Логический анализатор SignalTap

Логический анализатор

Логический анализатор — электронный прибор, который может записывать и отображать последовательности цифровых сигналов



SignalTap

SignalTap — встраиваемый в Cyclone IV FPGA логический анализатор, который позволяет в реальном времени отслеживать сигналы внутри FPGA

- Имеет доступ к любому сигналу в схеме
- Синтезируется вместе с отлаживаемой схемой
- Использует встроенную в FPGA память в качестве буфера
- Передает данные на ПК через JTAG
- Находится в Tools > Signal Tap Logic Analyzer

SignalTap

• JTAG Chain Configuration

Signal Configuration

- Data/Setup
- Instance
 Manager

nstance Manager: 📉 🎼		valid JTAG configurati	on			×	JTAG Chain Configuration: No device is on
stance 🛃 auto_signaltap_0	Status Not running	Enabled LEs: 0	Memory: 0 0 bits	Small: 0/0 NA	Medium: 0/30 Large: 0/0 NA NA NA		Hardware: Please Select Device: None Detected SOF Manager: ()
ado_signatup_0 Noce Type[Allas]	tane	Lockmode: A	low all Changes Enable Tigger Cone 0 1 ∀ Basic A	¢		Signal Configura Clock Data Sample depth dispersive Nodes Allocat Pipeline Facto Storage qual Type: Input port: Nodes Allocat Uput port: Nodes Allocat Type: Input port: Nodes Allocat	tion: 128 • RAM type: Auto t: [2 64 sample segments et: * Auto Manual: 0 r [3 fifter: El Continuous ated: * Auto Manual: 0 ata discontinuities torage qualifier
Mararshu Displays		Data Logi					
		auto_signaltap_0					

Настройка и использование SignalTap

- 1. Выбрать JTAG-программатор
- 2. Добавить желаемые сигналы
- 3. Выбрать тактовый сигнал
- 4. Определить количество семплов
- 5. Запустить компиляцию
- 6. Загрузить прошивку
- 7. Запустить анализ сигналов
- 8. Исследовать временные диаграммы сигналов

Попробуем на примере проекта examples/2019/07_uart

1) JTAG Chain Configuration

- JTAG-программатор выбирается точно так же, как и перед началом обычной прошивки
- Когда JTAG будет настроен, появится надпись "JTAG ready"

JTAG Chair	Configuration: No d	evice is si	elected
Hardware:	Please Select	\$	Setup
Device:	None Detected	\$	Scan Chain
>> SOF	Manager: 🛓 🕕		

JTAG Chair	Configuration: JTAG	JTAG ready							
Hardware:	USB-Blaster [1-3]	\$	Setup						
Device:	@1: 10CL006(Y Z)/10	CL0 \$	Scan Chain						
>> SOF	Manager: 👗 🕕								

2) Setup

- При двойном нажатии на поле
 Setup открывается окно выбора сигналов
- Можно ввести имя желаемого сигнала или найти его в списке
- После завершения выбора нажать Insert
- Сейчас выбраны сигналы start из модуля uart_tx и выходной пин TXD

Named:	*			✓ List 2
Options				
Filter	Design Entry (all name	25)		≜ Customize
ritten.	Design endy (an name	.5/		Customizen
Look in:	top		🖌 🛄 🗹 Include subentitie	es 🗹 Hierarchy view
Matching	Nodes:		Nodes Found:	
	Name	Assignments	Name	Assignments
Cop Cop Cop Cop Cop Cop Cop Cop	ddo dd1 .K qualo CD urt_data urt_start it m_addr m:rom urt_tx:uart_tx Addo Equalo .Selector0	Unassigned Unassigned PIN_24 Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned	uart_tx:uart_tx start	Unassigned

3) Signal Configuration

- В поле Clock нужно выбрать тактовый сигнал, по которому значения исследуемых сигналов будут сохраняться в буфер
- Сейчас выбран сигнал CLK

Clock: CLK Data Sample depth: 64 K	×
Data Sample depth: 64 K RAM type: Auto Segmented: 2 32 K sample segments Nodes Allocated: Auto Pipeline Factor: O Storage qualifier: Type: Input port:	
Sample depth: 64 K Segmented: 2 32 K sample segments Nodes Allocated: Auto Manual: 2 Pipeline Factor: 0 Storage qualifier: Type: Continuous Input port:	
Segmented: 2 32 K sample segments Nodes Allocated: Auto Manual: Pipeline Factor: Storage qualifier: Type: Continuous Input port: 	•
Nodes Allocated: Auto Manual: Pipeline Factor: O Storage qualifier: Type: Continuous Input port: 	¢
Pipeline Factor: 0 Storage qualifier: Type: Continuous Input port:	*
Storage qualifier: Type: Continuous	•
Type: Continuous :	
Input port:]
Nodes Allocated: Auto Manual: 2	3
☑ Record data discontinuities	
Disable storage qualifier	

