



Une école de l'IMT

Introduction à SystemVerilog

Pratique de la logique combinatoire

Processeurs et Architectures Numériques





Langages de description du matériel

HDL (Hardware Description Languages)

- Besoin de normaliser/formaliser les descriptions
 - des structures
 - des comportements (équations booléennes)
- Représenter des signaux évoluant dans le temps
- Simuler le parallélisme intrinsèque au matériel

Langages de description du matériel

Simulation/synthèse

Un langage, le même code, deux utilisations et deux outils :

SIMULATEUR : Programme permettant de simuler. On génère des stimuli sur les entrées et on visualise les valeurs des sorties.

SYNTHÉTISEUR : Programme qui transforme la représentation HDL en un assemblage de ressources physiques (portes logiques par exemple).

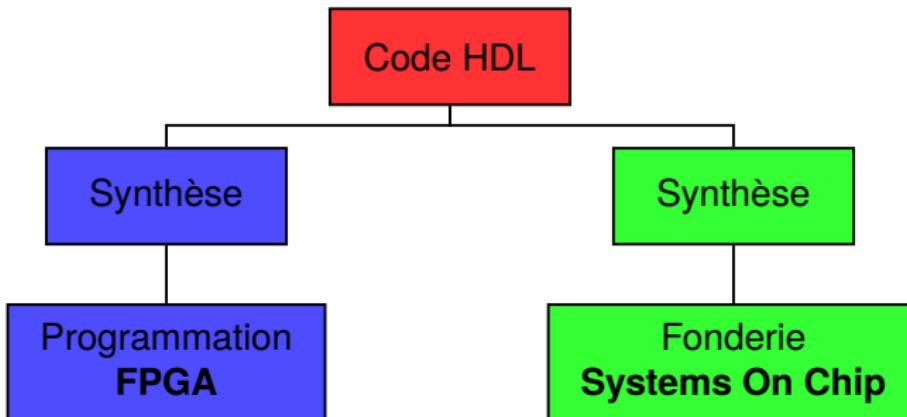
Langages de description du matériel

SoC/FPGA

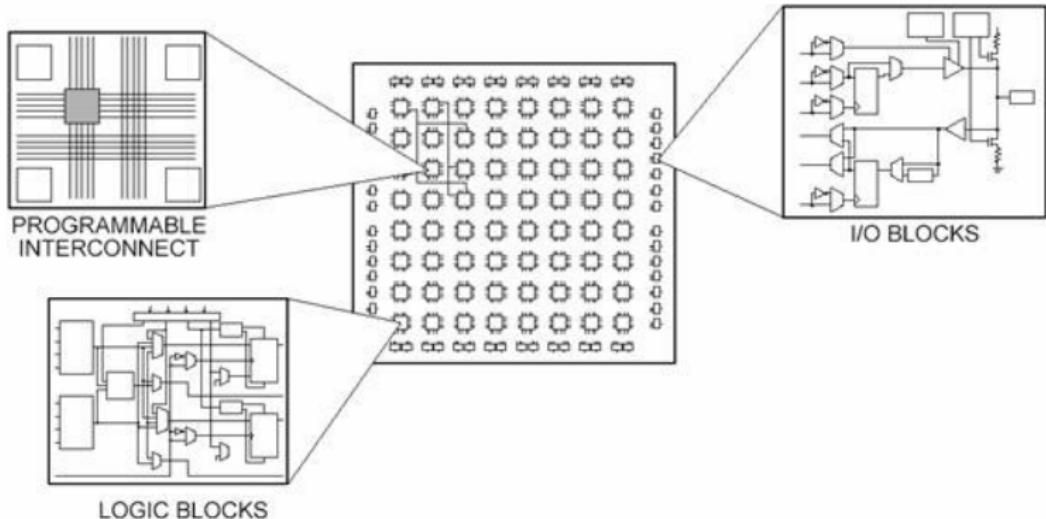
Un langage, le même code, deux cibles technologiques :

SoC System On Chip.

