

Отладочная плата LDM-HELPER-XC7Z015-1CLG485



### Отладочная плата

# HELPER

### Master-модуль

### LDM-HELPER-XC7Z015-1CLG485



СДЕЛАНО В РОССИИ

1

### СОДЕРЖАНИЕ

ВЕДЕНИЕ	3
Основные технические характеристики модуля	4
2 Маркировка и опции	4
3 Описание и работа	5
3.1 Принципиальная электрическая схема	5
3.2 Питание и настройка перемычек	7
3.3 Общая последовательность разработки конфигурации процес	сора
Zynq на примере простейшего проекта (UART)	8
3.3.1 Конфигурирование аппаратной части платформы Zynq	8
3.3.1.1 Создание блок схемы "дизайна" аппаратной части и	
настройка параметров его блоков	8
3.3.1.2 Генерация HDL "обёртки" для созданного "дизайна"	21
3.3.1.3 Добавление файлов физических ограничений в форм	iame
XDC	22
3.3.1.4 Синтез и имплементация созданного HDL описания	
"дизайна", создание файла прошивки	24
3.3.2 Написание программ для сконфигурированной аппаратной ч	асти
платформы Zynq	25
3.3.2.1 Импортирование спецификации аппаратной платф	ормы
и конфигурирование пакета поддержки платы	25
3.3.2.2 Написание прикладного программного обеспечения .	28
3.3.2.3 Формирование программы начального загрузчика FS	BL u
файлов прошивки для ПЛИС и флешки	34
3.4 Комплектация	38
3.5 Монтажные чертежи	39
3.6 Трассировка по слоям	40
Эксплуатация, хранение и транспортирование	45

Отладочная плата *LDM-HELPER-XC7Z015-1CLG485* представляет собой master-модуль к мультиплатформенной системе проектирования семейства *HELPER*. Она создана на базе ПЛИС SoC *XC7Z015* со встроенным аппаратным ядром Dual-core ARM Cortex-A9 фирмы Xilinx в корпусе *CLG485*.

Общий вид отладочной платы приведен на рисунке 1.



Рисунок 1. Общий вид отладочной платы LDM-HELPER-XC7Z015-1CLG485

### 1 Основные технические характеристики модуля

Параметр	Значение
Тип:	Отладочная плата
Архитектура ПЛИС:	SoC
Семейство ПЛИС:	XILINX Zynq-7000
Маркировка ПЛИС:	XC7Z015
Габаритные размеры (ДхШхВ):	130х110х25 мм
Макетное поле (шаг 2.54 мм):	Нет
Корпус ПЛИС:	CLG485
Количество линий I/O PS IO, GTP, SelectIO	
HR:	128, 4, 150
Тактовый генератор:	33,33 МГц
Режим программирования:	JTAG
Напряжение питания платы:	+5 B
Кол-во элементов LEs:	74K Artix-7 FPGA
Процессор:	Dual-core ARM Cortex-A9
Память:	8Gb x32 DDR3L SDRAM
	2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b
Периферия:	GPIO
Ethernet:	10\100\1000M (RTL8211E)
LED индикаторы выходов:	8
Кнопки входных линий:	5
Bec:	105 г

Вместе с master-модулем можно использовать slave-модули:

- Аналоговый модуль АЦП, ЦАП, Цифровые потенциометры;
- Радиочастотный модуль WiFi, ZigBee, Bluetooth;
- Навигационный модуль GPS, ГЛОНАСС;
- Мультимедиа модуль Аудиокодек, драйвер HDMI;
- Функциональный модуль Цифровой термометр, датчик давления, FRAM, MRAM, EEPROM, FLASH, IO экспандер, RTC, датчик тока, цифровой компас, гироскоп;
- Силовой модуль мощные низковольтные, маломощные высоковольтные с опторазвязкой ключи;
- Плата 5-ти осевого ЧПУ модуля (SPI-интерфейс);
- Плата сбора данных с 5-ти энкодеров (SPI-интерфейс).

Список модулей постоянно пополняется.

Возможно изготовление специализированных модулей по ТЗ заказчика.

### 2 Маркировка и опции

Отладочная плата поставляется в двух модификациях.

Маркировка: LDM-HELPER-XC7Z015-1CLG485-FULL (без microUSB разъема X13)

Маркировка: LDM-HELPER-XC7Z015-1CLG485-FULL-F (с microUSB разъемом X13)

### 3 Описание и работа

### 3.1 Принципиальная электрическая схема



www.ldm-systems.ru +7(495)500-89-20 info@ldm-systems.ru



www.ldm-systems.ru +7(495)500-89-20 info@ldm-systems.ru

### 3.2 Питание и настройка перемычек

Питание отладочной плата LDM-HELPER-XC7Z015-1CLG485 осуществляется от постоянного стабилизированного источника с напряжением +5 B, 0,5÷3 A (выводы Vin 22,23 или вывод +5Vusb 64 X2, разъем X15 или от USB порта X12, подключенного кабелем к порту USB персонального компьютера).

**Внимание!** Если для питания платы используется USB-порт ПК, убедитесь, что минимальный ток не будет ниже 1А. Рекомендуется использовать порт USB 3.0.

В таблице 1 приведены режимы включения джамперов, переключателей и их функции (ver.1.07).

Таблица 1

Джампер	Положение	Функционал
<b>C</b> A1	Нажата	Сброс по линии RESET
SAT	Не нажата	Рабочий режим
SA2	SELECT	Кнопка джойстика "Выбор"
SA3	UP	Кнопка джойстика "Вверх"
SA4	DOWN	Кнопка джойстика "Вниз"
SA5	LEFT	Кнопка джойстика "Влево"
SA6	RIGHT	Кнопка джойстика "Вправо"
S \ 7*	Нажата	Отключение питания от ПЛИС
SAT	Не нажата	Питание ПЛИС подключено
V1	1-2	VCCBAT = +1,8B
	2-3	VCCBAT = +VBAT
X2	-	Межплатные разъемы модуля
X3	-	Разъем интерфейса JTAG
X4	-	Разъем линий xADC
X5	-	MGTP_Tx/Rx(0)
X6	-	MGTP_Tx/Rx(2)
X7	-	MGTP_Tx/Rx(1)
X8	-	MGTP_Tx/Rx(3)
X9	-	MGTP_CLK(01)
X10	1-2	MIO[5] = 0 (JTAG) Загрузка через JTAG (X3)
X10	2-3	MIO[5] = 1 (QSPI Boot) Загрузка из QSPI памяти
X11	-	Разъем интерфейса RS-485
X12	-	Разъем интерфейса USB-UART (UART1)
X13		Опциональный разъем microUSB интерфейса USB
X14	-	Разъем ETHERNET 10/100/1000
X15	-	Разъем питания +5В
X16	-	Разъем пользовательских I/O с уровнем 1,8B/3,3B/User
¥17	1-2	+VCC_B35 = +1,8B
	2-3	+VCC_B35 напряжение с вывода X18.2
¥18	1-2	Выбор напряжения для Х17 = +3,3В
X10	2-3	Выбор напряжения для Х17 = напряжению с КТ4
¥10	1-2	+VCC501_ADJ = +3,3B
	2-3	+VCC501_ADJ = +2,5B
X20	-	Разъем пользовательских I/O с уровнем 3,3В
X21	-	Разъем HDMI канала

Режимы включения джамперов и их функции

\* - Не допускается периодическое нажатие с интервалом менее 1 сек.

### 3.3 Общая последовательность разработки конфигурации процессора Zynq на примере простейшего проекта (UART)

Конфигурирование платформы Zynq можно разделить на две основные части:

- 1) Конфигурирование аппаратной части платформы Zynq;
- Написание программ для сконфигурированной аппаратной части платформы Zynq.

Рассмотрим более подробно каждый из этапов.

### 3.3.1 Конфигурирование аппаратной части платформы Zynq

Данный этап выполняется средствами разработки ISE или Vivado IDE. Для выбранного нами кристалла XC7Z015-1CLG485 доступно конфигурирование только средствами Vivado IDE. Тестовый проект создавался в Vivado 2018.3.

