

# Создание многопроцессорной системы в Vitis: Zynq + MicroBlaze

Автор оригинала: Adam Taylor

Перевел: dvorozhbit

Рецензент: KeisN13





#### Оглавление

Аннотация	
Создание нового проекта	
Vitis Platform	7
Заключение	14

#### Инфо:

Данная статья представляет собой перевод. Оригинал статьи находится здесь <u>https://www.hackster.io/news/microzed-chronicles-microblaze-and-vitis-317aea401937</u>

Поддержите статью <u>Adam Taylor</u> - автора <u>оригинального руководства</u> и следите за выходом его <u>новых руководств</u>

Также не забудьте поддержать автора перевода

Хотите предложить материал для публикации? Напишите нам на fpga-systems@yandex.ru



### Аннотация

<u>Paнee</u> мы рассматривали то, как мы можем использовать Vitis для встроенных систем на Zynq и Zynq MPSoC.

Другим важным аспектом Vitis является возможность создать платформенное решение, которое включает в себя различные элементы обработки в PS и даже в PL.

Сегодня мы рассмотрим, как мы можем использовать Vitis для разработки платформенного решения на Zynq, которое использует и Cortex A9 в PS, и MicroBlaze в PL. Нашей целевой платой будет MicroZed 7020.

## Создание нового проекта

Первое, что нам нужно сделать, это создать новый проект, ориентированный на MicroZed 7020. После создания проекта следующим шагом будет создание блок-схемы и добавление в неё Zynq PS.

Reset A	ll Filters					Up	date B	loard Repositories
endor:	em.avnet.com 💙	Name:	All Remaining			✓ Boar	d Rev:	Latest 🗸
<u>S</u> earch:	Q-		~					
Display	Name			Preview	Vendor	File Version	Part	
MicroZ	ed 7010 Board				em.avnet.com	1.1	xc7z	010clg400-1
MicroZ	ed 7020 Board				em.avnet.com	1.1	xc7z	020clg400-1
MiniZe	ed				em.avnet.com	1.2	xc7z	007sclg225-1
Ultra96	5v1 Evaluation Platform				em.avnet.com	1.2	xczu	3eg-sbva484-1-e
Ultra96	5v2 Evaluation Platform				em.avnet.com	1.0	xczu	3eg-sbva484-1-e
ZedBo Add Da	ard Zynq Evaluation and I aughter Card Connection	Developme s	nt Kit	8000	em.avnet.com	1.4	xc7z	020clg484-1

После добавления системы обработки запустите автоматизацию блоков, чтобы сконфигурировать Zynq PS в PL для реализации MicroZed 7020.

À Run Block	Automation				
Automatica	Ily make connections in your desig	n by checking the boxes of the	e blocks to connect. Select a block on the left to	display its configuration options on	the right.
Q Z	ŧ 🔶	Description			
	P processing_system7_0	This option sets the boa This action cannot be ur for FIXED_IO, Trigger an NOTE: Apply Board Pres configuration. Instance: /processing_sy <b>Options</b> Make Interface External Apply Board Preset: Cross Trigger In: Cross Inigger Out:	ard preset on the Processing System. All current indone. Zynq? block automation applies current ind DDR interfaces. set will discard existing IP configuration - please ystem?_0 it: FIXED_IO, DDR it: FIXED_IO, DDR it: Disable v Disable v	properties will be overwritten by th board preset and generates externa uncheck this box, if you wish to retu	e board preset. I connections

Следующим шагом является добавление блока MicroBlaze Processing и настройка его на минимальную площадь.

