Langage VHDL et composants programmables

SIN1 - VHDL

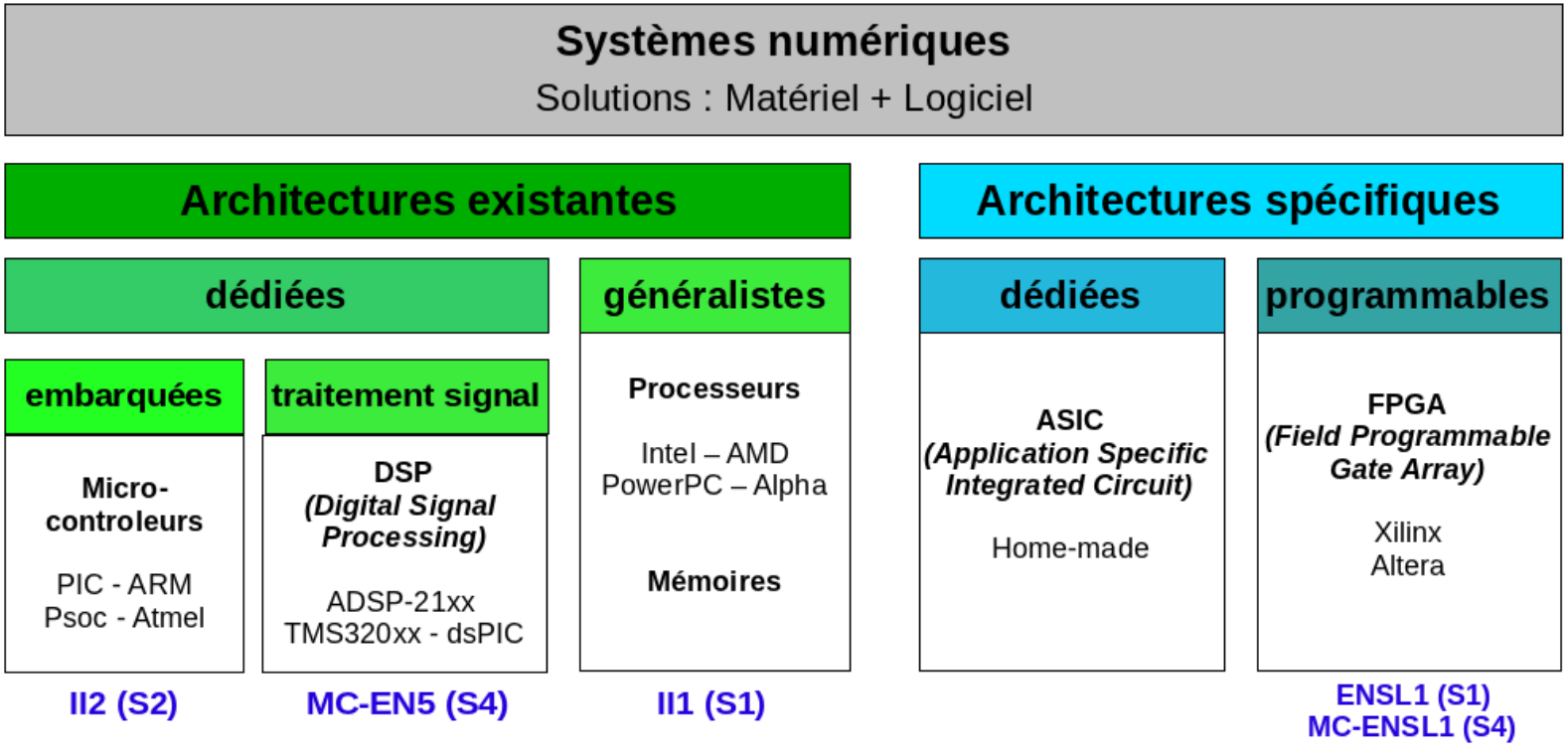
J. Villemejane - julien.villemejane@u-pec.fr