Конфигурирование аппаратной части можно подразделить на следующие под этапы:

- Создание блок схемы "дизайна" аппаратной части и настройка параметров его блоков;
- 2) Генерация HDL "обёртки" для созданного "дизайна";
- Добавление файлов физических ограничений в формате XDC. Ucf больше не поддерживается;
- 4) Синтез и имплементация созданного HDL описания "дизайна", создание файла прошивки для ПЛИС;

# 3.3.1.1 Создание блок схемы "дизайна" аппаратной части и настройка параметров его блоков

Назначение разрабатываемой системы приём тестовых данных по параллельной шине и отправка их по интерфейсу UART, входящему в состав периферии процессорного ядра Zynq. При разработке использовались платы LDM-HELPER-XC7Z015-1CLG485 и LDM-HELPER-uMB301 электрическая схема соединения, которых приведена на рисунке 2:



Рисунок 2. Схема соединения плат LDM-HELPER-XC7Z015-1CLG485 и LDM-HELPER-uMB301

После сборки электрической схемы можно приступать к созданию блок схемы "дизайна" аппаратной части и настройке параметров его блоков.

Vivado 2017.2	
Ele Flow Ioels Window Help Q- Cluck Access	
HLx Editions	
Quick Start	
oper Lampie Project 2	
Tasks	
Manäge IP > Open Hardware Manager > Xillinx Tcl Store >	
Learning Center	
Documentation and Tutorials > Quick Take Videos > Refease Notes Guide >	
Tcl Console	▼ B <sup>2</sup> C, C: 3amacs [00]
🕅 Hyck 📓 🗇 🍃 / 🔯 > 💿 🦮 📣 🕸 🚥 🗐 🛋 🔪 😳 🥔 🥔 🖉 Kirado 2017.2	RU 📒 🗗 🔵 🤤

1) Запустим Vivado и создадим новый проект:

À New Project		×
HLX Editions	Create a New Vivado Project         This witard will guide you through the creation of a new project.         To create a Vivado project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and choose a default part.	
<b>?</b>	< <u>Н</u> азад <u>Next</u> > Finish Cancel	

Project Name       Enter a name for your project and specify a directory where the project data files will be stored.         Project name:       project_1         Project jocation:       C/but/Xil_Prj         Create project subdirectory         Project will be created at:       C/but/Xil_Prj	À New Project		×
Enter a name for your project and specify a directory where the project data files will be stored.  Project name: project_1 Project jocation: C/buf/Xil_Prj Create project subdirectory Project will be created at C/buf/Xil_Prj	Project Name		
Project name:       project_1         Project Jocation:       C/buffXil_Prj         Create project subdirectory         Project will be created at: C/buffXil_Prj	Enter a name for yo	ur project and specify a directory where the project data files will be stored.	A
Project name:       project_1         Project jocation:       C/but/Xil_Prj         Create project subdirectory         Project will be created at C/but/Xil_Prj			
Project Jocation: C:/buf/Xil_Prj  Create project subdirectory Project will be created at: C:/buf/Xil_Prj	<u>P</u> roject name:	project_1	8
Create project subdirectory Project will be created at: C:/buf/XII_Prj	Project location:	C:/buf/Xil_Prj	0
Project will be created at: C:/buf/Xil_Prj	Create projec	t subdirectory	
	Project will be cr	eated at. C:/buf/Xil_Prj	
	$\sim$		
(?) < <u>Hasaa</u> <u>Next</u> Finish Cancel	(?)	< <u>Н</u> азад <u>Next</u> > Finish	Cancel

🌔 New Project		×
Project Name		
🍋 Choose Project Location		×
Recent: C:/buf/Xil_Prj	✓ Ø S S A S S Y	5
Directory: C:\buftXil_Prj		
<ul> <li>FTP</li> <li>Keil_v5</li> <li>PertLogs</li> <li>Program Files</li> <li>Program Files (x86)</li> <li>Python26</li> <li>Sistem</li> <li>Users</li> <li>Virtual_Michines</li> <li>Windows</li> <li>Xilinx</li> <li>DDS_Module</li> <li>MSVS_Prj</li> <li>Proteus_Prj</li> </ul>	Create New Folder	Î
<ul> <li>▲ Xil_Prj</li> <li>▲ inetpub</li> <li>▲ temp</li> <li>▲ дисковод BD-RE (D:)</li> <li>▲ CD-дисковод (E:)</li> </ul>	Select	v
(?)	< <u>Н</u> азад <u>N</u> ext> Finish Cancel	

À Nen	y Project
<b>Proje</b> Specif	y the type of project to create.
۲	RTL Project You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis. ✓ Do not specify sources at this time
0	Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.
0	I/O Planning Project Do not specify design sources. You will be able to view part/package resources.
0	Imported Project Create a Vivado project from a Synplify, XST or ISE Project File.
0	Example Project Create a new Vivado project from a predefined template.
?	< <u>Н</u> азад Finish Cancel

Remaining	~
DSPs Gb	GTPE2 nsceivers Transceiv
120 4	4
160 4	4
	s DSPs Gb Tra 120 4 160 4



2) После создания проекта окно приобретёт вид, указанный на рисунке 3:

À UART - [C:/buf/Xil_Prj/UART/UAR	T.xpr] - Vivado 2017.2												5
Eile Edit Flow Tools Window	Layout View Help Q+ Quick Acce	55										Ready	
	ο Σ % θ χ											III Default Layout	~
Flow Navigator 🗧 🔍 🗕	PROJECT MANAGER - UART												? >
Y PROJECT MANAGER	Sources	? _ 🗆 🖾 X	Project Summ	ary								? 🗆 🖸	зx
Settings		0											-
Add Sources	Design Sources		Settings Edit										
Language Templates	> 🗎 Constraints		Project nam	e: UA	NRT								
	~ 🖾 Simulation Sources		Project locat	Project location: C:/bufXil_Prj/UART									
	🖙 sim_1		Product fam Broject part	ly: Zy	nq-7000								
Create Block Design			Top module	name: No	t defined								
Ones Block Design			Target langu	age: VH	IDL								
Constrate Black Design			Simulator la	nguage: Mi	xed								
Generale block Design													
✓ SIMULATION	Historethu Librarias Compile Order		Synthesis						Implem	entation			
Run Simulation	incluring country compare order		Status:	Not started					Status		Not started		
	Properties	? _ 🗆 🖾 X	Messages:	No errors or	r warnings				Messa	ges:	No errors or warnings		
<ul> <li>RTL ANALYSIS</li> </ul>	N	+ + O	Part:	xc7z015clg4	485-1				Part		xc7z015clg485-1		
<ul> <li>Open Elaborated Design</li> </ul>			Strategy:	Vivado Synti	hesis Defaults				Strateg	y.	Vivado Implementation Defaults		
Y SYNTHERIS									Increm	ental compile:	None		
Run Synthesis													
Open Synthesized Design	Select an object to see or	operties	DRC Violatio	ns					Timing				41
<ul> <li>Obert Ohrmesting Design</li> </ul>	Celeti al objeccio see pi	openero		Run Imp	lementation to se	e DRC res	ults				Run Implementation to see timing re-	ults	
IMPLEMENTATION													
Run Implementation			Utilization						Power				
> Open Implemented Design													~
	Tcl Console Messages Log Rep	Design Runs ×										? _ 0	3 6
<ul> <li>PROGRAM AND DEBUG</li> </ul>	Q 素 ⊕ I4 ≪ ▶ ≫	+ %											
(III) Generate Bitstream	Name Constraints Status	WNS TNS WHS	THS TPWS	Total Power	Failed Routes	LUT F	F BR	Ms URAM	DSP St	rt Elapsed	Strategy		P
> Open Hardware Manager		ed									Vivado Synthesis Defaults (Vivado S	nthesis 2017)	ж
	▷ impl_1 constrs_1 Not start	ed									Vivado Implementation Defaults (Viv	ado Implementation 2017	J) X(
	< c						_						>

Рисунок 3. Основное окно Vivado

3) Для создания проекта блок схемы "дизайна" аппаратной части и настройки его параметров и блоков необходимо в панели "Flow Navigator" в подразделе "IP Integrator" выбрать пункт "Create Block Design". В результате окно приобретёт вид, указанный на рисунке 4:

ease specify name	of block design.	1
<u>D</u> esign name:	Zynq_Uart_Design	0
D <u>i</u> rectory:	🛜 <local project="" to=""></local>	~