Re-customize IP	
MicroBlaze (11.0)	ed.
IP Symbol Resources	Component Name microblaze_0
= Frequency = Area = Performance Resource Estimates	MicroBlaze* Usage Information
100.0 90.0 80.0 70.0 60.0 50.0 40.0 30.0	Select a predefined configuration with Select Configuration below. Information about the selected configuration can be found in the toolhip. <i>Each predefined configuration completely changes the MicroBlaze parameters.</i> To modify the configuration, click on the Next button, click on the Advanced button at the top to directly access parameters in a tabled interface, or click OX to accept the configuration and close the dialog.  Predefined Configuration Minimum Area Select Processor Implementation ③ 32 64 General Settings
20.0 10.0 0.0 0.0 0.0	Select implementation optimization AREA    Enable MicroBlaze Debug Module Interface  Use Instruction and Data Caches  Enable Exceptions
non x DAIE I	< Bark Next > Page 1 of 3 OK Cancel

<sup>Яндекс</sup> Деньги ڬ



Re-customize IP		
MicroBlaze (11.0)		À
Documentation 🗍 IP Location 🖌 Advance	đ	
P Symbol Resources	Component Name microblaze_0	
= Frequency = Area = Parformance	Instructions	^
Resource Estimates	Enable Barrel Shifter Enable Floating Point Unit NONE	- 1
100.0	Enable Integer Multiplier NONE V	
90.0	- Enable Integer Divider	
80.0	Enable Additional Machine Status Register Instructions	
70.0	- Enable Pattern Comparator	
60.0 E 60.0	Enable Reversed Load/Store and Swap Instructions	
2 50.0 40.0	Select Extended Addressing NONE	- 1
30.0	Optimization	
20.0	Select implementation optimization AREA V	
10.0	- Enable Branch Target Cache	
0.0	Branch Target Cache Size DEFAULT	
0.0	Fault Tolerance	
Resource Usage		v
MARK 8 DUMINE 8	< Back Ve	xt > Page 2 of 3
	OK	Cancel

Q ₹ ♦	Description			
	MicroBlaze connection a Debug Module, Periphe connected as needed. A Information about the o Options	automation generat ral AXI Interconnec preset MicroBlaze ptions can be foun	is local memory of selected size, and caches can be configured. Micro Interrupt Controller, a clock source, Processor System Reset are adde configuration can also be selected. In the tooltips.	Blaze ed and
	Preset Local Memory Local Memory ECC Cache Configuration	None 64KB None SKB SKB SkB	¥	
	Peripheral AXI Port	Enabled V	•	
	Clock Connection	/processing_syst	em7_0/FCLK_CLK0 (100 MHz) 👻	

После добавления MicroBlaze следующим шагом будет настройка Zynq A9 - это включает в себя отключение таймера APU и включение порта S\_AXI\_GP0 в PS - PL конфигурации.