IUT Créteil-Vitry
Département GEII
Université Paris-Est Créteil

Année universitaire 2013-2014

Plan du cours

Systèmes numériques



Systèmes numériques

Systèmes numériques

Solutions : Matériel + Logiciel

Architectures existantes

dédiées

embarquées

Micro-controleurs

traitement signal

DSP *(Digital Signal Processing)*

généralistes

Processeurs

TRAITEMENTS SEQUENTIELS

Architectures spécifiques

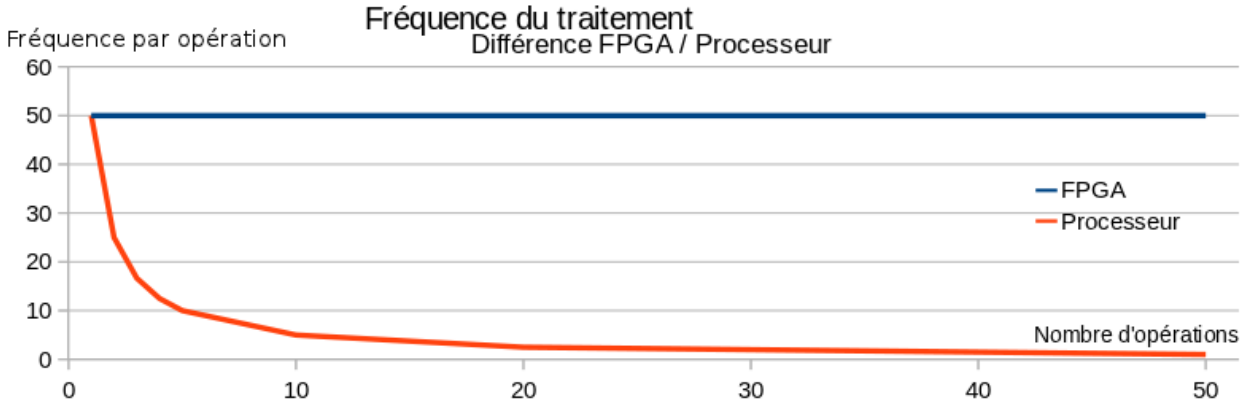
dédiées

ASIC *(Application Specific Integrated Circuit)*

programmables

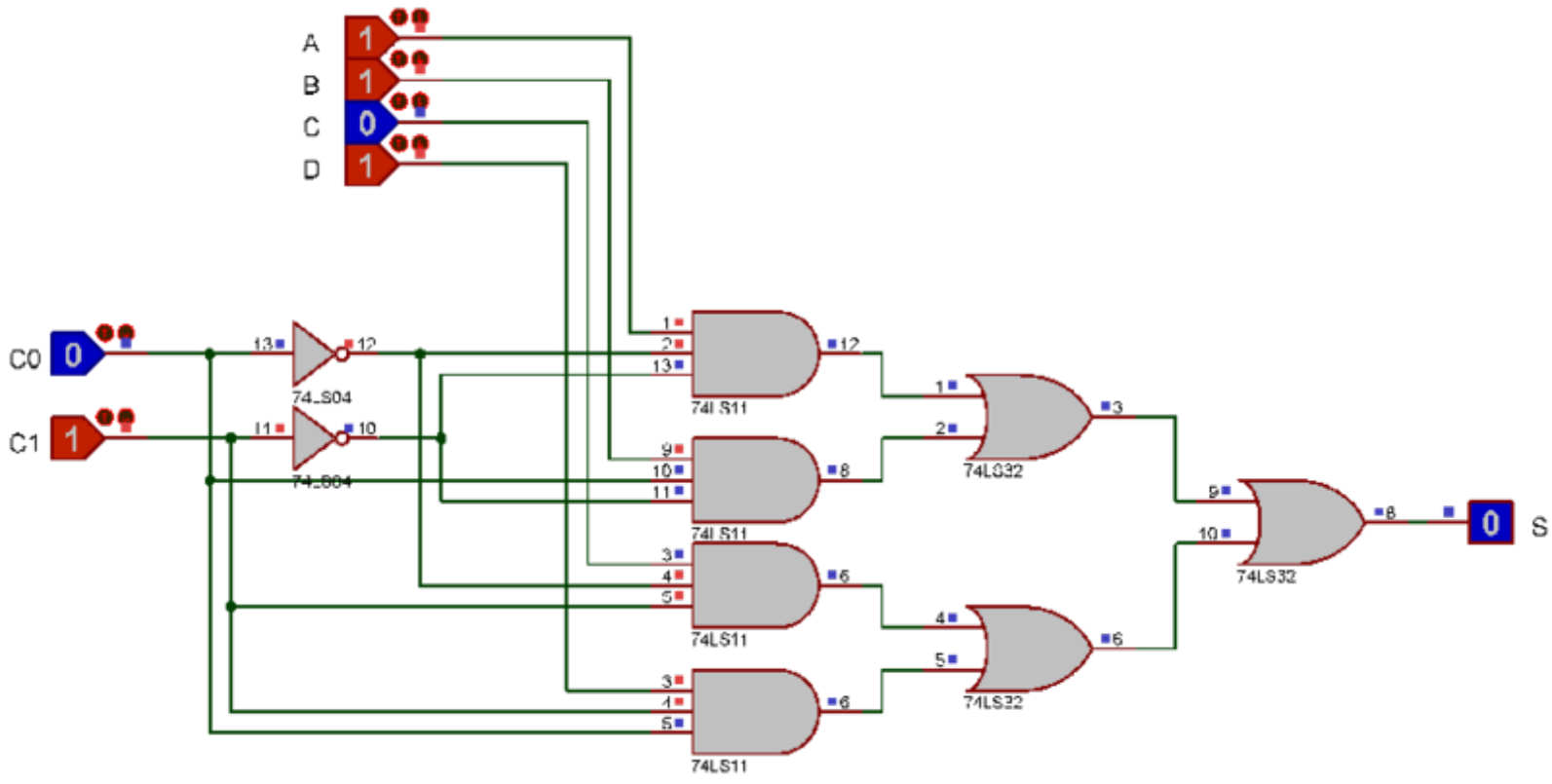
FPGA *(Field Programmable Gate Array)*

TRAITEMENTS PARALLELES



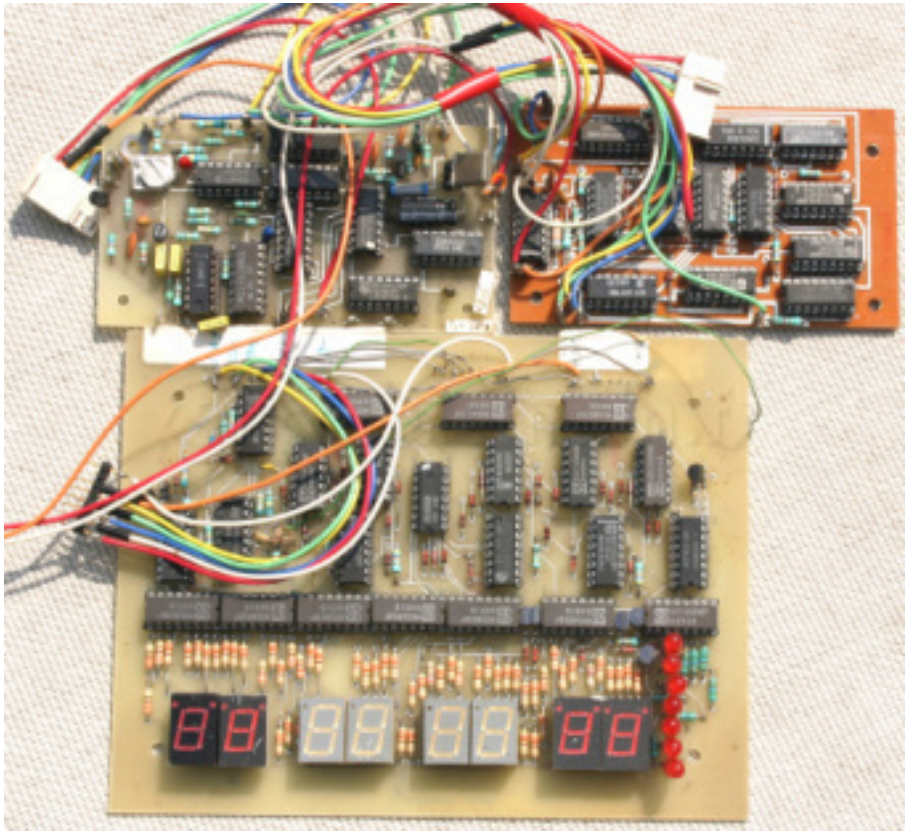
Synthèse structurelle

Structure logique



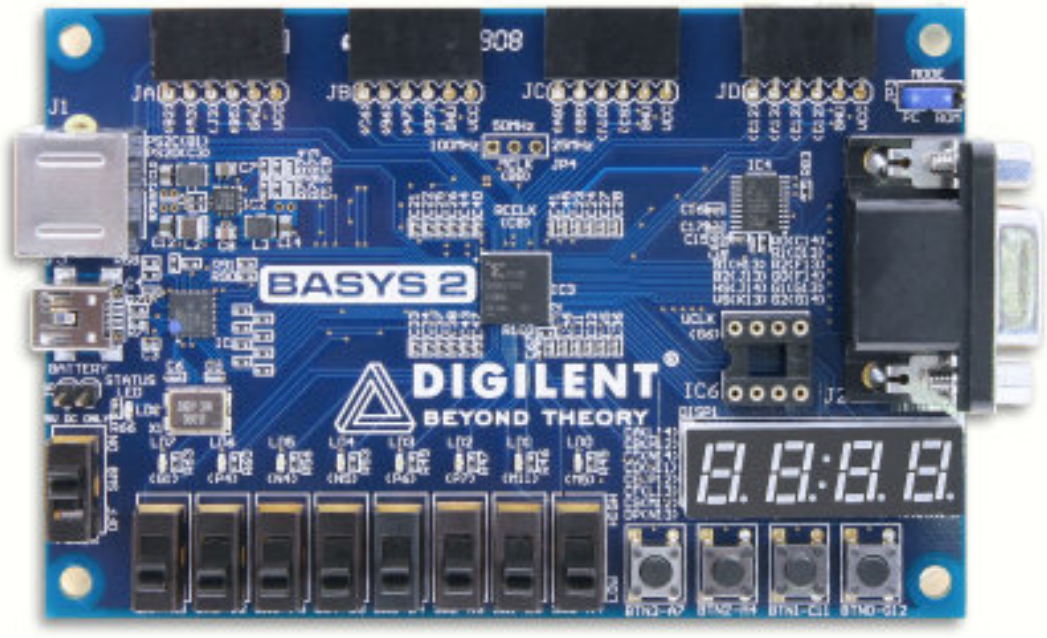
Synthèse structurelle

Réalisation pratique



Composant programmable

Réalisation pratique



Composant programmable

- Deux grands fabricants : Altera et Xilinx
- De **100.000** à **10.000.000 portes logiques**
- Composants destinés au **prototypage de systèmes numériques** complexes
- Bonne alternative aux circuits spécifiques, les **ASIC** (Application Specific Integrated Circuit), pour des petites ou moyennes séries

