

# Real Time Embedded Systems

## **Embedded System design example**

René Beuchat  
Laboratoire d'Architecture des  
Processeurs

*rene.beuchat@epfl.ch*

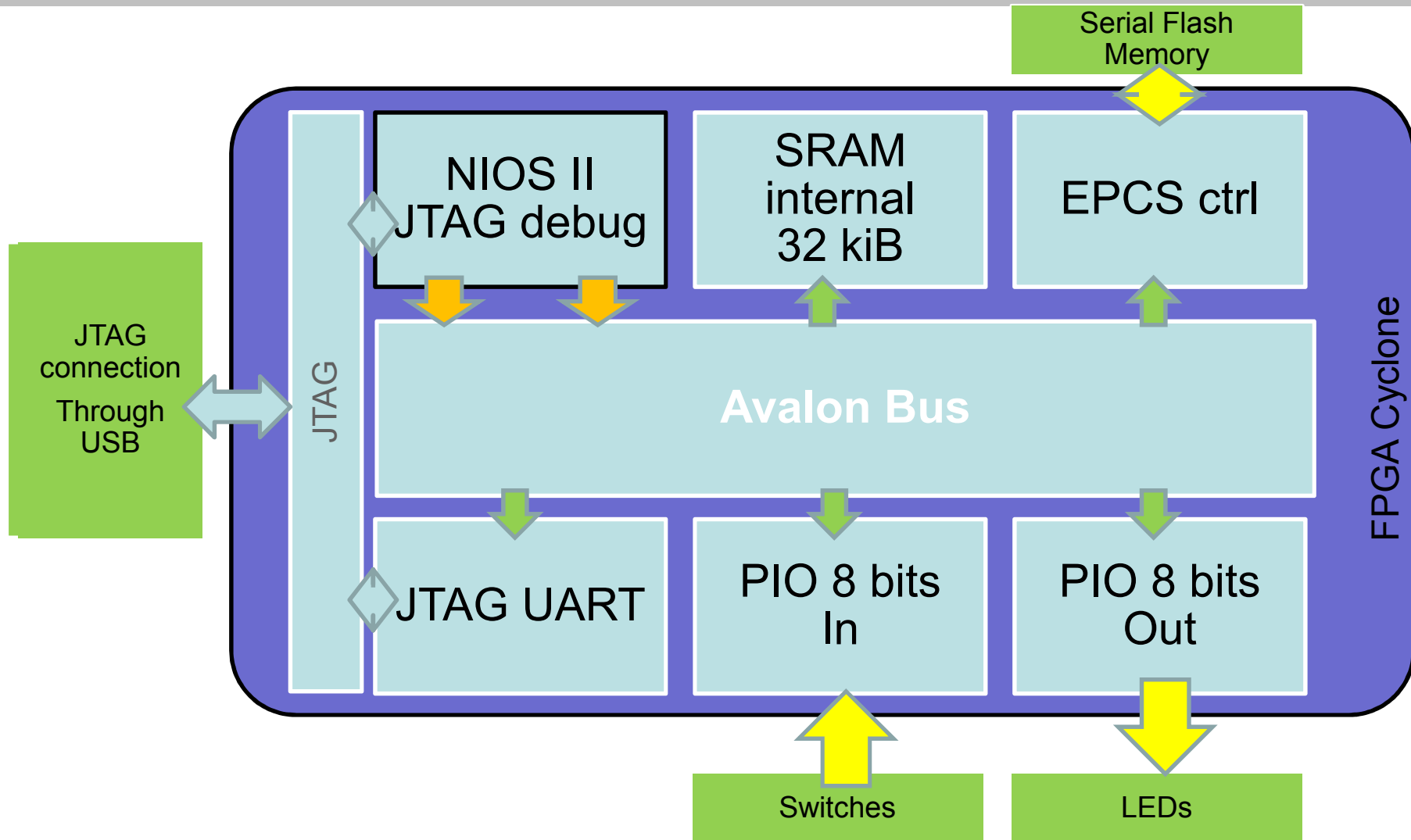
# System to design, requests

- Softcore processor: NIOSII standard version
  - 4kiB instruction cache
  - no data cache
- Memories:
  - on-chip memory SRAM, 16 kiB with 32 bits width
  - epcs controller for external Flash memory
- Programmable interfaces:
  - JTAG UART for debug purpose
  - Parallel port **In 8 bits**
  - Parallel port **Out 8 bits**