FPGA Field Programmable Gates Array



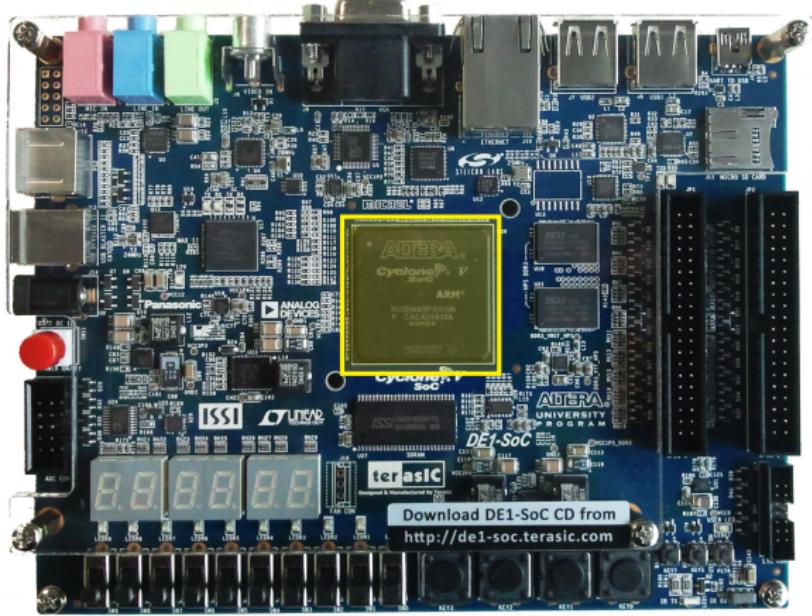
Qu'est ce qu'un FPGA ?



National Instrument : <http://www.ni.com/white-paper/6983/fr>

La maquette FPGA

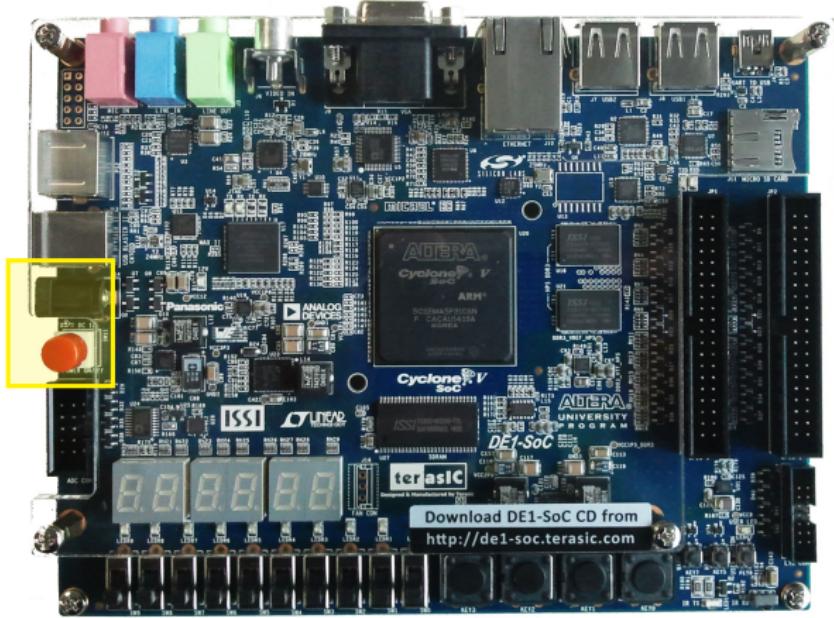
Terasic/Altera DE1-SoC



■ Le FPGA (Altera CycloneV-SoC)