www.FPGA-Systems.ru

ZYNQ7 Processing Sy	stem (5.5)							
Documentation 🌣 Pres	sets 🛛 IP Location 🌣 Import XPS Settin	ngs						
Page Navigator -	al brighter							Summary Repo
Zynq Block Design	Bank 0 I/O Voltage LV 🌱 Ba	nk 1 I/O Voltage 🛛 L\ 🛩						
PS-PL Configuration	← Q ≚ ≑ 4 0							
Peripheral I/O Pins	Search: Q-							
MIO Configuration	Peripheral	ю		S IO Typ	e S	P	P	
Clock Configuration	> Memory Interfaces			1				
	> I/O Peripherals							
DDR Configuration	Application Processor Unit							
SMC Timing Calculation	Timer 0	EMIO	~					
Interrupts	Timer 1							
	Watchdog							
	Programmable Logic Test and Det	200						
Re-customize IP YNQ7 Processing Syst	tem (5.5)						ОК	Cancel
Re-customize IP YNQ7 Processing Syst Documentation  Prese	tem (5.5) ets 🛸 IP Location 🏾 🏶 Import XPS Settings	<u>.</u>					ОК	Cancel
Re-customize IP ZYNQ7 Processing Syst Documentation Prese Page Navigator	tem (5.5) Its IP Location IP Import XPS Settings						OK	Cancel
Re-customize IP  EYNQ7 Processing Syst Documentation Prese Page Navigator  Zynq Block Design EPL Configuration	tem (5.5) IP Location P Import XPS Settings Information Import APS Settings Import APS Settings	· · · · · · · · · · · · · · · · · · ·					OK	Cancel
Re-customize IP  EYNQ7 Processing Syst Documentation Prese Page Navigator  Zynq Block Design PS-PL Configuration	tem (5.5) Its PLocation PI Import XPS Settings Information Import XPS Settings Import XPS Settings Import XPS Settings Import XPS Settings Import XPS Settings						СК	Cancel
Re-customize IP	tem (5.5) ts P Location P Import XPS Settings	Select			Dese	cription	OK	Cancel
Re-customize IP  YYNQ7 Processing Syst  Documentation Page Navigator  S-PL Configuration  Peripheral I/O Pins  MIO Configuration	tem (5.5) ts Di Location Di Import XPS Settings	Select			Desc	cription	OK	Cancel
Re-customize IP  YYNQ7 Processing Syst  Documentation Page Navigator  Aynq Block Design  PS-PL Configuration  Peripheral I/O Pins  MIO Configuration  Clock Configuration	tem (5.5) ts ☐ IP Location ♥ Import XPS Settings * Q ₹ € Search: Q: Name > General > AXI Non Secure Enablement > GP Share AXI Interface	Select			Dese Y Enal	cription ble AXI Not	OK	Cancel
Re-customize IP	tem (5.5) ts ☐ IP Location ♥ Import XPS Settings * Q ₹ € Search: Q: Name > General > AXI Non Secure Enablement ♥ GP Slave AXI Interface \$ AXI QPD interface	Select			Des Enal	cription ble AXI Not	OK	Cancel
Re-customize IP  ZYNQ7 Processing Syst  Documentation Proceeding Prese  Page Navigator  Page Navigator  Ps-PL Configuration  Peripheral I/O Pins  MIO Configuration  Clock Configuration  DDR Configuration	tem (5.5) ts ☐ IP Location ♥ Import XPS Settings ♥ Q ₹ ♥ Search: Q Name > General > AXI Non Secure Enablement ♥ GP Slave AXI Interface \$ AXI GPD interface \$ AXI GPD interface \$ AXI GPD interface	Select			Desi Enal Enal	cription ble AXI Not bles Genera	OK	Cancel
Re-customize IP  EYNQ7 Processing Syst Documentation Page Navigator  Zynq Block Design PS-PL Configuration Peripheral I/O Pins MIO Configuration Clock Configuration DDR Configuration SMC Timing Calculation	tem (5.5) ts ☐ IP Location ♀ Import XPS Settings	Select			Des v Enal Enal Enal	cription ble AXI Not bles Genera	OK	Summary Rep
Re-customize IP  EYNQ7 Processing Syst Documentation Page Navigator  Page Navigator PS-PL Configuration  Clock Configuration  Clock Configuration  DDR Configuration SMC Timing Calculation Interrupts	tem (5.5) ts ■ IP Location ♥ Import XPS Settings * Q ₹ € Search: Q: Name > General > AXI Non Secure Enablement ✓ GP Slave AXI Interface S AXI GP0 Interface S AXI GP1 Interface > AXI GP1 Interface > HP Slave AXI Interface > AP Slave AXI Interface	Select 0			Dess V Enal Enal	cription AXI Not	OK n Secure Trans al purpose 32- al purpose 32-	Summary Rep
Re-customize IP	tem (5.5) HS ■ IP Location ♥ Import XPS Settings PROFESSION Search: Qc Name > General > AXI Non Secure Enablement > GP Slave AXI Interface S AXI GPD Interface S AXI GPD Interface > AXI GPD Interface > AXI GPD Interface > AXI GPD Interface > AXI SPD Interface > AXI SPD Interface > AXI SPD Interface > AXI SPD Interface > MP Slave AXI Interface > AXF Slave AXI Interface > AXF Slave AXI Interface > MD Slave AXI Interface	Select 0			Dess V Enal Enal	cription ble AXI No.	OK a Secure Trans al purpose 32- al purpose 32-	Cancel

PayPa

деньги 🗹

Затем мы можем запустить автоматизацию соединения, которая создаст базовую систему MicroBlaze и подключит ее к Zynq PS.