A UART - [C:/buf/Xil_Prj/UART/UA	ART.xpr] - Vivado 2017.2		_ <del>.</del> .
Eile Edit Flow Icols Window	Layout Yew Help Qr Guick Access		Ready
●, Ⅲ ◆ → Ⅲ Ⅲ ×	<b>□ ▶, ₩ Φ Σ # # #</b>		🖽 Default Layout 🛛 🗸
Flow Navigator 🚦 0 ? .	BLOCK DESIGN - Zynq_Uart_Design		? ×
✓ PROJECT MANAGER	Sources Design X Signals 2 D Pl	Diagram	2008
O Settings			
Add Sources			•
Language Templates	<ul> <li>zincoacosign</li> </ul>		
IP Catalog			
Y IP INTEGRATOR			
Create Block Design			
Open Block Design			
Generate Block Design			
✓ SINULATION			
Run Simulation		This design is emply. Press the 🕂 button to add IP.	
<ul> <li>RTL ANALYSIS</li> </ul>	Dromenties 2 D P V		
> Open Elaborated Design			
✓ SYNTHESIS			
Run Synthesis			
> Open Synthesized Design		De la	
	Select an object to see properties		
<ul> <li>IMPLEMENTATION</li> </ul>			
<ul> <li>Run impremensation</li> <li>Constitution</li> </ul>			
<ul> <li>Open impremented Design</li> </ul>	Tel Canada y Hannan Las Desute Desire Des		
Y PROGRAM AND DEBUG	Console A messages Log Reports Design runs		7 - 0 6
👪 Generate Bitstream			0
> Open Hardware Manager	<pre>into: [12,120:11-111] b0 user in the TP Second b0 provide the transmission of the TP Second b0 create property target_language VOL [nurses_project] create_b0(assing "Type_DET_beign" from the transmission of the transmission of the transmission for the transmission of the transmission of the transmission for the transmission of the transmission of the transmission of the transmission of the transmission of the transmission of the transmission of the transmission of transmission of the transmission of transmission of</pre>	112:70111ss/91ysdo/2017.2/04/07/ 00100122 . Memory (MB): peak = 798.324 ; gain = 28.496 2/2ymg_Dart_Dealgu/Zymg_Dart_Dealgu.bdD = 00100116 . Memory (MB): peak = 685.331 ; gain = 18.266	Ĵ
	f Type a Tol command here		

Рисунок 4. Окно Create Block Design

Последовательно нажимая на кнопку "+", позволяющую добавлять IP ядра добавить в проект ядра с следующими названиями:

- 1) ZYNQ7 Processing System;
- 2) Processor System Reset;
- 3) AXI Interconnect;
- 4) AXI GPIO.

В итоге окно Create Block Design приобретёт следующий вид:



Затем необходимо произвести настройки ядер AXI Interconnect и AXI GPIO. В первом из них нужно отключить один из master интерфейсов, а во втором установить разрядность шины не 32, а 4 бита и включить все порты на вход. Выполнение этих действий поясняют рисунки 5 и 6.

XI Interconnect (2.1)		1
Documentation 📄 IP Location		
Component Name axi_interconnect_0		
Top Level Settings Slave Interfaces		
Number of Slave Interfaces	1 ~	Î
Number of Master Interfaces	1 🗸	
Interconnect Optimization Strategy	Custom 🗸	
AXI Interconnect includes IP Integrator a When the endpoint IPs attached t in width, clock or protocol, a conv If a converter IP is inserted, IP inte configures the converter to match To see which conversion IPs haw 'expand hierarchy' buttons to expl	utomatic converter insertion and configuration. to the interfaces of the AXI Interconnect differ enter IP will automatically be added inside the ini grator's parameter propagation automatically the design. a been inserted, use the IP integrator ore inside the AXI Interconnect hierarhcy.	erconnect.
NOTE:Addressing information for AXI Inf	erconnect is specified in the IP Integrator addres	ss editor.
Enable Advanced Configuration Op	ions	

Рисунок 5. Настройка ядра AXI Interconnect

umentation 🛛 📄 IP Location					
now disabled ports	Component Name axi_gpio	_0			
	GPIO				
	All Inputs				
	All Outputs				
	GPIO Width	4	0	[1 - 32]	
	Default Output Value	0x0000000	0	[0x00000000,0xFFFFFFF]	
+ S_AXI	Default Tri State Value	0xFFFFFFF	0	[0x00000000,0xFFFFFFF]	
s_axi_aclk GPIO +	Enable Dual Channel				
	GPIO 2				
	All Inputs				
	All Outputs				
	GPIO Width	32		[1 - 32]	
	Default Output Value	0x00000000	0	[0x00000000,0xFFFFFFF]	
	Default Tri State Value	0xFFFFFFF	0	[0x00000000,0xFFFFFFF]	
	Enable Interrupt				

Рисунок 6. Настройка ядра AXI Gpio

После установки данных параметров необходимо соединить между собой IP разрабатываемого "дизайна" в соответствие с рисунком 7:



Рисунок 7. Схема разрабатываемого "дизайна"

Порт btns\_5bit, конфигурируется путем применения команды **Make External** из выпадающего меню, вызываемого правым щелчком мыши по порту IP AXI Gpio.

В последнюю очередь конфигурируются параметры IP ZYNQ7 Processing System. В проекте необходимо настроить DDR память, а из периферийных интерфейсов необходимы QSPI, для подключения FLASH и непосредственно UART1. AXI интерфейс включен по умолчанию в режиме master. Процесс конфигурирования ядра поясняют рисунки 8-13.

Documentation 🌣 Pr	esets	IP Location 🏽 🍈 Import XPS Settings		
Page Navigator	PS-P	L Configuration		Summary Repo
Zynq Block Design	+	Search: Q-		
PS-PL Configuration	Q,	Name	Select	Description
Desinherel I/O Dine	-	✓ General		
Penpheral I/O Pins	-	UART0 Baud Rate	115200	Baud rate is generated with internally fixed UART
MIO Configuration	\$	UART1 Baud Rate	115200	Baud rate is generated with internally fixed UART
		PL AXI idle Port		Enables idle AXI signal to the PS used to indicate
Clock Configuration		DDR ARB bypass Port		Enables DDR urgent/arb signal used to signal a
DDR Configuration		PS-PL Debug interface		Enables PL debug signals to PS and vice-versa
		FTM Trace data interface		Enables FTM Trace AXI stream interface used to
SMC Timing Calculatic		FTM Trace buffer	0	Generates a FIFO to hold trace data
nterrupts		FTM Data edge detector	0	Stores trace data in the FIFO when the data chan
		FTM Trace buffer FIFO size	128	FTM Trace buffer FIFO size
		FTM Trace buffer clock delay	12	Number of clock cycles interval for a trace data out
		Include ACP transaction checker		Enables ACP transaction checker.
		Trace data/control signal pipeline width	8	Enables configurable number of pipeline stages
		Power-on reset(POR) 4k timer		Enables power-on reset(POR) 4k timer. By defau
		Processor event interface		Enables event bus which provides a low-latency a
		> Address Editor		
		> Enable Clock Triggers		



Page Navigator -	DDR Configuration		Summary Re
Zynq Block Design	Enable DDR		Conners ; co
PS-PL Configuration	← Q ≚ ≑		
Peripheral I/O Pins	Search: Q-		
MIQ Configuration	Name	Select	Description
Mic configuration	✓ DDR Controller Configuration		
Clock Configuration	Memory Type	DDR 3 (Low Voltage)	Y Type of memory interface. Refer to UG585 Zynq Technical Reference
DDR Configuration	Memory Part	MT41K256M16 RE-125	Memory component part number. For unlisted parts choose "Custom
	Effective DRAM Bus Width	32 Bit 💊	Data width of DDR interface, not including ECC data width. Refer to U
SMC Timing Calculation	ECC	Disabled	Enables error correction code support. ECC is supported only for an
Interrupts	Burst Length	8	Minimum number of data beats the controller should use when com
	DDR	533.333	Memory clock frequency. The allowed freq range is (200.000000 : 53
	Internal Vref		Enables internal voltage reference source. Disable to use external Vr
	Juntion Temperature (C)	Normal (0-85)	Intended operating temperature range. Controls the DDR refresh inte
	> Memory Part Configuration		
	> Training/Board Details	User Input	·
	Additive Latency (cycles)	0	Additive Latency (cycles). Increases the efficiency of the command an
	> Enable Advanced options		Enable Advanced DDR QoS settings

Рисунок 9. Настройка контроллера DDR (DDR Controller Configuration)