# Design Environment

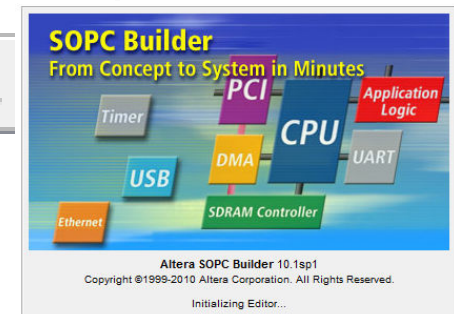
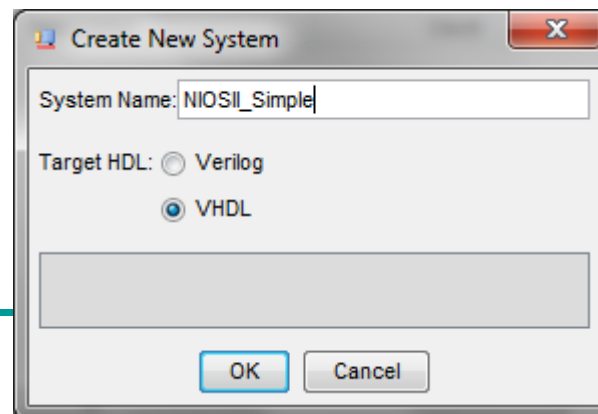
- Goal: Design of an Embedded system on FPGA
  - Specifically for Altera Cyclone device
  - System design with:
    - SOPC or Qsys environment on QuartusII
  - VHDL for specific modules design
  - ModelSim-Altera simulation tools
  - Software in C with:
    - NIOS IDE (Legacy) or
    - NIOS SBT

# General Bloc Schematic



# How to design it ?

- QuartusII → New project
  - Cyclonell -> EP2C20F484C8
- New schematic File
  - Files → New → Bloc Diagram/Schematic
  - Files → Save as... (*Labo\_NIOS\_InOut.bdf*)
- SOPC Builder/Qsys (clic-clic or 
  - Select VHDL
  - Give a name



