LAB 4:
Avalon Interface

By: Negar Aghapour

17 December 2022

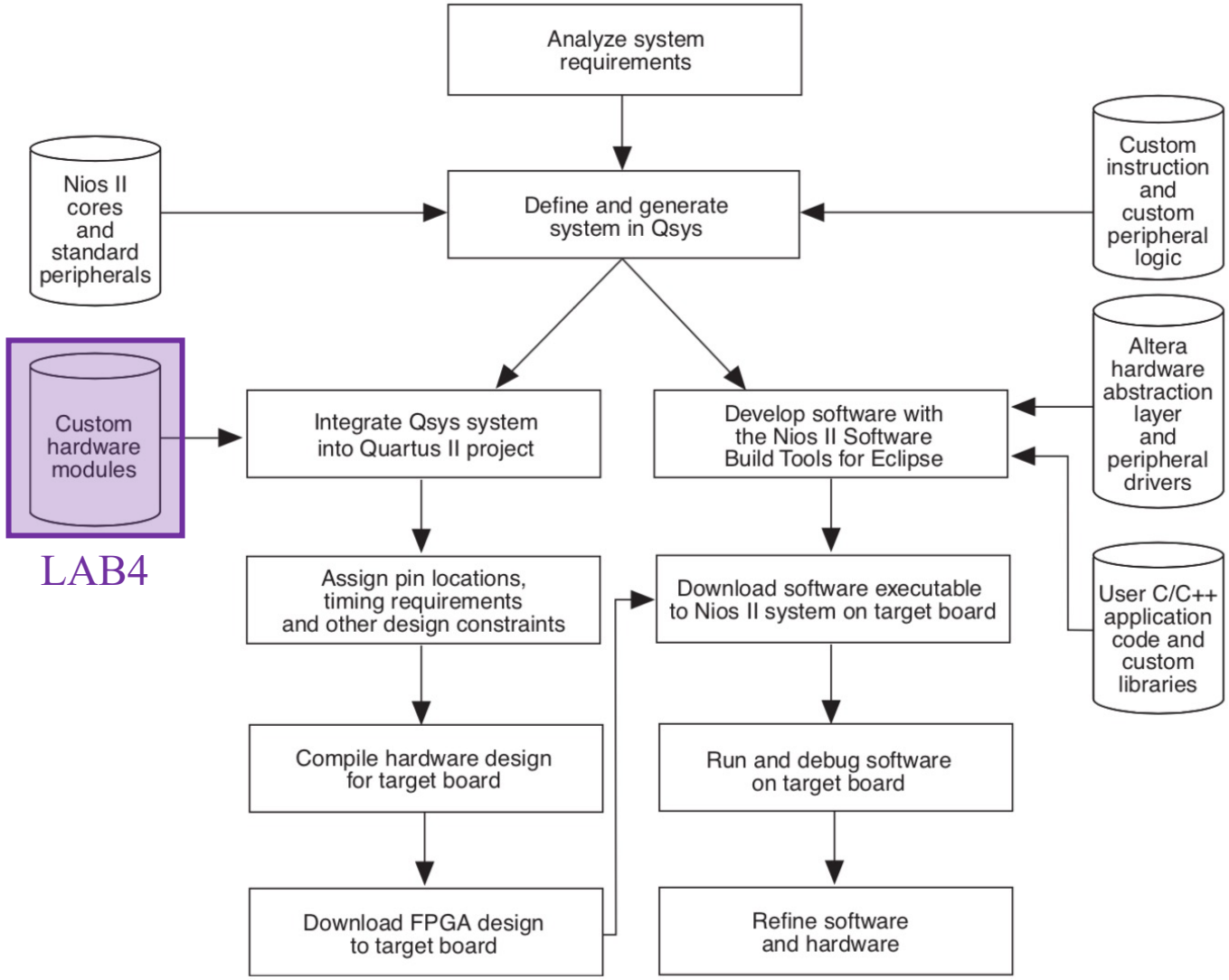
FPGA-based Embedded System Design

School of Electrical and Computer Engineering, College of Engineering
University of Tehran

