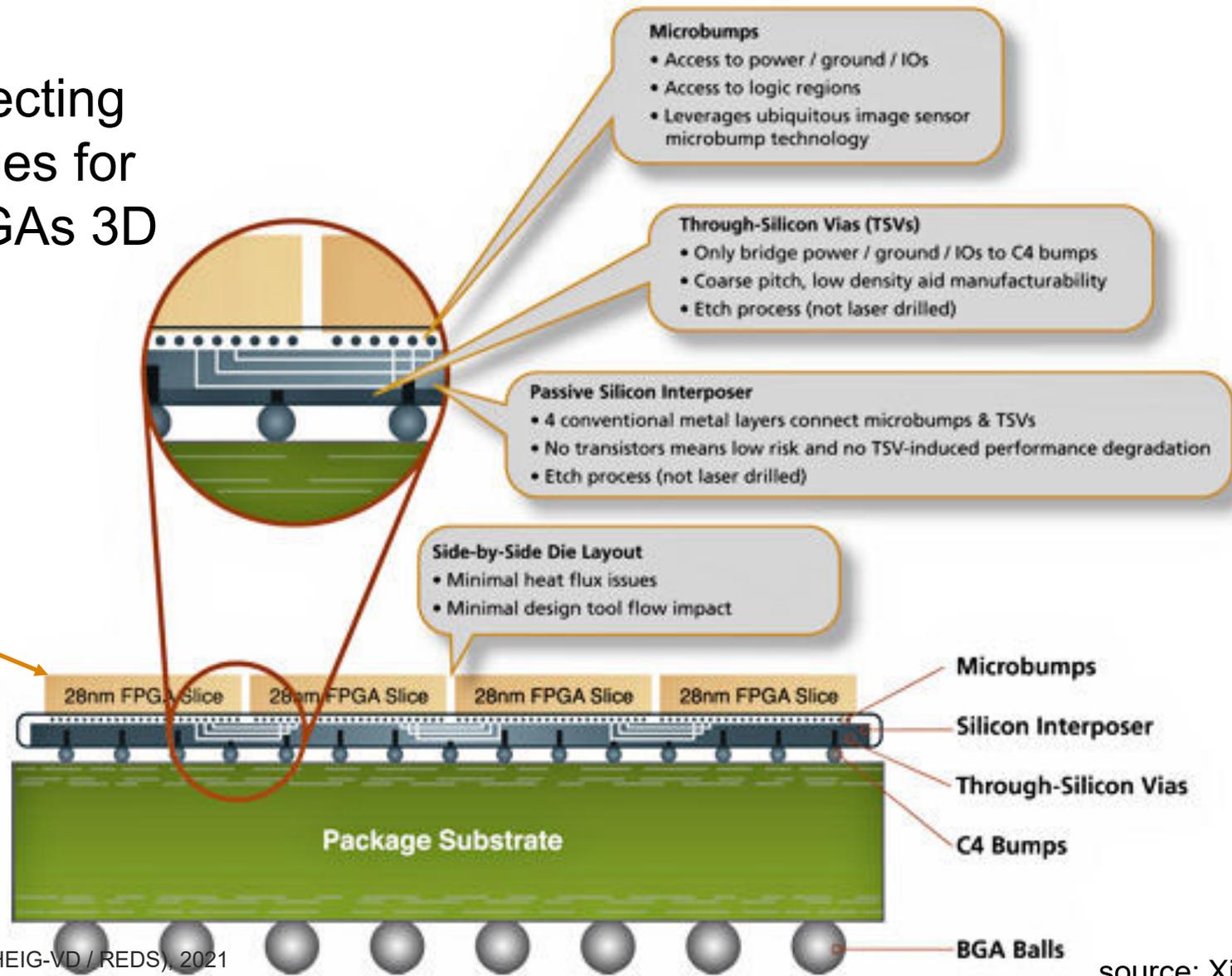
source: Xilinx

- Up to 2000K logic cells offered in the largest Virtex-7 T device

# Stacked Silicon Interconnect technology

Interconnecting multiple dies for Xilinx FPGAs 3D devices

one die



# Nouvelle technologie: Stratix-10



Stratix® 10 FPGAs and SoCs to support data rates up to 56 Gbps. Altera is demonstrating today the FPGA industry's first dual-mode 56-Gbps pulse-amplitude modulation with 4-levels (PAM-4) and 30-Gbps non-return-to-zero (NRZ) transceivers.

