



Développez vos systèmes embarqués sur SoC FPGA

Comment embarquer Linux et développer en VHDL vos applicatifs dédiés au traitement d'image sans pénaliser le CPU.

Jean-Marie CODOL
Développeur

Submarine Open Technologies
Montpellier

Avril 2022



Développez vos systèmes embarqués sur SoC FPGA

Comment embarquer Linux et développer en VHDL vos applicatifs dédiés au traitement d'image sans pénaliser le CPU.

Partie 2

Comment programmer un SoC FPGA chez Intel

Comment un FPGA réalise l'implémentation d'un algorithme de traitement du signal

Fin d'introduction pour les non-développeurs FPGA

Plan

JOUR 1

§001 Faire communiquer Linux avec un FPGA sur le SoC intel DE10-nano

- distribution fournie par Altera
- créer une image SD (en partie sur un serveur distant)
- compilation croisée (eclipse CDT embedded)
- activer les bridges HPS <-> FPGA par un device tree
- utiliser le bridge HPS2FPGA

J 2

§002 Partager une zone de RAM entre Linux et un FPGA sur le SoC intel DE10-nano

- projet de lecture/écriture en RAM sur FPGA
- projet de lecture/écriture en RAM sur CPU

J 3

§003 Cas pratique avec seuillage d'image

- projet openCV
- échange sur mémoire RAM

Quand utiliser un SoC FPGA

Le SoC FPGA : Une proposition de résolution du problème d'occupation CPU



Qu'est-ce qu'un SoC FPGA ?

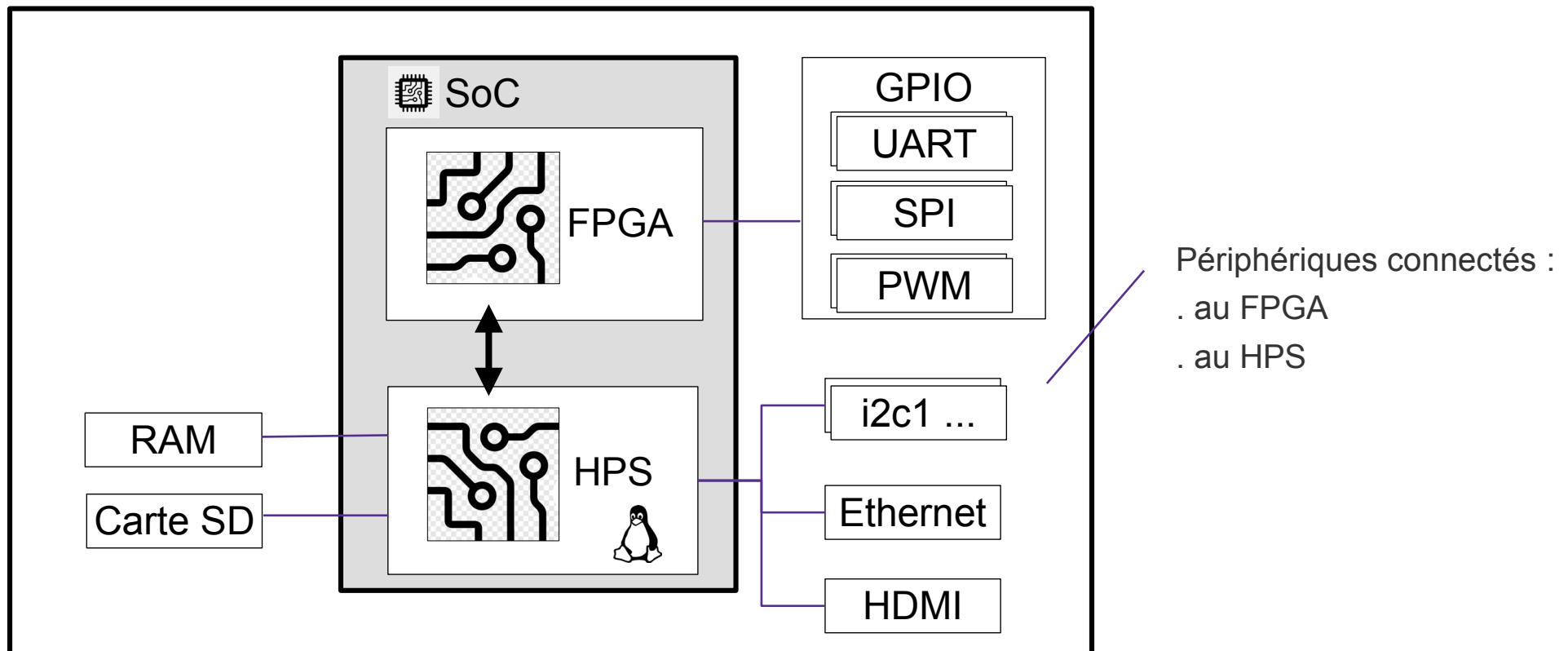
Les appareils SoC FPGA intègrent les architectures de processeur et de FPGA dans un seul appareil.