Design, Verification & Debugging of
Embedded Systems LAB

DVDES

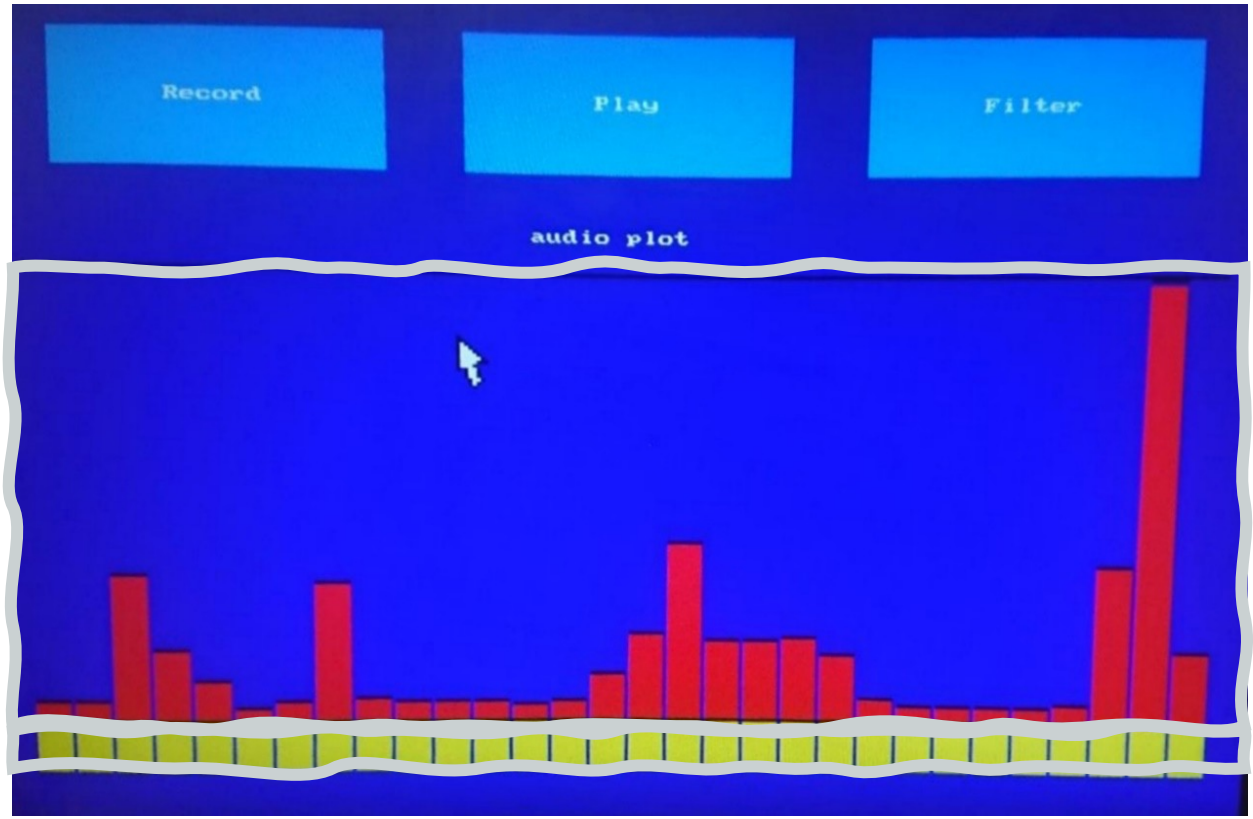
Nios II development flow



LAB4

Using Lab2 Audio Player (Record and Play)

Playing voice displays



Software

Software

RECORD

1. Record voice for 10 seconds
2. Divide recording buffer into N (given) parts
3. Calculate average of each part
4. Display recorded voice amplitude (red rectangles)
   - Scale the height of the rectangles

DVDES

Software

PLAY

1. Play Sound for 10 seconds
2. Display playing sound location (yellow rectangles)
3. Measure processing time

DVDES

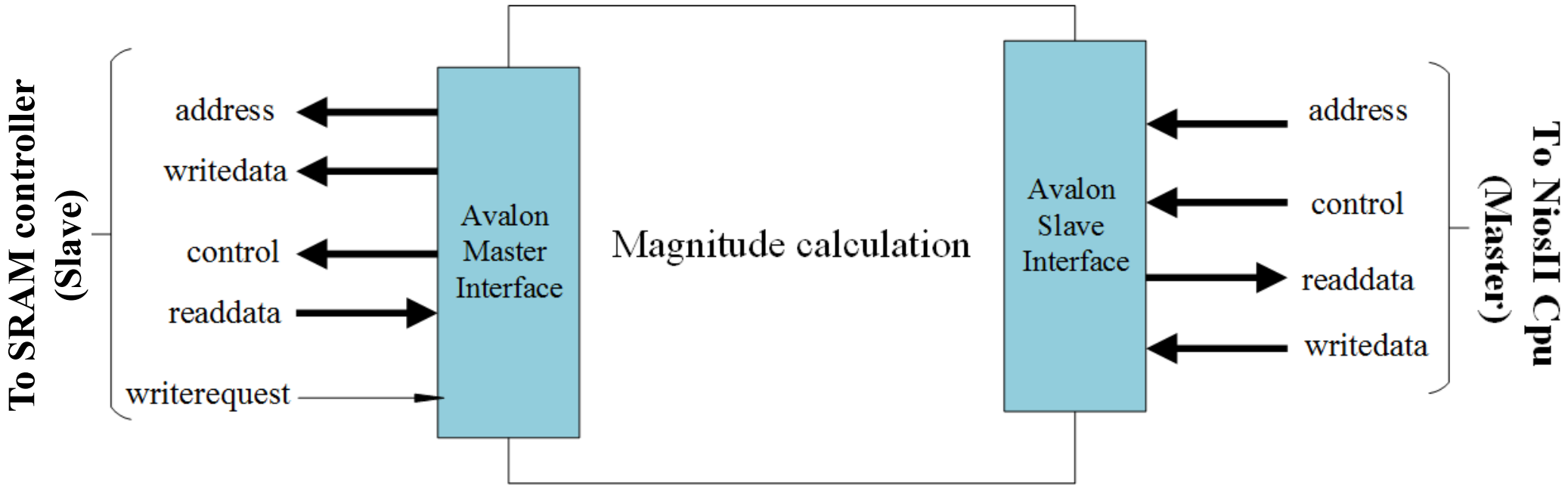
Hardware

Hardware

1. Calculate the summation of sound volume by hardware
   - Get number of parts (Num)
   - Get size of each part
   - Get the address of right buffer
   - Get the address of left buffer
   - Get the address of the output data
2. Calculate the average and sketch the diagram by software
3. Measure calculation time

