# C1 - Introduction

### Yann DOUZE VHDL

### VHDL = LOGIC DESIGN

- Just Like Building a Circuit on Your Breadboard!!
- Also known as a "Hardware Description Language"

```
LIBRARY ieee:
     USE ieee.std logic 1164.all;
    ENTITY LabExCG4 IS
                u, v, w, x, y : IN BIT;
10
                      : IN STD LOGIC VECTOR(2 DOWNTO 0);
11
                 m : OUT BIT);
12
     END LabExCG4:
13
    ARCHITECTURE Behavior OF LabExCG4 IS
    BEGIN
    PROCESS(s)
17
     BEGIN
18
19
20
21
22
             WHEN "100" => m <= v;
24
             WHEN OTHERS => m <= v;
25
         END CASE:
26
     END PROCESS:
     END Behavior:
                                  (NOT Actual equivalent Circuit - For Concept Demo only)
```

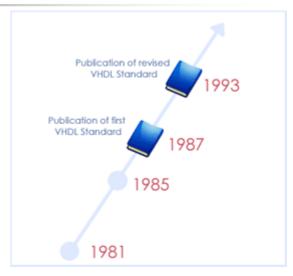


# Qu'est ce que le VHDL?

- VHDL: VHSIC Hardware Description Language
- VHSIC : Very High Speed Integrated Circuit
- VHDL : langage de description matérielle, décrit la structure et le comportement d'un circuit numérique.
- Langage standard de description de circuits ou de systèmes numériques en vue de:
  - Modélisation (simulation) des circuits ou systèmes
  - Synthèse (génération automatique) de circuit numérique.
  - Descriptions de programmes de test (banc de test)
  - Description de type hiérarchique (netlist)



- 1981 Lancé par le USA DoD (Department of Defense) pour résoudre la crise du cycle de vie du matériel.
- 1983-85 Développement de la base du langage par Intermetrics, IBM et TI.
- 1986 Toutes les droits transférées à l'IEEE (Institute of Electrical and Electronics Engineers)
- 1987 Publication des normes IEEE Standard 1076-1987
- 1994 Norme révisée VHDL-1076-1993 (Tout le nécessaire, utilisé par 90% des designer)
- 2002 VHDL Norme IEEE 1076-2002
- 2009 VHDL 2008 Norme IEEE 1076-2008



https://www.doulos.com/knowhow/vhdl\_designers\_guide/a\_brief\_history\_of\_vhdl



# **Autres langages proches**

- Verilog est plus ancien. La syntaxe est proche de celle du langage C. Très utilisé aux USA et en Asie
- VHDL-AMS Langage de modélisation mixte numérique-analogique IEEE.1076.1-1999. Il est entièrement compatible avec le VHDL. Utilisé uniquement pour la modélisation.
- System C
- System Verilog



# Modélisation ou synthèse?

- Modélisation
  - Tout le langage. Logique + Temporel
  - Un modèle peut être comportemental, structurel ou de type data-flow.
- Synthèse
  - Le VHDL de synthèse est un sous-ensemble du VHDL généraliste
  - La synthèse demande une bonne connaissance du circuit et de la technologie.



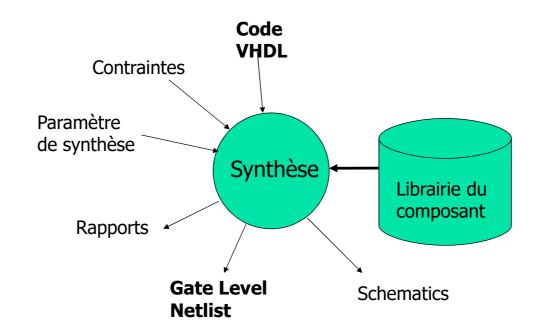
end DECOD1 4;

# La synthèse (1)

```
entity DECOD1 4 is
     port(INO, IN1: in std logic;
          DO, D1, D2, D3: out std_logic);
                                         Synthèse
```

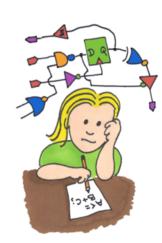
H (EN = 'T) then M or A: N or M and S; and it; 12 or N or C; If (OE = '1') then '11 or N;



