

# Disegno e implementazione di progetti in FPGA Zynq di Xilinx: un esempio pratico

Riccardo Travaglini

Ignazio D'Antone, Stefano Meneghini, Luigi Rignanese, Mirco Zuffa

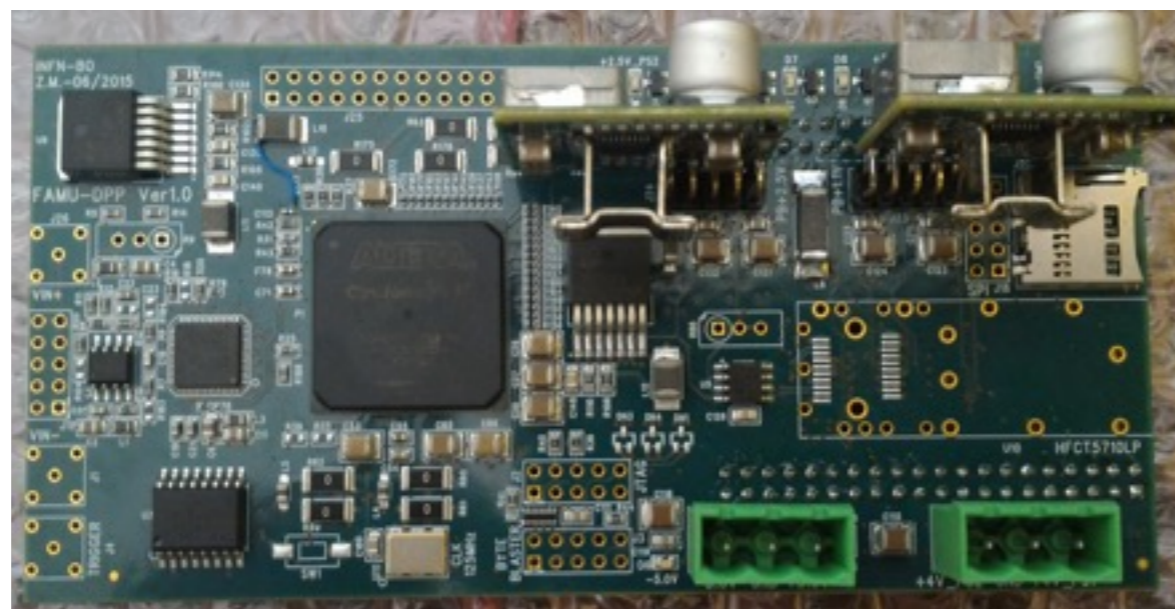
INFN - sezione di Bologna - Centro di Elettronica  
Workshop su Applicazioni FPGA in ambito Astrofisica  
Sessione 2 - Torino 18 maggio 2016

- Attivita' a supporto degli esperimenti in Sezione
- Progettazione, sviluppo, test e installazione di elettronica per sistemi di front-end, trigger e acquisizione
- 14 componenti
- Ventennale esperienza di lavoro con FPGA (Altera, Microsemi, Xilinx):
  - ✓ progettazione di PCB (Orcad, Pads, Expedition, HyperLynx)
  - ✓ sviluppo di firmware (VHDL, C++, Handel-C, Schematic)
  - ✓ software per microprocessori embedded (C)

<http://www-ceb.bo.infn.it>

# Progetto di R&D per un canale di acquisizione a 1 GS/s Motivazioni e Stato dell'arte @INFN-BO

- Elevati rate di campionamento favoriscono:
  - Discriminazione a forma di impulso
  - Reiezione di pile-up nell'acquisizione di rivelatori veloci
- Esempi di rivelatori:
  - PMT con fronti di salita veloci (  $\sim 10$  ns)
  - SiPM
- 2015: realizzata una scheda con ADC commerciale (12 bit @ 500 MS/s) per l'esperimento FAMU\*



6 layer  
130x65 mm  
ADC: AD9434  
Altera Cyclone V 5CGXC5  
(custom fw)  
out to USB 3.0 mezzanine

\*<https://webint.ts.infn.it/en/research/exp/famu.html>

- Attivita' di R&D del centro di elettronica per realizzare un dispositivo a 1 GS/s
- Design: ADC-interleaved con due ADC a 500 MS/s
- Obiettivi principali:
  - Studio delle problematiche della realizzazione di un PCB per il campionamento a 1 GS/s
  - Esperienza nella realizzazione e calibrazione di ADC-interleaved
  - Esperienza nell'acquisizione ed elaborazione real-time tramite FPGA Zynq di Xilinx
- Stato del progetto:
  - identificato i requisiti di progetto e i componenti tramite simulazioni e misure con schede commerciali (trattato nel seguito della presentazione)
  - terminato lo schematico
  - ordinati i componenti
  - layout in corso

# Studio preliminare: acquisizione a 500 MS/s con schede commerciali

Obiettivi dello studio preliminare: finalizzare il firmware di acquisizione, verificare le performance variando condizioni di progetto (ad es. utilizzando diverse sorgenti di clock con forme d'onda note)



**Digilent ZedBoard**

Zynq Z7020

Dual Core Arm Cortex-A9 max 866 MHz

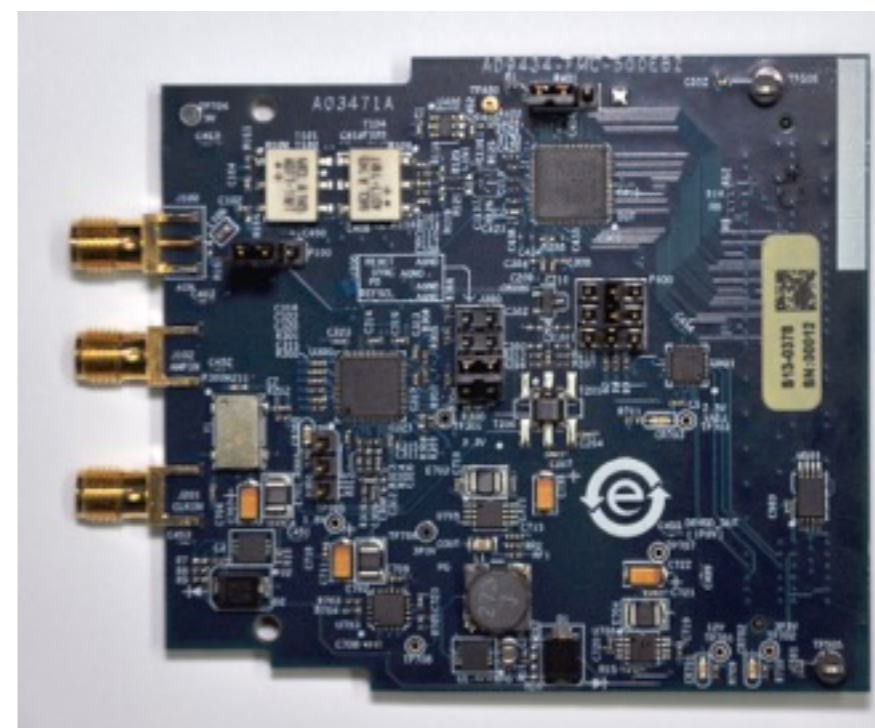
Fs: ~100k

Lut: ~53k

Block RAM: 516kb

220 DSP48

0 transceivers



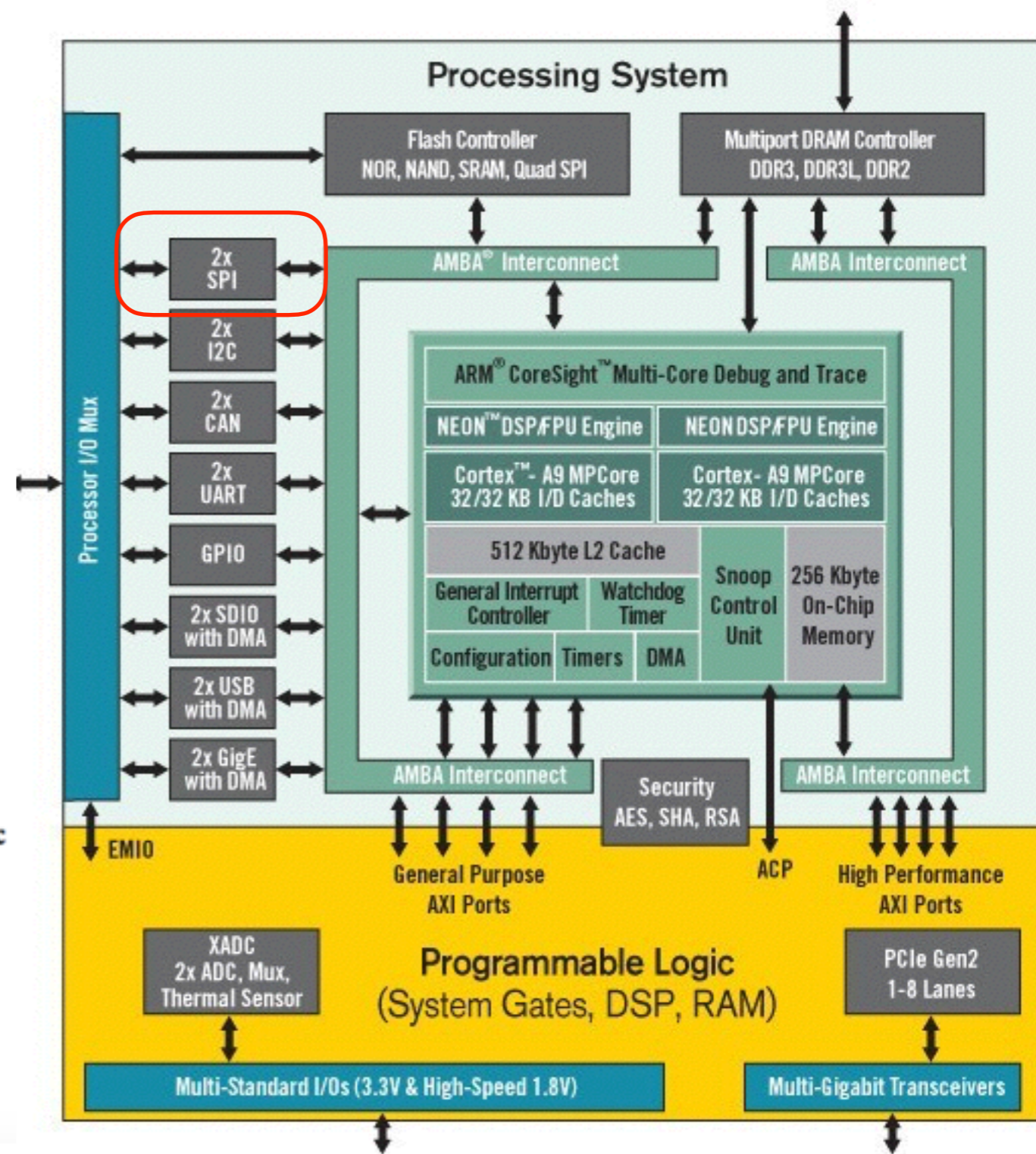
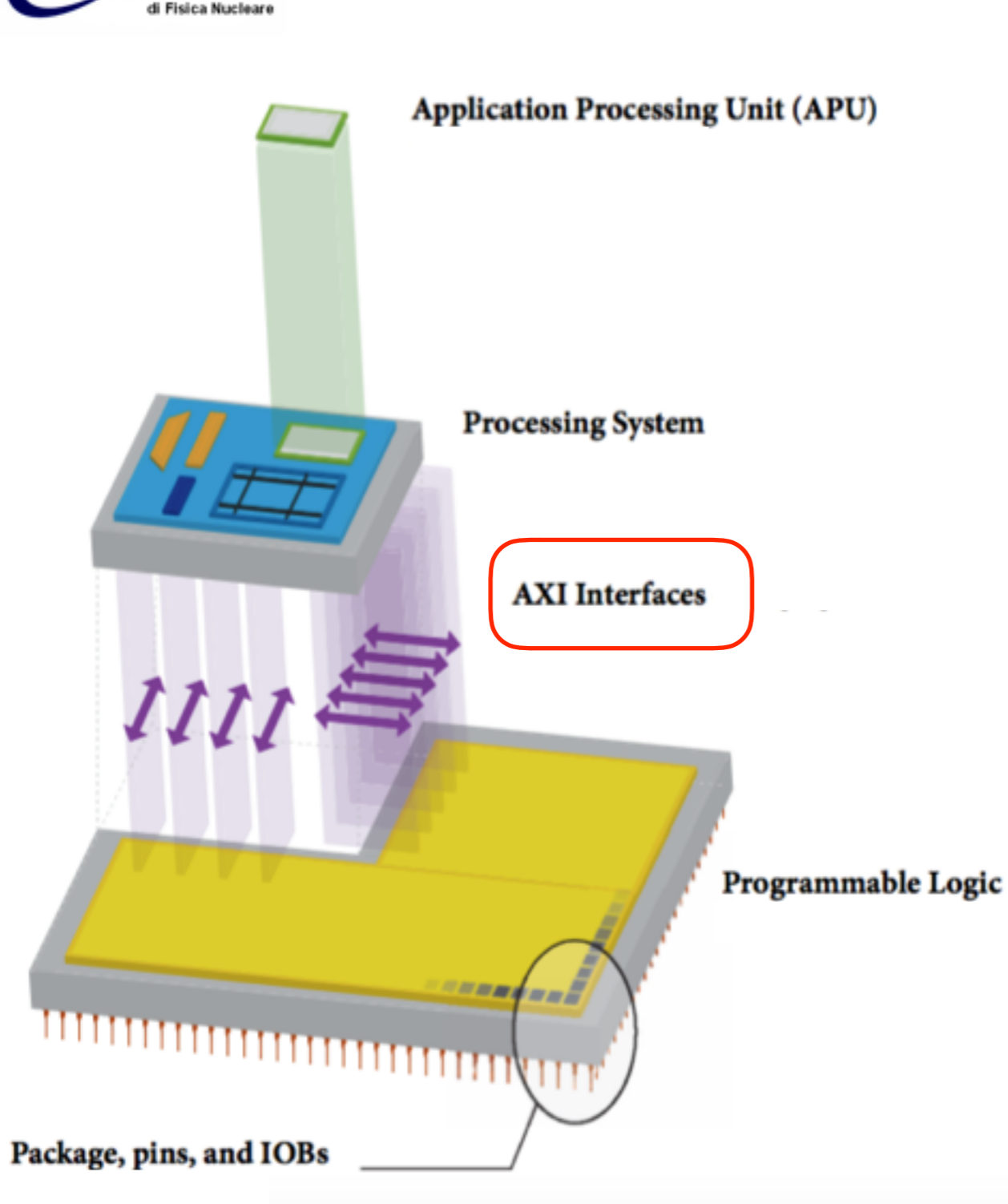
**Analog Devices AD9434-FMC-500EB**