Hardware

Communication with SRAM and CPU is required



Hardware (Slave Interface)

Registers:

Slave Address	31	30..12	11..1	0	
00	Done	Size	Num	Go	**Config. Reg.**
01					**Right Addr.**
10					**Left Addr.**
11					**Out Addr.**

Config. Reg. addr. = #Base addr. + 0x00

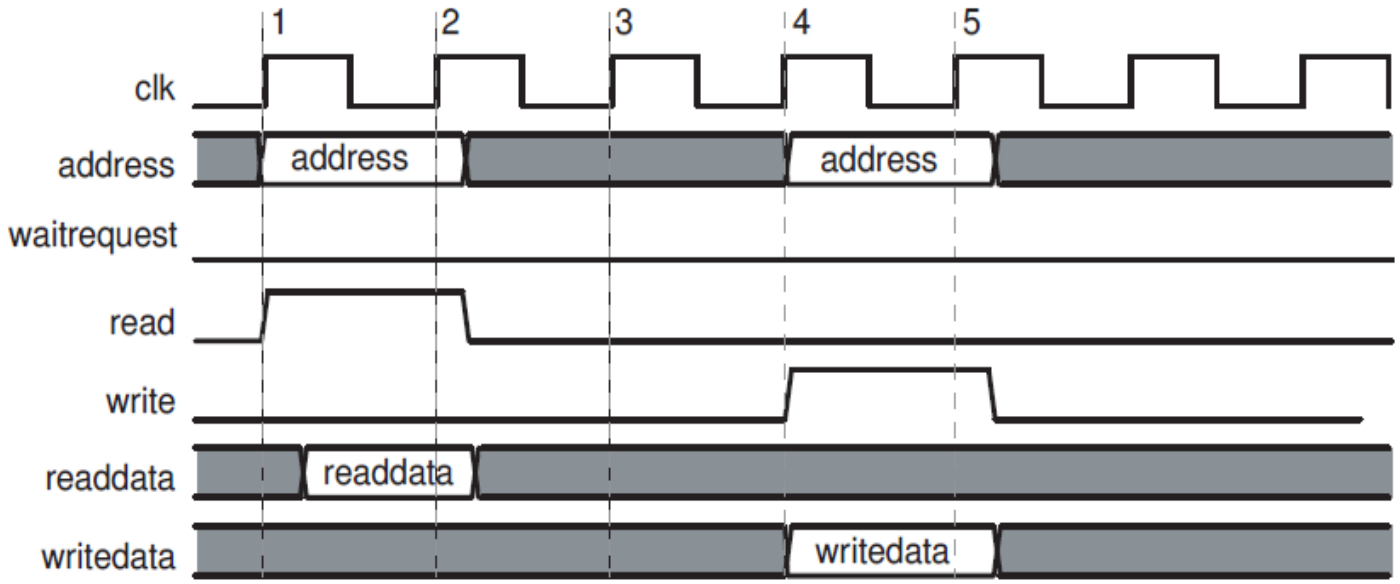
Right Buf. Reg. addr. = #Base addr. + 0x04