CPLD Complex Programmable Logic Device

FPGA Field Programmable Gate Array

ASIC Application Specific Integrated Circuit

Composant programmable

- Deux grands fabricants : Altera et Xilinx
- De **100.000** à **10.000.000 portes logiques**
- Composants destinés au **prototypage de systèmes numériques** complexes
- Bonne alternative aux circuits spécifiques, les **ASIC** (Application Specific Integrated Circuit), pour des petites ou moyennes séries

CPLD	Complex Programmable Logic Device
FPGA	Field Programmable Gate Array
ASIC	Application Specific Integrated Circuit

Composant programmable

Avantages et inconvénients

Avantages

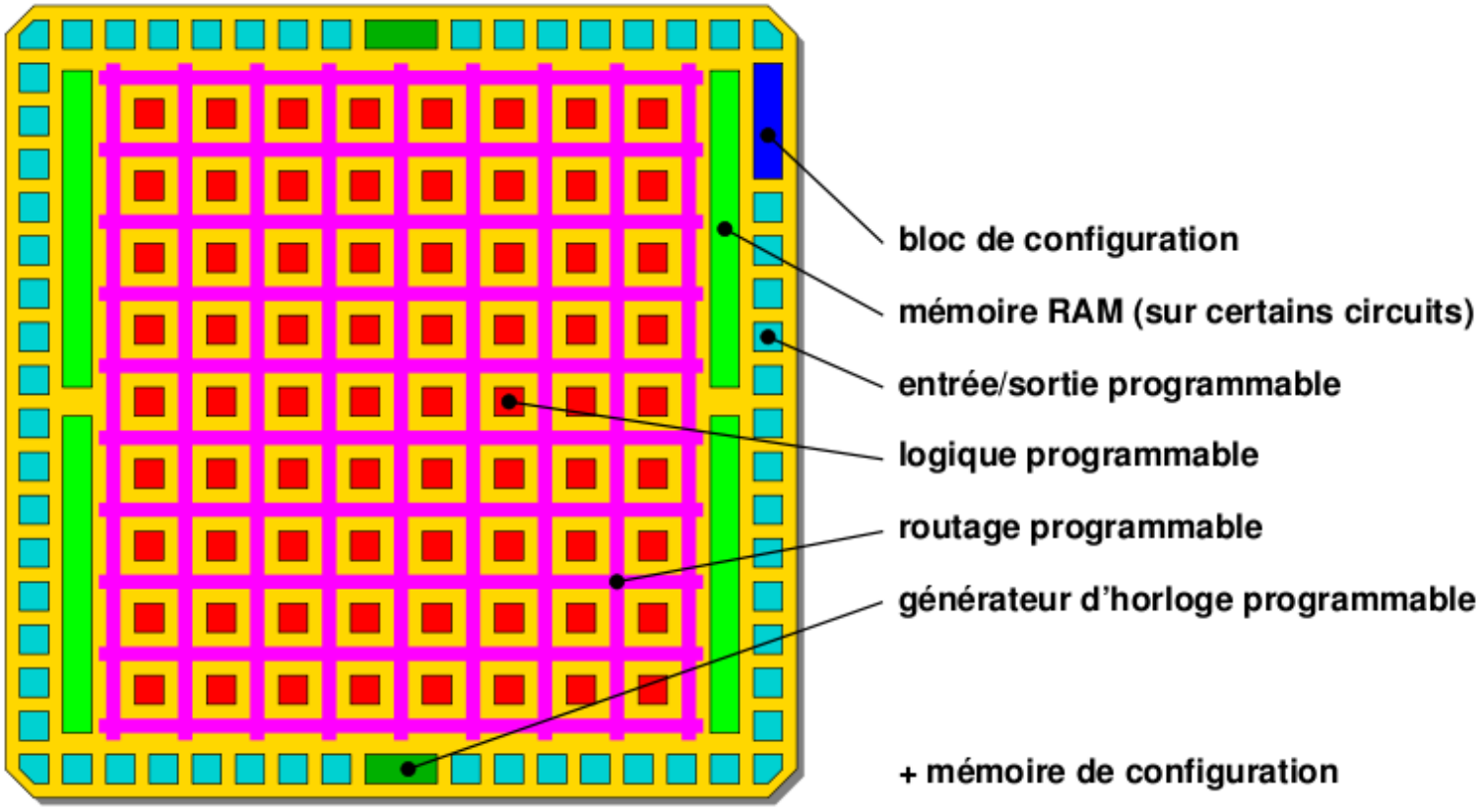
- Très haute densité
- Grande vitesse (100 MHz à quelques GHz)
- Très grand nombre d'entrées/sorties (boîtiers BGA)

Inconvénients

- Prix élevé (mais en baisse)
- Alimentation difficile (plusieurs tensions, courants élevés, connexions multiples)
- Volatiles (cellules RAM)
- Circuits imprimés (PCB - Printed Circuit Board) multicouches