L'intégration de la fonctionnalité de gestion de haut niveau des processeurs et des opérations rigoureuses en temps réel, du traitement extrême des données ou des fonctions d'interface d'un FPGA (Field Programmable Gate Array) dans un seul appareil permet d'obtenir une plateforme informatique embarquée encore plus puissante.

Par conséquent, ils offrent une meilleure intégration, une faible puissance, une taille de carte réduite et une communication à bande passante plus élevée entre le processeur et le FPGA. Ils disposent également de nombreux périphériques, d'une mémoire sur puce, d'un réseau logique de type FPGA et d'émetteurs-récepteurs à haut débit.

Quand utiliser un SoC FPGA

Le SoC FPGA : Une proposition de résolution du problème d'occupation CPU



Quand utiliser un SoC FPGA

Le SoC FPGA : Une proposition de résolution du problème d'occupation CPU

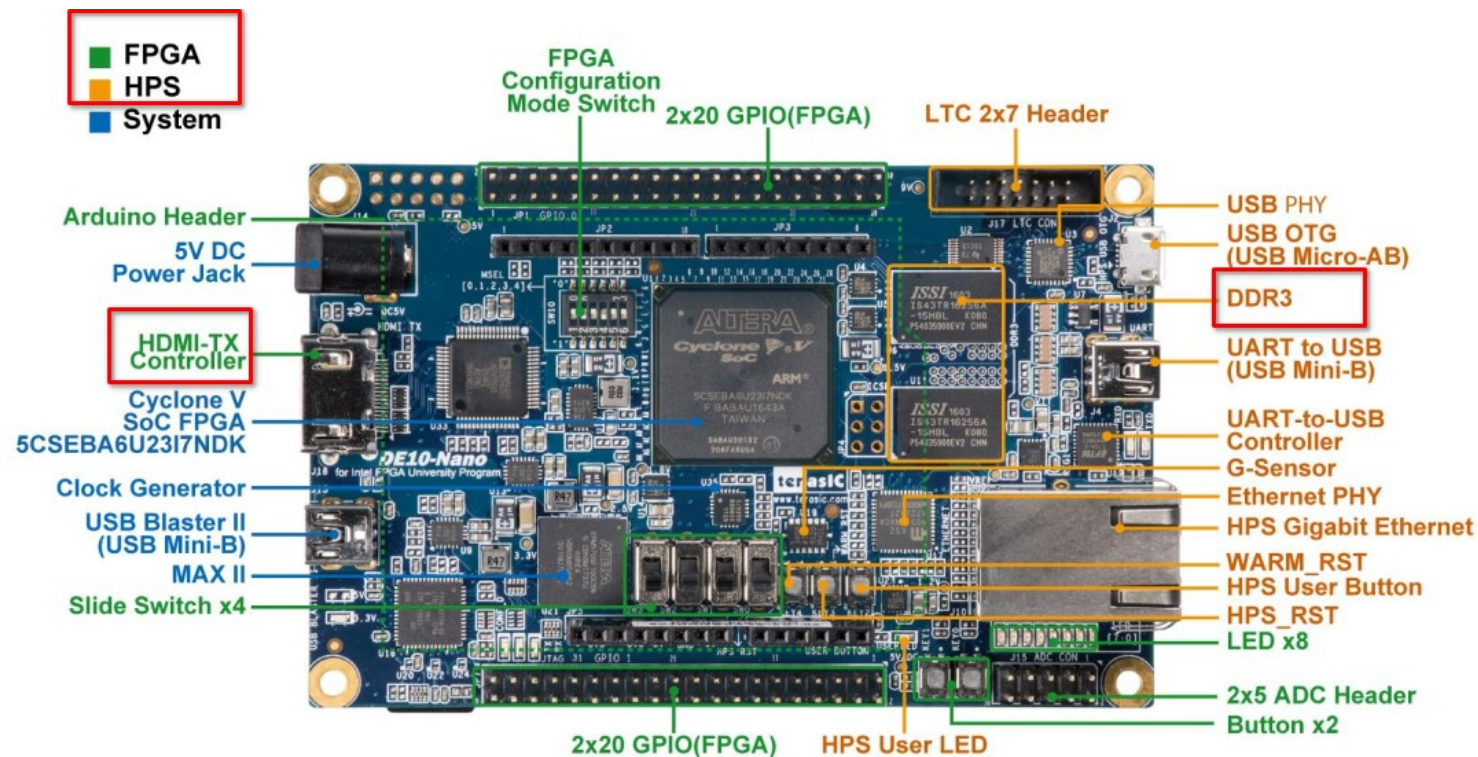
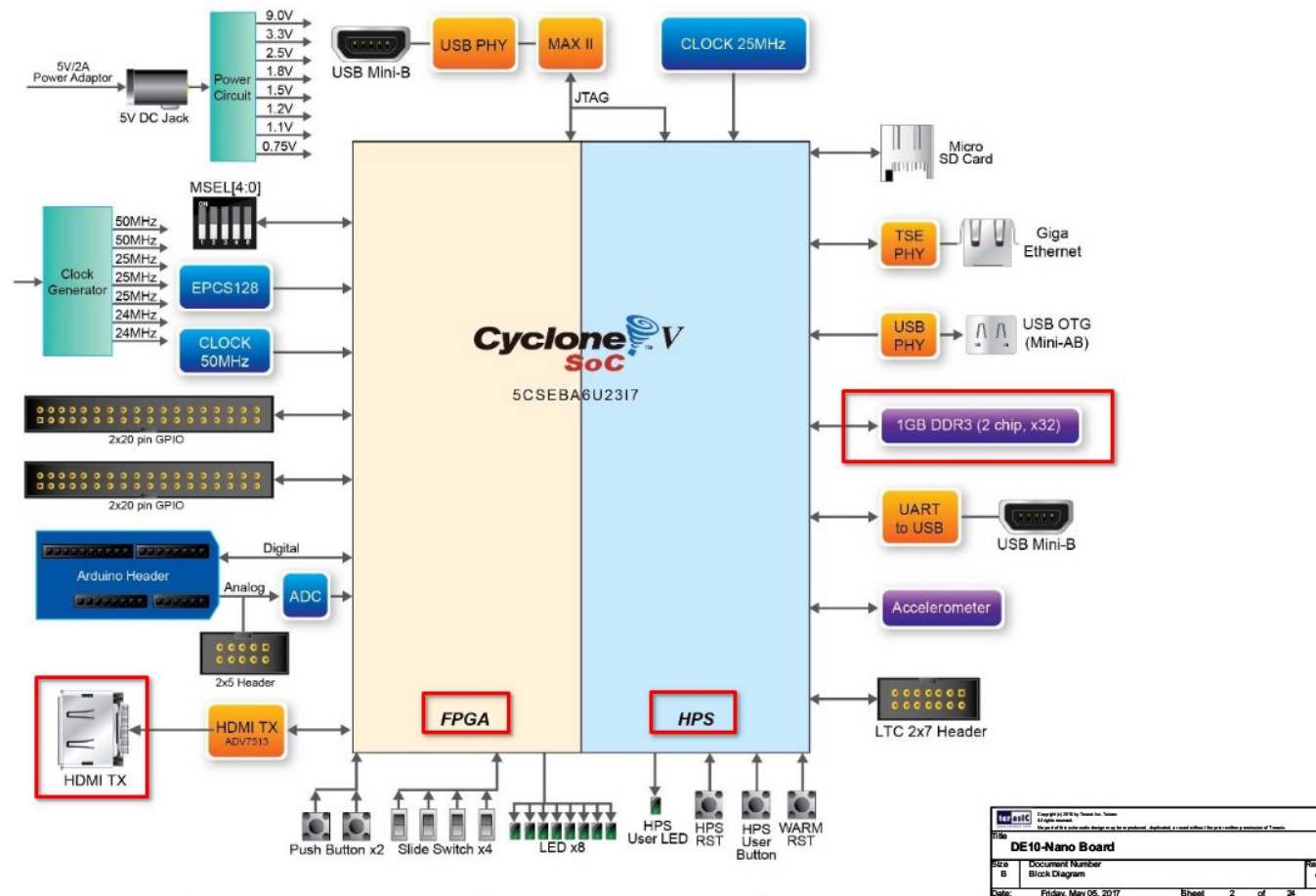