Когда мы подключаем порт MicroBlaze Peripheral AXI к порту ведомого (slave) GP на Zynq PS, MicroBlaze получает доступ к нескольким периферийным устройствам и памяти, которые доступны для Zynq PS.





Завершающими этапами являются создание обёртки HDL, генерация битстрима и экспорт XSA.

# Vitis Platform

PayPa

После запуска Vitis первое, что мы хотим сделать, это создать проект приложения. Это также создаст новый системный проект - именно к этому системному проекту мы добавим домены.

Первый домен будет создан в платформе автоматически при создании проекта приложения.

and a New Anglia	ject		
Enter a name for your	Application project.		
Project name: Zynq_n	b		
Use default location	n		
ocation: C:\hdl_proje	cts\vitis_mb\workspace\Zynq_mb		Browse
Choose file s	ystem: default		
System project: Cre	ate New V Zynq_mb_system		
Curtam	Custom Declast 0	• A system project is	a container for multiple
System Project 1 Baremetal App Baremetal Baremetal	System Project 2	<ul> <li>A system project is applications that wo platform at the same</li> <li>A domain is the BS isomorphic processo</li> <li>A platform contain</li> </ul>	a container for multiple uld run on different domains of a a time. P/OS that controls one or more ors. s one or more domains.
System Project 1 Baremetal App Baremetal Domain	System Project 2 Linux App 1 Linux App 2 Al Engine App A72 Linux Domain Platform	<ul> <li>A system project is applications that wo platform at the same</li> <li>A domain is the BS isomorphic processo</li> <li>A platform contain</li> <li>A workspace can co unlimited system pro</li> </ul>	a container for multiple uld run on different domains of a e time. P/OS that controls one or more ors. s one or more domains. ontain unlimited platforms and ojects

Добавьте платформу: нажмите кнопку «+» и укажите в диалоговом окне экспортируемый XSA (только что созданный с помощью Vivado). Когда будет предложено выбрать процессор для домена, выберите процессор Cortex A9\_0.

. Il Croste			
y create a	new platform fro	om hardware (X	SA)
Board vck190 zc702 zc706 zcu102 zed design_1	Flow Embedded Embedded Embedded Embedded Embedded	Vendor Xilinx Xilinx Xilinx Xilinx Xilinx Xilinx	Path C:Xilinxi\Vfiis\2019.2\data\embeddedsw\lib\fixed_hwplatforms\vck190.xsa C:Xilinxi\Vfiis\2019.2\data\embeddedsw\lib\fixed_hwplatforms\vc702.xsa C:Xilinxi\Vfiis\2019.2\data\embeddedsw\lib\fixed_hwplatforms\vc2706.xsa C:Xilinxi\Vfiis\2019.2\data\embeddedsw\lib\fixed_hwplatforms\vc2.x102.xsa C:Xilinxi\Vfiis\2019.2\data\embeddedsw\lib\fixed_hwplatforms\vc4.xsa C:\Xilinxi\Vfiis\2019.2\data\embeddedsw\lib\fixed_hwplatforms\vc4.xsa C\\filprojects\vitis_mb\design_1_wrapper.xsa
	Board vck190 zc702 zc706 zcu102 zcd design_1	Board Flow vek190 Embedded zc702 Embedded zcr06 Embedded zcu102 Embedded zed Embedded design_1 Embedded	Board Flow Vendor vck190 Embedded Xilinx zc702 Embedded Xilinx zcr06 Embedded Xilinx zcu102 Embedded Xilinx zed Embedded Xilinx design_1 Embedded Xilinx

Следуйте подсказкам в диалоговом окне и создайте пример "hello world", нацеленный на ядро Cortex-A9 0.

Это откроет новый проект приложения и системный проект для платформы. Под платформой вы увидите, что домен Zynq уже существует вместе с FSBL.

Чтобы иметь возможность добавить приложение MicroBlaze, нам нужно добавить новый домен в платформу. Откройте файл проекта platform.spr и нажмите кнопку +, чтобы добавить новый домен.