La maquette FPGA

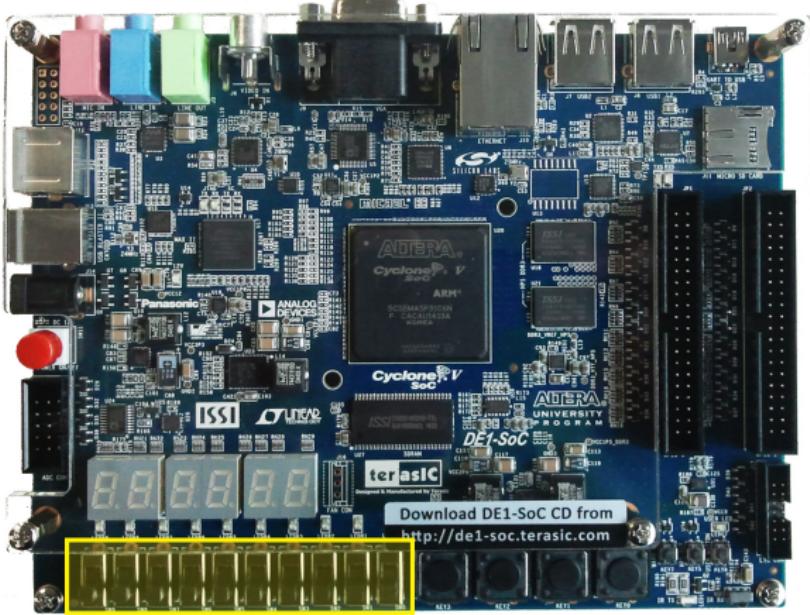
Terasic/Altera DE1-SoC



■ L'alimentation électrique

La maquette FPGA

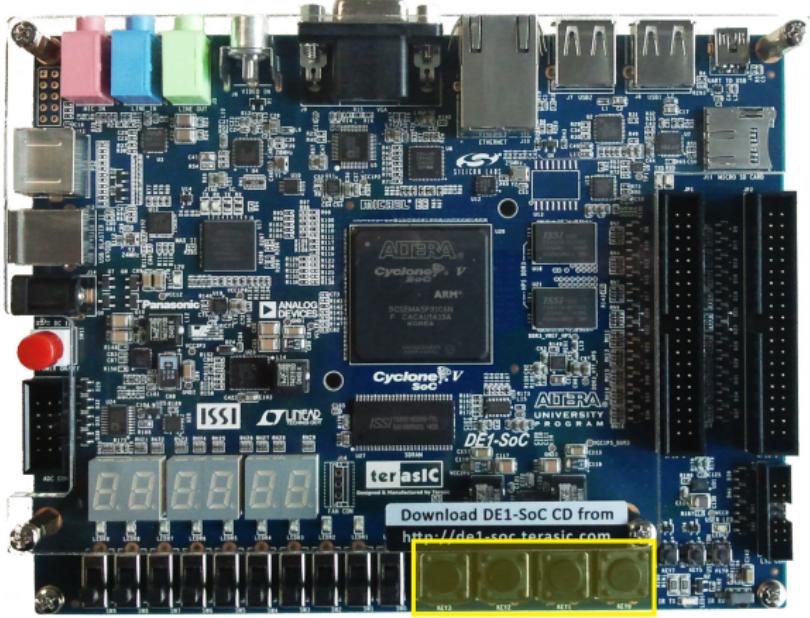
Terasic/Altera DE1-SoC



■ Des interrupteurs (sw)