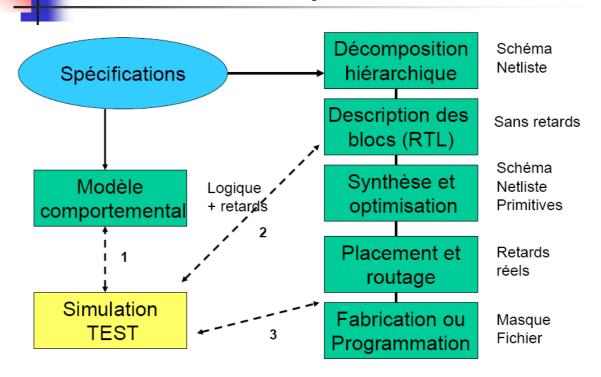


# Avertissement pour la synthèse

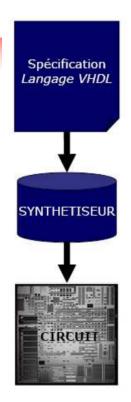
- La synthèse est très sensible à la manière dont est écrit le VHDL.
- Pour la synthèse, il faut respecter certaines règles de codage.
- Un bon design ne peut venir que d'un bon code VHDL (l'outil ne synthèse ne fait pas des miracles)
- La synthèse ne peut pas remplacer l'expertise humaine.
- "Le VHDL de synthèse est un sous-ensemble du VHDL généraliste"



# Flot de conception



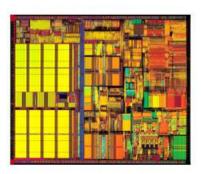
## Les cibles matérielles spécialisés

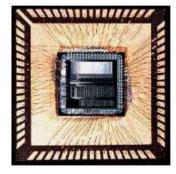


- ASIC : Application Specific Integrated Circuit
  - Numérique, analogique ou mixte (télécommunication)

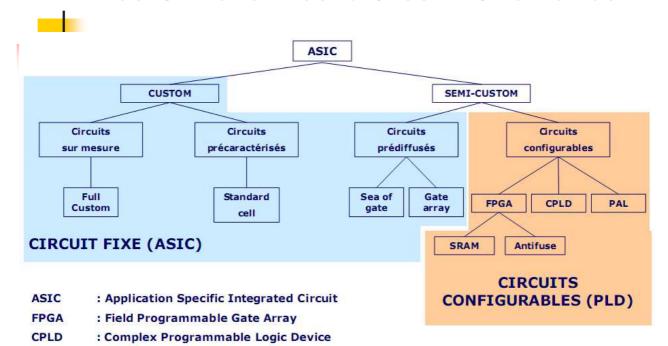


- Spécialisé pour une application
- Réalisation complexe (de la spécification haut niveau à la synthèse physique)
- Extrêmement performant : dédié+ réalisation parallèle + technologie de pointe
- Circuit = cahier des charges