4) Signal Configuration

- В поле Sample depth нужно выбрать размер буфера — сколько раз будут сохраняться исследуемые сигналы
- В данном случае будет 65536 семплов

lock: CLK		
Data		
Sample de	oth: 64 K 🗢 RAM type: Auto	\$
Segmer	ted: 2 32 K sample segments	\$
Nodes Allo	cated: Auto O Manual: 2	*
Pipeline Fa	ctor: 0	\$
Storage q	ualifier:	
Type:	Continuous	\$
Input por	t	
Nodes Al	located: Auto O Manual: 2	*
Recor	d data discontinuities	

5) Start Compilation

• Сверху есть кнопка Start Compilation



• Теперь в отлаживаемую схему добавится модуль SignalTap

Entity:Instance	Entity:Instance
Cyclone IV E: EP4CE6E22C8	Cyclone IV E: EP4CE6E22C8
🗄 🕸 top 📥	E abo top
🕀 🤤 rom:rom	sld_hub:auto_hub
abo uart_tx:uart_tx	🕀 👯 sld_signaltap:auto_signaltap_0
	E abc rom:rom
	uart_tx:uart_tx

6) JTAG Chain Configuration

• Как и во время обычной прошивки нужно выбрать файл и загрузить его в FPGA

JTAG Chair	Configuration:	JTAG rea	dy	×
Hardware:	USB-Blaster [1	-3]	Setup	
Device:	@1: 10CL006()	Y Z)/10CL0	\$	Scan Chain
>> SOF I	Manager:	1 t_file	es/ua	rt_tx.sof

 После этого Instance Manager сообщит о готовности выгружать сигналы на ПК (надпись "Ready to acquire") и отобразит экземпляры SignalTap (в данном случае он один — auto_signaltap_0)

Instance Manager: 🍡		×						
Instance	Status	Enabled	LEs: 633	Memory: 1310	Small: 0/0	Medium: 17/30	Large: 0/0	
🖳 🤁 auto_signaltap_0	Not running	V	633 cells	131072 bits	0 blocks	16 blocks	0 blocks	

7) Instance Manager

• При нажатии Run Analysis начнется сохранение значений сигналов в память FPGA, после чего они будут отправлены на ПК по JTAG

Instance Manage	🕽 🔳 🎦 R	eady to acqu	uire				×
Instance	Status	Enabled	LEs: 633	Memory: 13107 Small: 0/0	Medium: 17/30	Large: 0/0	
auto_signaltap_0	Not running	•	633 cells	131072 bits 0 blocks	16 blocks	0 blocks	
						emperatural .	
Instance Manager:	ଚ 🔳 🖿 🗛	cquisition in	progress				×
Instance	Status	Enabled	LEs: 633	Memory: 13107 Small: 0/0	Medium: 17/30	Large: O/O	
auto_signaltap_0	Acquiring pre.	🗹	633 cells	131072 bits 0 blocks	16 blocks	0 blocks	



• Теперь в окне Data видны временные диаграммы сигналов TXD и start, сохраненные по сигналу CLK

log: Trig (2019/12/23 15:57:24 (0:0:0.0 elapsed)																	
Type Al	as Name	-8192	-4096	0	4096	8192	12288	16384	20480	24576	28672	32768	36864	40960	45056	49152	53248	57344
-	TXD										18 54 -							
-	uart_tx:uart_tx start																	
🥦 Data	a Setup																	

- Длина диаграммы 65536 семплов
- Виден сигнал start, запускающий передатчик UART
- Видна передача стартового бита, битов данных (01000110 буква F) и стопового бита



Особенности SignalTap

- Исследование любых сигналов загруженной в FPGA схемы на компьютере без необходимости выводить их на пины
- Сохранение сигналов по определенным пользователем условиям
- Изменение настроек SignalTap требует частичной или полной перекомпиляции проекта и перепрошивки FPGA
- Ограничение на количество семплов, связанное с количеством доступной памяти в FPGA

GitHub

github.com/viktor-prutyanov/drec-fpga-intro