La maquette FPGA

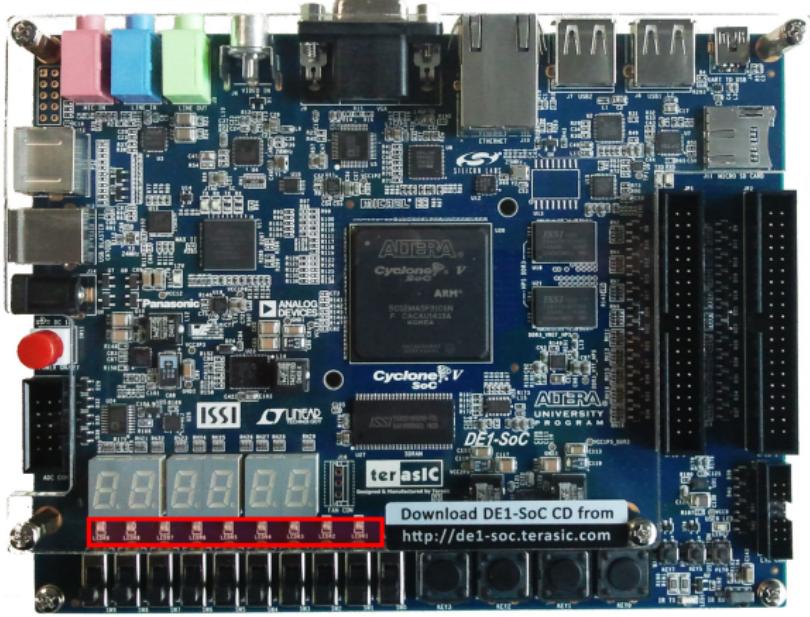
Terasic/Altera DE1-SoC



- Des boutons poussoirs (key)