Documentation 🌣 Preset	em (5.5) s 📄 IP Location 🌼 Import XPS Setting:	s	
Page Navigator —	DDR Configuration		Summary Re
Zynq Block Design	Enable DDR		
PS-PL Configuration	← Q ≚ ≑		
Peripheral I/O Pins	Search: Q-		
MIQ Configuration	Name	Select	Description
ino comgaratori	> DDR Controller Configuration		
Clock Configuration	<ul> <li>Memory Part Configuration</li> </ul>		
DDR Configuration	DRAM IC Bus Width	16 Bits	Width of individual DRAM components.
DDICComgulation	DRAM Device Capacity	4096 MBits	Storage capacity of individual DRAM components.
SMC Timing Calculation	Speed Bin	DDR3_1066F	Speed bin of the individual DRAM components.
nterrunte	Bank Address Count (Bits)	3	Number of bank address pins.
interrupto	Row Address Count (Bits)	15	Number of row address pins.
	Col Address Count (Bits)	10	Number of column address bits.
	CAS Latency (cycles)	7	Column Access Strobe (CAS) latency in memory clock cycles. It refer
	CAS Write Latency (cycles)	6	CAS write latency setting in memory clock cycles.
	RAS to CAS Delay (cycles)	7	tRCD. Row address to column address delay time. It is the time req
	Precharge Time (cycles)	7	tRP. Precharge Time is the number of clock cycles needed to termin
	tRC (ns)	48.75	Row cycle time (ns)
	tRASmin (ns)	35.0	Minimum number of memory clock cycles required between an Active
	tFAW (ns)	40.0	Determines the number of activates that can be performed within a c
	> Training/Board Details	User Input 🗸	
	Additive Latency (cycles)	0 🛞	Additive Latency (cycles). Increases the efficiency of the command an
	> Enable Advanced options		Enable Advanced DDR QoS settings

Рисунок 10. Настройка контроллера DDR (Memory Part Configuration)

Cancel

OK

#### ZYNQ7 Processing System (5.5)

6	Documentation	Presets	IP Location	Timport XPS Settings
---	---------------	---------	-------------	----------------------

Page Navigator —	DDR Configuration			Summary Repo				
Zynq Block Design	Enable DDR							
PS-PL Configuration	<b>←</b> Q							
Peripheral I/O Pins	Search: Q-							
MIO Configuration	Name	Select		Description				
and the state of the state of the	> DDR Controller Configuration							
Clock Configuration	> Memory Part Configuration							
	✓ Training/Board Details	User Input	~					
DDR Conliguration	V DRAM Training							
SMC Timing Calculation	Write leveling			Enables Write Leveling calibration, which adjusts write DQS relative $t_{\cdot\cdot}$				
iterrupts	Read gate			Enables Read Gate calibration, which adjusts valid RD DQS window.				
	Read data eye			Enables Read Data Eye calibration, which adjusts the read DQS to th				
	<ul> <li>DQS to Clock Delay (ns)</li> </ul>							
	DQS0	0.023	0	DQS to Clock delay [0] (ns). The DQS path delay subtracted from the				
	DQS1	0.023	٢	DQS to Clock delay [1] (ns). The DQS path delay subtracted from the				
	DQS2	0.023	$\otimes$	DQS to Clock delay [2] (ns). The DQS path delay subtracted from the				
	DQS3	0.023	$\otimes$	DQS to Clock delay [3] (ns). The DQS path delay subtracted from the				
	<ul> <li>Board Delay (ns)</li> </ul>							
	DQ[7:0]	0.248	0	Board delay [0] (ns). The midrange of data (DDR_DQ, DDR_DM) trac				
	DQ[15:8]	0.244	۲	Board delay [1] (ns). The midrange of data (DDR_DQ, DDR_DM) trac				
	DQ[23:16]	0.253	0	Board delay [2] (ns). The midrange of data (DDR_DQ, DDR_DM) trac				
	DQ[31:24]	0.270	0	Board delay [3] (ns). The midrange of data (DDR_DQ, DDR_DM) trac				
	Additive Latency (cycles)	0	0	Additive Latency (cycles). Increases the efficiency of the command an				
	> Enable Advanced options			Enable Advanced DDR QoS settings				
	> Enable Advanced options			Enable Advanced DDR QoS settings				

Рисунок 11. Настройка контроллера DDR (Training/Board Details) для платы версии ver.1.07

YNQ7 Processing Syst	em (5.5) s 🕞 IP Location 🔅 Import XPS S	Settings										
Page Navigator —	MIO Configuration											Summary Rep
Zynq Block Design	Bank 0 I/O Voltage LVCMOS	3.3V 🗸	Bank	1 I/O Volta	ge LVCMOS 2.5V	~						
PS-PL Configuration Peripheral I/O Pins	← Q ¥ ♦ •ť	9										
MIO Configuration	Peripheral	ю		Signal	Ю Туре		Speed		Pullup		Direction	Polarity
-	V VART 1	MIO 5253	~									
Clock Configuration	🗌 Modem signals											
DR Configuration	UART 1	MIO 52		tx	LVCMOS 2.5V	~	fast	~	enabled	~	out	



2

OK

Cancel

#### ZYNQ7 Processing System (5.5)

<ol> <li>Documentation</li> </ol>	Presets	IP Location	Timport XPS Settings
-----------------------------------	---------	-------------	----------------------

nd Block Design	Bank 0 I/O Voltage LVCMOS 3.3V	✓ Bar	nk 1 I/O Voltage	LVCMOS 2.5V V				
S-PL Configuration	← Q ≍ ≑ 4 0							
eripheral I/O Pins	Search: Q-							
IO Configuration	Peripheral	10	Signal	IO Type	Speed	Pullup	Direction	Polarity
CT 1000000000000000000000000000000000000	<ul> <li>Memory Interfaces</li> </ul>							
Clock Configuration	V 🕑 Quad SPI Flash	MIO 1 6 🗸 🗸						
DR Configuration	√● Single SS 4-bit IO	MIO 16 🗸						
	Data Mode	x4 ~						
MC TIMING Calculation	Quad SPI Flash	MIO 1	qspi0_s	LVCMOS 3.3V 🗸 🗸	fast 🗸	ena 🗸	out	
Interrupts	Quad SPI Flash	MIO 2	qspi0_io[	LVCMOS 3.3V 🗸 🗸	fast 🗸	disabled	inout	
	Quad SPI Flash	MIO 3	qspi0_io[	LVCMOS 3.3V 🗸	fast 🗸	disabled	inout	
	Quad SPI Flash	MIO 4	qspi0_io[	LVCMOS 3.3V 🗸 🗸 🗸	fast 🗸	disabled	inout	
	Quad SPI Flash	MIO 5	qspi0_io[	LVCMOS 3.3V 🗸 🗸 🗸	fast 🗸	disabled	inout	
	Quad SPI Flash	MIO 6	qspi0_sclk	LVCMOS 3.3V 🗸 🗸	fast 🗸	disabled	out	
	> O Dual Quad SPI (4 bit)							
	> O Dual Quad SPI (Paral							
	V 🕑 Feedback Clk	MIO 8 🗸 🗸						
	Quad SPI Flash	MIO 8	qspi_fbclk	LVCMOS 3.3V 🗸 🗸 🗸	fast 🗸	disabled	out	
	> SRAM/NOR Flash							
	> NAND Flash							
	> I/O Peripherals							
	> Application Processor Unit							
	> Programmable Logic Test and	Debug						

Рисунок 13. Включение интерфейса QSPI Flash

После настройки периферии необходимо определить адресное пространство, которое ей выделяется. Для этого необходимо в окне "Create Block Design" из вкладки "Diagram" перейти во вкладку "Address Editor" и применить команду "Auto Assign Address". Выполнение этого действия изображает рисунок 14:

X

19



Рисунок 14. Автоматическое назначение адресов периферии

На этом создание блок схемы "дизайна" аппаратной части и настройка параметров его блоков завершены. Можно приступать к процессу генерации HDL "обёртки".

### 3.3.1.2 Генерация HDL "обёртки" для созданного "дизайна"

Для генерации HDL "обёртки" для созданного "дизайна" необходимо перейти к вкладке **"Sources"** окна **"Block Design"**. В разделе **"Design Sources"** выделить строку с названием только что созданного "дизайна" и щелчком правой кнопки мыши вызвать контекстное меню, в котором выбрать пункт **"Create HDL Wrapper"**. После чего запустится процесс генерации HDL "обёртки" в автоматическом режиме. Процесс создания обёртки поясняется на рисунке 15:



Рисунок 15. К процессу генерации HDL "обёртки"

В результате будет создан файл описания на языке VHDL – рисунок 16:



Рисунок 16. Результат генерации HDL "обёртки"

На этом генерация HDL "обёртки" завершена. Для перехода к процессам синтеза, имплементации и генерации файла прошивки ПЛИС, необходимо добавить файл физических ограничений для части системы, реализуемой на распределённой логике.