Figure 2-1 DE10-Nano development board (top view)

de10-nano user manual

Quand utiliser un SoC FPGA

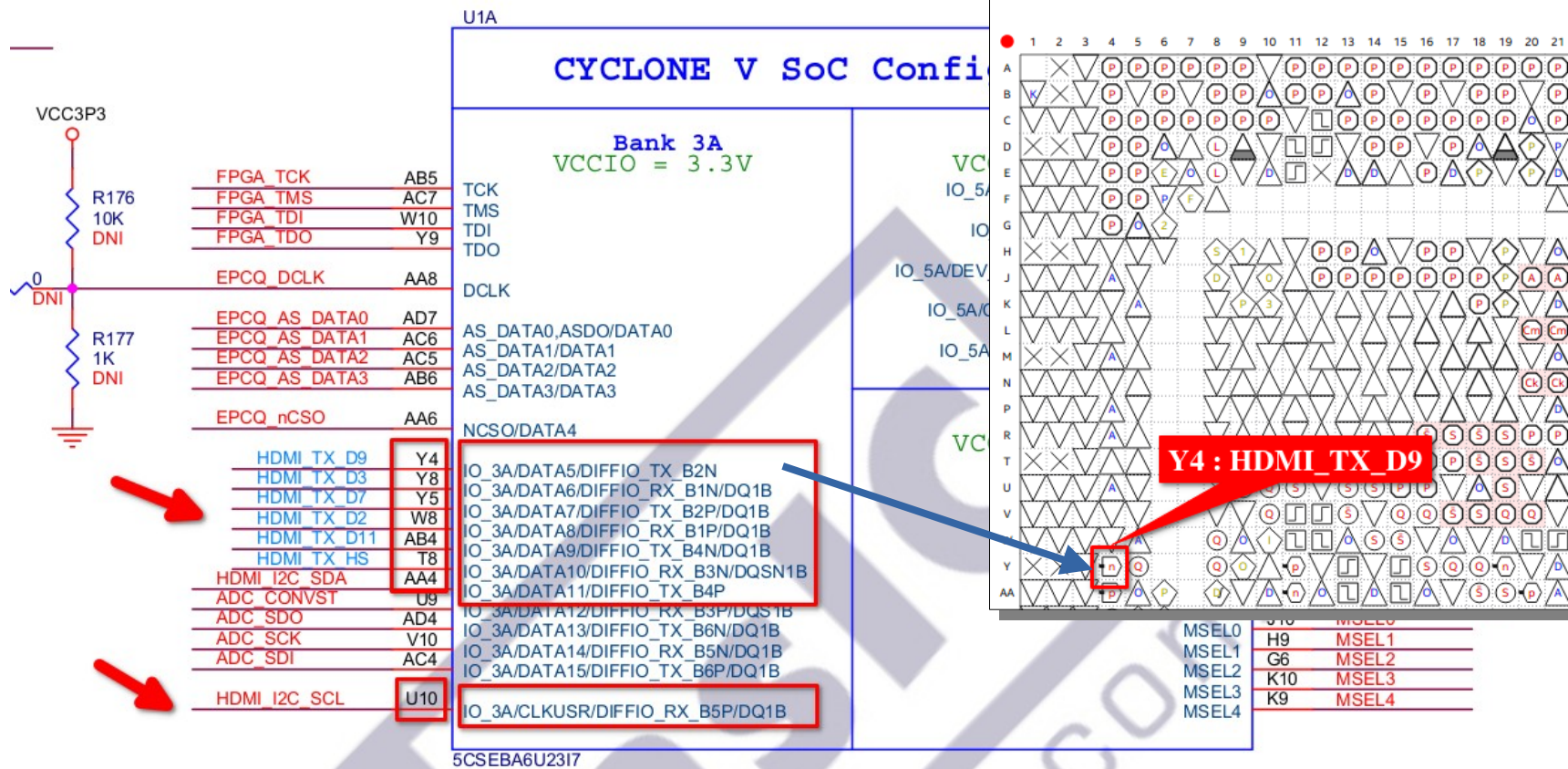
Le SoC FPGA : Une proposition de résolution du problème d'occupation CPU



de10-nano schematics

Quand utiliser un SoC FPGA

Le SoC FPGA : Une proposition de résolution du problème d'occupation CPU



de10-nano schematics

Le SoC FPGA : Une proposition de résolution du problème d'occupation CPU



Quand utiliser un SoC FPGA

Le SoC FPGA : Une proposition de résolution du problème d'occupation CPU

3 remarques ... ce que nous allons faire techniquement

Comment programmer un SoC FPGA chez Intel

Nous allons ...