AD9434 ADC

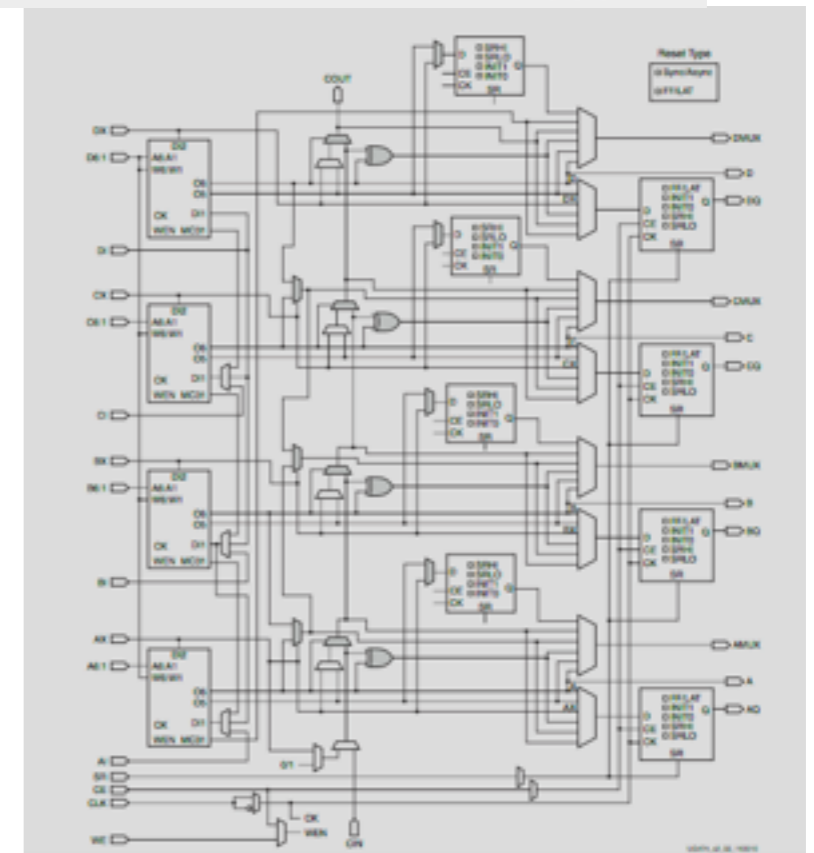
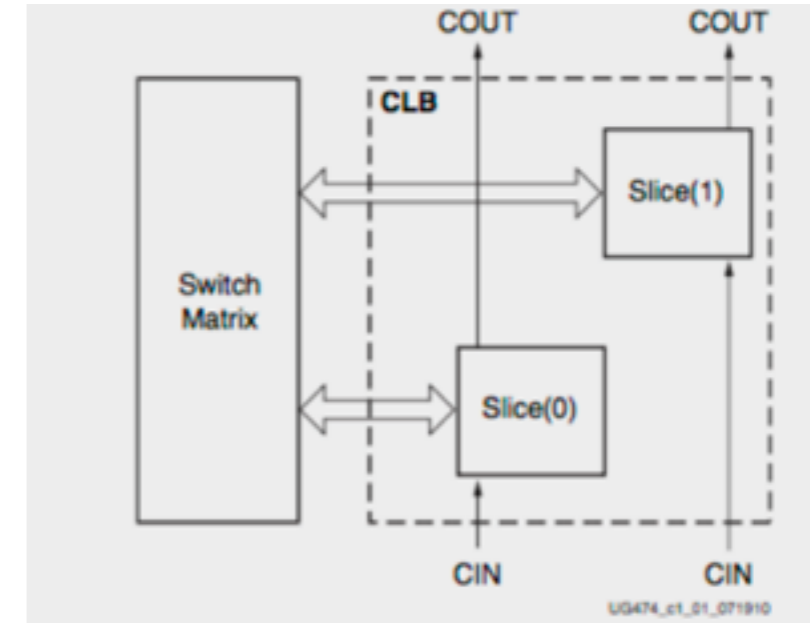
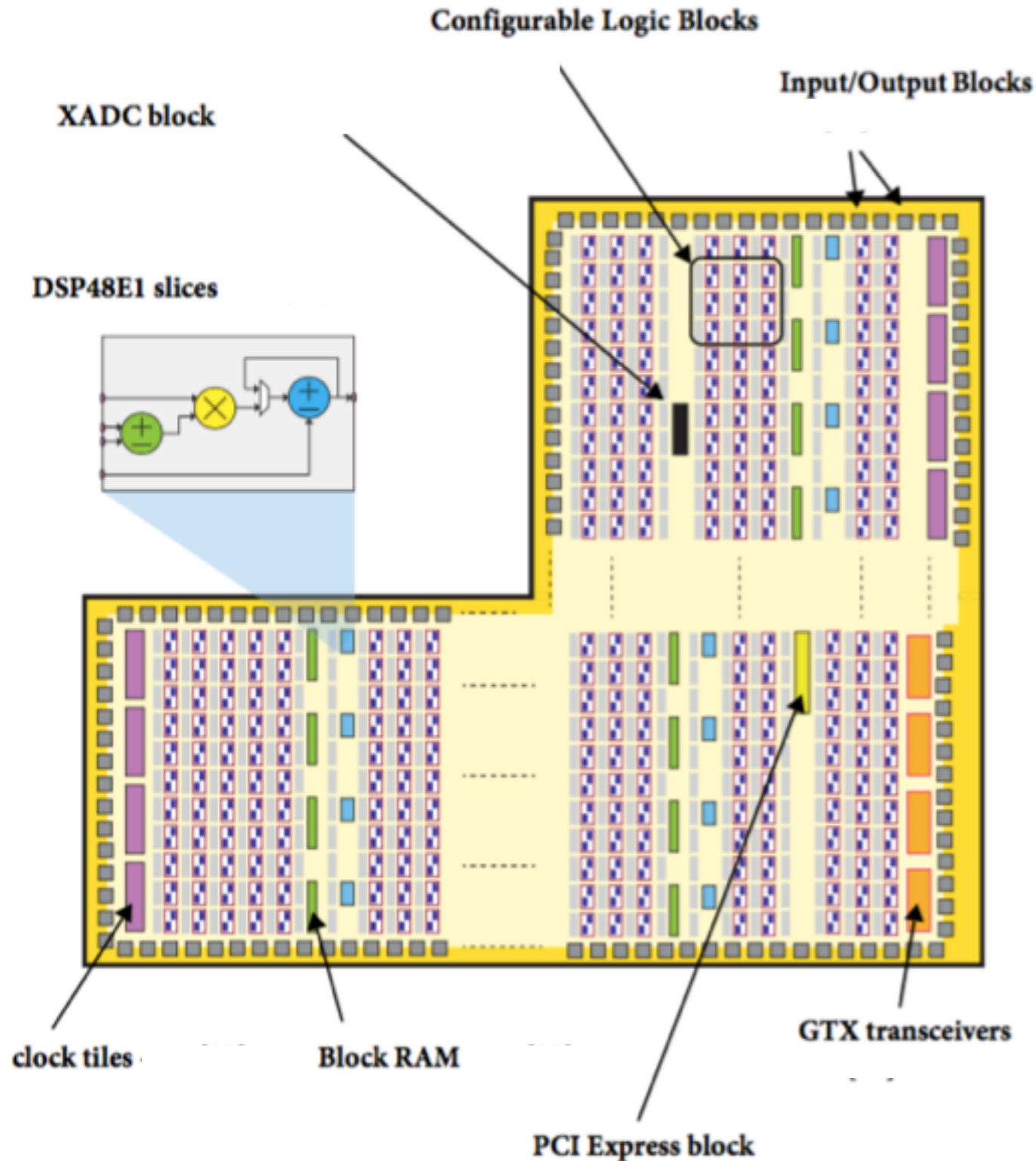
1 Channel

12-bit

500 MS/s



# Xilinx Zynq - Programmable Logic



Logica programmabile per la serie 7 comune a tutti gli FPGA

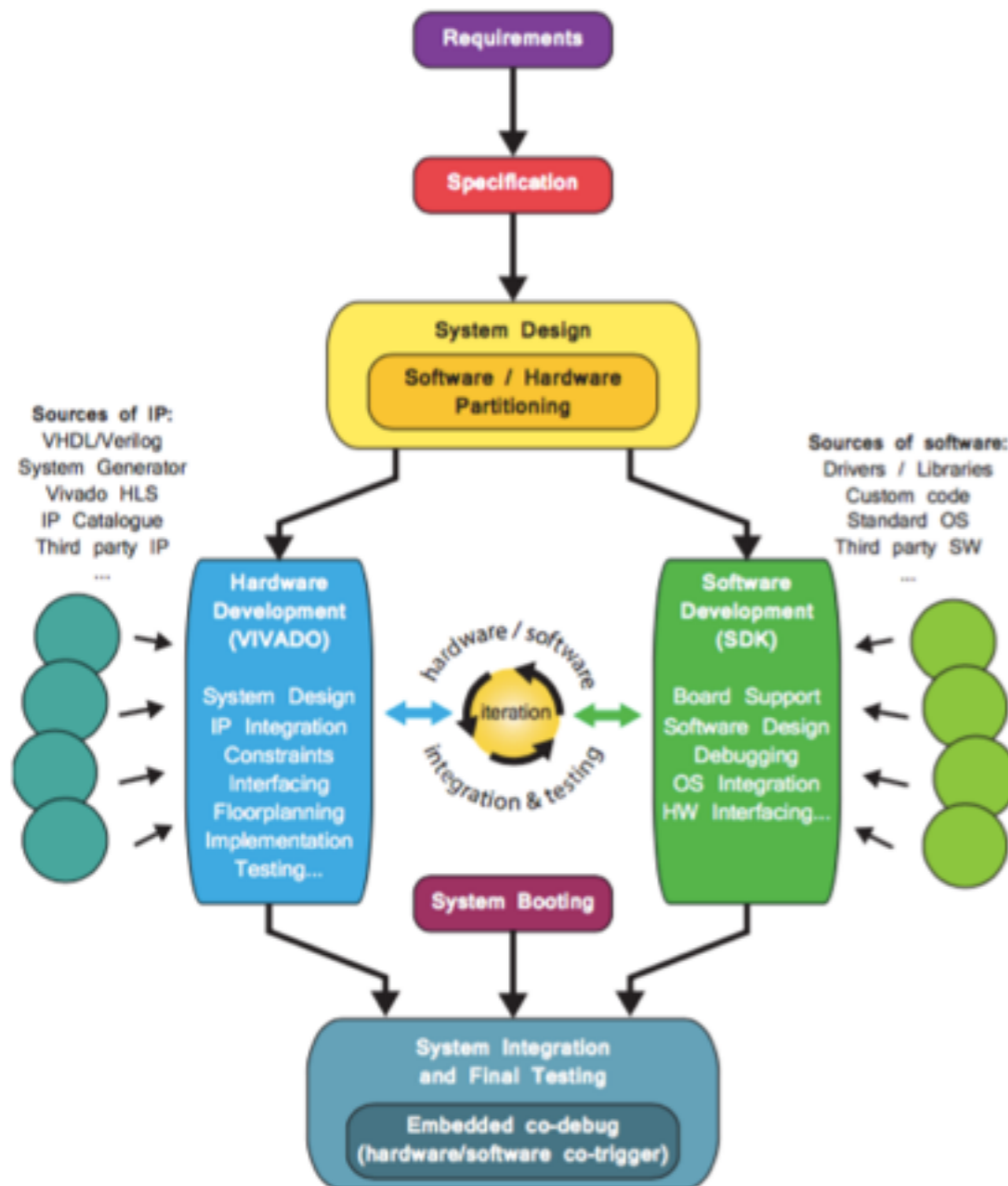
# "Standard" Zynq design strategy

## Software

Sistema operativo  
Task sequenziali general purpose  
User Application  
GUIs

## Firmware

Algoritmi che richiedono  
elevato flusso di dati  
Algoritmi paralleli  
=> offloading co-processors  
  