### 3.3.1.3 Добавление файлов физических ограничений в формате XDC

Создание файла физических ограничений, осуществляется путём нажатия кнопки "Add Sources", находящейся в панели "Flow Navigator", подменю "Project Manager". Ограничения формировались с использованием принципиальной схемы платы LDM-HELPER-XC7Z015-1CLG485. Более подробно процесс создания файла физических ограничений поясняют рисунки 17-19:

Add Sources		×
HLx Editions	Add Sources         This guides you through the process of adding and creating sources for your project         Add or greate constraints         Add or create design sources         Add or create gimulation sources	
E XILINX ALL PROGRAMMABLE.		
?	< <u>Назад</u> Finish	Cancel

Рисунок 17. К процессу генерации файла физических ограничений

<u>Specify constraint set:</u>	👼 constrs_1 (act	ive) 🗸	T'1	VI	
+,   -   +   +		Create a new c project	constraints file and add it to	o your 🍌	
		<u>F</u> ile type:	XDC	~	
		F <u>i</u> le name: Fil <u>e</u> location:	Pins	© ~	
		?	OF	Cancel	

Рисунок 18. К процессу генерации файла физических ограничений



Рисунок 19. К процессу генерации файла физических ограничений

Полный текст файла физических ограничений приведён ниже:

```
set_property PACKAGE_PIN AA20[get_ports {gpio_tri_i[0]}];set_property PACKAGE_PIN R17[get_ports {gpio_tri_i[1]}];set_property PACKAGE_PIN T17[get_ports {gpio_tri_i[2]}];set_property PACKAGE_PIN U19[get_ports {gpio_tri_i[3]}];
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {gpio_tri_i[0]}];
set_property IOSTANDARD LVCMOS33 [get_ports {gpio_tri_i[1]}];
set_property IOSTANDARD LVCMOS33 [get_ports {gpio_tri_i[2]}];
set_property IOSTANDARD LVCMOS33 [get_ports {gpio_tri_i[3]}];
```

После добавления файла физических ограничений можно переходить к процессам синтеза. имплементации и генерации файла прошивки.

# 3.3.1.4 Синтез и имплементация созданного HDL описания "дизайна", создание файла прошивки

Синтез, имплементация, а также создание файла прошивки осуществляются средствами Vivado IDE в автоматическом режиме. Для запуска каждого этапа необходимо вызвать команды "Run Synthesis", "Run Implementation" и "Generate Bitstream" находящихся в соответствующих разделах панели "Flow Navigator".

После выполнения всех перечисленных действий можно считать завершённой разработку аппаратной части системы и перейти к созданию программной части.

## 3.3.2 Написание программ для сконфигурированной аппаратной части платформы Zynq

Проектирование программной части разрабатываемой системы ведётся в САПР SDK, входящей в комплект с IDE Vivado. Проектирование программной части также можно разбить на под этапы:

- Импортирование спецификации аппаратной платформы, созданной в Vivado на предыдущем этапе в SDK и конфигурирование пакета поддержки платы;
- 2) Написание прикладного программного обеспечения;
- Формирование программы начального загрузчика FSBL и файлов прошивки для ПЛИС и флешки.

### 3.3.2.1 Импортирование спецификации аппаратной платформы и конфигурирование пакета поддержки платы

После завершения процесса генерации файла прошивки ПЛИС для написания прикладных программ, предназначенных для работы на базе разработанной аппаратной платформы, необходимо экспортировать описание аппаратной части в среду разработки SDK. Для этого в главном меню IDE Vivado необходимо открыть вкладку "File" и выбрать команду "Export->Export Hardware". При этом появиться диалоговое окно, изображённое на рисунке 20:

Export Ha	rdware	
Export hardwa development t	re platform for software ools.	1
Include b	itstream	
Export to:	🛜 <local project="" to=""></local>	~
100		



После выполнения процедуры экспорта спецификации аппаратной части необходимо запустить SDK выполнив в Vivado команду "File->Launch SDK". После запуска пакета SDK его основное окно примет вид, указанный на рисунке 21:

UART.sdk - C/C++ - Zynq_Uart_Design_wrapper_hw_platy	form_0/system.hdf - Xilinx SDK Window Help												_ <u>-</u> ×
·····································	8 / • • • • • •										2	ick Access	<b>1</b>
🕒 Project Explorer 🛛 📄 🎭 🔻 🗢 🗖	system.hdf 🛙									- 0	Be Outline 23 🔁 Docu	Make	- 0
Zynq_Uart_Design_wrapper_hw_platform_0	Zynq_Uart_Design_wrappe	er_hw_platforr	m_0 Hard	ware Plat	form Spe	cification				-			
12	Design Information										An outime is not available.		
	Target FPGA Device: 72015 Part: xc72015 Created With: Vivado. Created On: Thu Sep Address Map for processor p	clg485-1 2017.2 > 14 07:45:43 2017 s7_cortexa9_[0-	1]										
	Cell	Base Addr	High A	ddr S	lave I/f	Mem/Res	,						
	ps7 intc dist 0	0xf8f01000	0xf8f0	1fff		REGISTE	R	_					
	ps7 scutimer 0	0xf8f00600	0xf8f0	061f		REGISTE	R						
	ps7 slcr 0	0xf8000000	0xf800	offf		REGISTE	R						
	axi gpio 0	0x41200000	0x4120	offff	S AXI	REGISTE	R						
	ps7 scuwdt 0	0xf8f00620	0xf8f0	06ff	-	REGISTE	R						
	ps7 l2cachec 0	0xf8f02000	0xf8f0	2fff		REGISTE	R						
	ps7 scuc 0	0xf8f00000	0xf8f0	00fc		REGISTE	R						
	ps7 qspi linear 0	0xfc000000	0xfcff	m		FLASH							
	ps7 pmu 0	0xf8893000	0xf889	3fff		REGISTE	R						
	ps7 afi 1	0xf8009000	0xf800	9fff		REGISTE	R						
	ps7 afi 0	0xf8008000	0xf800	Sfff		REGISTE	R						
	ps7 gspi 0	0xe000d000	0xe000	dfff		REGISTE	R						
	ps7 afi 3	0xf800b000	0xf800	bfff		REGISTE	R						
	ps7 afi 2	0xf800a000	0xf800	afff		REGISTE	R						
	ps7 globaltimer 0	0xf8f00200	0xf8f0	02ff		REGISTE	R						
	ps7 dma s	0xf8003000	0xf800	3fff		REGISTE	R						
	ns7 ion hus config 0	0xe0200000	0xe020	offf		REGISTE	R						
	ps7 xadc 0	0xf8007100	0xf800	7120		REGISTE	R						
	ns7 ocmc 0	0xf800c000	0xf800	efff		REGISTE	R			-			
	Overview												
🖞 Target Connections 🛛 🦛 🖉 🗁 🗖	🖹 Problems 🕄 🧔 Tasks 🖾 Co	onsole 🔲 Propert	ies 🕒 SDF	C Terminal		~ •		SDK Log 🛙				<b>1</b> 8	
🗉 🗁 Hardware Server	0 items						_	07:10:55 INFO	: Registering comman	nd hand	lers for SDK TCF service	s	-
⊕	Description A	1	Resource	Path		Location	Ty	07:10:56 INFO 07:10:58 INFO 07:10:58 INFO 07:11:01 INFO 07:11:01 INFO	: Launching XSCT ser : XSCT server has st : Successfully done : Successfully done : Processing command	settin settin line	sct.bat -interactive C:\ successfully. g XSCT server connection g SDK workspace option -hwspec C:/buf/Xi	buf\Xil_Prj\U channel l_Prj/UART/UA	JART\U
	•						Þ	4					*

Рисунок 21. Вид SDK после экспорта описания аппаратной платформы

Так как спецификация аппаратной части уже загружена в SDK, необходимо создать пакет поддержки платы, по сути, являющийся набором драйверов, сопрягающих разработанную платформу с программным обеспечением. Для создания пакета поддержки платы необходимо выполнить следующую команду "File->New->Board Support Package". В результате откроется диалог, представленный на рисунках 22-23:

🗱 New Board Support Package Project	<u>_ 🗆 ×</u>
Xilinx Board Support Package Project	h
Create a Board Support Package.	( <u>)</u>
Project name: standalone_bsp_0	
Use default location	
Location: C:\buf\Xil_Prj\UART\UART.sdk\standalone_bsp_0	Browse
Choose file system: default 💌	
Target Hardware	
Hardware Platform: Zynq_Uart_Design_wrapper_hw_platform_0	✓ New
CPU: ps7_cortexa9_0	•
Compiler: 32-bit 💌	
Board Support Package OS	
freertos901_xilinx standalone features such as caches, interrupts and exceptions as well as the basic fe hosted environment, such as standard input and output, profiling, abort and	atures of a d exit.
? Finish	Cancel

Рисунок 22. Первый диалог при создании пакета поддержки платы

Board Support Package	Settings			×
Board Support Package Settin	ngs			
Control various settings of yo	our Board Support Pac	:kage.		
Overview     Standalone     Laiffs     drivers     Lps7_cortexa9_0	standalone_bsp_0         OS Type:       stana         OS Version:       6.3         Target Hardware         Hardware Specific         Processor:         Supported Libraria         Check the box ne         navigator on the 1	ialone  cation: C:\bufXi ps7_cort es eft	Standalone is a simple, low-level software layer. It prov such as caches, interrupts and exceptions as well as the environment, such as standard input and output, profiling 1_Prj\UART\UART.sdk\Zynq_Uart_Design_wrapper_hv exa9_0 s you want included in your Board Support Package. You	ides access to basic processor features e basic features of a hosted g, abort and exit. v_platform_0\system.hdf can configure the library in the
	Name	Varsian	Description	
	Name	Version	Description	
	Intrinitat	1.2	hulD TCD/ID Stack library hulD ut 4.1	
		1.0	Open Amp Library	
	V wilffs	3.6	Generic Fat File System Library	
	vilflash	4.3	Xiliny Elash library for Intel/AMD CEL complia	
	□ xilisf	5.8	Xilinx In-system and Serial Flash Library	
	L xilmfs	2.3	Xilinx Memory File System	
	L xilom	2.1	Power Management API Library for ZyngMP	
	T xilrsa	1.3	Xilinx RSA Library	
	T xilskey	62	Xilinx Secure Key Library	
	_ maney	0.2	man occar ney horary	
•				OX Cancel

Рисунок 23. Второй диалог при создании пакета поддержки платы

В первом диалоге выбирается имя пакета поддержки платы, процессорное ядро для которого он создаётся и устанавливается используемая операционная система. По умолчанию в текущем проекте не предусмотрено использование ОС. В диалоге номер два (рисунок 23) в пакет добавляются специализированные библиотеки расширения. В данном случае необходима лишь библиотека "**xilffs**" для последующего создания проекта первичного загрузчика. После настройки данных параметров автоматически компилируется пакет поддержки платы и окно SDK приобретает следующий вид:

UART.sdk - C/C++ - standalone_bsp_0/system.mss - Xilinu File Edit Navignte Search Project Run Xilinu Tools	SDK Window Help									_ 8 X		
	@ A ▼ \$ \$ \$ \$ ▼ \$ *							Qu	ick Access	R 100		
🕒 Project Explorer 🛛 🗧 🎭 🔽 🗢 🗖	system.hdf						- 0	Be Outline 22 Docu	Make	- 0		
19 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	standalone_bsp_0 Board Supp	oort Package					-	An outline is not available.				
	Modify this BSP's Settings Re-ge											
	Target Information	Target Information										
	This Board Support Package is comp	piled to run on the followin	ng target.									
	Hardware Specification: C/buf/Xil_ Target Processor: ps7_cortes	Hardware Specification: C:bufXil_PrjUARTUART.sdkZynq_Uart_Design_wrapper_hw_platform_0/system.hdf Target Processor: ps7_cortexa9_0										
	Operating System	Operating System										
	Board Support Package OS.											
	Name: standalone											
	Version: 6.3											
	Description: Standalone is a sim exceptions as well	Description: Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and										
	Documentation: standalone_v6_3	Documentation: standalone_v6_3										
	Peripheral Drivers											
	Drivers present in the Board Support	Package.										
	axi_gpio_0 gpio	Documentation Imp	ort Examples									
	ps7_afi_0 generic											
	ps7_afi_1 generic											
	ps7_ah_2 generic	Documentation										
	ps/_an_5 generic	ps7_ah_3 generic Documentation										
	ns7 dev cfg 0 devcfg											
	ps7 dma ns dmaps	ps7_dev_cig_o develg <u>Documentation Import Examples</u> ps7_dma_ns_dmaps Documentation Import Examples										
	ps7_dma_s dmaps	Documentation Imp	ort Examples									
	Overview Source	Documentation					*	J				
👛 Target Connections 🛛 🥔 🥔 🕒 🗖	Problems 🕄 🧔 Tasks 🖾 Conse	le 🔲 Properties 🕒 SD	K Terminal	~ -	٥	SDK Log 23						
⊪ 🗁 Hardware Server	0 errors, 1 warning, 1 other					07:31:02 INFO : Registering co	mmand hand	llers for SDK TCF service		*		
E-      Linux TCF Agent	Description *	Resource	Path	Location	Ty	07:31:03 INFO : Launching XSCT	server: )	successfully	uf\Xil_Prj\L	IART\U		
	⊕ Marnings (1 item)     ⊕ i Infos (1 item)			07:31:00 INFO : ASCT Server		mand line	option -hwspec C:/buf/Xi	_Prj/UART/UA	ART.sd			
						07:31:09 INFO : Successfully o	one settir	g SDK workspace				
						<b>.</b>				•		

Рисунок 24. Вид SDK после генерации пакета поддержки платы

На этом можно считать оконченным этап импортирования спецификации аппаратной платформы и конфигурирования пакета поддержки платы.

### 3.3.2.2 Написание прикладного программного обеспечения

После создания пакета поддержки платы по экспортированной спецификации аппаратной платформы можно приступить к созданию прикладного программного обеспечения. Для создания данного проекта необходимо выполнить команду "File-> New->Application Project". После чего появиться окно, изображенное на рисунке 25:

💵 New Project	
Application Project Create a managed mal	te application project.
Project name: Uart	
Use default location	1
Location: C:\buf\Xil_I	rj\UART\UART.sdk\Uart Browse
Choose file s	ystem: default 🔽
OS Platform: standa	lone
Target Hardware	
Hardware Platform:	Zynq_Uart_Design_wrapper_hw_platform_0
Processor:	ps7_cortexa9_0
Target Software	
Language:	○ C C++
Compiler:	32-bit 💌
Hypervisor Guest:	N/A 🔽
Board Support Packa	ge: O Create New Uart_bsp
	♥ Use existing standalone_bsp_0
<u>.</u>	
?	< Back Next > Finish Cancel

Рисунок 25. Первый диалог при создании проекта прикладной программы

В нём можно определить название проекта, процессорное ядро на котором он будет исполняться и используемый для этого пакет поддержки платы. После выбора соответствующих пунктов, появиться окно с шаблонами проектов для выбранного языка программирования, показанное на рисунке (в данном случае С):



Рисунок 26. Второй диалог при создании проекта прикладной программы

По умолчанию в данном проекте будет использован шаблон "Hello World". После нажатия кнопки "Finish" будет создан проект с шаблоном "Hello World" и окно SDK приобретёт вид, указанный на рисунке 27:

UART.sdk - C/C++ - Uart/src/helloworld.c - Xilinx SDK										_ <del>_</del> ×
File Edit Navigate Search Project Run Xilinx Tools	Window Help									-
									Quick Access	B    40
Project Explorer 🛛 📄 🖏 🔻 🖓 🖓 🖓	system.hdf	helloworld.c 🛙					- 8	E Outline 🛙 🔁 D	ocu ® Make	- 0
Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalore_beg_0 Standalo	<pre># Copyright (c) 2009 - 2014 # Filesond.c: simple test     This sourcest configure in     This user (configure in     This user (configure in     This user (configure in     This user (configure in     This contrast) is a configure in     This user (configure in     This contrast) is a configure in     This contrast (configure in     This contrast)     This contrast, and (contrast)     This contrast, and (contrast, and (contrast)     This contrast, and (contrast)     This contrast, and (contrast, and (contrast)     This contrast, and (contrast, and (contrast))     This contrast, and (contrast, and (contrast))     This contrast, and (contrast, and (contra</pre>	<pre>cline, inc. All rig application s UMIT 1655 to band titalized by this appl to band rate 112000 e only in MW design figured by boatcom/bar by boatcom/bar</pre>	rate 9600. Lication, since					1 stáo h 1 platórní 1 si sil print 6 main() int		****
👛 Target Connections 🛙 🍰 🝰 🗖 🗖	🛐 Problems 😂 🧟 Tasks 🕒 Conso	le 🔲 Properties 🕒 SD	K Terminal	~ 🗆	٥	SDK Log 🛙				
🗄 🗁 Hardware Server	0 errors, 20 warnings, 1 other		1			07:31:02 INFO	: Registering command hand	lers for SDK TCF see	vices	-
E C Linux TCF Agent	Description A	Resource	Path	Location	Ty	07:31:03 INFO 07:31:06 INFO	: Launching XSCT server: x: : XSCT server has started :	sct.bat -interactive	C:\buf\Xil_Prj\U	JART\U
	<ul> <li>Marnings (20 items)</li> <li>i Infos (1 item)</li> </ul>					07:31:09 INFO 07:31:09 INFO 07:31:09 INFO 07:31:09 INFO	: Processing command line ( : Successfully done setting : Successfully done setting	sption -hwspec C:/bu g XSCT server conner g SDK workspace	f/Xil_Prj/UART/UA	ART.sd
						-				-
	•				•					•

Рисунок 27. Вид SDK после генерации проекта прикладной программы с шаблоном "Hello World"

\* The above copyright notice and this permission notice shall be included in
 \* all copies or substantial portions of the Software.

\* Use of the Software is limited solely to applications:

\* (a) running on a Xilinx device, or

\* (b) that interact with a Xilinx device through a bus or interconnect.

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

\* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

\* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL

\* XILINX BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,

\* WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF

\* OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

\* SOFTWARE.

\*

\* Except as contained in this notice, the name of the Xilinx shall not be used

\* in advertising or otherwise to promote the sale, use or other dealings in

\* this Software without prior written authorization from Xilinx.

### /\*

\* helloworld.c: simple test application

\* This application configures UART 16550 to baud rate 9600.

\* PS7 UART (Zynq) is not initialized by this application, since

\* bootrom/bsp configures it to baud rate 115200

\*

\* | UART TYPE BAUD RATE

\*

\* uartns550 9600

```
* uartlite Configurable only in HW design
  ps7 uart 115200 (configured by bootrom/bsp)
*/
//-----
//Подключаемые файлы:
#include <stdio.h>
#include <sleep.h>
#include "platform.h"
#include "xparameters.h"
#include "xgpio.h"
//-----
// Определение идентификатора входного порта данных
#define GPIO INPUT DEVICE ID XPAR AXI GPIO 0 DEVICE ID
//-----
// Декларация функции чтения данных из входного порта
int GpioInputData(u16 DeviceId, u32 *DataRead);
XGpio GpioInput; /* The driver instance for GPIO Device configured as I/P */
//-----
int main()
{
init platform();
 while(1)
 {
     int Status;
     u32 InputData;
  Status = GpioInputData(GPIO INPUT DEVICE ID, &InputData);
  if (Status != XST SUCCESS)
  {
  return XST_FAILURE;
  };
  print(&InputData);
  sleep(1);
 };
 cleanup platform();
 return 0;
};
//-----
// Определение функции чтения данных из входного порта
int GpioInputData(u16 DeviceId, u32 *DataRead)
{
int Status:
 Status = XGpio_Initialize(&GpioInput, DeviceId);
 if (Status != XST_SUCCESS)
 {
 return XST FAILURE;
 };
 XGpio SetDataDirection(&GpioInput, 1, 0xFFFFFFF);
 *DataRead = XGpio DiscreteRead(&GpioInput, 1);
 return XST SUCCESS;
};
     _____
//_.
```

### При этом окно SDK примет вид, указанный на рисунке 28:

UART.sdk - C/C++ - Uart/src/helloworld.c - Xilinx SDK	The last The							_ 5 X
Fue Latt Navigate Search Project Run Rubix Tools	winaow Help							
	<pre>// Operanews with the status; while(1) {// Accorange dynamics (# ) // Accorange dynamics (# ) //</pre>	INPUT DEVICE ID, I INPUT	HNG VICE_ID  Device configu &InputData);  NODIA ; F); 	red as I/P */				Cutter 23 PLOCI. S MARC. I S PLOCI. S PLOCI. S MARC. I S PLOCI. S PLOCI. S MARC. I S PLOCI. S PLOCI. S PLOCI. S MARC. I S PLOCI. S PLOCI. S PLOCI. S MARC. I S PLOCI. S PLOCI.
👛 Target Connections 😂 🥔 🦧 💭 🗖	😰 Problems 🛛 🧟 Tasks 📮 Console	🔲 Properties 📮 SD	OK Terminal	~ ~ [		SDK Log 🖾		
Hardware Server	0 errors, 21 warnings, 2 others	1.	1	1	1	07:31:02 INFO	: Registering command hand	lers for SDK TCF services
E Contraction of the Contraction	Description A	Resource	Path	Location	Ty	07:31:03 INFO 07:31:06 INFO	: Launching XSCT server: x: : XSCT server has started of	<pre>sct.bat -interactive C:\buf\Xil_Prj\UART\U successfully.</pre>
🛞 🏠 QEMU TcfGdbClient	<ul> <li> <b>a</b> Warnings (21 items)      </li> <li> <b>i</b> Infos (2 items)      </li> </ul>					07:31:09 INFO 07:31:09 INFO 07:31:09 INFO	: Processing command line of : Successfully done setting : Successfully done setting	ption -hwspec C:/buf/Xil_Prj/UART/UART.sd g XSCT server connection channel g SDK workspace
								-
	•				•	•		

Рисунок 28. Вид SDK после компиляции проекта прикладной программы

Таким образом, мы создали проект прикладного программного обеспечения, осуществляющий чтение данных с входного параллельного порта данных и перенаправляющего принятые данные в приёмопередатчик UART. Далее необходимо создать проект загрузчика первого уровня.

### 3.3.2.3 Формирование программы начального загрузчика FSBL и файлов прошивки для ПЛИС и флешки

Программа загрузчика начального уровня FSBL, представляет из себя, проект прикладного приложения, созданного с шаблоном "**Zynq FSBL**", которой входит в состав шаблонов SDK предоставляемых по умолчанию (рисунок 29).



Рисунок 29. Шаблон для проекта загрузчика начального уровня

Дальнейшие действия полностью аналогичны процессу создания прикладного приложения за исключением того, что нет необходимости дописывать свой код в проект. В результате компиляции проекта в его папке также появиться файл с расширением **\*.elf**. (рисунок 30)



Рисунок 30. Вид SDK после компиляции проекта загрузчика начального уровня

Далее необходимо сформировать файлы прошивки для ПЛИС и флэшки. Заметим, что файлы прошивки для первой уже были созданы. Ими являются \*.bit файл, сгенерированный в Vivado и \*.elf файл, сгенерированный в проекте прикладного программного обеспечения. Поэтому на данном этапе уже можно попробовать запустить проект. Для этого необходимо сначала загрузить в ПЛИС \*.bit файл, а затем запустить файл \*.elf, проекта прикладного программного обеспечения. Для загрузки \*.bit файла из среды SDK необходимо выполнить команду "Xilinx Tools - > Programm FPGA" (предварительно необходимо подключить плату к ПК через JTAG программатор (например XB-XUP USB-JTAG) и подать на неё питание). Для подключения JTAG программатора к плате можно использовать переходник с IDC10 на PBS-12 (LDM-ADAP-JTAG Xilinx - в комплектацию не входит) рисунок 31.