Remarque 1. nous allons développer en VHDL : langage "compréhensible" par un développeur C

```
1 /*
2  * test.c
3  *
4  * Created on: 14 avr. 2022
5  * Author: jean-marie codol
6  */
7
8
9 char signed_adder(char a, char b) {
10     char result = a+b ;
11     return result ;
12 }
13
```

```
1 -- Quartus Prime VHDL Template
2 -- Signed Adder
3
4 library ieee;
5 use ieee.std_logic_1164.all;
6 use ieee.numeric_std.all;
7
8 entity signed_adder is
9
10     generic
11     (
12         DATA_WIDTH : natural := 8
13     );
14
15     port
16     (
17         a      : in signed ((DATA_WIDTH-1) downto 0);
18         b      : in signed ((DATA_WIDTH-1) downto 0);
19         result : out signed ((DATA_WIDTH-1) downto 0)
20     );
21
22 end entity;
23
24 architecture rtl of signed_adder is
25 begin
26
27     result <= a + b;
28
29 end rtl;
30
```

<https://vhdlguide.readthedocs.io/>

<https://www.nandland.com/vhdl/tips/tip-convert-numeric-std-logic-vector-to-integer.html>

www.nandland.com

Comment programmer un SoC FPGA chez Intel

Nous allons ...

Remarque 2. Nous allons configurer le SoC FPGA +/- comme un micro-contrôleur

signed_adder.vhd => signed_adder.sof => FPGA

signed_adder.sof => signed_adder.jic => Chargé au démarrage (MSEL)

signed_adder.sof => signed_adder.rbf => Chargé par u-boot ou Linux (MSEL)

The left screenshot shows the Intel Quartus Programmer interface. The 'Hardware Setup' window is open, displaying a table of files and a diagram of the 5CSEBA6U23 device. The table has columns for File, Device, and Checksum. The file 'output_files/DE10_NANO_SoC_GHRD.sof' is highlighted. The diagram shows the 5CSEBA6U23 device connected to TDI and TDO pins.

File	Device	Checksum
output_files/DE10_NANO_SoC_GHRD.sof	5CSEBA6U23	019E7A70

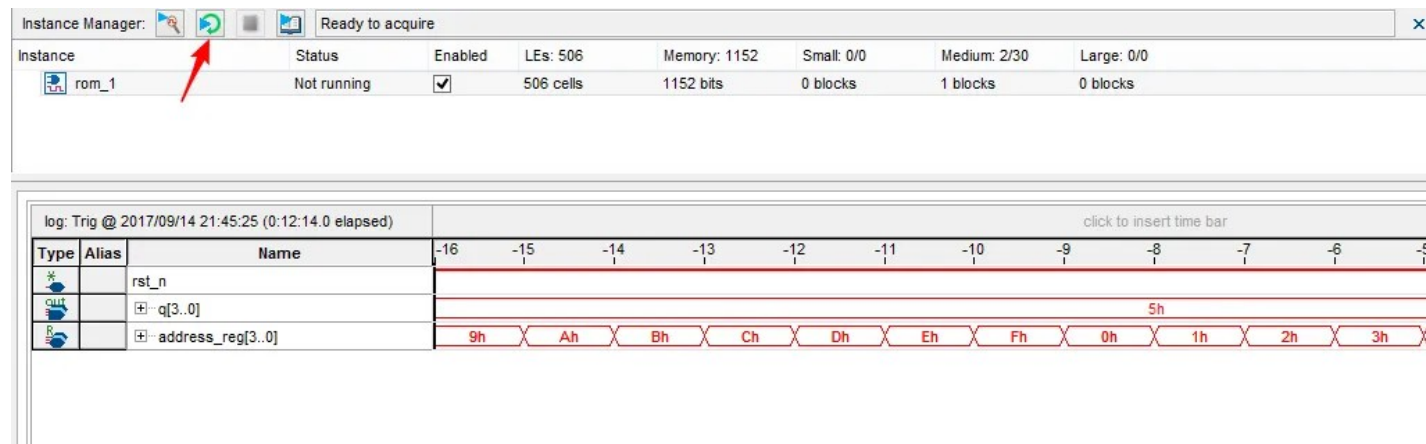
The right screenshot shows the Intel Quartus Programmer interface. The 'Hardware Setup' window is open, displaying a table of files and a diagram of the 5CSEBA6 device connected to an EPCS64 flash. The table has columns for File, Device, Checksum, Usercode, Program/Configure, Verify, Blank-Check, Examine, Security Bit, Erase, and ISP CLAMP. The file 'output_files/add...' is highlighted. The diagram shows the 5CSEBA6 device connected to TDI and TDO pins, with an EPCS64 flash connected to its output.