Left Buf. Reg. addr. = #Base addr. + 0x08

Output Buf. Reg. addr. = #Base addr. + 0x0C

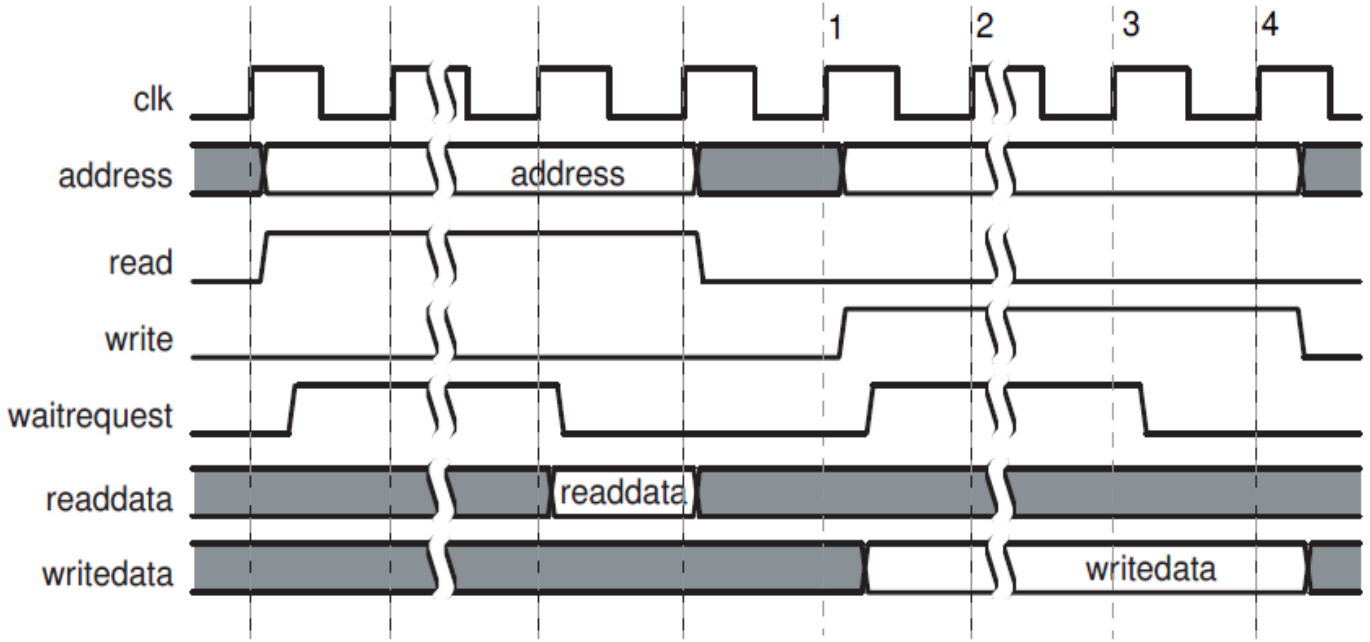
Hardware (Slave Interface)

Without waitrequest (wait cycle = 0)
Assume chip select = 1



Hardware (Master Interface)

With waitrequest (from SRAM)
Assume chip select = 1



What to do

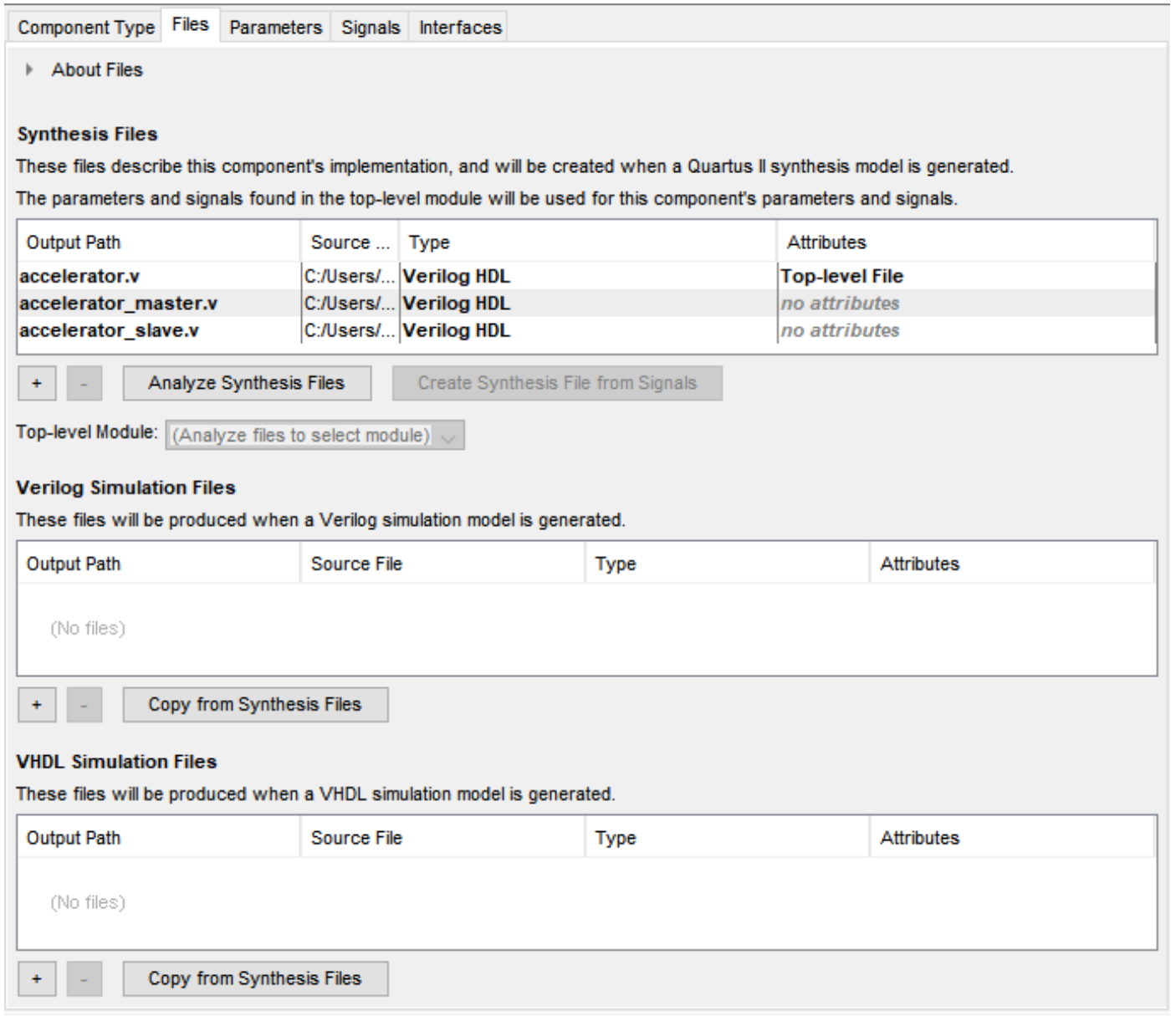
Write module in HDL.