Рисунок 31. Вид переходника LDM-ADAP-JTAG Xilinx

В результате появиться окно, изображённое на рисунке 32:

🗱 Program FPGA					×
Program FPGA	and the FL	F files that reside in BRAM men	0.077		<b>→</b> -□
Specify the bist can		Thes that reside in DRAW men	liory		40-0
Hardware Configura	tion				
Hardware Platform:	Zynq_Uart	_Design_wrapper_hw_platform	_0 -		
Connection:	Local		•	New	
Device:	Auto Dete	ct		Select	
Bitstream:	Zynq_Uart	_Design_wrapper.bit		Search	Browse
T Partial Bitstream					
BMM/MMI File:				Search	Browse
-Software Configurati	ion				
Processor		ELF/MEM File to Initialize in H	Block R.	AM	-
2		Γ	Prog	am	Cancel
U.		L			

Рисунок 32. Программирование ПЛИС через JTAG в SDK

После успешной загрузки конфигурационной последовательности, необходимо на уже прошитой плате запустить проект прикладного программного обеспечения. Для этого необходимо в панели "**Project Explorer**" выбрать папку с прикладной программой и в её под директории "**Binaries**" выбрать файл с расширением \*.elf. В контекстном меню нужно выбрать следующую команду: "**Run AS-> Launch on Hardware (GDB)**" (рисунок 33).



Рисунок 33. Запуск проекта прикладной программы на уже сконфигурированной аппаратной части

После выполнения данной операции прикладное приложение будет запущенно на аппаратной платформе.

Таким образом, осталось сформировать лишь файл прошивки для флэшки и прошить её. Для создания файла прошивки флэш памяти необходимо выделить в панели "**Project Explorer**" папку прикладной программы и в контекстном меню выбрать команду "**Create Boot Image**" в результате выполнения которой появиться окно, изображённое на рисунке 34:

es Create Boot In	nage	
Create Boot Image Creates Zynq Boo	t Image in .bin format from given FSBL elf and partition files in specified output folder.	<b>1</b>
Architecture: Zyn	9 💌	
• Create new BIF	file C Import from existing BIF file	
Basic Security		
Output BIF file pa	nh: C:\buf\Xil_Prj\UART\UART.sdk\Uart\bootimage\Uart.bif	Browse
UDF data:		Browse
🗆 Split	Output format: BIN •	
Output path:	C:\buf\Xil_Prj\UART\UART.sdk\Uart\bootimage\BOOT.bin	Browse
Boot image partitio	n:	
Boot image partitic File path (bootloader) Cilb	ns Encrypted #X3 PyTUARTUART.adrFSBI.DebutFSBL.eff one	Authe
Boot image partitic File path (bootloader) Cibu CibufXd_PrjU/	ns If 2 py UARTUART of #SBL Debug #SBL eff none RFUART of Zyna L none	Authe none none
Boot image partitic File path (bootloader) C:lb C:buf.X3_PrjU/ C:buf.X3_PrjU/	ns fX3 PyUAXTUAXT add#SHUD-bng#SBL off fX3 PyUAXTUAXT add#SHUD-bng#SBL off snose NRTUART add Zong Uart add NRTUART add Uart Debug Uart add	Authe none none none Add
Boot image partitic File path (bootloader) C*b C*buf/Xd_PrjU/ C*buf/Xd_PrjU/	ns It. If Debyg 95511. eff If Debyg 95511. eff none RET UART side Zyng, Uart, Design, wrapper, Jwo, glatform, O Zyng, U none RET UART side Uart Debug Uart off none	Authe none none Add
Boot image partitic File path (beotloader) Ci-br C:buf/X3_PrjU/ C:buf/X3_PrjU/	ns Facrysted ANJ PyUAKTUAKT sik FSBI (Debug FSBL of none KRTUART sik Zyna, Uart, Design, wrapper, Jw., platform, 0/Zyna, U none RRTUART sik Uart Debug Uart of none	Authe none none Add Delet Edir
Boot image partitic File path (bootloader) Cib (bootloader) Cib (bootloade	ns #X0_PjUARTUART.od/#SBI.Debug#SBI.off none ARTUART.od/#SBI.Debug#SBI.off none ARTUART.od/#JonegUart.off none NRTUART.od/UartDebugUart.off none	Authe none none Add Delet Edit
Boot image partitic File path [bootloader] Cib C:buf X3_PrjU/ C:buf X3_PrjU/	ns The security of the securety of the securety of the securety of the securety of the secure	Authe aone none Add Edit Up Down
Boot image partitic File path (bootloader) Cile Cilear/Xd_Pay(U/ Cilear/Xd_Pay(U/	ns Therrypted If Ng_PyUAKTUAKT off: SSULDebug §SSUL elf none NRTUART sfk/Zyng_Uart_Design_wrapper_hw_platform_0/Zyng_U none NRTUART sfk/Uart Debug Uart elf none	Authe none none Add Edit
Boot image partitic File path (locodoader) C-ib C-ibert Xal_put C-ibert Xal_put 4	ns Facryped MAI DylUAKTUAKT sdc FSBL Debug FSBL eff none NRTUART sdc Zyng, Uar, Design, wrapper Jw. platform, 0 Zyng, U none NRTUART sdc Uart Debug Uart eff none	Authe none none Edit Up Dow

Рисунок 34. Диалог создания образа для флэш памяти

Этот диалог показывает, какие файлы войдут в загрузочный образ для флэшки. По умолчанию в него обязательно входит файл **\*.elf** загрузчика первого уровня, файл **\*.bit** конфигурации аппаратной части, разрабатываемой платформы и файл **\*.elf** проекта прикладного программного обеспечения. Также можно задать выходной формат файла и его месторасположение. Для данного проекта выходной файл должен иметь расширение **\*.mcs**.

После генерации данного файла в папке проекта прикладного программного обеспечения появиться под директория "**bootimage**", в которой и будет находиться сгенерированный файл. Его загрузку можно осуществить средствами SDK выполнив команду "Xilinx Tools->Program Flash". При этом появиться окно изображенное на рисунке 35:

SOK		×
Program Flash Mem	be .mcs/MCS or .bin/BIN format	
Hardware Platform:	Zynq_Uart_Design_wrapper_hw_platform_0	•
Connection:	Local	New
Device:	Auto Detect	Select
Image File:		Browse
Offset:		
Flash Type	qspi_single	-
FSBL File:		Browse
<ul> <li>Convert ELF to be</li> <li>Blank check after</li> <li>Verify after flash</li> </ul>	ootloadable SREC format and program erase	
?	Program	ancel

Рисунок 35. Программирование флэш памяти в SDK

Таким образом, был показан цикл разработки проекта на аппаратной – программной платформе ZYNQ с применением средств Xilinx Vivado и Xilinx SDK. Данный цикл, применим к проектам любой сложности, реализуемого на платформе ZYNQ.

### 3.4 Комплектация

- отладочная плата LDM-HELPER-XC7Z015-1CLG485;
- CD-диск с описанием к плате, демонстрационными примерами и дополнительным программным обеспечением.

### 3.5 Монтажные чертежи

Слой ТОР



Слой ВОТТОМ



### 3.6 Трассировка по слоям

Слой ТОР



Слой LAYER 1



www.ldm-systems.ru +7(495)500-89-20 info@ldm-systems.ru

### Слой LAYER 2



Слой LAYER 3



www.ldm-systems.ru +7(495)500-89-20 info@ldm-systems.ru

### Слой LAYER 4



Слой LAYER 5



### Слой LAYER 6



Слой ВОТТОМ





### Присоединительные размеры

### 4 Эксплуатация, хранение и транспортирование

#### Требования к условиям эксплуатации:

Изделие при испытаниях, перевозке, хранении и эксплуатации не наносит вреда окружающей среде и здоровью человека. Сохраняет свои параметры во всем диапазоне рабочих температур от 0°С до +70°С в закрытом помещении с относительной влажностью воздуха не более 80 %, без конденсата, при изменении напряжения первичного источника электропитания в допустимых пределах. По электромагнитной совместимости изделие соответствует всем требованиям для аппаратуры данного класса.

#### Требования к условиям хранения:

Изделие должно храниться в складских помещениях, защищенных от воздействий атмосферных осадков, на стеллажах в упаковке производителя при отсутствии в воздухе паров кислот, щелочей и других веществ, вызывающих коррозию. Условия хранения изделия по ГОСТ 15150-69: температура воздуха от +5°C до +40°C, относительная влажность до 80% при температуре +25°C. Предельный срок хранения в указанных условиях - три года.

#### Требования к условиям транспортирования:

Транспортирование изделия разрешается в упаковке производителя всеми видами транспорта, за исключением негерметизированных отсеков самолета, без ограничения расстояния.

Транспортирование упакованных изделий может производиться в крытых вагонах и автомашинах, трюмах судов и герметичных кабинах самолетов при температуре воздуха от -20°C до +70°C. При любом способе транспортирования необходимо предусмотреть крепление ящика к кузову (платформе) транспортного средства с помощью крепежной арматуры.