La maquette FPGA

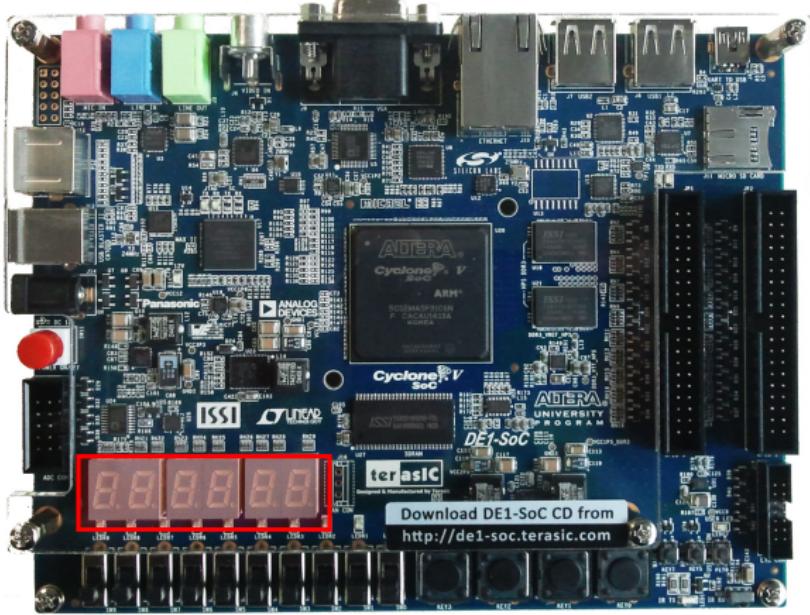
Terasic/Altera DE1-SoC



■ Des LEDs rouges

La maquette FPGA

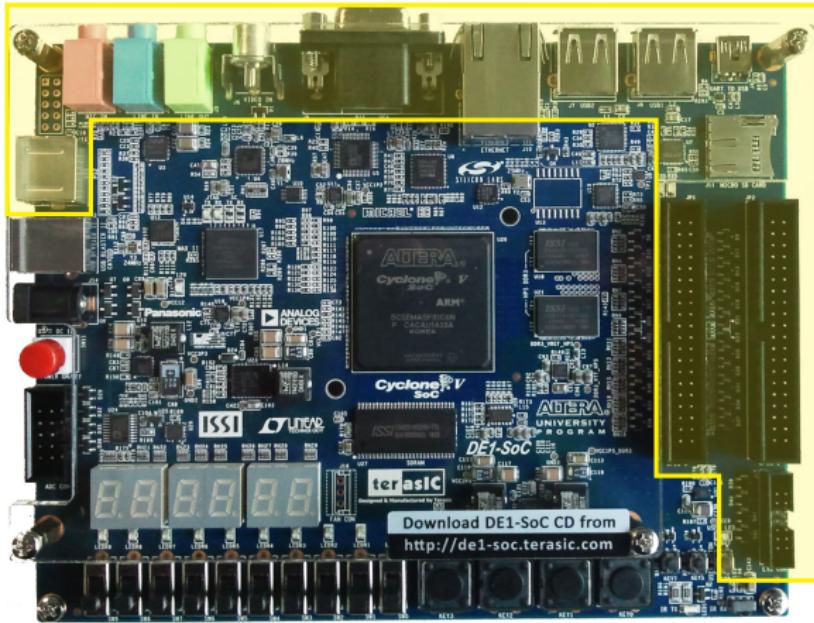
Terasic/Altera DE1-SoC



- Des afficheurs 7 segments

La maquette FPGA

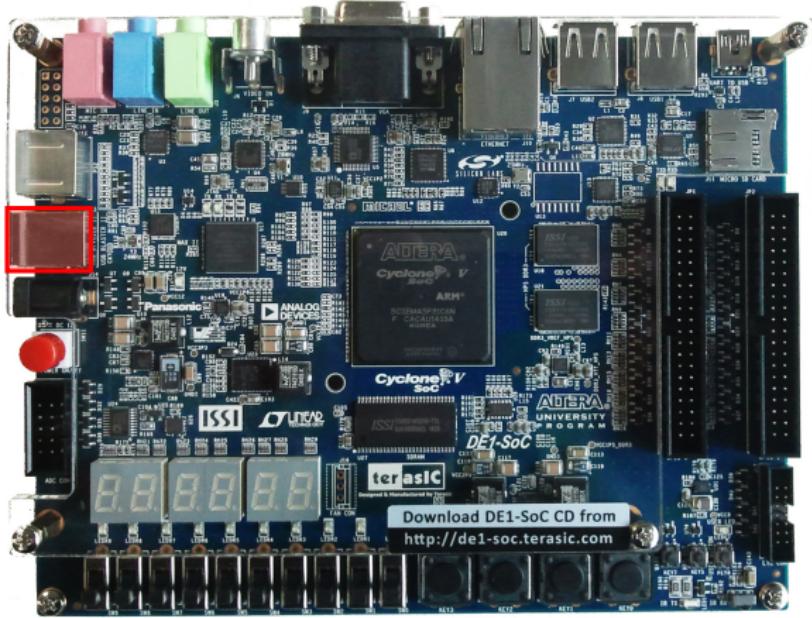
Terasic/Altera DE1-SoC



- D'autres Entrées/Sorties (USB, VGA, Audio, Ethernet, etc.)

La maquette FPGA

Terasic/Altera DE1-SoC



- Interface USB de programmation



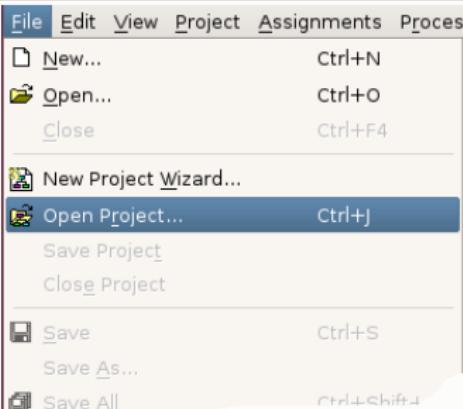
Comment utiliser la maquette ?

- Introduction à SystemVerilog
 - <https://sen.enst.fr/elecinf102/IntroductionSystemVerilog>
- Téléchargez l'archive du projet Quartus
- Décompressez l'archive
 - Dans un terminal “`tar xjvf bac_a_sable.tar.bz2`”
- Lancez Quartus
 - Toujours dans un terminal “`quartus &`”

Comment utiliser la maquette ?