Periferiche custom



**Diversi tool per lo sviluppo di FW e SW (parallelizzabile)**



# Esempio di disegno di FPGA Zynq con Vivado (ADC a 500 MS/s)

- Nota: progetto derivato da esempio con Microblaze su board demo ML605 con Spartan 6 (<https://wiki.analog.com/resources/fpga/xilinx/fmc/ad9434>)

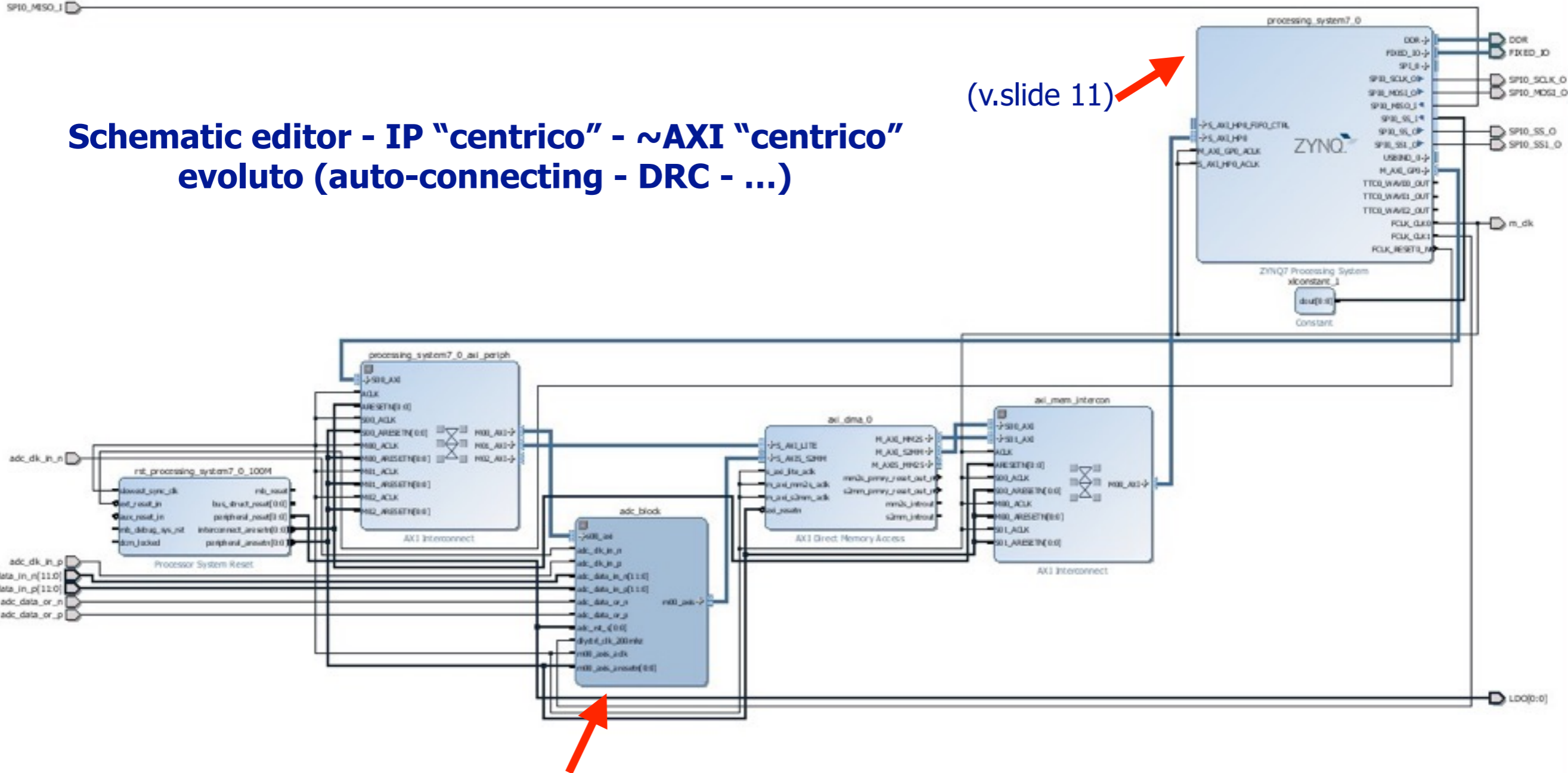
nelle slides successive:

- IP integrator: Block Design per la configurazione del Processing System e l'implementazione nell Programmable Logic di IP
  - Xilinx, 3rd party : AXI interconnect, DMA, reset system...
  - Custom : AD9434\_to\_AXI...
- Block design wrapper in VHDL (automaticamente generato)
- Modifica integrando interfaccia tra Zynq SPI e implementazione I<sup>2</sup>C-like dell'ADC (VHDL)
- Sintesi
- Inserimento di core per il debug di segnali (Integrated Logic Analyzers)
- Piazzamento dei pin di I/O
- Timing constraints
- Implementazione e Bitfile

# IP integrator - Block Design

(v.slide 11)

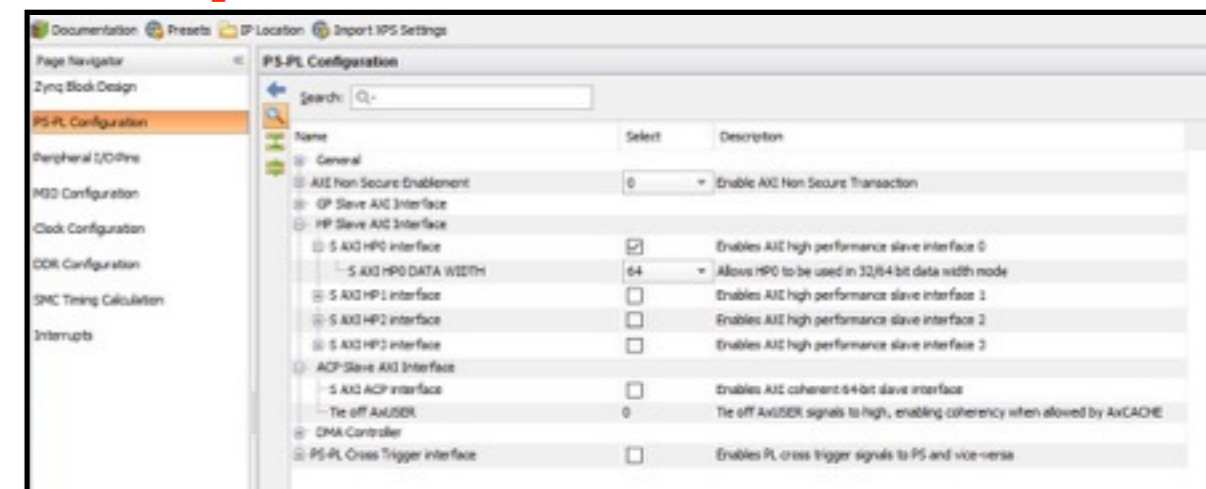
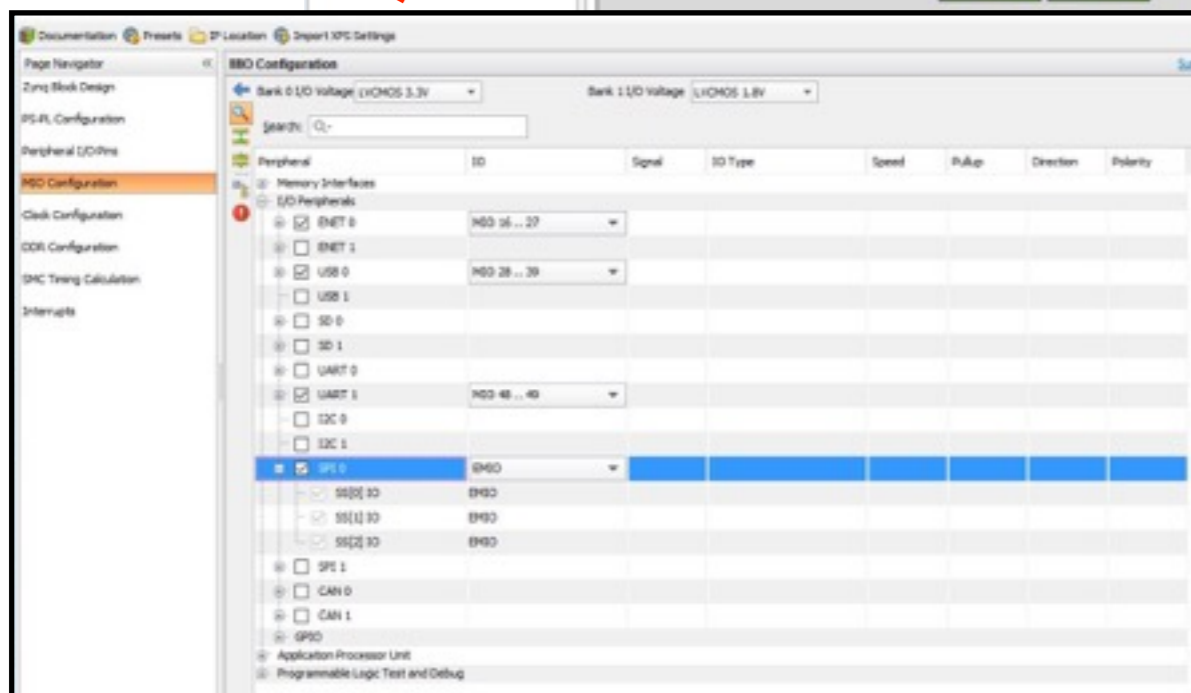
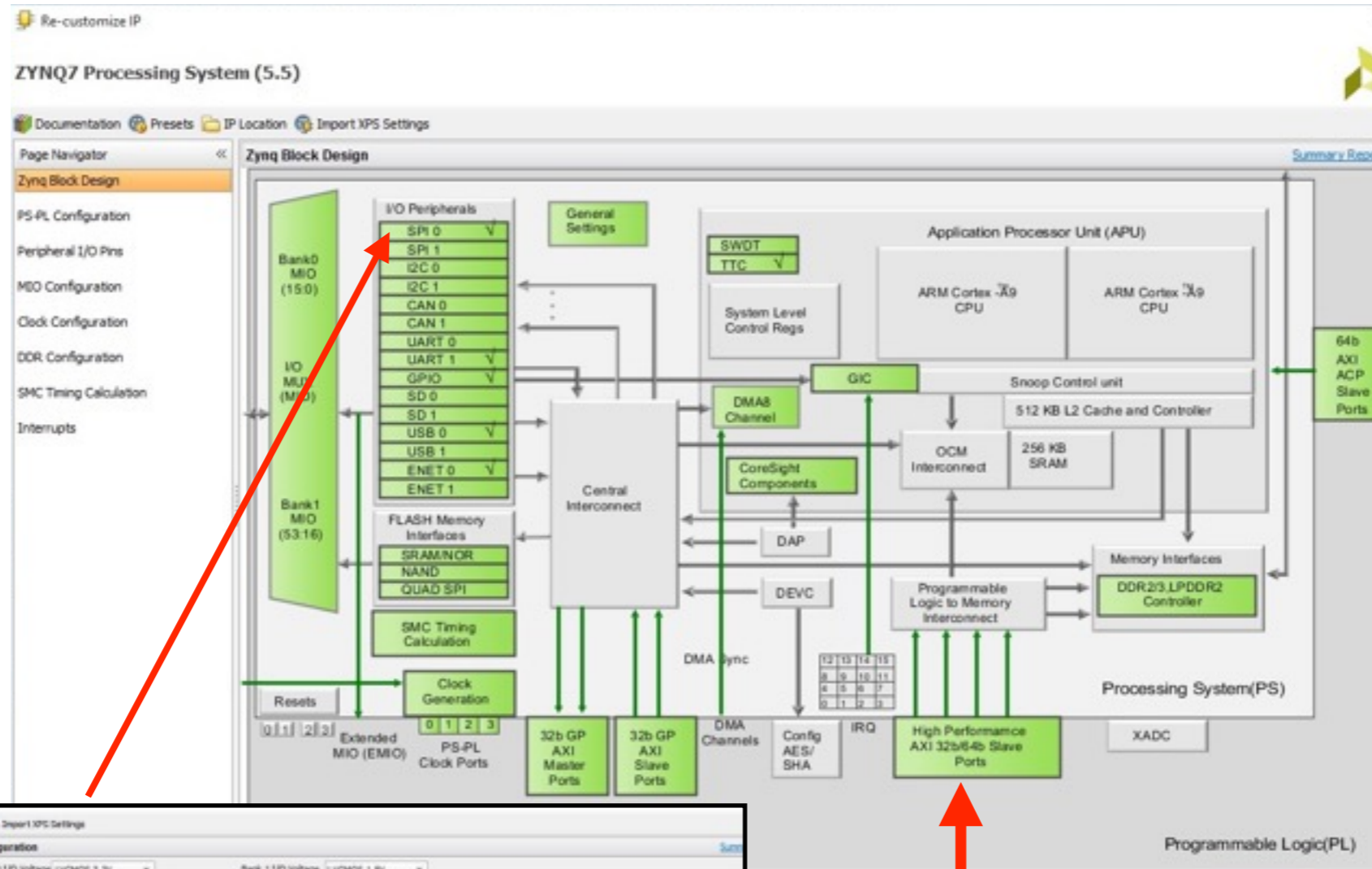
**Schematic editor - IP "centrico" - ~AXI "centrico"  
evoluto (auto-connecting - DRC - ...)**