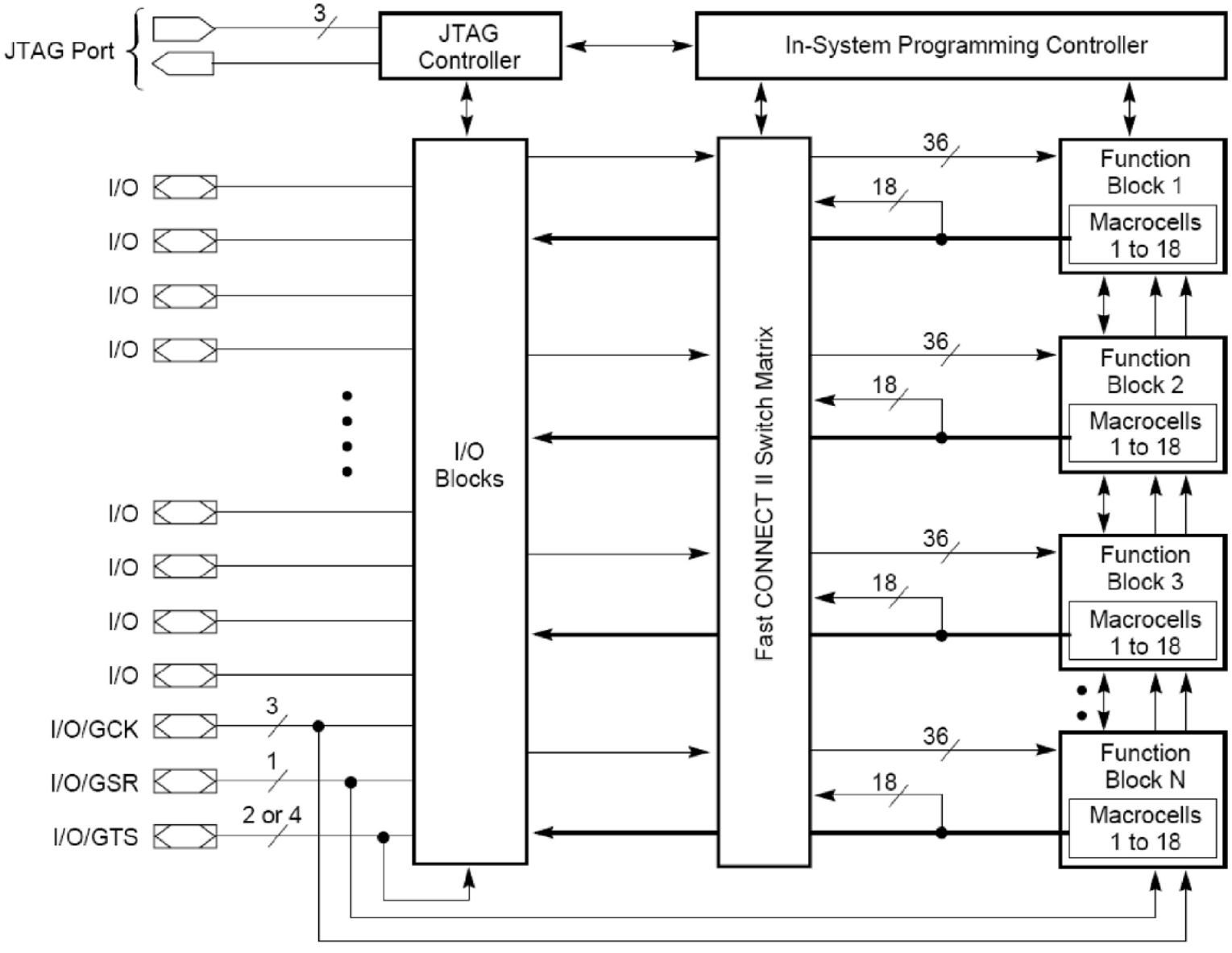
Composant programmable

Structure interne - FPGA - Xilinx



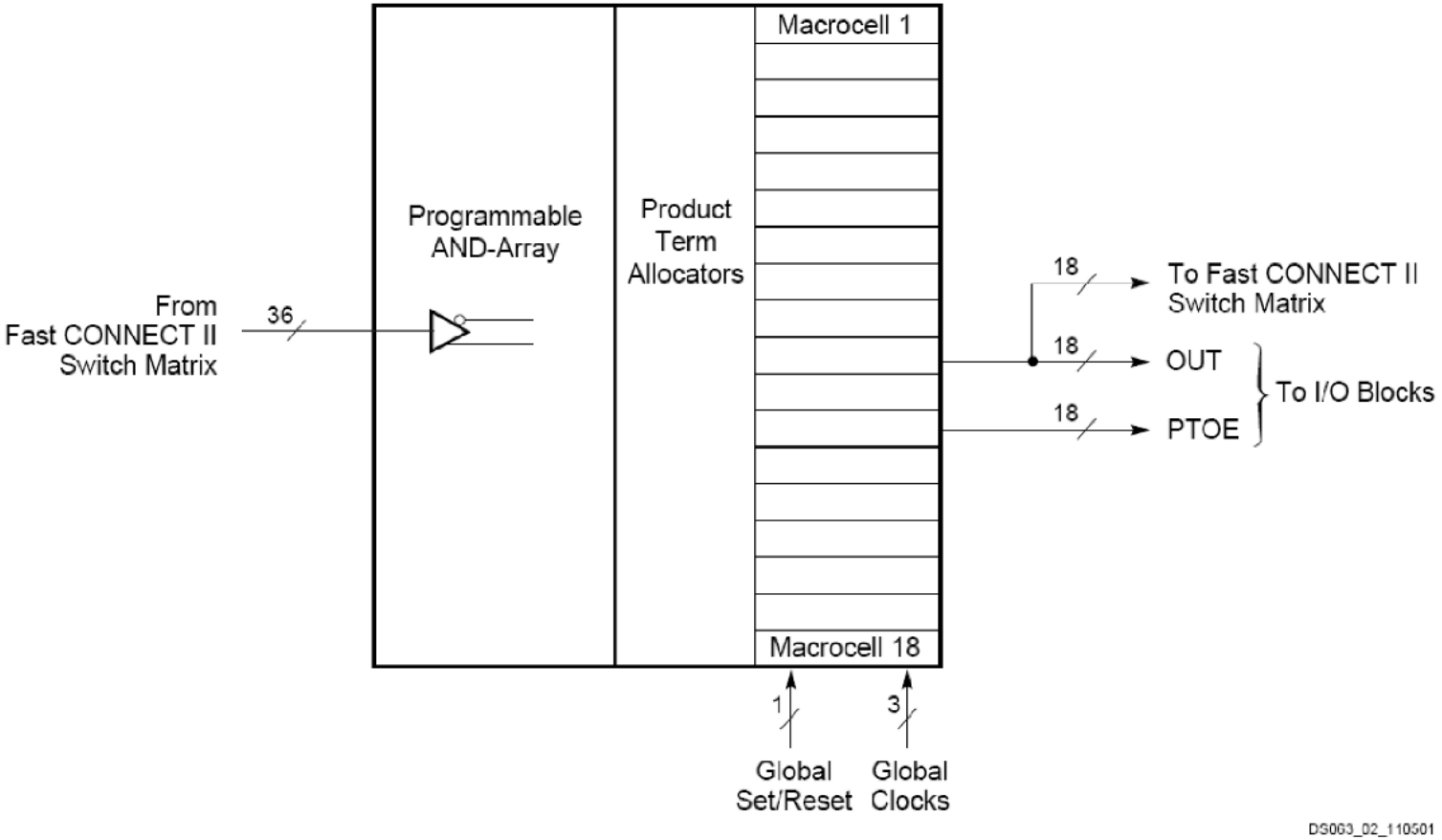
Composant programmable

Architecture du XC9500 - CPLD - Xilinx



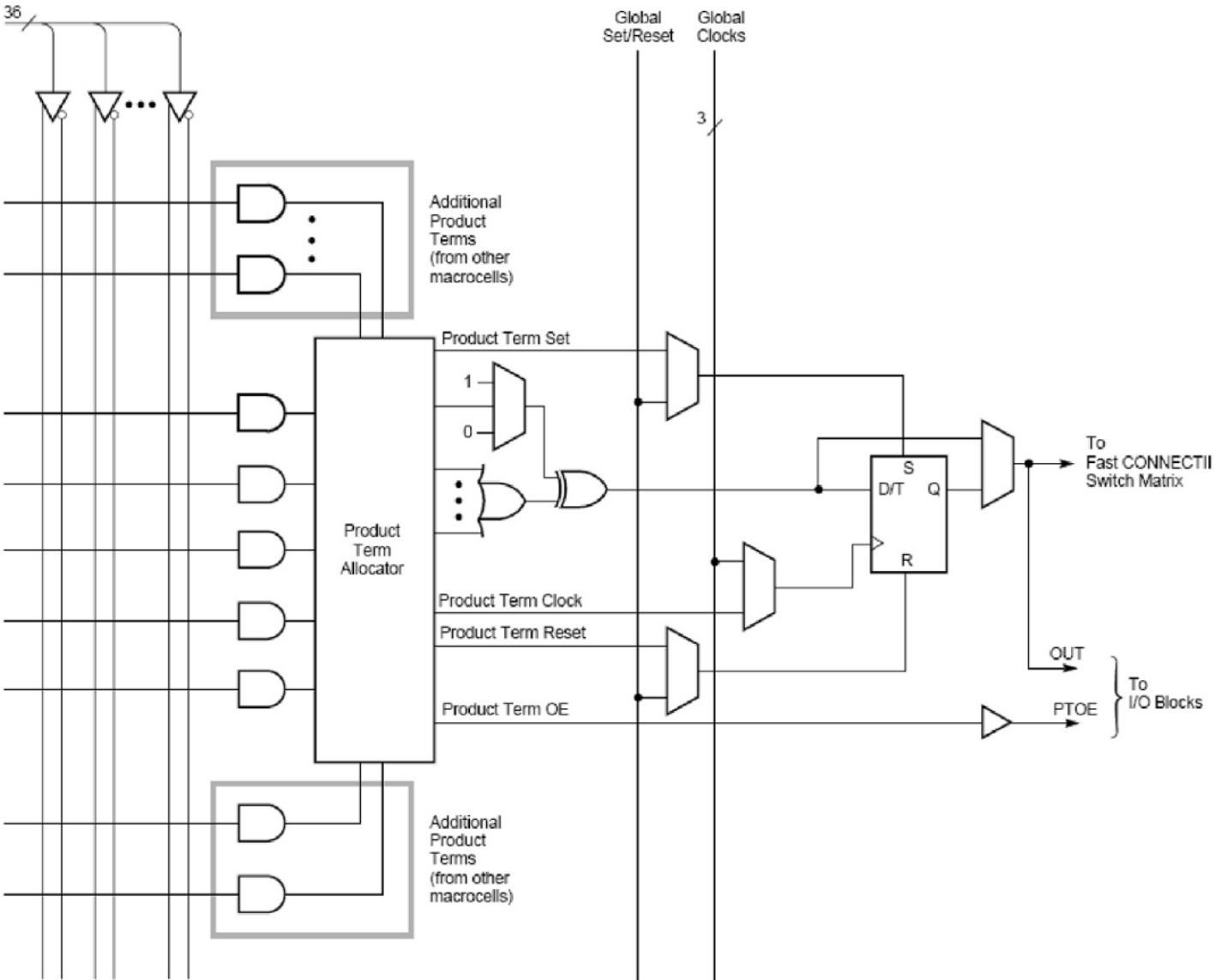