Debug and synthesis modules.

Add magnitude calculation circuit to Qsys as new component

Generate new .sof file and program the FPGA

Qsys


Component Type
Files
Parameters
Signals
Interfaces
About Files
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 the top-level module will be used for this component's parameters and signals.
Output Path
Source ...
Type
Attributes
accelerator.v
C:/Users/...
Verilog HDL
Top-level File
accelerator_master.v
C:/Users/...
Verilog HDL
no attributes
accelerator_slave.v
C:/Users/...
Verilog HDL
no attributes
+
-
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
Type
Attributes
(No files)
+
-
Copy from Synthesis Files
VHDL Simulation Files
These files will be produced when a VHDL simulation model is generated.
Output Path
Source File
Type
Attributes
(No files)
+
-
Copy from Synthesis Files

DVDES

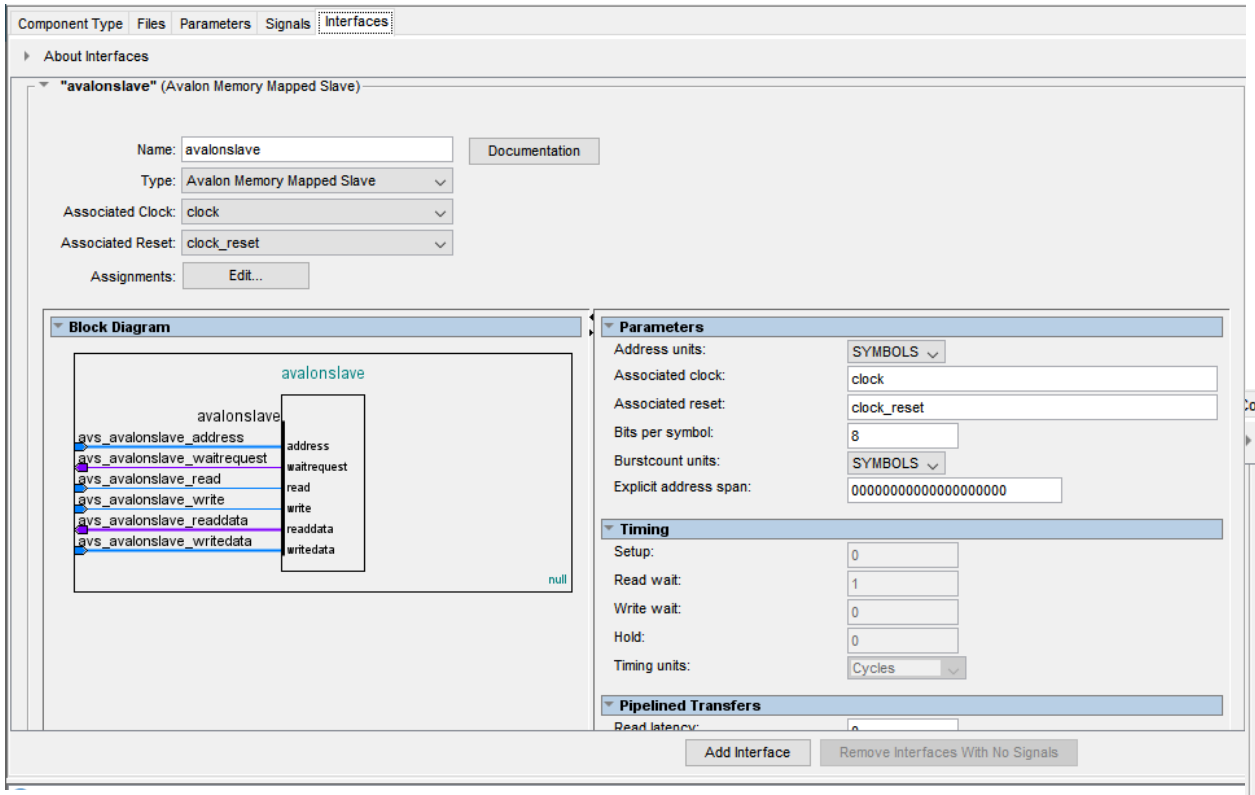
Qsys