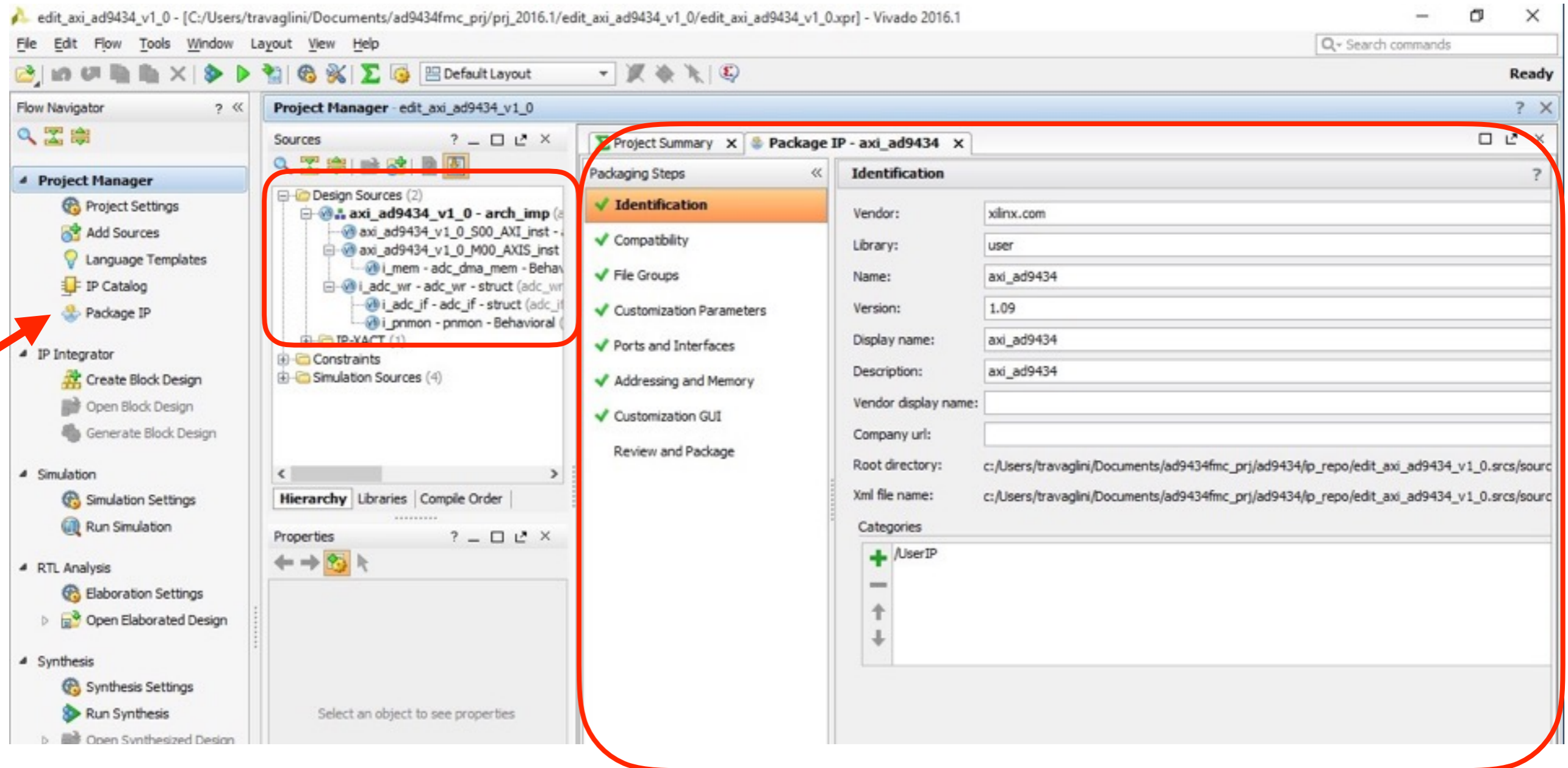


**IP custom che acquisisce 12 bit in parallelo a 500 MHz (con Serdes),  
parallelizza 1 a 4 bus a 125 MHz (48 bit) e implementa AXI master a 64 bit (verso il DMA)**

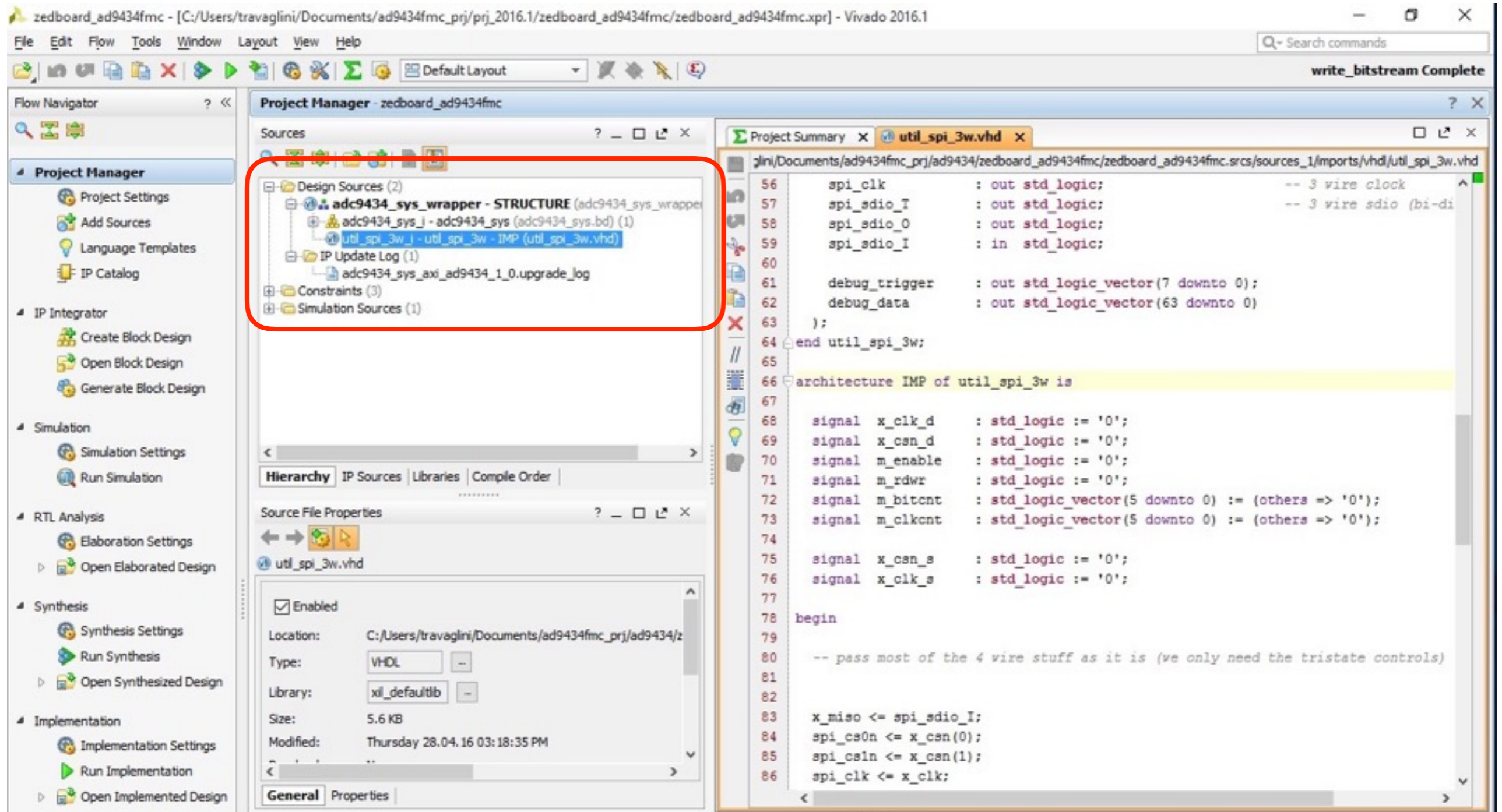
(VHDL puro - v.slide 12)

# Zynq Processing System IP core





**Progetto di Vivado configurato per generare un IP e creato con un "wizard" dal progetto originale**



The screenshot displays the Vivado 2016.1 interface. The Project Manager window shows the project hierarchy for 'zedboard\_ad9434fmc'. A red box highlights the 'Design Sources' section, where the file 'util\_spi\_3w.vhd' is listed under the 'adc9434\_sys\_wrapper - STRUCTURE'.

The Source File Properties window for 'util\_spi\_3w.vhd' shows the following details:

- Enabled:
- Location: C:/Users/travaglini/Documents/ad9434fmc\_prj/ad9434/z
- Type: VHDL
- Library: xil\_defaultlib
- Size: 5.6 KB
- Modified: Thursday 28.04.16 03:18:35 PM