Composant programmable

Blocs fonctionnels



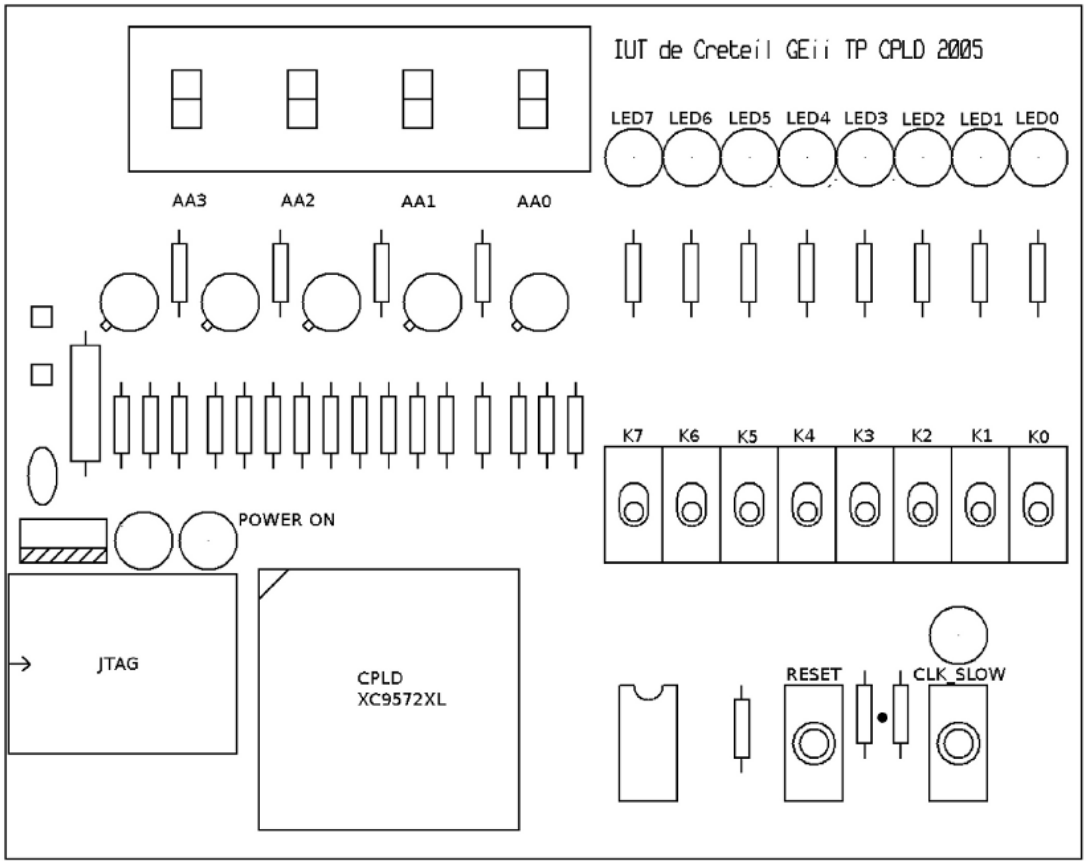
Composant programmable

Macro-cellules



Composant programmable

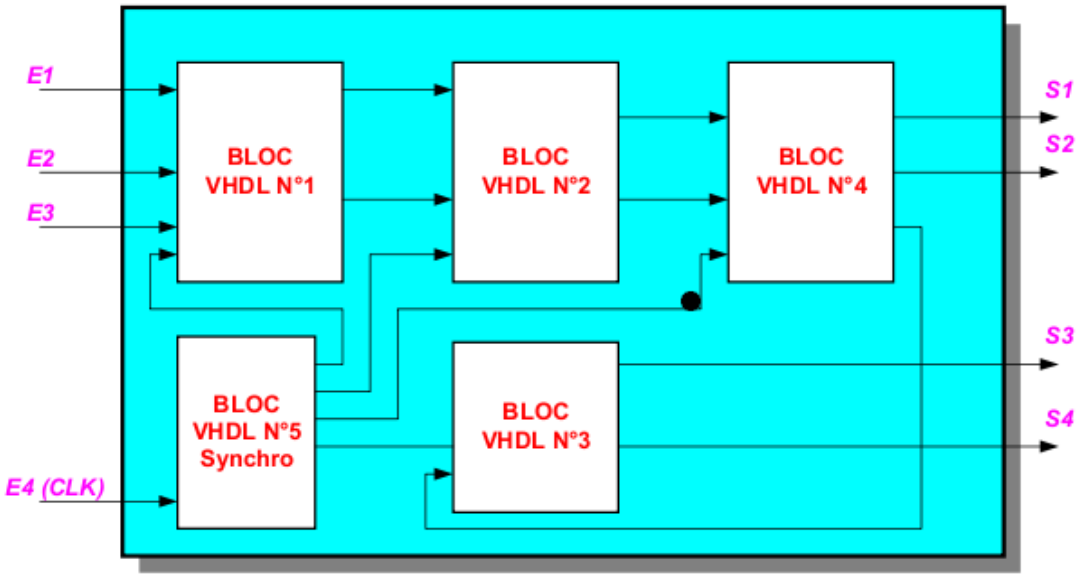
Carte d'étude



Pour plus d'informations, reportez-vous au support de TP.