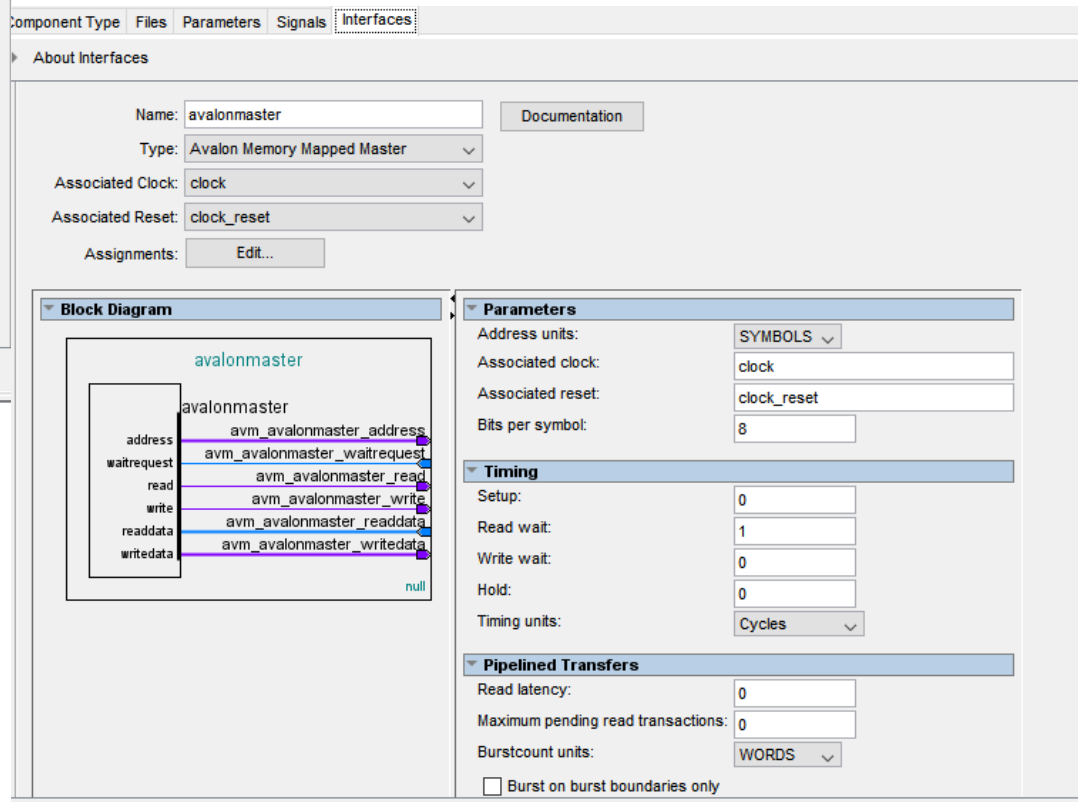
Component Type | Files | Parameters | Signals | Interfaces

About Signals

Name	Interface	Signal Type	Width	Direction
csi_clock_clk	clock	clk	1	input
csi_clock_reset	clock_reset	reset	1	input
avs_avalonslave_address	avalonslave	address	avs_av...	input
avs_avalonslave_waitreq...	avalonslave	waitrequest	1	output
avs_avalonslave_read	avalonslave	read	1	input
avs_avalonslave_write	avalonslave	write	1	input
avs_avalonslave_readdata	avalonslave	readdata	avs_av...	output
avs_avalonslave_writedata	avalonslave	writedata	avs_av...	input
avm_avalonmaster_addr...	avalonmaster	address	avm_av...	output
avm_avalonmaster_waitr...	avalonmaster	waitrequest	1	input
avm_avalonmaster_read	avalonmaster	read	1	output
avm_avalonmaster_write	avalonmaster	write	1	output
avm_avalonmaster_read...	avalonmaster	readdata	avm_av...	input
avm_avalonmaster_write...	avalonmaster	writedata	avm_av...	output