Projet Quartus

Ouvrir le projet

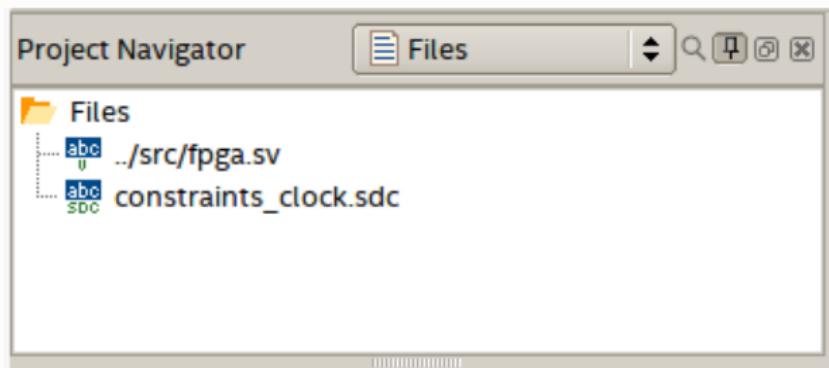


- Fichier **fpga.qpf**
- Description de la maquette
 - le modèle de FPGA
 - les entrées/sorties
- Fichiers de HDL (*SystemVerilog*)

Comment utiliser la maquette ?

fpga.sv

Ouvrir le fichier



Comment utiliser la maquette ?

fpga.sv

```
'default_nettype none

module fpga (
    ///////////////////// Push Button /////////////////////
    input logic [3:0] key,           // Pushbutton[3:0]
    ///////////////////// DPDT Switch ///////////////////
    input logic [9:0] sw,            // Toggle Switch[9:0]
    ///////////////////// 7-SEG Dispaly ///////////////////
    output logic [6:0] hex0,         // Seven Segment Digit 0
    output logic [6:0] hex1,         // Seven Segment Digit 1
    output logic [6:0] hex2,         // Seven Segment Digit 2
    output logic [6:0] hex3,         // Seven Segment Digit 3
    output logic [6:0] hex4,         // Seven Segment Digit 4
    output logic [6:0] hex5,         // Seven Segment Digit 5
    ///////////////////// LED /////////////////////
    output logic [9:0] ledr          // LED Red[9:0]
);

// ajouter votre code à partir d'ici

endmodule
```

Comment utiliser la maquette ?

fpga.sv

```
'default_nettype none

module fpga (
    ////////////////// Push Button //////////////////
    input logic [3:0] key,           // Pushbutton[3:0]
    ////////////////// DPDT Switch //////////////////
    input logic [9:0] sw,            // Toggle Switch[9:0]
    ////////////////// 7-SEG Dispaly //////////////////
    output logic [6:0] hex0,         // Seven Segment Digit 0
    output logic [6:0] hex1,         // Seven Segment Digit 1
    output logic [6:0] hex2,         // Seven Segment Digit 2
    output logic [6:0] hex3,         // Seven Segment Digit 3
    output logic [6:0] hex4,         // Seven Segment Digit 4
    output logic [6:0] hex5,         // Seven Segment Digit 5
    ////////////////// LED //////////////////
    output logic [9:0] ledr          // LED Red[9:0]
);
// ajouter votre code à partir d'ici

endmodule
```



Comment utiliser la maquette ?

fpga.sv

La description

```
'default_nettype none

module fpga (
    ///////////////////// Push Button /////////////////////
    input logic [3:0] key,                         // Pushbutton[3:0]
    ///////////////////// DPDT Switch ///////////////////
    input logic [9:0] sw,                           // Toggle Switch[9:0]
    ///////////////////// 7-SEG Dispaly ///////////////////
    output logic [6:0] hex0,                        // Seven Segment Digit 0
    output logic [6:0] hex1,                        // Seven Segment Digit 1
    output logic [6:0] hex2,                        // Seven Segment Digit 2
    output logic [6:0] hex3,                        // Seven Segment Digit 3
    output logic [6:0] hex4,                        // Seven Segment Digit 4
    output logic [6:0] hex5,                        // Seven Segment Digit 5
    ///////////////////// LED /////////////////////
    output logic [9:0] ledr,                        // LED Red[9:0]
);
// ajouter votre code à partir d'ici
endmodule
```

Comment utiliser la maquette ?