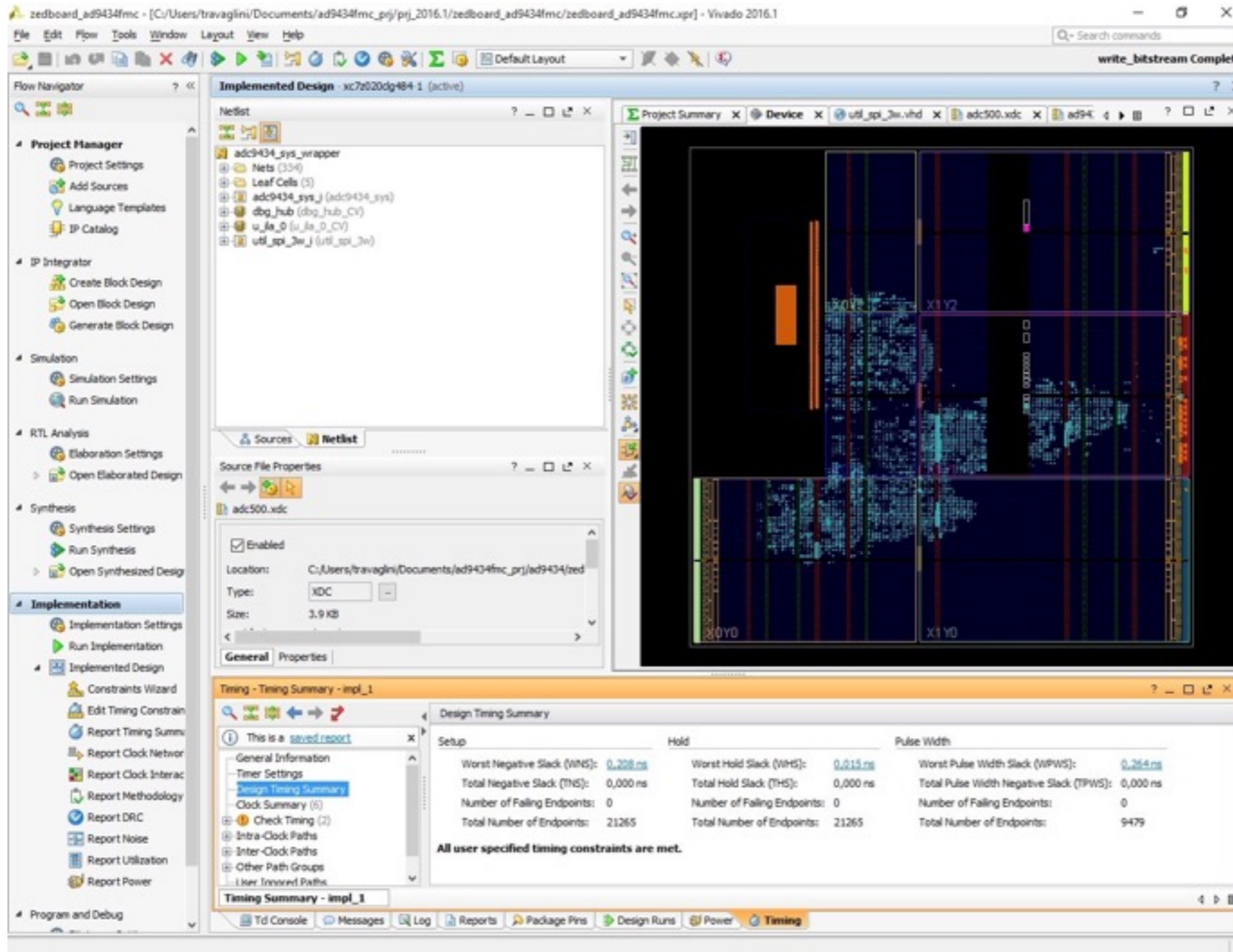
The VHDL code editor shows the module definition for 'util\_spi\_3w' with the following code:

```

56 spi_clk      : out std_logic;           -- 3 wire clock
57 spi_sdio_T   : out std_logic;           -- 3 wire sdio (bi-di
58 spi_sdio_0   : out std_logic;
59 spi_sdio_I   : in  std_logic;
60
61 debug_trigger : out std_logic_vector(7 downto 0);
62 debug_data    : out std_logic_vector(63 downto 0)
63 );
64 end util_spi_3w;
65
66 architecture IMP of util_spi_3w is
67
68 signal x_clk_d      : std_logic := '0';
69 signal x_csn_d      : std_logic := '0';
70 signal m_enable     : std_logic := '0';
71 signal m_rdrwr      : std_logic := '0';
72 signal m_bitcnt     : std_logic_vector(5 downto 0) := (others => '0');
73 signal m_clkcnt     : std_logic_vector(5 downto 0) := (others => '0');
74
75 signal x_csn_s      : std_logic := '0';
76 signal x_clk_s      : std_logic := '0';
77
78 begin
79
80 -- pass most of the 4 wire stuff as it is (we only need the tristate controls)
81
82
83 x_miso <= spi_sdio_I;
84 spi_cs0n <= x_csn(0);
85 spi_cs1n <= x_csn(1);
86 spi_clk <= x_clk;

```

## Convivenza di IP integrator e moduli VHDL



The screenshot shows the Vivado 2016.1 IDE interface. The top menu bar includes File, Edit, Flow, Tools, Window, Layout, View, and Help. The main workspace is divided into several panes:

- Flow Navigator:** Shows the project workflow from Project Manager to Program and Debug.
- Project Manager:** Lists project settings, sources, and IP integrator options.
- Implementation:** The current step, showing implementation settings and the implemented design.
- Netlist:** Displays the design's netlist structure, including components like adc9434\_sys\_wrapper, nets, leaf cells, and various modules.
- Timing - Timing Summary - impl\_1:** A detailed report showing timing constraints are met. The summary table is as follows:

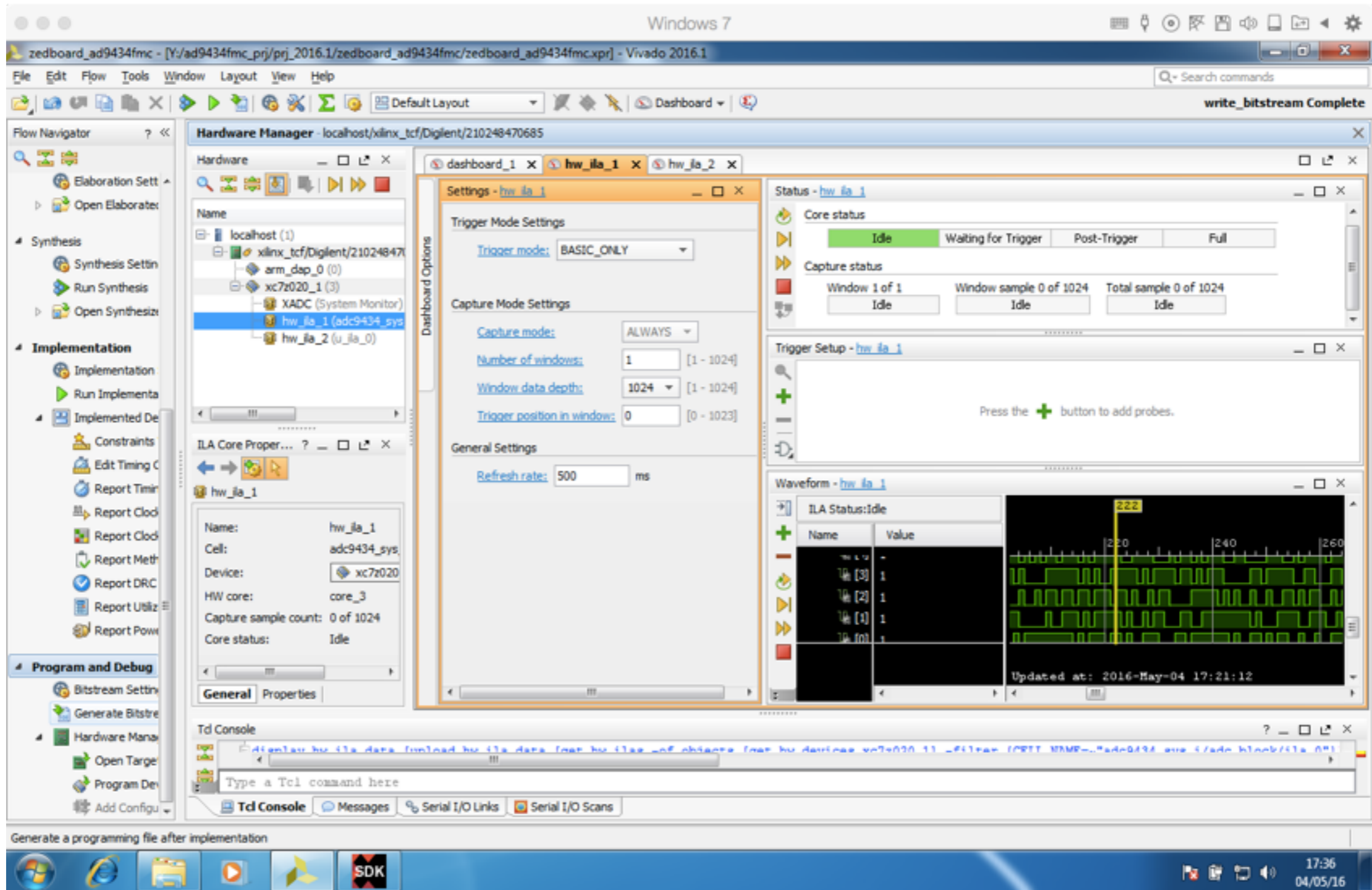
Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 0,208 ns	Worst Hold Slack (WHS): 0,015 ns	Worst Pulse Width Slack (WPWS): 0,264 ns
Total Negative Slack (TNS): 0,000 ns	Total Hold Slack (THS): 0,000 ns	Total Pulse Width Negative Slack (TPWS): 0,000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 21265	Total Number of Endpoints: 21265	Total Number of Endpoints: 9479

The bottom status bar shows various tool icons like Td Console, Messages, Log, Reports, Package Pins, Design Runs, Power, and Timing.

## Flusso di sviluppo di Vivado tradizionale

- Sintesi
- Setup Debug
- Pin Placement
- Timing Constraints
- Implementazione
- Bitfile

# Debugging con Integrated Logic Analyzer(s) 1/3



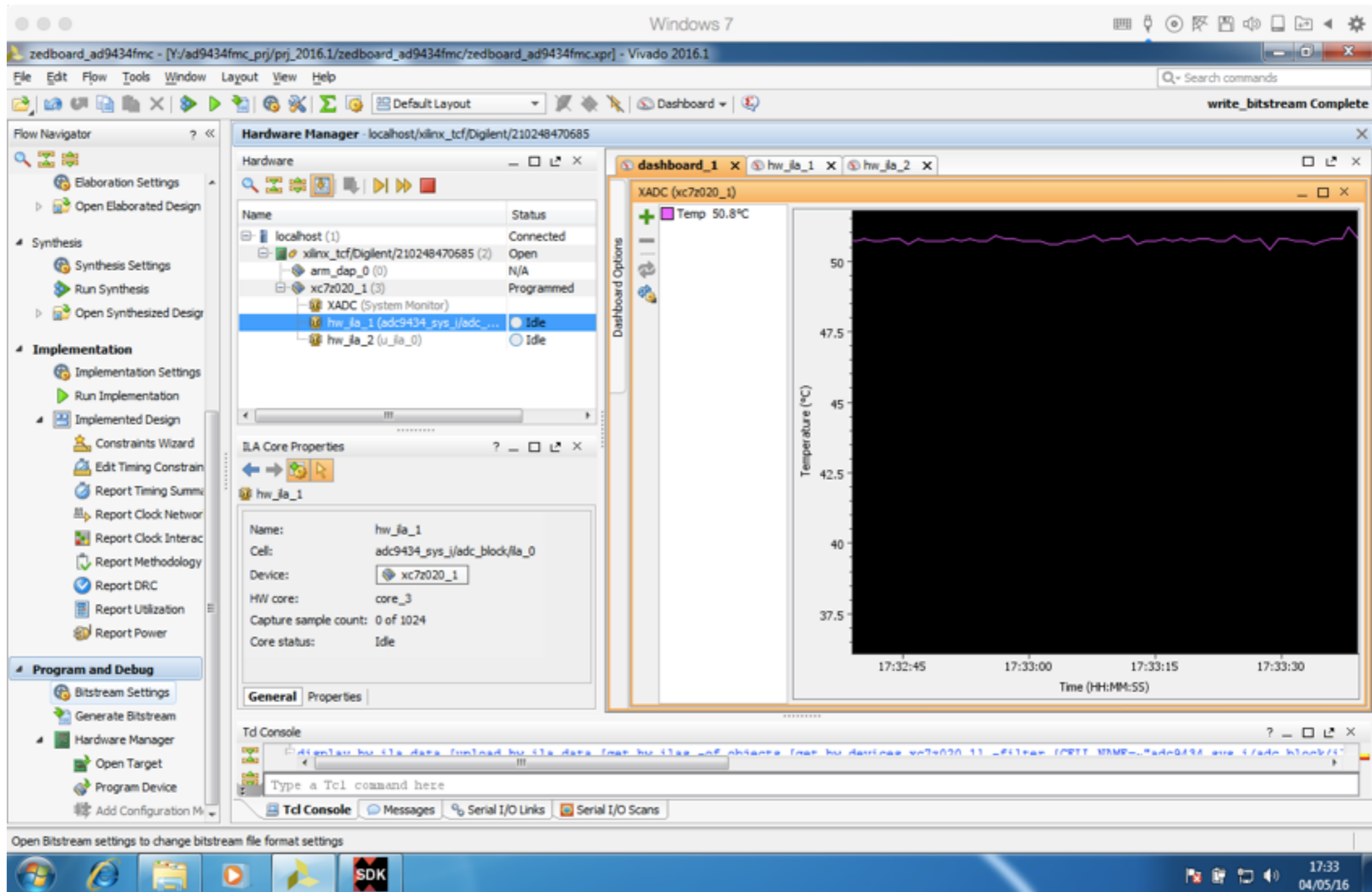
The screenshot displays the Vivado 2016.1 IDE interface for configuring an Integrated Logic Analyzer (ILA) core. The main window is titled "Settings - hw\_ila\_1" and is divided into several sections:

- Trigger Mode Settings:** The "Trigger mode" is set to "BASIC\_ONLY".
- Capture Mode Settings:** The "Capture mode" is set to "ALWAYS". The "Number of windows" is 1, "Window data depth" is 1024, and "Trigger position in window" is 0.
- General Settings:** The "Refresh rate" is set to 500 ms.
- Status - hw\_ila\_1:** Shows the core status as "Idle". The capture status for "Window 1 of 1" is also "Idle".
- Waveform - hw\_ila\_1:** Displays a waveform with a table of data points. The table shows the ILA Status as "Idle" and several data points with values of 1.