#### Les différentes cibles matérielles



PAL

GAL

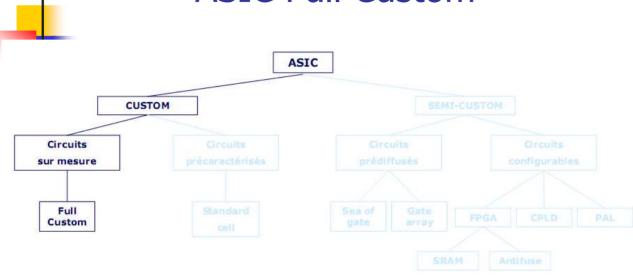
SRAM

: Programmable Array Logic

: Generic Array Logic = PAL

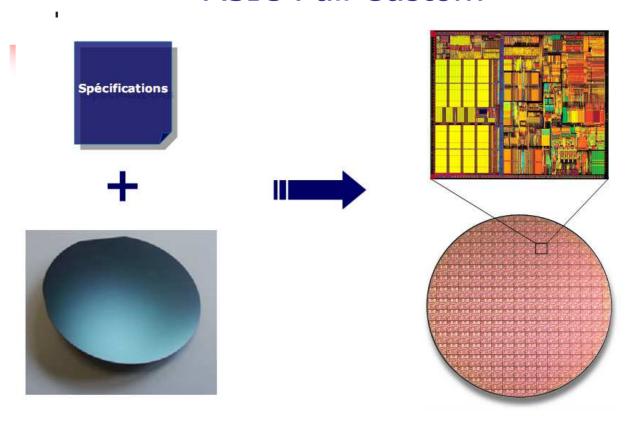
: Static Random Acess Memory

#### **ASIC Full Custom**

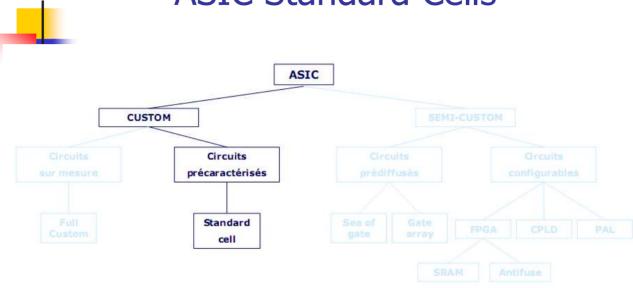


ASIC : Application Specific Integrated Circuit

### **ASIC Full Custom**

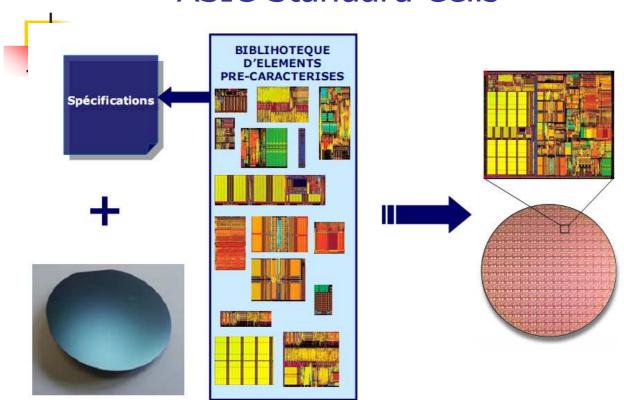


#### **ASIC Standard Cells**

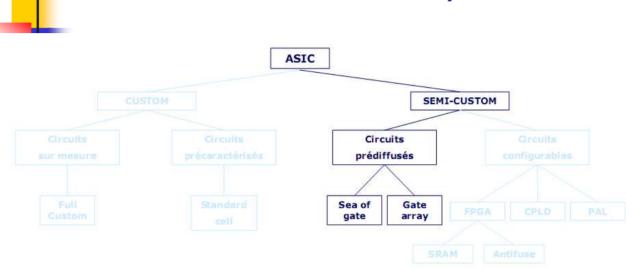


ASIC : Application Specific Integrated Circuit

### **ASIC Standard Cells**

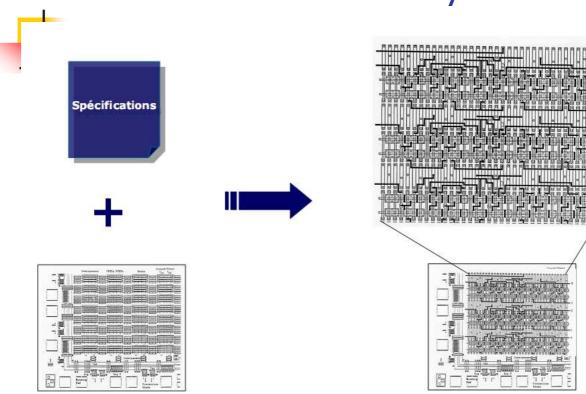


### **ASIC Gate Array**

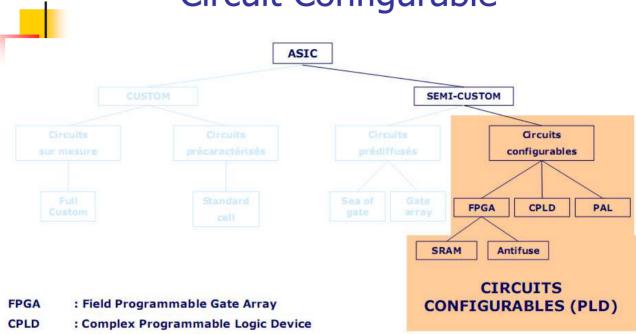


ASIC : Application Specific Integrated Circuit

# **ASIC Gate Array**



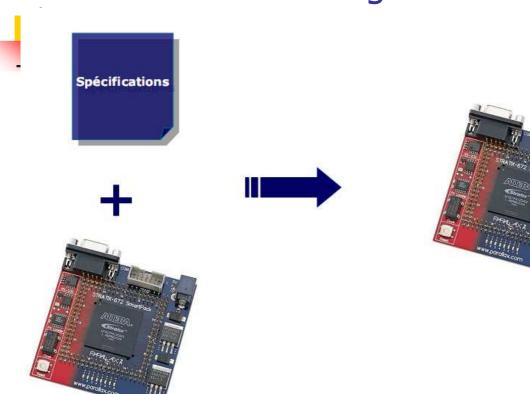
### Circuit Configurable



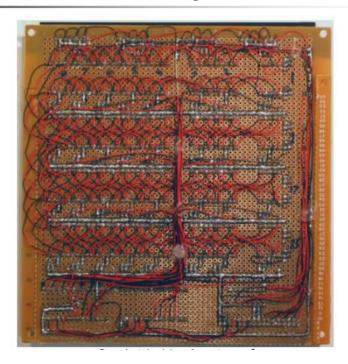
PAL : Programmable Array Logic
GAL : Generic Array Logic = PAL

