#### SISTEMI EMBEDDED

Tutorial on creating and using a custom component in a Qsys system

Federico Baronti

Last version: 20180423

#### Introduction

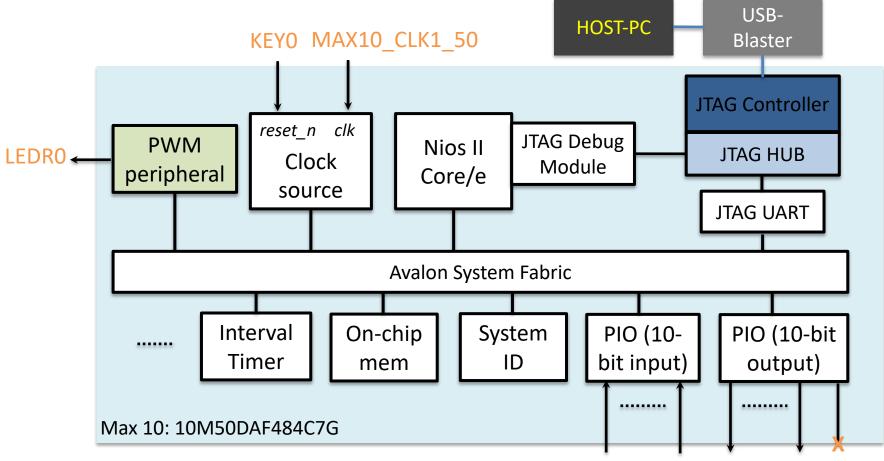
• Problem:

– Use a *custom component* within a Qsys system

- Solution:
  - Provide the *custom component* with standardized interfaces: clock, reset, memory-mapped, ...
  - Add the *custom component* to the Qsys Library
  - Instantiate, configure and connect the *custom component* as any other component in the system
  - Provide an appropriate device driver to control the *custom component* from the software application

#### Example: PWM peripheral

• Provide my\_first\_computer (enriched with Watchdog, Interval Timer, JTAG UART) with a PWM peripheral to control the brightness of a LED



SW9 ... SW0 LEDR9...LEDR1

### PWM Peripheral (1)

- Memory-mapped with 3 regs: CONTROL (1 bit), PERIOD (32-bit) and COMPARE (32-bit)
- It is based on a **32-bit up-counter** 
  - The counter is forced to 0 when reaches the content of the PERIOD reg
- The PWM output is set when the counter goes from the PERIOD value to 0 and cleared when reaches the COMPARE value

### PWM Peripheral (2)

- CONTROL reg allows you to enable/disable the PWM output (when disabled the output is low)
- The PERIOD reg allows you to set the PWM period  $T_{PWM} = T_{clk} * (PERIOD+1)$

A write to the PERIOD reg clears the counter

- The COMPARE reg allows you to set duty cycle  $\boldsymbol{\delta}$ 

 $\delta = \frac{COMPARE + 1}{PERIOD + 1}$ 

 The COMPARE reg is buffered and is updated when the counter goes from *PERIOD* to 0

### PWM Peripheral (3)

• Module interface

```
module unipi_se_pwm (
        // inputs:
        address,
        clk,
         reset n,
         read,
        write,
        writedata,
        // outputs:
         readdata,
         pwm_out
         );
```

// parameters
parameter RESET\_PERIOD = 0;
parameter RESET\_COMPARE = 0;
parameter RESET\_PWM\_ENABLE = 0;

The HDL code is provided (unipi\_se\_pwm.v)

#### PWM Peripheral (4)

Signal name		Avalon interface	Role	
clk	IN	Clock	clock	
reset_n	IN	Reset	reset_n	
address	IN	Memory-Mapped slave	address	
read	IN	Memory-Mapped slave	read	
write	IN	Memory-Mapped slave	write	
writedata	IN	Memory-Mapped slave	writedata	
readdata	OUT	Memory-Mapped slave	readdata	
pwm_out	OUT	Conduit	Export	

### Add custom component to Qsys (1)

- Component Library / Project / New Component
  - *Component Type* Name your custom
     component

		wm_hw.tcl*	
ile Templates Component Typ	e Files Parameters	Signals Interfaces	
About Comp			
, About boing	ponone rypo		
Name:	SE_Unipi_pwm		
Display name:	SE_Unipi_pwm		
Version:	1.0		
Group:	SE_Unipi		▼
Description:			
Created by:			
con:			
Documentation:	Title	URL	
	+ -		
) Info: No erro	rs or warnings.		
Info: No erro	rs or warnings.		
Info: No erro	rs or warnings.		