The Tcl Console at the bottom shows the command: `display_hw_ila_data [upload_hw_ila_data [get_hw_ila_of_objects [get_hw_devices xc7z020_1] -filter {PART_NAME =~ "adc9434_sys"} /adc_block/ila_0]]`

## Interfaccia di configurazione del Logic Analyzer

# Debugging con Integrated Logic Analyzer(s) 2/3



The screenshot shows the Vivado 2016.1 IDE interface. The Hardware Manager window displays the hardware configuration for a Digilent ZedBoard. The hardware list includes:

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210248470685 (2)	Open
arm_dap_0 (0)	N/A
xc7z020_1 (3)	Programmed
XADC (System Monitor)	
hw_ila_1 (adc9434_sys_1/adc_...)	Idle
hw_ila_2 (u_ila_0)	Idle

The XADC (System Monitor) window shows a temperature graph with the following data:

Time (HH:MM:SS)	Temperature (°C)
17:32:45	~50.8
17:33:00	~50.8
17:33:15	~50.8
17:33:30	~50.8

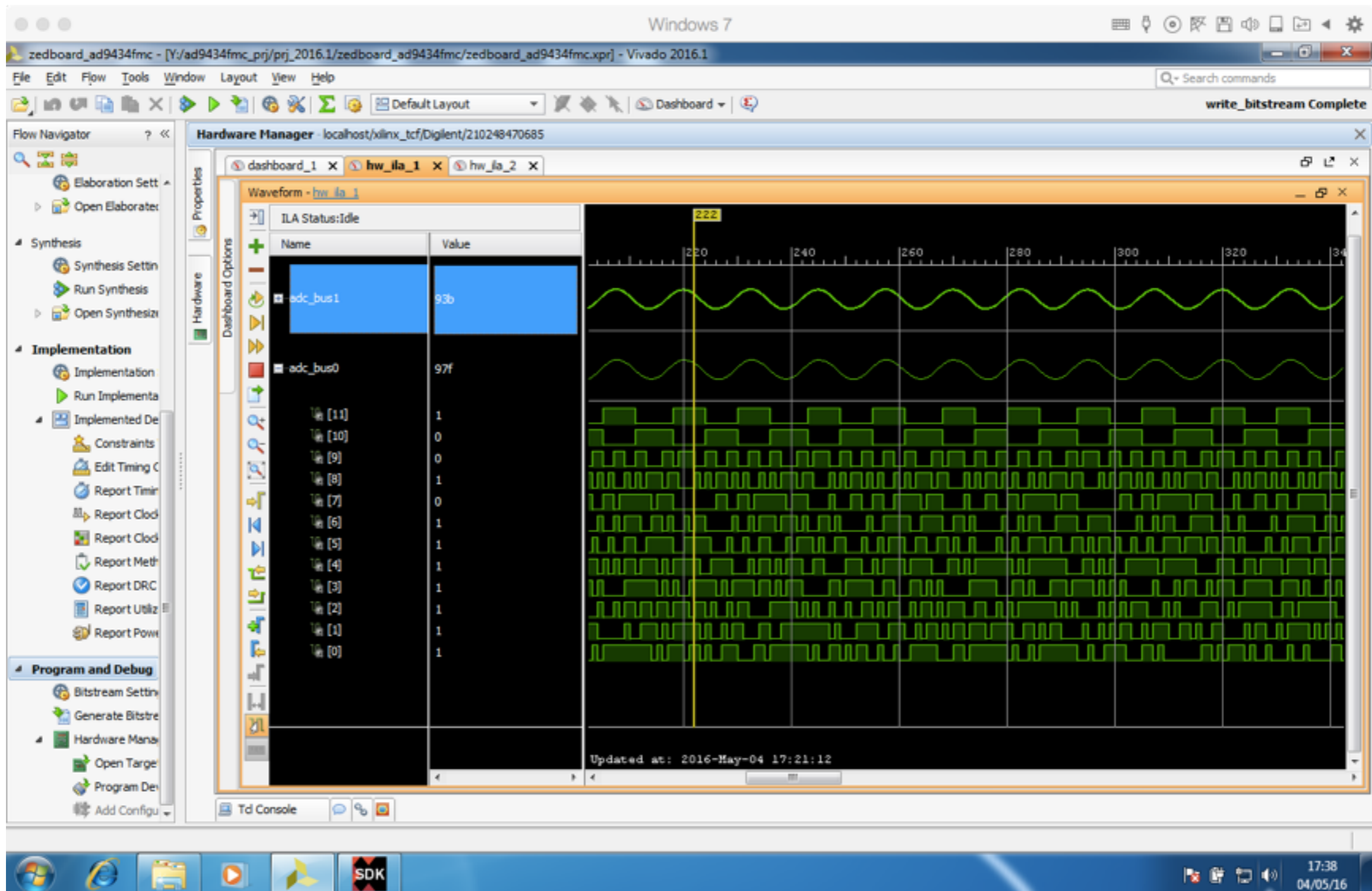
The Td Console window shows the following command:

```
digilent_hw_ila_data_unload_by_ila_data_load_by_ila_obj_objects_load_by_device xc7z020_1 -filter (CRTL_NAME=="adc9434_sys_1/adc_block/ila_1")
```

## Accesso al sensore di temperatura dell'FPGA



# Debugging con Integrated Logic Analyzer(s) 3/3



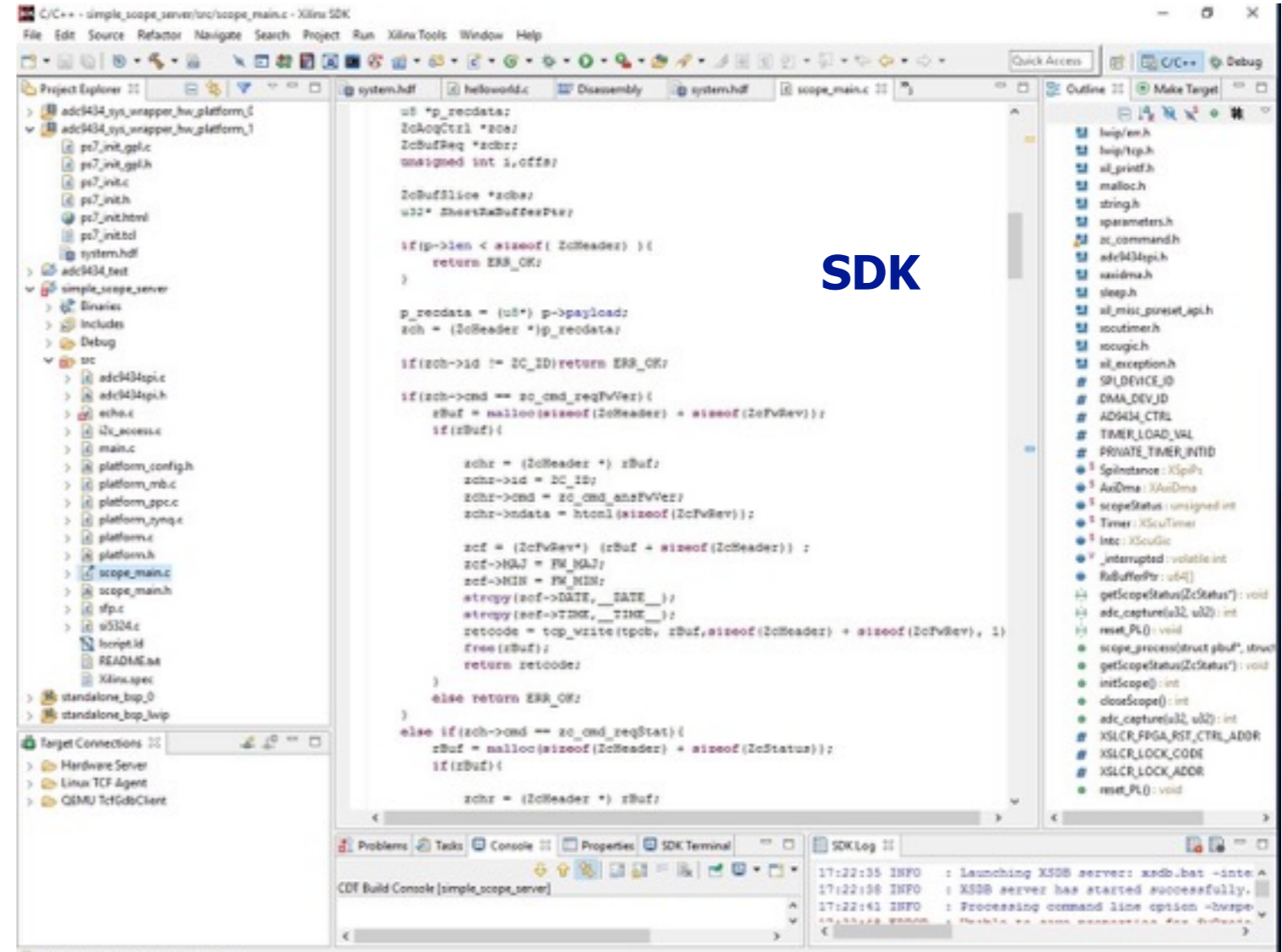
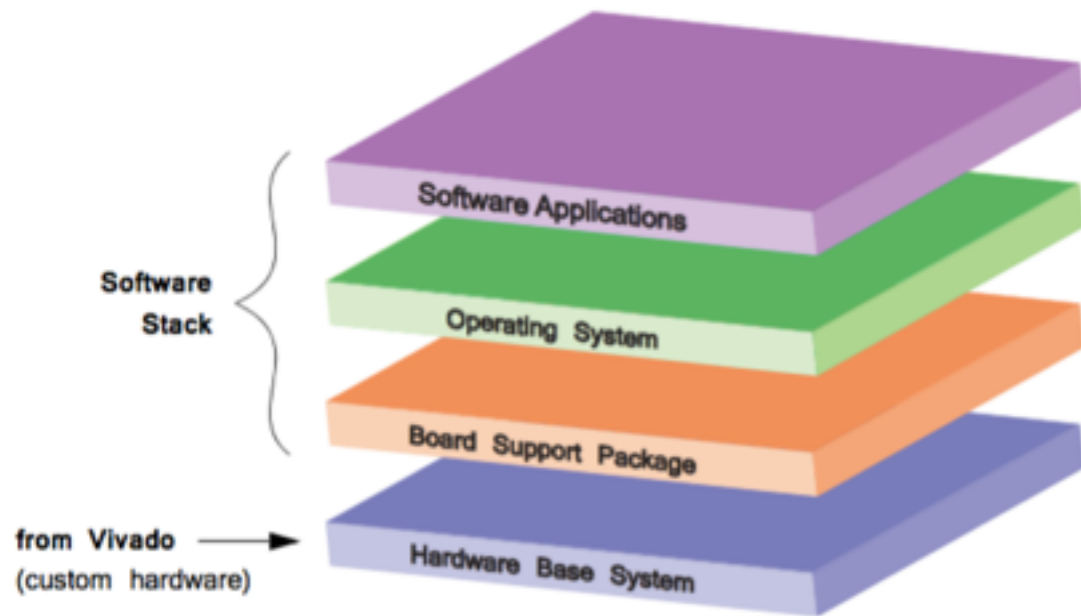
The screenshot displays the Vivado 2016.1 Integrated Logic Analyzer (ILA) interface. The main window shows a waveform capture for 'hw\_ila\_1'. The waveform is divided into two sections: digital signals (bottom) and analog signals (top). The digital signals are represented by green square waves, and the analog signals are represented by green sine waves. A vertical yellow cursor is positioned at time 222. The digital signals are labeled with bit indices [11] through [0]. The analog signals are labeled 'adc\_bus1' and 'adc\_bus0'. The values for these signals are shown in the table below.

Name	Value
adc_bus1	93b
adc_bus0	97f
[11]	1
[10]	0
[9]	0
[8]	1
[7]	0
[6]	1
[5]	1
[4]	1
[3]	1
[2]	1
[1]	1
[0]	1

The interface also shows the 'Flow Navigator' on the left, the 'Hardware Manager' at the top, and the 'Tcl Console' at the bottom. The status bar indicates the update time: 'Updated at: 2016-May-04 17:21:12'.