Customers can use Stratix 10 FPGAs to build next-generation communications and networking infrastructure that support 50G, 100G, 200G, 400G and terabit applications.

# New technology: PAM-4 modulation

Pulse-amplitude modulation: PAM-4 modulation uses four distinct amplitude levels, each of which represents a combination of two bits (00, 01, 11, 10)

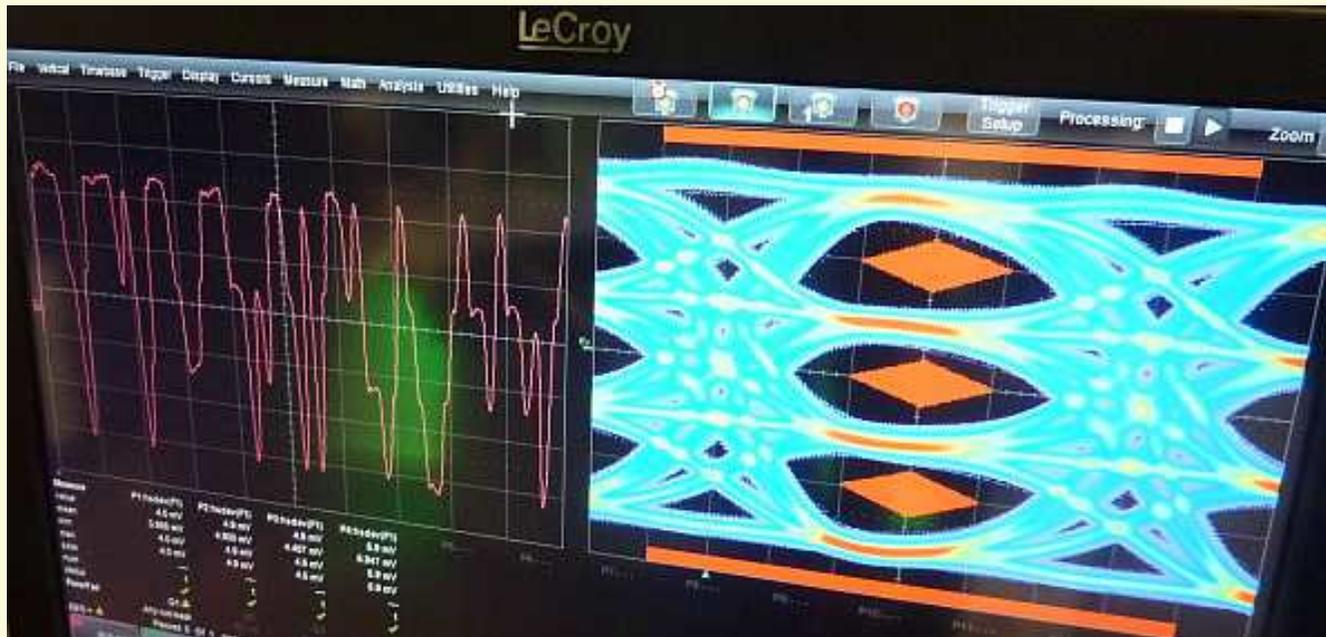
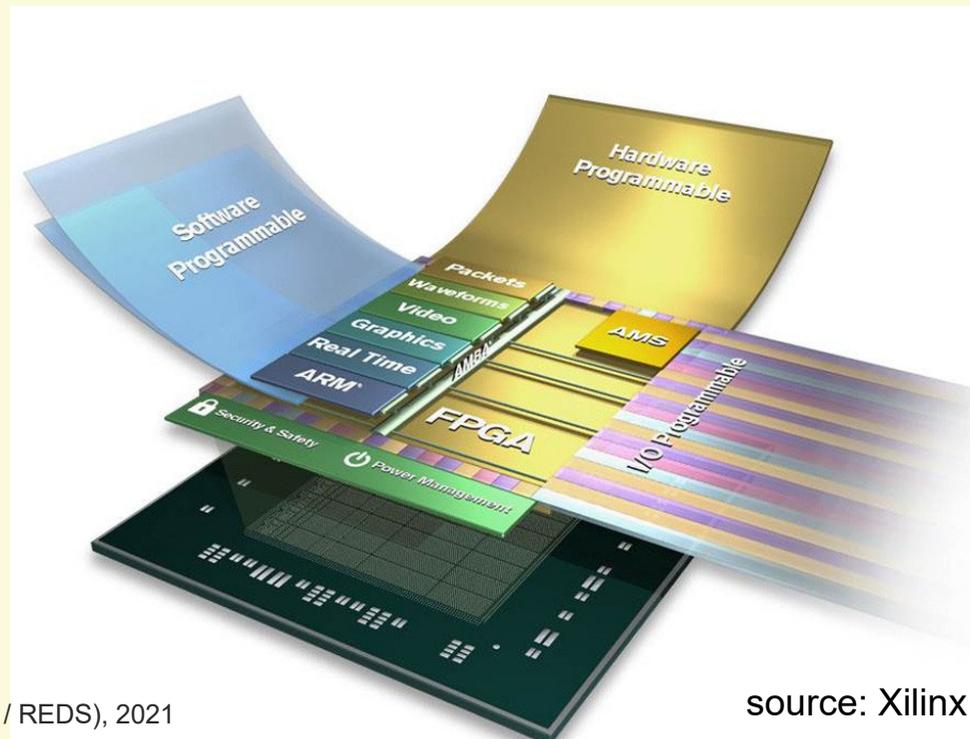