SRAM : Static Random Acess Memory

# Circuit Configurable

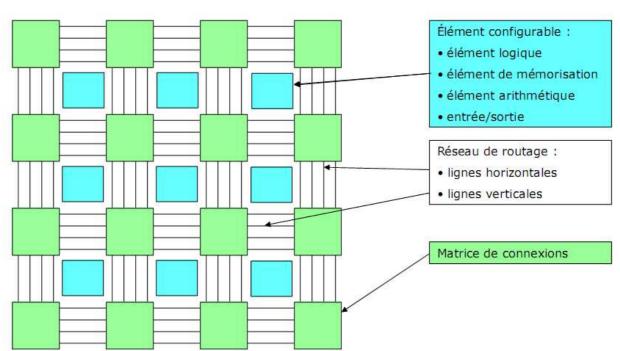


# FPGA: le concept!





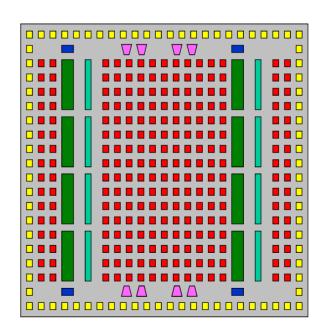
### **FPGA**





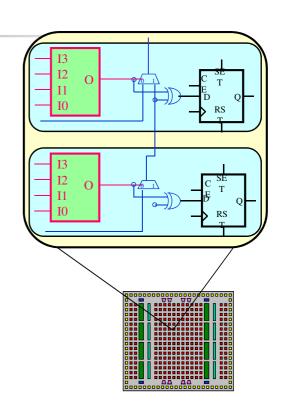
#### **FPGA Architecture**

- Logic Fabric
  - Gates and flip-flops
- Embedded Blocks
  - Memory
  - DSP/Multipliers
  - Clock management (PLL)
  - High speed serial I/O
  - Soft/hard processors
- Programmable I/Os
- In-system programmable



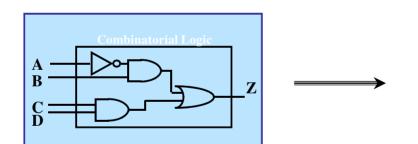


- Logic Cell (Xilinx) / Logic Element (Altera)
  - Lookup table (LUT)
  - Flip-Flop
  - Carry logic
  - Muxes (not shown)
- Slice
  - Two Logic Cells/Element





# Look-Up Table (LUTs)



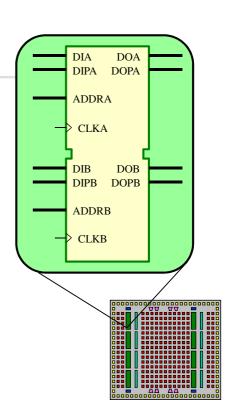
A	В	C	D	Z
0	0	0	0	0
0 0 0 0 0	0 0 0 0 1	0 0 1 1 0	0 1 0 1 0 1	0 0 0 1 1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
1	1	0	0 1 0	0
1	1	0 0	1	0
1 1 1 1	1 1 1	1	0	0 0 0
1	1	1	1	1