## Acquisizione dei segnali con visualizzazione digitale e analogica

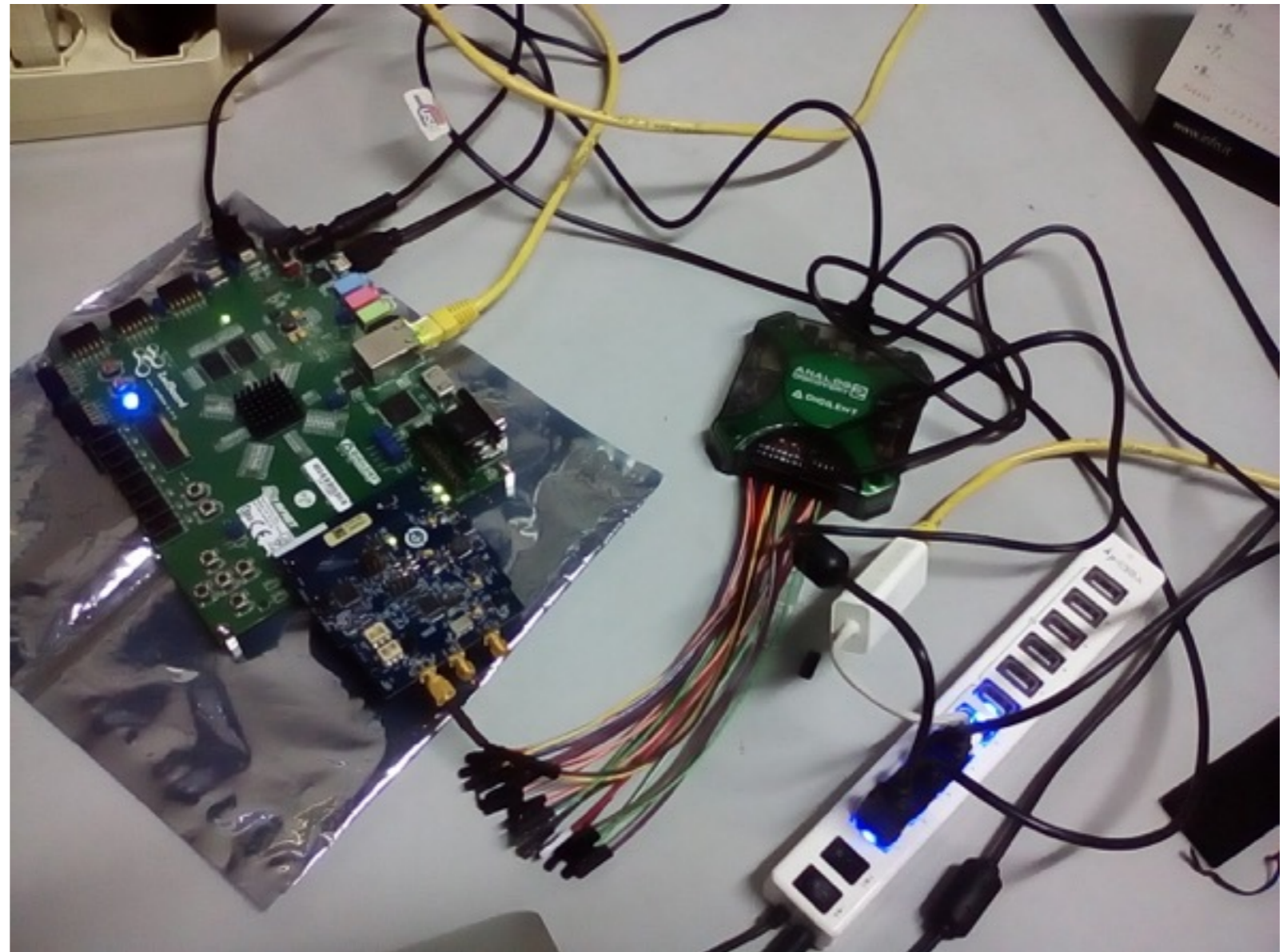
# Sviluppo del software per il microprocessore



**Applicazione Sw: server Ethernet che risponde a pacchetti con codifica custom per configurare i dispositivi (SPI) e spedisci dati acquisiti in memoria (tramite DMA)**

**Sviluppato a partire dal template di progetto "echo server" di SDK**

# Test setup



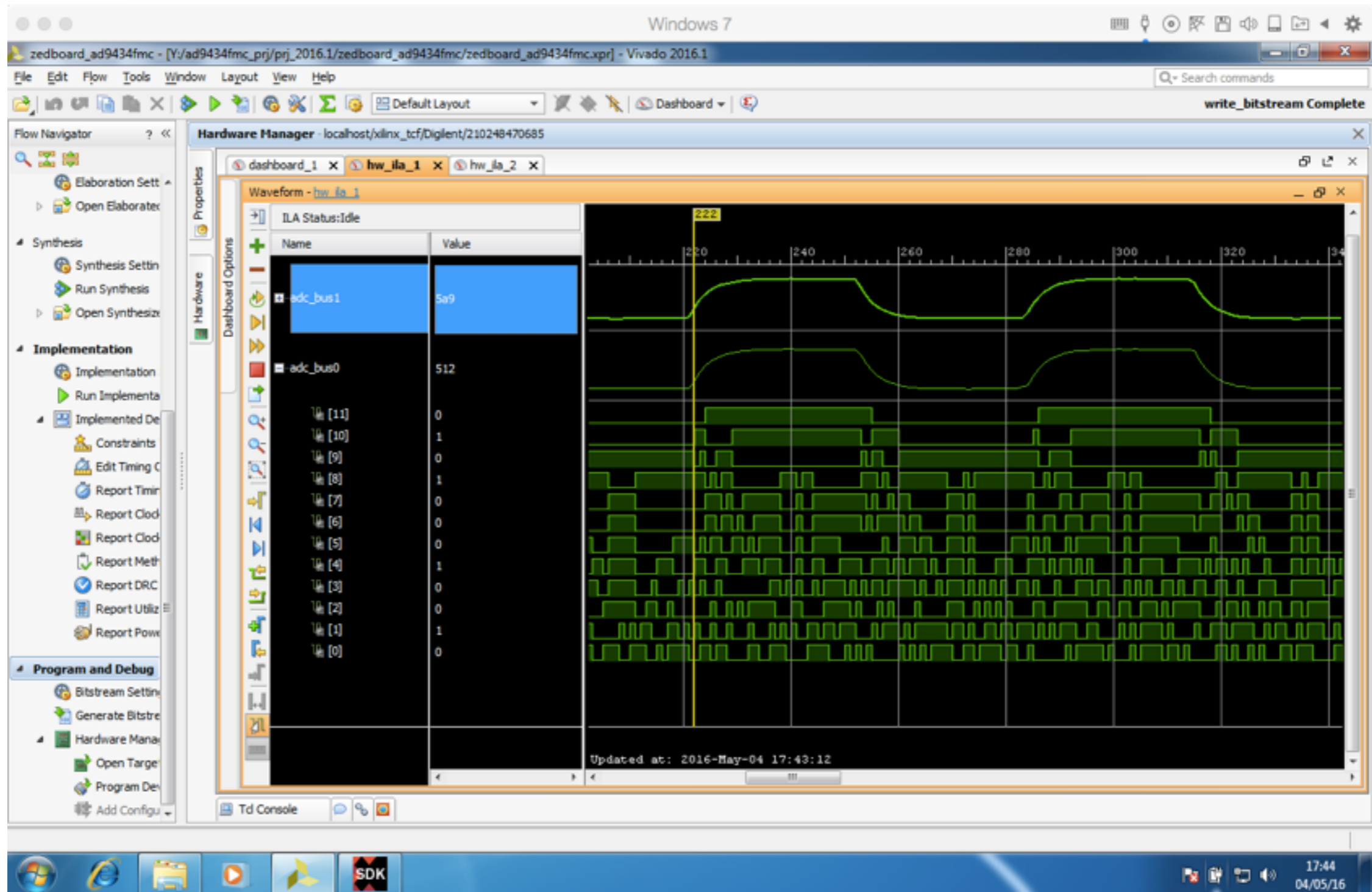


Analog Discovery 2  
Digilent

100MSPS USB Oscilloscope,  
Logic Analyzer,  
Arbitrary Function Generator  
(20MHz BW)

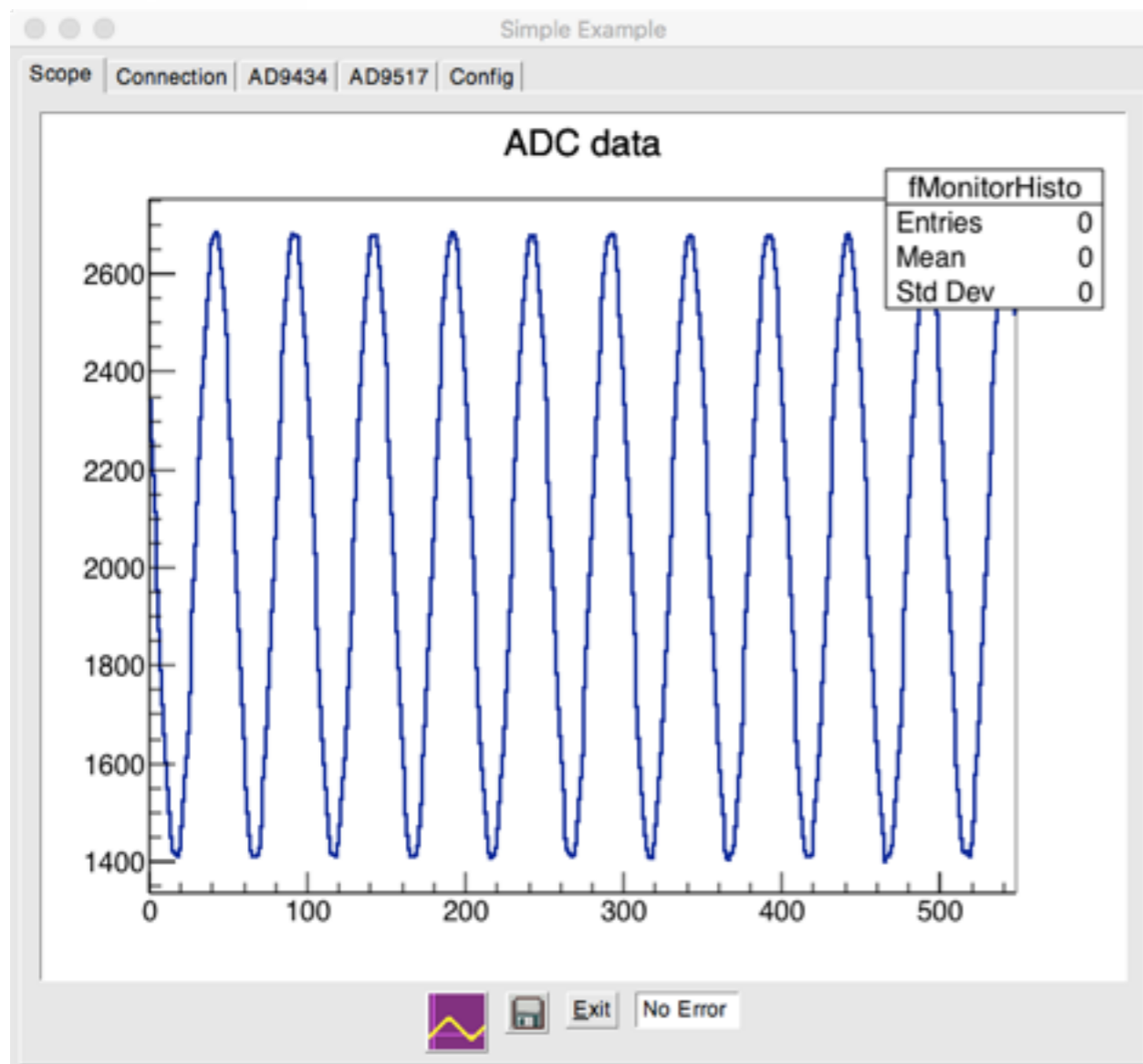
<http://store.digilentinc.com/analog-discovery-2-100msps-usb-oscilloscope-logic-analyzer-and-variable-power-supply/>