Structure d'un projet

- Des **modules** VHDL (ou composants)
- Des **fichiers de simulation** (test-bench)
- Des **fichiers de contraintes**



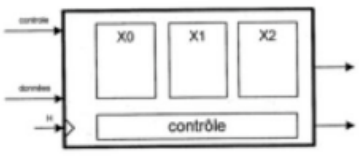
Phases de développement
1/2

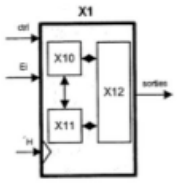
Specification (Définition globale)

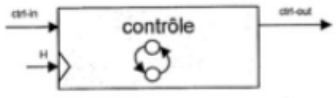
TOP

DOWN

Analyse fonctionnelle (sur papier!)





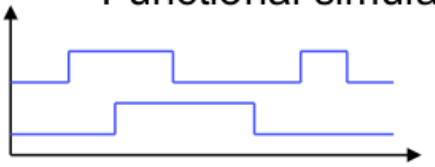


Description VHDL des blocs

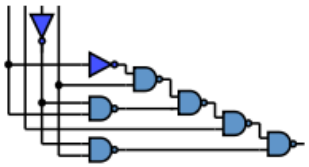
```
Library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity RC5_core is
        port(
            clock, reset, encr_decr: in std_logic;
            data_input: in std_logic_vector(31 downto 0);
            data_output: out std_logic_vector(31 downto 0);
            out_full: in std_logic;
            key_input: in std_logic_vector(31 downto 0);
            key_read: out std_logic;
        );
end AES_core;
```

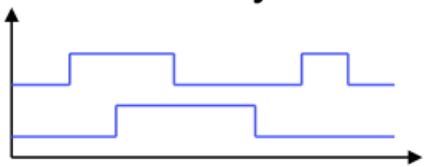
Functional simulation



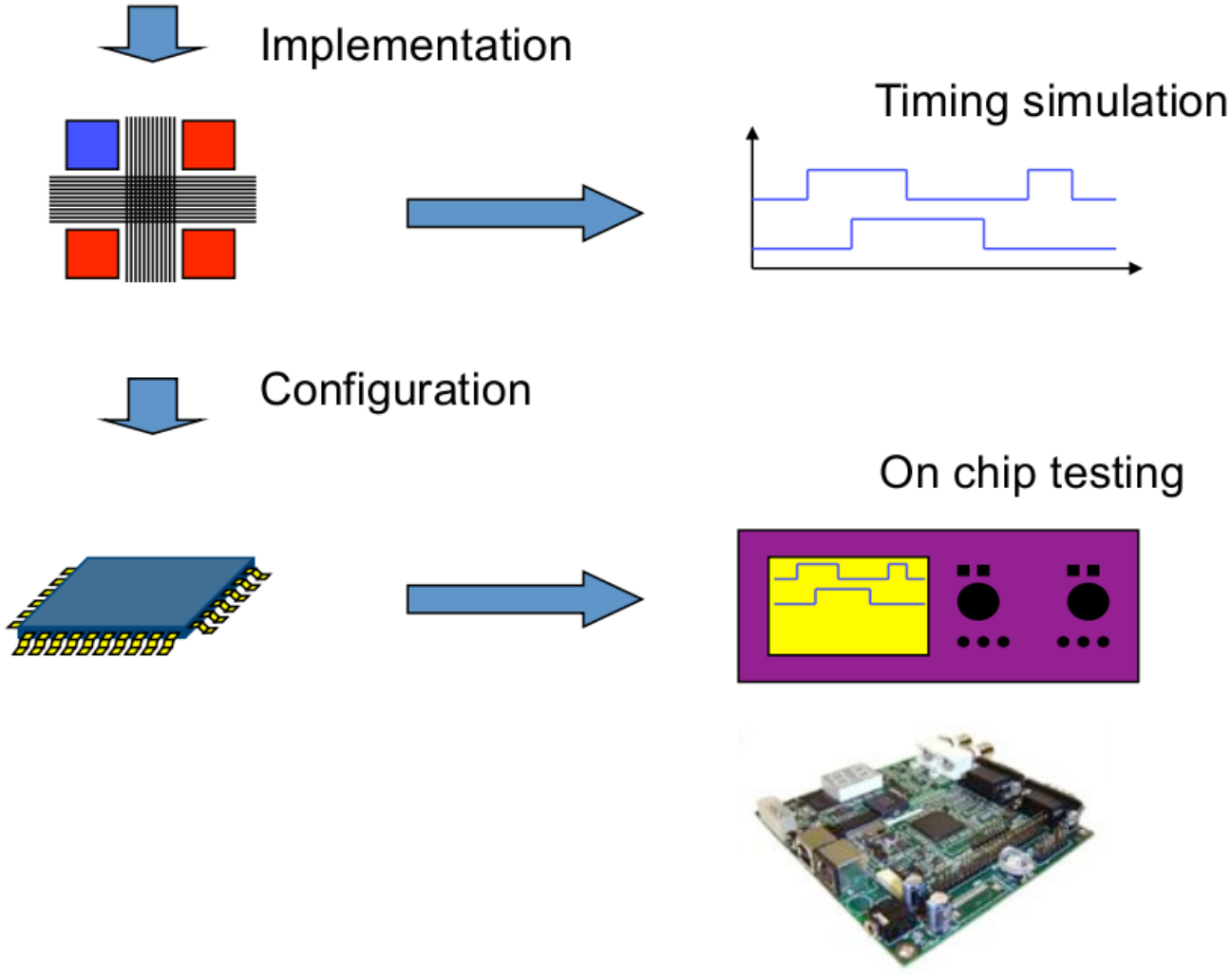
Synthese



Post-synthesis simulation



Phases de développement
2/2



Structure d'un module VHDL

La description d'un système numérique par le biais du langage VHDL passe par 3 étapes différentes :

- la déclaration des ressources externes (bibliothèques) ;
- la description de l'**entité** du système, correspondant à la liste des entrées/sorties ;
- la description de l'**architecture** du système, correspondant à la définition des fonctionnalités du système.

L'ensemble est contenu dans un fichier source portant l'extension `*.vhd`.