# **Blocs Memory**

#### Block RAM

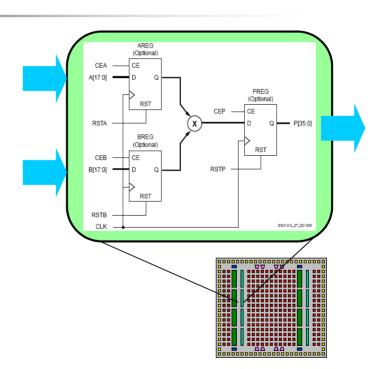
- RAM or ROM
- True dual port
  - Separate read and write ports
- Independent port size
  - Data width translation
- Excellent for FIFOs





#### Multipliers

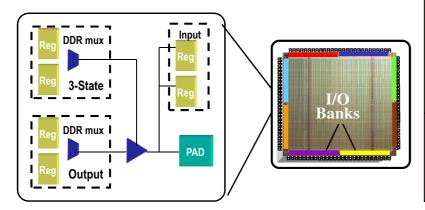
- Signed or unsigned
- Optional pipeline stage
- Cascadable





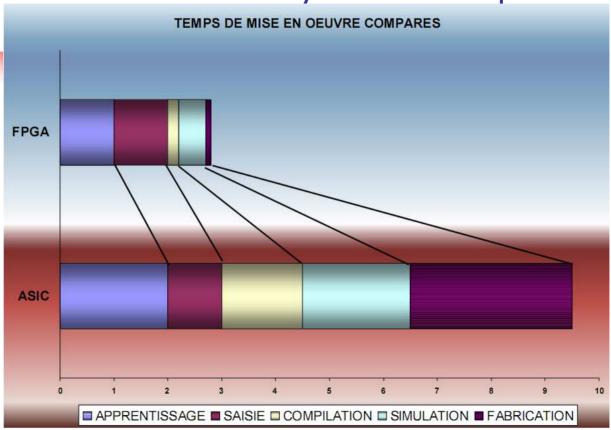
# Programmable I/Os

- Single ended
- Differential / LVDS
- Programmable I/O standards
  - Multiple I/O banks



	-		
	Standard	Output V <sub>CCO</sub>	Input V <sub>REF</sub>
Single ended	LVTTL	3.3V	-
	LVCMOS33	3.3V	-
	LVCMOS25	2.5V	-
	LVCMOS18	1.8V	-
	LVCMOS15	1.5V	-
	LVCMOS12	1.2V	-
	PCI 32/64 bit 33MHz	3.3V	-
	SSTL2 Class I	2.5V	1.25V
	SSTL2 Class II	2.5V	1.25V
	SSTL18 Class I	1.8V	0.9V
	HSTL Class I	1.5V	0.75V
	HSTL Class III	1.5V	0.9V
	HSTL18 Class I	1.8V	0.9V
	HSTL18 Class II	1.8V	0.9V
	HSTL18 Class III	1.8V	1.1V
	GTL		0.8V
	GTL+	-	1.0V
Differential	LVDS2.5	2.5V	-
	Bus LVDS2.5	2.5V	-
	Ultra LVDS2.5	2.5V	-
	LVDS_ext2.5	2.5V	-
	RSDS	2.5V	-
	LDT2.5	2.5V	

### Durée dans le cycle de conception



#### Les ASIC

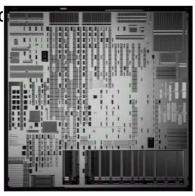


- hautes intégrations
- hautes performances (vitesse, low-power)
- coûts faibles pour de gros volumes de prod
- personnalisation
- Sécurité industrielle

#### INCONVENIENTS



- prix du 1er exemplaire
- pas d'erreur possible
- non-flexible
- time-to-market élevé
- fabrication réservée aux spécialistes (fondeur)



#### Les FPGA



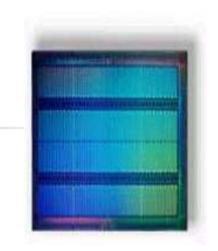
#### AVANTAGES

- Possibilité de prototypage
- time-to-market faible
- adaptabilité aux futurs évolutions grâce à la reconfiguration
- flexibilité



#### INCONVENIENTS

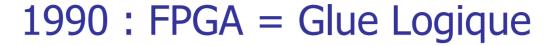
- intégration limité par les ressources de routage
- performances
- prix à l'unité élevé pour de grosses productions

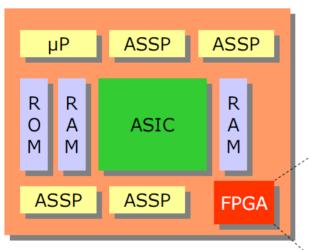


#### Evolution des méthodes de conception



- Toujours plus d'intégration (SoC)
- Les FPGA sont de plus en plus performant et de moins en moins cher donc de plus en plus utilisé.
- Les FPGA viennent lentement remplacer les circuits ASIC.
- Illustration ...