Qsys



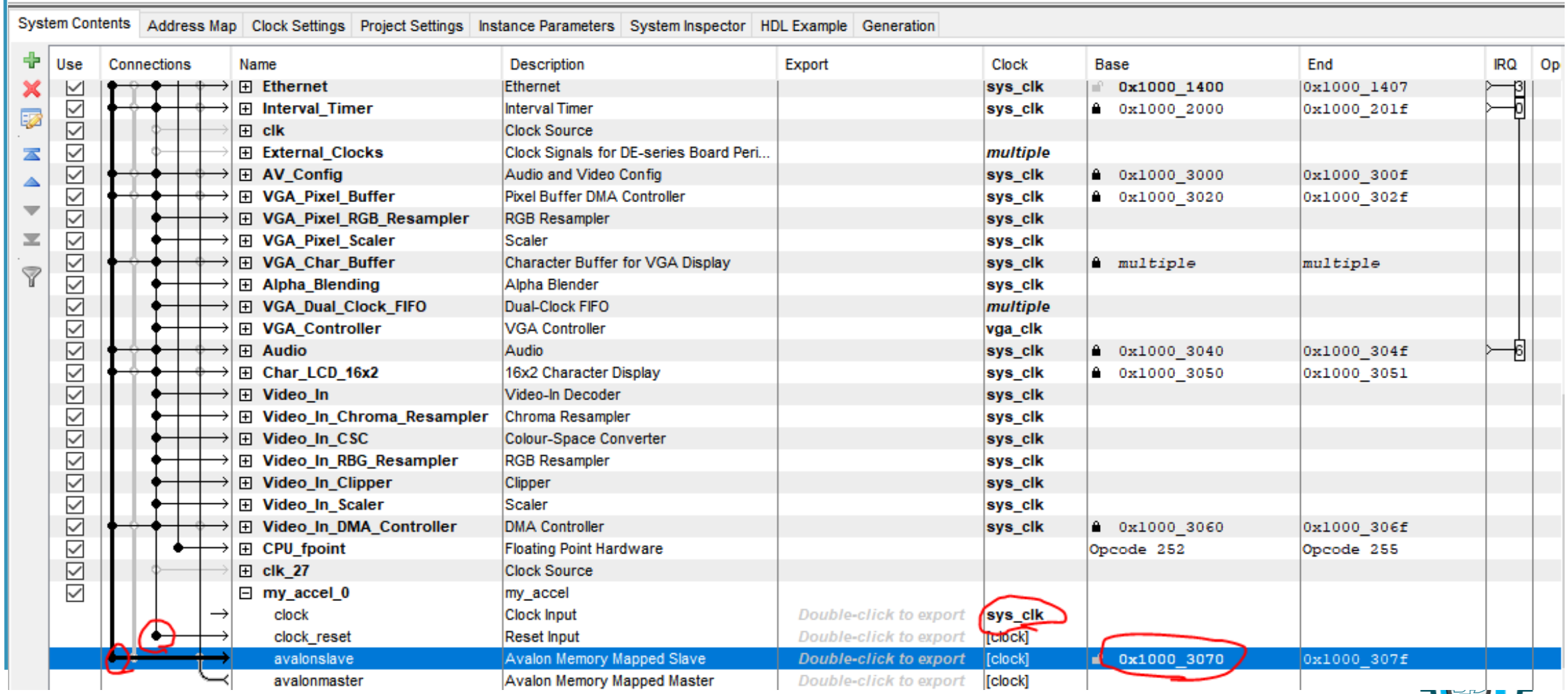


Remove modules with no signal

Use	C...	Name	Description	Export	Clock	Base	End	IRQ	Opcode Name
☑	→	SRAM	SRAM/SSRAM Controller		sys_clk	0x0800_0000	0x0807_ffff		
☑	→	SD_Card	SD Card Interface		sys_clk	0x0b00_0000	0x0b00_03ff		
☑	→	Flash	Altera UP Flash Memory IP Core		sys_clk	multiple	multiple		
☑	→	Red_LEDs	Parallel Port		sys_clk	0x1000_0000	0x1000_000f		
☑	→	Green_LEDs	Parallel Port		sys_clk	0x1000_0010	0x1000_001f		
☑	→	HEX3_HEX0	Parallel Port		sys_clk	0x1000_0020	0x1000_002f		
☑	→	HEX7_HEX4	Parallel Port		sys_clk	0x1000_0030	0x1000_003f		
☑	→	Slider_Switches	Parallel Port		sys_clk	0x1000_0040	0x1000_004f		
☑	→	Pushbuttons	Parallel Port		sys_clk	0x1000_0050	0x1000_005f	1	
☑	→	Expansion_JP1	Parallel Port		sys_clk	0x1000_0060	0x1000_006f	11	
☑	→	Expansion_JP2	Parallel Port		sys_clk	0x1000_0070	0x1000_007f	12	
☑	→	PS2_Port	PS2 Controller		sys_clk	0x1000_0100	0x1000_0107	7	
☑	→	USB	USB Controller		sys_clk	0x1000_0110	0x1000_011f	2	
☑	→	JTAG_UART	JTAG UART		sys_clk	0x1000_1000	0x1000_1007	8	
☑	→	Serial_Port	RS232 UART		sys_clk	0x1000_1010	0x1000_1017	10	
☑	→	IrDA_UART	IrDA UART		sys_clk	0x1000_1020	0x1000_1027	9	
☑	→	Ethernet	Ethernet		sys_clk	0x1000_1400	0x1000_1407	3	
☑	→	Interval_Timer	Interval Timer		sys_clk	0x1000_2000	0x1000_201f	0	
☑		clk	Clock Source						
☑		External_Clocks	Clock Signals for DE-series Board Peri...		*multiple*				
☑	→	AV_Config	Audio and Video Config		sys_clk	0x1000_3000	0x1000_300f		
☑	→	VGA_Pixel_Buffer	Pixel Buffer DMA Controller		sys_clk	0x1000_3020	0x1000_302f		
☑		VGA_Pixel_RGB_Resampler	RGB Resampler		sys_clk				
☑		VGA_Pixel_Scaler	Scaler		sys_clk				
☑	→	VGA_Char_Buffer	Character Buffer for VGA Display		sys_clk	multiple	multiple		
☑		Alpha_Blending	Alpha Blender		sys_clk				
☑		VGA_Dual_Clock_FIFO	Dual-Clock FIFO		*multiple*				
☑		VGA_Controller	VGA Controller		vga_clk				
☑	→	Audio	Audio		sys_clk	0x1000_3040	0x1000_304f	6	
☑	→	Char_LCD_16x2	16x2 Character Display		sys_clk	0x1000_3050	0x1000_3051		
☑		Video_In	Video-In Decoder		sys_clk				
☑		Video_In_Chroma_Resampler	Chroma Resampler		sys_clk				
☑		Video_In_CSC	Colour-Space Converter		sys_clk				
☑		Video_In_RBG_Resampler	RGB Resampler		sys_clk				
☑		Video_In_Clipper	Clipper		sys_clk				
☑		Video_In_Scaler	Scaler		sys_clk				
☑	→	Video_In_DMA_Controller	DMA Controller		sys_clk	0x1000_3060	0x1000_306f		
☑		CPU_fpoint	Floating Point Hardware			Opcode 252	Opcode 255		
☑		clk_27	Clock Source						
☑		my_accel_0	my_accel						
	→	clock	Clock Input	*Double-click to export*	sys_clk				
	→	clock_reset	Reset Input	*Double-click to export*	[clock]				
	→	avalonslave	Avalon Memory Mapped Slave	*Double-click to export*	[clock]	0x1000_3070	0x1000_307f		
	→	avalonmaster	Avalon Memory Mapped Master	*Double-click to export*	[clock]				