# Add custom component to Qsys (2)

• Component Library / Project / New Component

*Files* Select the HDL
 file(s)
 Analyze Synthesis
 Files

👃 Component Editor - SE_Unipi_	pwm_hw.tcl*						
File Templates							
Component Type Files Parameters Signals Interfaces							
<ul> <li>About Files</li> </ul>							
Synthesis Files							
These files describe this component's implementation, and will be created when a Quartus II synthesis model is generated.							
The parameters and signals found in	n the top-level module will be used fo	or this component's paran	neters and signals.				
Output Path	Source File	Туре	Attributes				
unipi_se_pwm.v	IPs/Unipi_SE_pwm/Verilog/unipi_s	Verilog HDL	Top-level File				
Top-level Module: (Analyze files to Verilog Simulation Files	+       -       Analyze Synthesis Files       Create Synthesis File from Signals         Top-level Module:       (Analyze files to select module)       -         Verilog Simulation Files       -       -         These files will be produced when a Verilog simulation model is generated.       -						
Output Path	Source File	Туре	Attributes				
(No files)							
+ - Copy from Synthesi	is Files						
VHDL Simulation Files							
These files will be produced when a	-	3.					
Output Path	Source File	Туре	Attributes				
(No files)							
+ - Copy from Synthesis Files							
Info: No errors or warnings.							
	Help Prev	Next 🕨 🛛 Fi	nish				

## Add custom component to Qsys (3)

Component Library / Project / New Component

Signals
 Assign each signal of the module
 to the appropriate
 Avalon Interface
 and the relevant
 signal role

Component Type File	s Parameters Signals	Interfaces			
About Signals					
Name	Interface		Signal Type	Width	Direction
ddress	avalon_slave_	0	address	2	input
vrite	avalon_slave_	0	write	1	input
vritedata	avalon_slave_	avalon_slave_0 writedata 32 input			
eaddata	avalon_slave_	0	readdata	32	output
ead	avalon_slave_	0	read	1	input
lk	clock		clk	1	input
eset_n	reset		reset_n	1	input
owm_out	conduit_end		export	1	output
		Add Signal	Remove Signal		
Info: No errors or w	arnings.	Add Signal	Remove Signal		

### Add custom component to Qsys (4)

- Component Library / Project / New Component
- *Interfaces* Configure each
   Avalon Interface

A0

A0

Read Waveforms

Write Waveforms

clk read write address

clk read write address

readdata

writedata

	👃 Component Editor - SE_Unipi_pwm_hw.tcl*		×			
	File Templates					
	Component Type Files Parameters Signals Interfaces					
	About Interfaces	faces				
each	* "avalon_slave_0" (Avalon Memory Mapped Slave)       Name:     avalon_slave_0       Documentation       Type:     Avalon Memory Mapped Slave					
_						
erface						
CITACC						
	Associated Clock: clock		=			
	Associated Reset: reset					
	Assignments: Edit					
	Block Diagram		_			
	Address units:	WORDS -	_			
	avalon slave 0 Associated clock:	WORDS	_			
	Associated reset:		_			
	avalon_slave_0 	reset	_			
	vrite address Burstcount units:	WORDS -				
<u> </u>	writedata[310] writedata Explicit address span:	000000000000000000000000000000000000000				
	read read		_			
Х	null		_			
	Read wait:	1				
	Write wait:	0				
	пою	0				
	Timing units:	Cycles 🗸				
	V Pipelined Transfers					
	Read latency:	0				
	Maximum pending read transactions	: 0				
	Burst on burst boundaries only		+			
			•			
X						
	_					

## Add custom component to Qsys (5)

• Component Library / Project / New Component

– Parameters	👍 Compor	nent Editor - SE_U	nipi_pwm_hw.te	cl*				
- Furumeters	File Templat	File Templates						
Set default	Component	Component Type Files Parameters Signals Interfaces						
volues	▹ About F	About Parameters						
values,								
		Name	Default Value	Edita	Туре	Group	Tooltip	
		RESET_PERIOD	499999	<b>V</b>	integer			
	Parameters:	RESET_COMPARE RESET_PWM_EN		<b>√</b>	integer integer			
		AUTO_CLOCK_C	-1	<b>V</b>	inteq 🖵		clock_rate for inter	
Preview - SE_Unipi_pwm								
SE_Unipi_pwm			Remove P	arameter				
SE_Unipi_pwm		Documentation						
Block Diagram	Parameters RESET_PERIOD:		•					
Show signals	RESET_COMPARE:	499999 249999						
SE_Unipi_pwm_0	RESET_PWM_ENABL	E: 1						
avalon_slave_0								
clock								
reset								
conduit_end conduit								
SE_Unipi_pwm								

#### Putting into practice (1)

- Instantiate, configure and connect the SE\_unipi\_pwm component
- Generate the Qsys system
- Back to Quartus II
  - Update the nios\_system instance to include the pwm output port and connect it to LEDR[0]
- Compile the project and configure the FPGA
  - If you have used the default values for the PWM Peripheral, the LEDR[0] should be on with average brightness

#### Putting into practice (2)

- Time to start an Eclipse project
- Write a program that allows you to control the brightness of LEDR[0] by means of the SW7-Sw0
  - Try to use the provided device drivers: unipi\_se\_pwm.c, unipi\_se\_pwm.h