Structure d'un module VHDL

La description d'un système numérique par le biais du langage VHDL passe par 3 étapes différentes :

- la déclaration des ressources externes (bibliothèques) ;
- la description de l'**entité** du système, correspondant à la liste des entrées/sorties ;
- la description de l'**architecture** du système, correspondant à la définition des fonctionnalités du système.

L'ensemble est contenu dans un fichier source portant l'extension `*.vhd`.

Structure d'un module VHDL

Déclaration des ressources externes

- Permet d'**inclure des librairies** de types prédéfinis ou fonctions
- Réalisée automatiquement pour les bibliothèques courantes

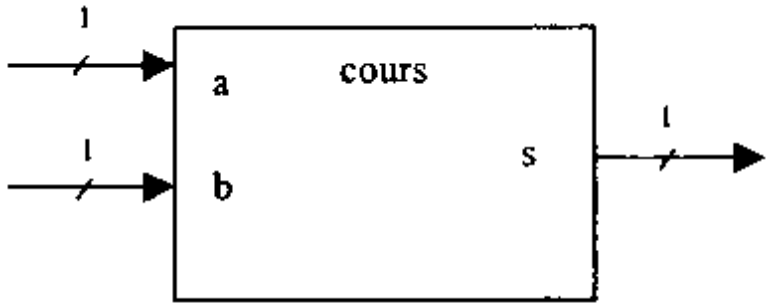
On retrouve en en-tête du fichier source `*.vhd` les instructions suivantes :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

Structure d'un module VHDL

Entité

- Description des **ports d'entrées / sorties** (avec sa direction - *in*, *out*, *inout*, *buffer* - et son type)




```
entity cours is
  port(
    a, b :  in STD_LOGIC; -- commentaire
    s :    out STD_LOGIC
  );
end cours;
```

Structure d'un module VHDL

Architecture

- Reliée à une entité
- Description du **fonctionnement** du système
  - *comportementale*
  - *structurelle*

```
architecture Behavorial of cours is
-- declaration des signaux
begin
  processus1;
  processus2;
  ...
end Behavorial;
```

Objets et types en VHDL

Objets et opérateurs

Objets

- **signal** objet physique, associé à des évènements
- **variable** intermédiaire de calcul, non physique
- **constant**

Opérateurs

LOGIQUES : `and`, `nand`, `or`, `nor`, `xor`, `xnor`, `not`
DÉCALAGE : `sll`, `slr`, `sla`, `sra`, `rol`, `ror`
RELATIONNELS : `=`, `/=`, `<`, `>`, `<=`, `>=`
ARITHMÉTIQUES : `+`, `-`, `*`, `/`, `MOD`
CONCATENATION : **&**
AFFECTATION : **¡=**

Objets et types en VHDL

Types et notations

Types

Types de base : **bit**, **bit_vector**, **integer**, **boolean**
Types IEEE : **std_logic**, **std_logic_vector**, **signed**, **unsigned**
Types définis par l'utilisateur :

- type énuméré, exemple : `type jour is (lu, ma, me, je, ve, sa, di);` (souvent utilisé dans les machines à état)
- sous-type : `subtype octet is bit_vector(0 to 7);`

Notations

bit : `'0'` ou `'1'`
bit_vector : `"0100"`
ASCII : `"Texte"`
Décimal : `423`
Hexadécimal : `x"1A"`

Instructions en VHDL

Hors processus

Affectation simple

```
x <= a;
```

Affectation conditionnelle

```
x <=  a when cond1 else
      b when cond2 else
      ...
      z;
```

Affectation sélective

```
with expr select
  x <=  a when val1,
        b when val2,
        ...
        z when others;
```

Instructions en VHDL

Dans un processus - Syntaxe

Processus

Label : `process`(liste des signaux de sensibilité)
Nom des objets internes : types ; – si nécessaire
`begin`
...
`end process;`

- Instructions exécutées **séquentiellement**
- Processus **activé** par un des signaux de la **liste de sensibilité**

Instructions en VHDL

Dans un processus

Structure IF... ELSIF ... ELSE ...

```
if x="00" then
  y <= '0';
elsif x="01" then
  y <= '1';
end if;
```

Structure CASE

```
case x is
  when "00" => y <= "00";
  when "01" => y <= "10";
  when others => y <= "11";
end case;
```