DVDES

Qsys


System Contents
Address Map
Clock Settings
Project Settings
Instance Parameters
System Inspector
HDL Example
Generation
Use
Connections
Name
Description
Export
Clock
Base
End
IRQ
Op
Ethernet
Ethernet
sys_clk
0x1000_1400
0x1000_1407
Interval_Timer
Interval Timer
sys_clk
0x1000_2000
0x1000_201f
clk
Clock Source
External_Clocks
Clock Signals for DE-series Board Peri...
multiple
AV_Config
Audio and Video Config
sys_clk
0x1000_3000
0x1000_300f
VGA_Pixel_Buffer
Pixel Buffer DMA Controller
sys_clk
0x1000_3020
0x1000_302f
VGA_Pixel_RGB_Resampler
RGB Resampler
sys_clk
VGA_Pixel_Scaler
Scaler
sys_clk
VGA_Char_Buffer
Character Buffer for VGA Display
sys_clk
multiple
multiple
Alpha_Blending
Alpha Blender
sys_clk
VGA_Dual_Clock_FIFO
Dual-Clock FIFO
multiple
VGA_Controller
VGA Controller
vga_clk
Audio
Audio
sys_clk
0x1000_3040
0x1000_304f
Char_LCD_16x2
16x2 Character Display
sys_clk
0x1000_3050
0x1000_3051
Video_In
Video-In Decoder
sys_clk
Video_In_Chroma_Resampler
Chroma Resampler
sys_clk
Video_In_CSC
Colour-Space Converter
sys_clk
Video_In_RBG_Resampler
RGB Resampler
sys_clk
Video_In_Clipper
Clipper
sys_clk
Video_In_Scaler
Scaler
sys_clk
Video_In_DMA_Controller
DMA Controller
sys_clk
0x1000_3060
0x1000_306f
CPU_fpoint
Floating Point Hardware
Opcode 252
Opcode 255
clk_27
Clock Source
my_accel_0
my_accel
clock
Clock Input
Double-click to export
sys_clk
clock_reset
Reset Input
Double-click to export
[clock]
avalonslave
Avalon Memory Mapped Slave
Double-click to export
[clock]
0x1000_3070
0x1000_307f
avalonmaster
Avalon Memory Mapped Master
Double-click to export
[clock]

DVDES

How to use in software

system.h

```
623 #define ALT_SYS_CLK INTERVAL_TIMER
624 #define ALT_TIMESTAMP_CLK none
625
626
627 /*
628  * my_accel_0 configuration
629  *
630  */
631
632 #define ALT_MODULE_CLASS_my_accel_0 my_accel
633 #define MY_ACCEL_0_BASE 0x10003070
634 #define MY_ACCEL_0_IRQ -1
635 #define MY_ACCEL_0_IRQ_INTERRUPT_CONTROLLER_ID -1
636 #define MY_ACCEL_0_NAME "/dev/my_accel_0"
637 #define MY_ACCEL_0_SPAN 16
638 #define MY_ACCEL_0_TYPE "my_accel"
639
640
641 /*
642  * sysid configuration
643  *
644  */
645
646 #define ALT_MODULE_CLASS_sysid altera_avalon_sysid_qsys
```

DVDES

How to use in software

- Volatile variables
- Use hardware with functions like these:

```
void amplitude_operation(int size, int num, int rbuff_addr, int lbuff_addr, int dest_addr)
{
    amplitude_circute_stop();
    amplitude_circute_set_size(size);
    // also for your debugging make int amplitude_circute_get_size(); (optional)
    amplitude_circute_set_num(num);
    // also for your debugging make int amplitude_circute_get_num(); (optional)
    amplitude_circute_set_rbuff_addr(rbuff_addr);
    // also for your debugging make int amplitude_circute_get_lbuff_addr(); (optional)
    amplitude_circute_set_lbuff_addr(lbuff_addr);
    // also for your debugging make int amplitude_circute_get_rbuff_addr(); (optional)
    amplitude_circute_set_dest_addr(dest_addr);
    // also for your debugging make int amplitude_circute_get_dest_addr(); (optional)
    amplitude_circute_start();
    while(amplitude_circute_get_status() == busy);
    return;
}
```

DVDES

LAB4

Compare measured time by software and hardware

Thanks for your attention

DVDES