Premier module

```
'default_nettype none

module fpga (
    /////////////////////
    input logic [3:0] key,
    /////////////////////
    input logic [9:0] sw,
    /////////////////////
    output logic [6:0] hex0,
    output logic [6:0] hex1,
    output logic [6:0] hex2,
    output logic [6:0] hex3,
    output logic [6:0] hex4,
    output logic [6:0] hex5,
    /////////////////////
    output logic [9:0] ledr
);

    always @(*) ledr <= sw;

endmodule
```

- Les LEDs rouges sont reliées aux interrupteurs
- Fonction identité

Comment utiliser la maquette ?

Compiler/programmer

Compilez

- en appuyant sur le bouton



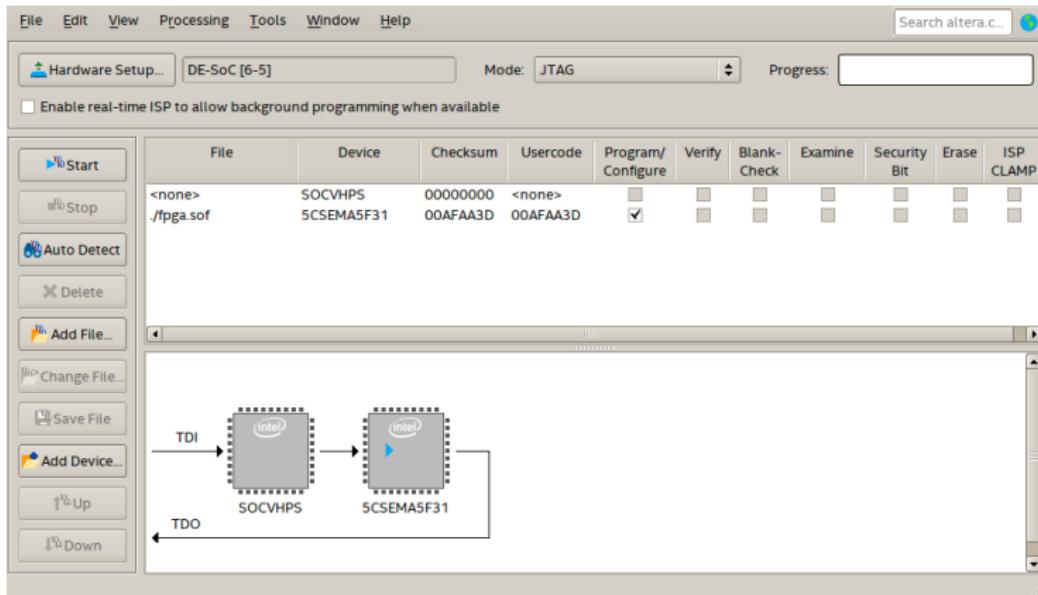
Lancez le programmeur

- en appuyant sur le bouton



Comment utiliser la maquette ?

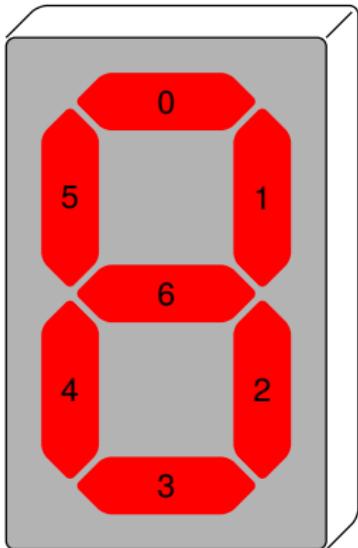
Le “programmeur”



Exercice

L'afficheur 7 segments

Le décodeur 7 segments



- Permet d'afficher de l'hexadécimal ($0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$)
- L'entrée est sur 4 bits ($0 \rightarrow F$)
- La sortie est sur 7 bits
 - Chaque bit contrôle un segment
 - Si le bit est à 0 le segment est allumé
- Donnez la table de vérité
- ...sur la maquette ...