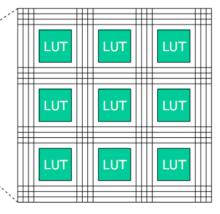


ASIC: Application Specific Integrated Circuit
 ASSP: Application Specific Standard Product

FPGA: Field Programmable Gate Array

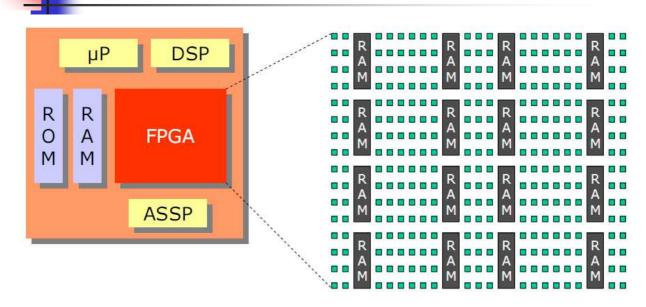
• RAM: Random Acess Memory

• ROM: Read Only Memory



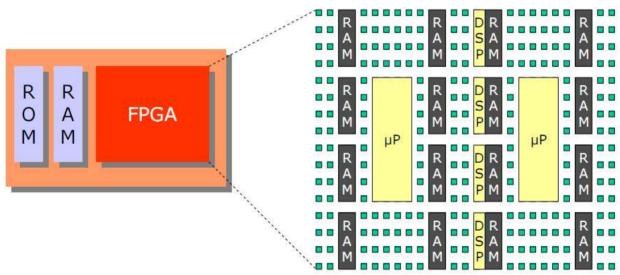
#### 2000 : FPGA = une alternative au ASIC

(augmentation de la capacité)





# Depuis 2005 : FPGA + Cœurs de processeurs





#### Conclusion sur les cibles matérielles

#### Avantages



- Grandes performances : consommation de puissance et fréquence de fonctionnement
- Parallélisme, pipeline ...
- Possibilité de traitement temps réel
- Spécialisation du circuits à l'application
- Possibilité d'avoir des composants flexibles : FPGA
- Choix large de techno, boitier, gamme ...

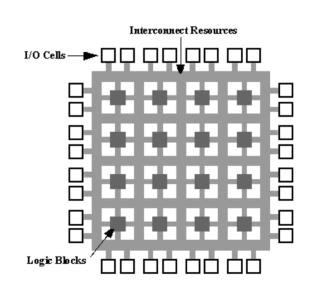
#### Inconvénients

- Conception plus ou moins complexe et longue
- Les coûts de conception peuvent être élevés (ASIC et FPGA de taille importante)
- Nécessite des méthodes de conception rigoureuse.
- Demande une plus grande expertise



# Principaux Fabricants de FPGA

- Xilinx
- Altera
- Lattice Semiconductor
- Actel (microsemi)
- Cypress
- Atmel
- QuickLogic





#### Différence entre un µC et un FPGA

Un μC:

- Exécute les instructions d'un programme de manière séquentielle.
- Un FPGA :
  - Description matérielle d'un système.
  - Un FPGA peut comporter un ou plusieurs μC ou μP, on parle alors de SoC ou SoPC.



# Qu'est ce qu'une IP?

- IP (Intellectual Propriety)

- composant virtuel
- Fonction décrit par un langage HDL
- Vendeur d'IPs :
  - ARM
  - www.design-reuse.com
- IPs open source :
  - www.opencores.org

# VHDL WWW

- VHDL sur Internet
  - http://vhdl33.free.fr/
  - https://www.doulos.com/knowhow/vhdl\_d esigners\_guide/
  - http://www.opencores.org/
    - Composants RTL open source (IPs)
  - http://www.freemodelfoundry.com/
    - Composants behaviour (modélisation) pour la simulation.