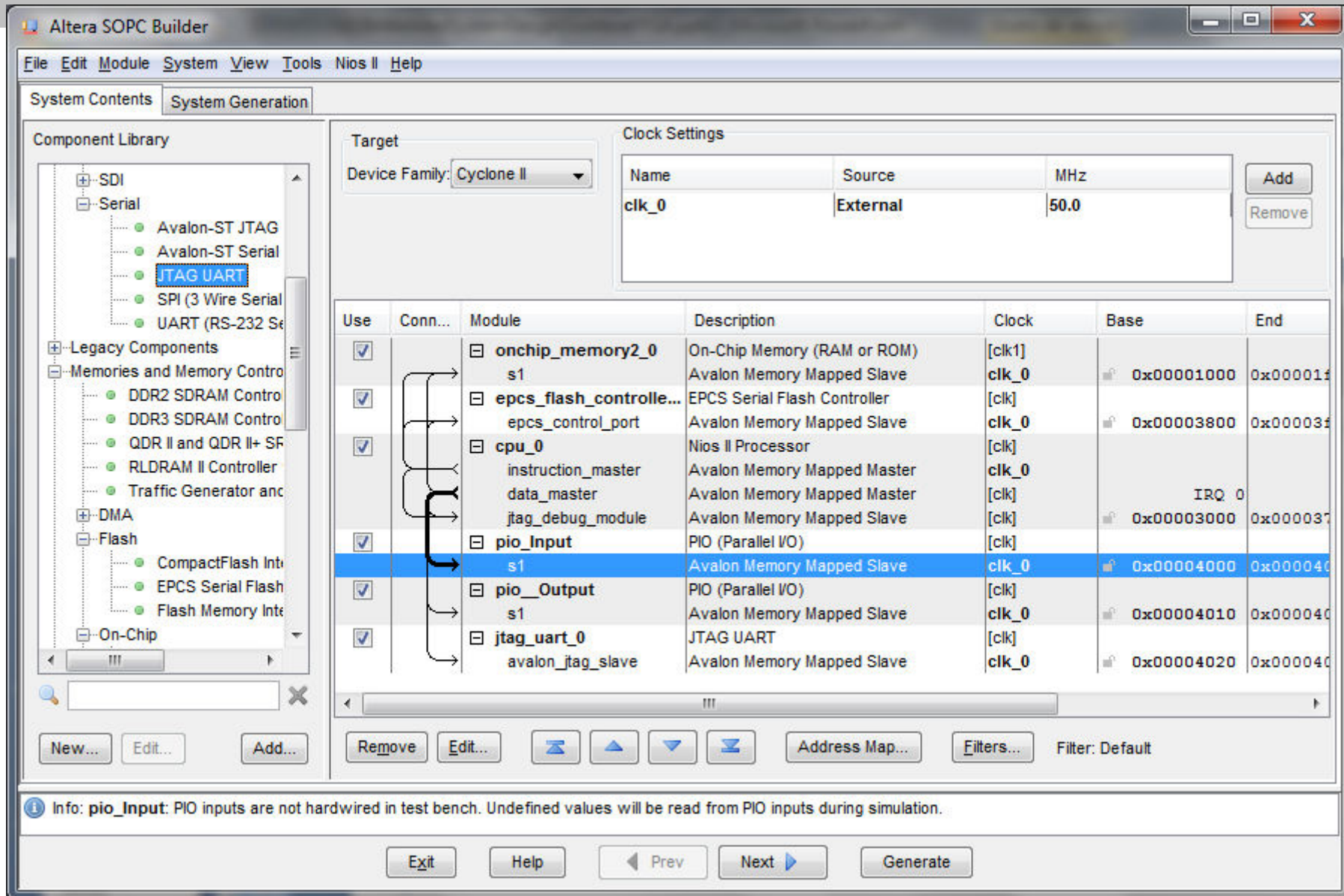
## SOPC Add Components

- → Memories → On-Chip → RAM 4kiBytes
- → Memories → Flash → EPCS Ctrl
- → Processors → NIOSII → /s
  - Reset Vectors → On-Chip SRAM
  - Interrupt Vectors → On-Chip SRAM
  - Cache Instruction → 4kiB
  - JTAG → Level 2

## SOPC Add Components

- → Peripherals → uC Peripheral → PIO
  - 8 bits Input
  - 8 bits Output
- → Interface Protocols → Serial →
  - JTAG UART
- System → Auto Assign Base Addresses

# SOPC View



Altera SOPC Builder

File Edit Module System View Tools Nios II Help

System Contents System Generation

Component Library

- SDI
  - Serial
    - Avalon-ST JTAG
    - Avalon-ST Serial
    - JTAG UART**
    - SPI (3 Wire Serial
    - UART (RS-232 Se
- Legacy Components
- Memories and Memory Contro
  - DDR2 SDRAM Contro
  - DDR3 SDRAM Contro
  - QDR II and QDR II+ SF
  - RLDRAM II Controller
  - Traffic Generator anc
- DMA
- Flash
  - CompactFlash Int
  - EPCS Serial Flash
  - Flash Memory Inte
- On-Chip

Target

Device Family: Cyclone II

Clock Settings

Name	Source	MHz
clk_0	External	50.0

Use Conn... Module Description Clock Base End

<input checked="" type="checkbox"/>		onchip_memory2_0	On-Chip Memory (RAM or ROM)	[clk1]		
<input checked="" type="checkbox"/>		s1	Avalon Memory Mapped Slave	clk_0	0x00001000	0x00001f
<input checked="" type="checkbox"/>		epcs_flash_controle...	EPCS Serial Flash Controller	[clk]		
<input checked="" type="checkbox"/>		epcs_control_port	Avalon Memory Mapped Slave	clk_0	0x00003800	0x00003f
<input checked="" type="checkbox"/>		cpu_0	Nios II Processor	[clk]		
<input checked="" type="checkbox"/>		instruction_master	Avalon Memory Mapped Master	clk_0		
<input checked="" type="checkbox"/>		data_master	Avalon Memory Mapped Master	[clk]		IRQ 0
<input checked="" type="checkbox"/>		jtag_debug_module	Avalon Memory Mapped Slave	[clk]	0x00003000	0x00003f
<input checked="" type="checkbox"/>		pio_input	PIO (Parallel I/O)	[clk]		
<input checked="" type="checkbox"/>		s1	Avalon Memory Mapped Slave	clk_0	0x00004000	0x00004f
<input checked="" type="checkbox"/>		pio__Output	PIO (Parallel I/O)	[clk]		
<input checked="" type="checkbox"/>		s1	Avalon Memory Mapped Slave	clk_0	0x00004010	0x00004f
<input checked="" type="checkbox"/>		jtag_uart_0	JTAG UART	[clk]		
<input checked="" type="checkbox"/>		avalon_jtag_slave	Avalon Memory Mapped Slave	clk_0	0x00004020	0x00004f

Remove Edit... Address Map... Filters... Filter: Default

Info: pio\_input: PIO inputs are not hardwired in test bench. Undefined values will be read from PIO inputs during simulation.

Exit Help Prev Next Generate



# System Generation

- Generate
  - Produce Avalon bus module and all the interconnections
- Include in the schematic
  - Add Pins Input/Output
  - Clk with PLL (24 MHz to 50MHz)
  - Tcl files for pin numbering
- Compile

- Correct the errors (if any)
- Nice ! The design is ready
  
- Download on the board
  - (FPGA4U/robot or other)
  
- Run NIOSII IDE and write the software
  - Copy In port on Out Port by software