## Onda quadra a 2 MHz generata con Analog Discovery 2

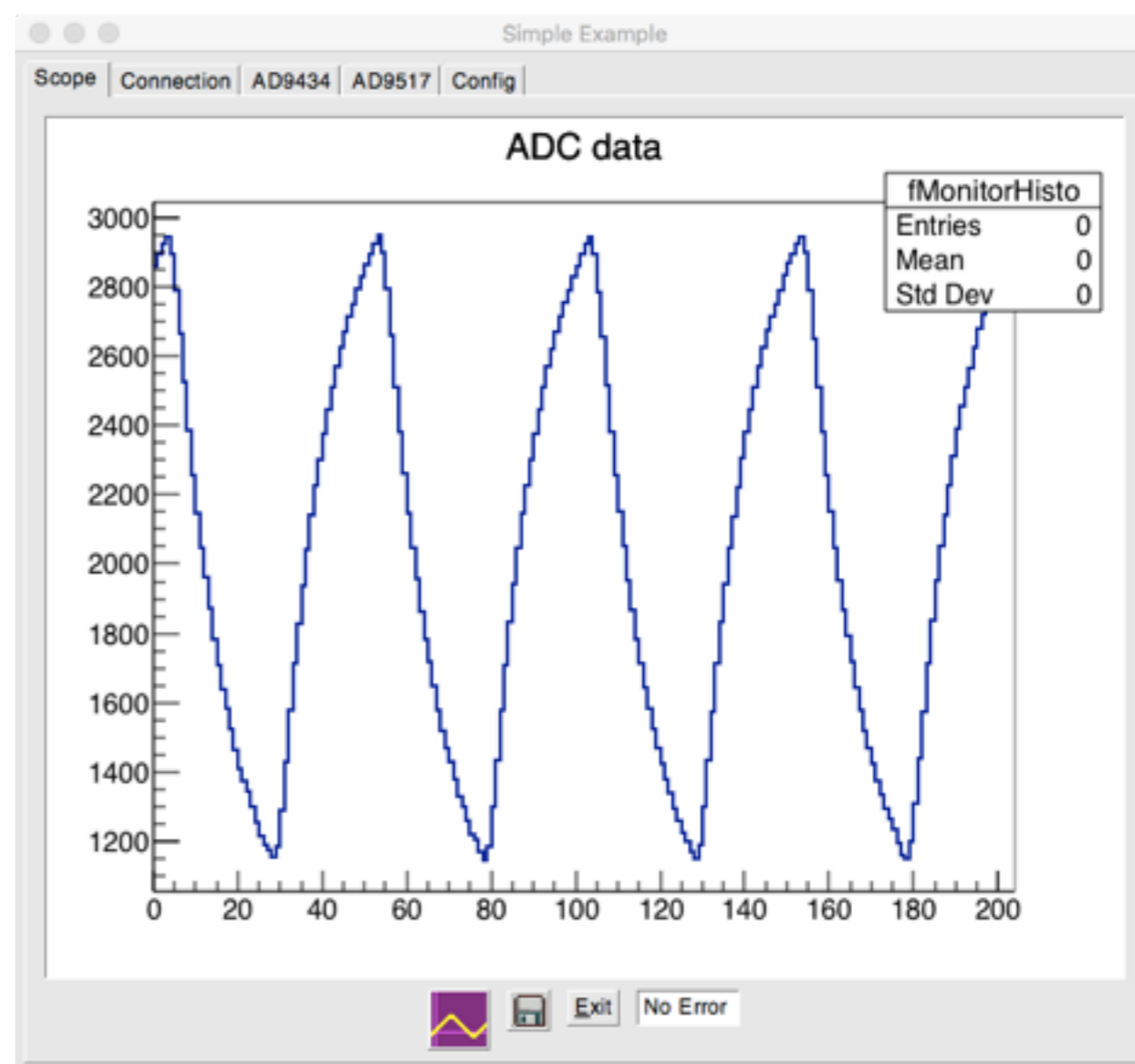


**1 periodo : ~ 60 campioni a 125 MHz (~ 2 MHz)**





**input sinusoida a 10 MHz**  
**1 periodo : 50 campioni a 500 MHz**



**input triangolare a 5 MHz**  
**1 periodo: 100 campioni a 500 MHz**

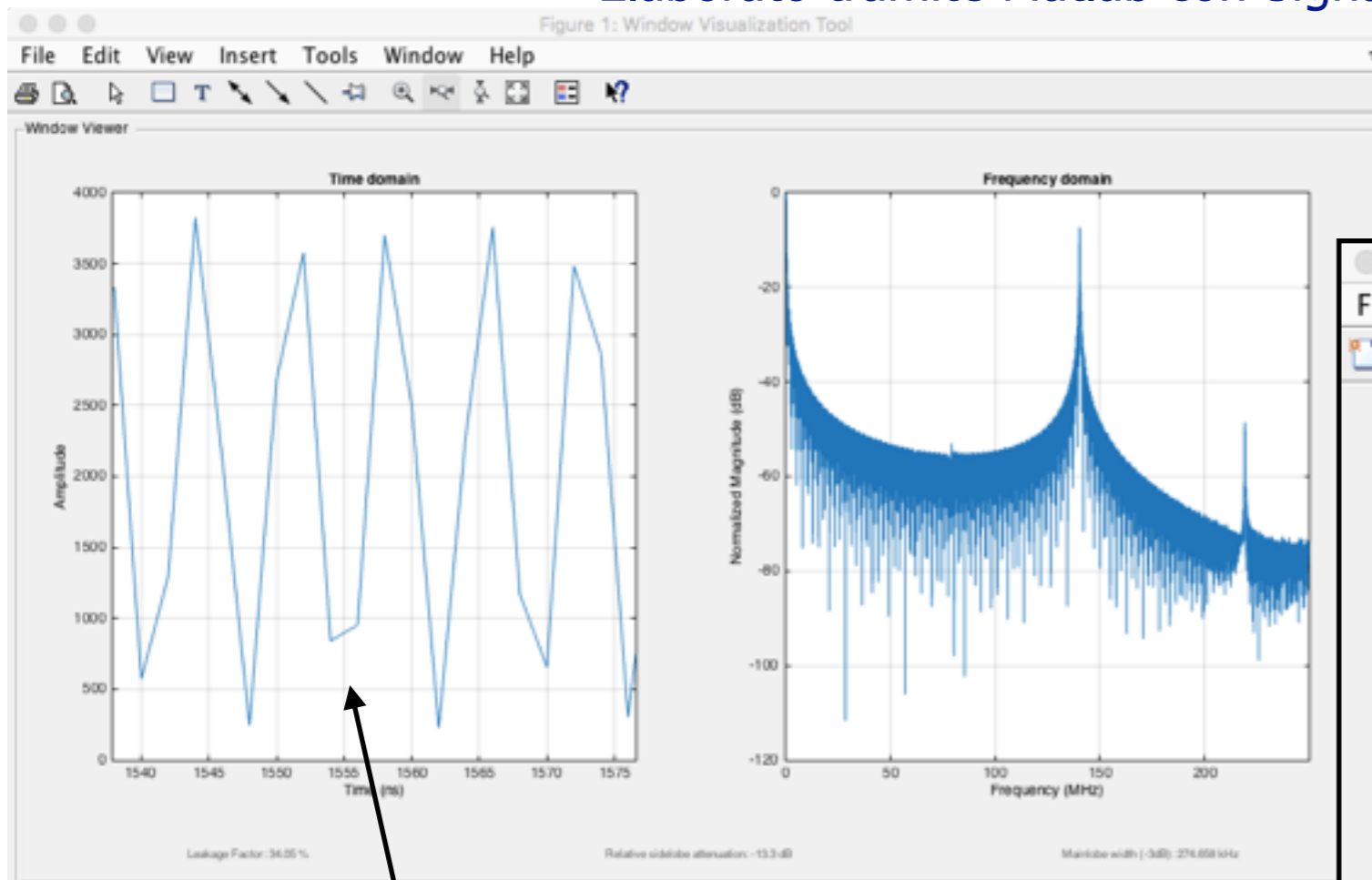
Graphical User Interface scritta in ROOT\* per la configurazione dei dispositivi  
l'acquisizione di campioni, il monitoring e il salvataggio su file (ascii)

\* <https://root.cern.ch>

# Esempio di Analisi

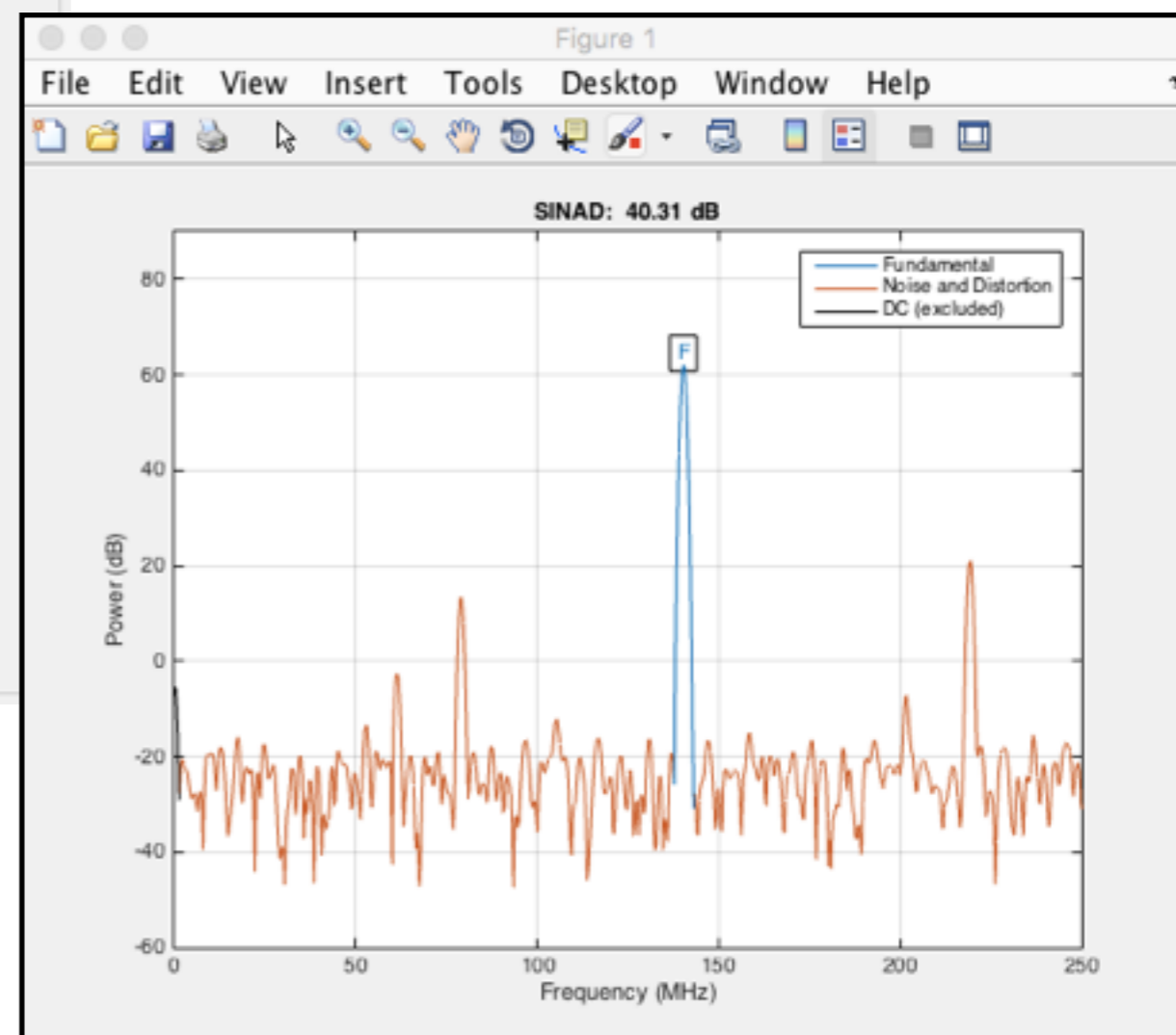
Acquisizione di un segnale sinusoidale a 140.3 MHz  
(1600 campioni)

Elaborato tramite Matlab con Signal Processing Toolbox



zoom

(sottoinsieme dei 1600 campioni)





# Considerazioni conclusive sull'utilizzo di FPGA Xilinx Zynq

😊 Documentazione e tutorials

😊 Facilita' di apprendimento

😊 Board commerciali di sviluppo

😊 Tempi di design

😊 Implementazione di protocolli  
standard (SPI, Ethernet...)

😊 Sistemi di test

😊 Acquisizione ed elaborazioni dati

🧠 Tolleranza radiazioni

🧠 Utilizzo nello spazio

😊 Vivado

🧠 Debug con Integrated Logic  
Analyzer

🧠 SDK

😊 Integrazione con moduli VHDL

🧠 Design AXI-“centrico”

😊 IP integrator

🧠 Configurazione e sw di alcuni IP  
cores (DMA)

😊 Creazione custom IP

🧠 Update di custom IP

- The Zynq Book (free PDF) : <http://www.zynqbook.com>
- The Zynq Book Tutorial (free) <http://www.zynqbook.com/download-tuts.html>
- Zynq-7000 All Programmable SoC. (Z-7010, Z-7015, and Z-7020): DC and AC Switching Characteristics. DS187 (v1.17) November 24, 2015
- LogiCORE IP AXI DMA v7.1 PG021
- Zynq-7000 AP SoC Technical Reference Manual UG585
- Vivado Design Suite AXI Reference Guide UG1037
- <http://zedboard.org>