Figure [www.edn.com](http://www.edn.com): PAM-4 uses four distinct voltages to represent two bits (left), which produces three eye diagrams (right).

# Xilinx: Zynq UltraScale MPSoC

The UltraScale™ MPSoC Architecture, built on TSMC's 16nm FinFET process technology, enables next generation Zynq® UltraScale MPSoCs. This new architecture provides 4 core processors with:

- 64 bits with support for virtualization
- graphics/video processing,
- next generation interconnect and memory



# Sites internet PLDs

---

- Informations des vendeurs d'outils EDA  
<http://www.vhdl.org/>
- Liste des outils VHDL  
<http://www.asic-world.com/vhdl/tools.html>
- Site donnant la liste des vendeurs de PLDs (2011!)  
<http://www.fpgacentral.com/vendor/>

# Sites internet PLDs

---

- Divers documentations
  - [http://en.wikipedia.org/wiki/Programmable\\_logic\\_device](http://en.wikipedia.org/wiki/Programmable_logic_device)
  - [https://en.wikipedia.org/wiki/Field-programmable\\_gate\\_array](https://en.wikipedia.org/wiki/Field-programmable_gate_array)
  - [http://en.wikipedia.org/wiki/Hardware\\_description\\_language](http://en.wikipedia.org/wiki/Hardware_description_language)
  - [http://en.wikipedia.org/wiki/List\\_of\\_Verilog\\_simulators](http://en.wikipedia.org/wiki/List_of_Verilog_simulators)
  - <http://www.doulos.com/knowhow>
- Documentation sur les processeurs:
  - [https://en.wikipedia.org/wiki/Soft\\_microprocessor](https://en.wikipedia.org/wiki/Soft_microprocessor)

# Sites internet vendeur PLDs

---

- <http://www.achronix.com/>
- <http://www.actel.com/>
- <http://www.altera.com/>
- <http://www.atmel.com/>
- <http://www.cypress.com/>
- <http://www.latticesemi.com/>
- <http://www.quicklogic.com/>
- <http://www.siliconbluetech.com/>
- <http://www.xilinx.com/>

# FIN présentation

---

## Questions