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
<none>	SOCVHPS	00000000	<none>							
Factory default enha...	5CSEBA6	00C64585	00C64585	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
output_files/add...	EPCS64	634D9B1C		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

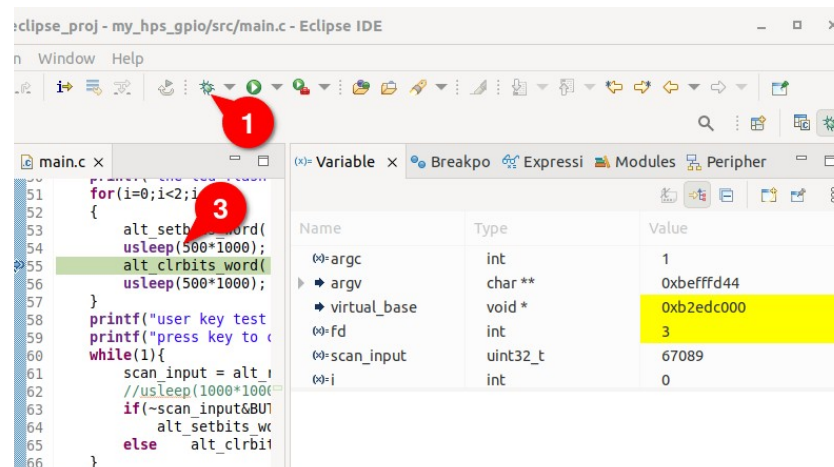
Comment programmer un SoC FPGA chez Intel

Nous allons ...

Remarque 3. Nous allons déboguer avec SignalTap et le débogueur distant c++ sous Eclipse
Analyseur logique SignalTap



Débogueur



Comment programmer un SoC FPGA chez Intel

Programmer un FPGA en HDL (VHDL / Verilog)

Résumé – questions ?

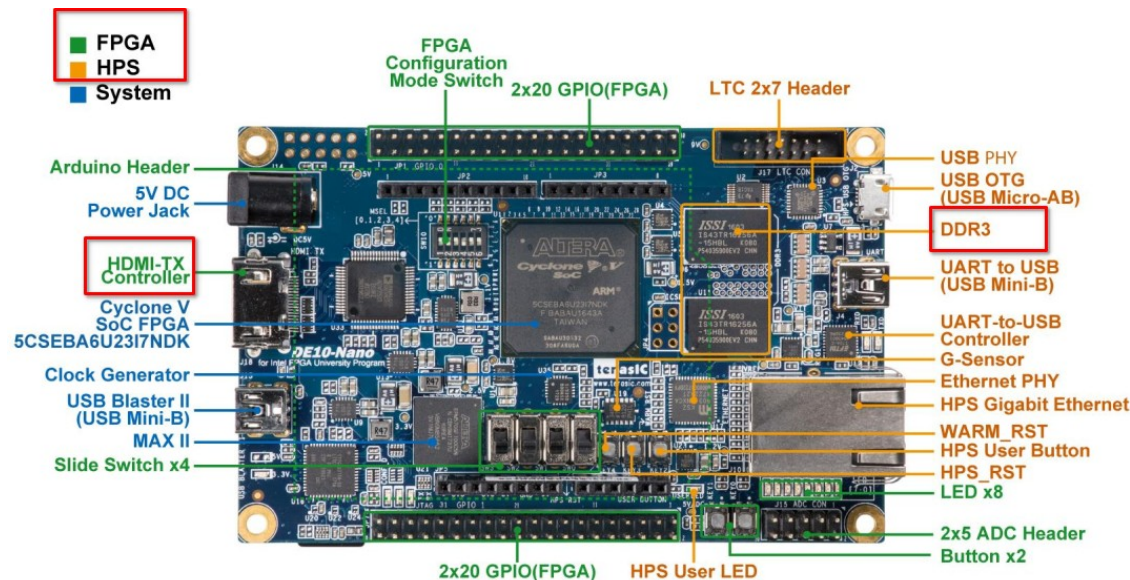


Figure 2-1 DE10-Nano development board (top view)