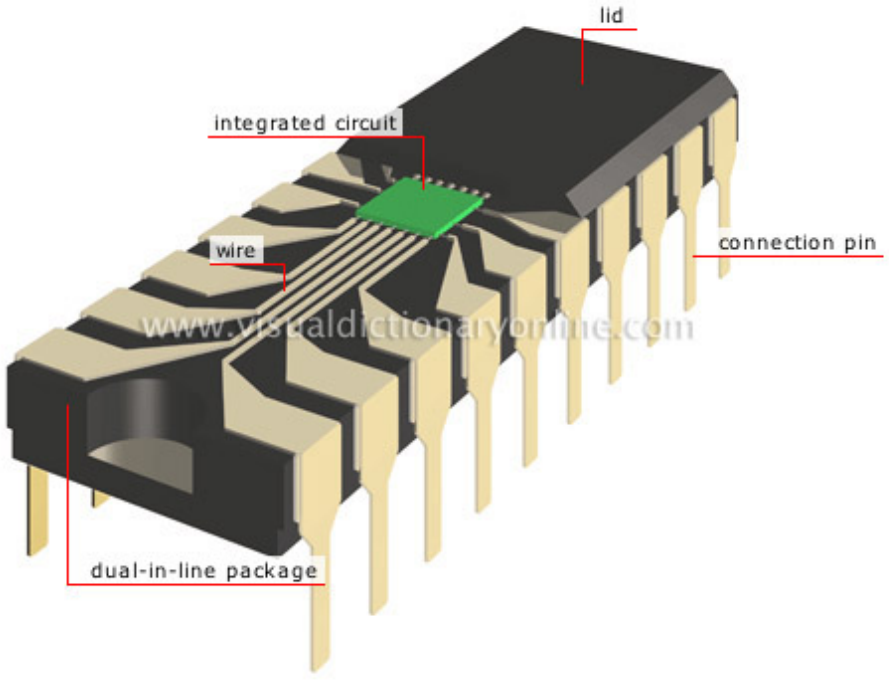
Quelques circuits intégrés

Intégration de transistors

Date	Nom	Nombre de transistors	Finesse de gravure (µm)	Fréquence de l'horloge	Largeur des données	MIPS
1971	4004	2 300		108 kHz	4 bits/4 bits bus	
1974	8080	6 000	6	2 MHz	8 bits/8 bits bus	0,64
1979	8088	29 000	3	5 MHz	16 bits/8 bits bus	0,33
1982	80286	134 000	1,5	6 à 16 MHz (20 MHz chez AMD)	16 bits/16 bits bus	1
1985	80386	275 000	1,5	16 à 40 MHz	32 bits/32 bits bus	5
1989	80486	1 200 000	1	16 à 100 MHz	32 bits/32 bits bus	20
1993	Pentium	3 100 000	0,8 à 0,28	60 à 233 MHz	32 bits/64 bits bus	100
1997	Pentium II	7 500 000	0,35 à 0,25	233 à 450 MHz	32 bits/64 bits bus	300
1999	Pentium III	9 500 000	0,25 à 0,13	450 à 1 400 MHz	32 bits/64 bits bus	510
2000	Pentium 4	42 000 000	0,18 à 0,065	1,3 à 3,8 GHz	32 bits/64 bits bus	1 700
2004	Pentium 4D « Prescott »	125 000 000	0,09 à 0,065	2.66 à 3,6 GHz	32 bits/64 bits bus	9 000
2006	Core 2™ Duo	291 000 000	0,065	2,4 GHz (E6600)	64 bits/64 bits bus	22 000
2007	Core 2™ Quad	2*291 000 000	0,065	3 GHz (Q6850)	64 bits/64 bits bus	2*22 000 (?)
2008	Core 2™ Duo (Penryn)	410 000 000	0,045	3,33 GHz (E8600)	64 bits/64 bits bus	~24 200
2008	Core 2™ Quad (Penryn)	2*410 000 000	0,045	3,2 GHz (QX9770)	64 bits/64 bits bus	~2*24 200
2008	Intel Core i7 (Nehalem)	731 000 000	0,045 (2008) 0,032 (2009)	2,66 GHz (Core i7 920) 3,33 GHz (Core i7 Ext. Ed. 975)	64 bits/64 bits bus	?
2009	Intel Core i5/i7 (Lynnfield)	774 000 000	0,045 (2009)	2,66 GHz (Core i5 750) 2,93 GHz (Core i7 870)	64 bits/64 bits bus	76383
2010	Intel Core i7 (Gulftown)	1 170 000 000	0,032	3,33 GHz (Core i7 980X)	64 bits/64 bits bus	147600

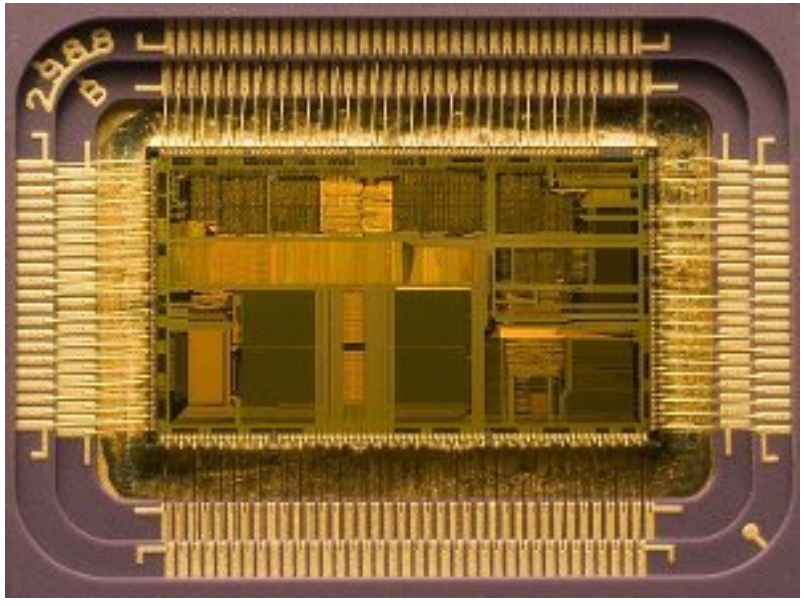
Quelques circuits intégrés

Circuit intégré



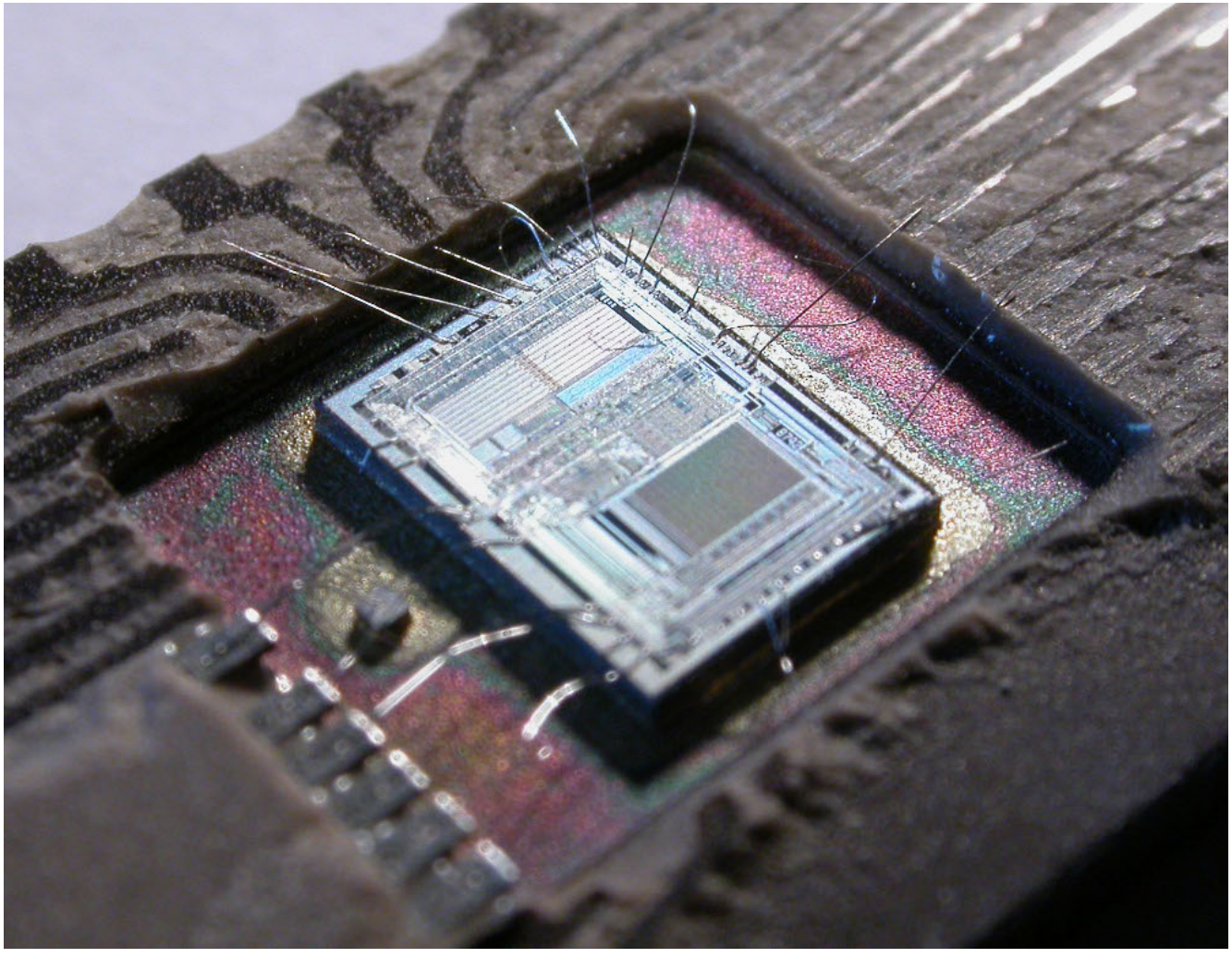
Quelques circuits intégrés

Processeur Intel



Quelques circuits intégrés

Processeur Intel



Quelques circuits intégrés

Processeur IBM