Введите имя для домена и выберите процессор MicroBlaze.

Workspace - desig	gn_1_wrapper/platform.spr - Vitis IDE ct Window Help						– α ×
	• 1 · • • • • • • • • • • • •						Quick Access
Explorer 💷	8 <b>%</b> 4 ***	≚ Zynq_mb_system Strang_mb	design_1	_wrapper 🖾			
✓ ■ design_1_wrap > ○ bitstream	oper (Out-of-date)	type filter text	BBOX	Domain: zynq_fst	a		
> 🗁 export > 🗁 hw		<ul> <li>Ops7_cortexa9_0</li> <li>Strong febl</li> </ul>		OS:	standalone		
> elogs > eps7_cortexa	a9_0	Board Support Package		Processor: Supported Runtimes:	ps7_cortexa9_0 C/C++ ~		
resources		Board Support Package	19_0	Display Name:	zyng_fsbl		1
✓ platform.sp	e			Description:	FSBL Application BSP - Auto	o Generated.	
	tandidane on po?, contexe@_01 i b poj pysem.apoj em (System) per (Plantom)	When Domain in Vesign Domain Create a new domain. Name Display Name: OS: Vesios: Processos: Supported Runtimes: ?	mb_p( mb_p) standalor standalor microblac C/C++	se ne(7.1) ee_0	<ul> <li>✓</li> <li>✓</li> <li>✓</li> <li>OK</li> </ul>	Carcel	~
		Main Hardware Specification					
		🖾 Console 🕮 🔝 Problems 🔝 Vitis Lo	g 🛈 Guidan	ce			÷ ÷ 🕅 🖬 🖬 🖬 🖬 🕶 • 😁 •
		Build Console (Zynq_mb_system, Debu	1				
		4					3

Как только это будет выполнено, под платформой вы увидите два домена - один для Zynq A9, а другой для MicroBlaze.



Теперь мы можем обновить платформу и перекомпилировать BSP и FSBL.

Чтобы создать второе приложение для MicroBlaze, выберите системный проект и щелкните его правой кнопкой мыши, чтобы добавить проект приложения.



Откроется диалоговое окно создания проекта. Введите имя проекта - имя системного проекта должно быть заполнено автоматически.

Enter a name for	your Application project.		L
Project name: m	b_app		
Use default lo	cation		
ocation: C:\hdl_	projects\vitis_mb\workspace\mb_app		Browse
Choose	file system: default		
System project	Zynq_mb_system ~	• A system project is	a container for multiple
System project System Project Baremeta App	System Project 2	A system project is applications that wo platform at the same A domain is the BS isomorphic processo	a container for multiple uld run on different domains of e time. P/OS that controls one or more rrs.

PayPal

<sup>яндекс</sup> Деньги ڬ

Когда будет предложено выбрать домен, выберите только что созданный домен MicroBlaze и снова выберите пример «Hello World».

New Appl	ication Project			>
omain				••••
Provide the	domain and other softw	are details for your project		
Domain:	mb_pl	~		
Language	: C ~			
CPU:	microblaze_0			
OS:	standalone			

Эти шаги приведут к созданию двух проектов приложений в рамках системного проекта.



Измените приложение hello world в обоих проектах, чтобы отразить, с какого процессора отправляются сообщения.

#include <stdio.h> #include "platform.h" #include "xil\_printf.h" int main() { init\_platform(); print("Hello World From Zynq\n\r"); cleanup platform(); return 0; ł #include <stdio.h> #include "platform.h" #include "xil\_printf.h" int main() { init platform(); print("Hello World From MicroBlaze\n\r"); cleanup\_platform(); return 0;

Мы хотим запустить код MicroBlaze из его внутренней BRAM. Выберите файл компоновщика (lscript.ld), доступный в исходном каталоге, и удалите его.



В меню Xilinx выберите параметр «Создать сценарий компоновщика».



Откроется диалоговое окно, в котором будет создан новый скрипт компоновщика. Убедитесь, что Code, Data, Heap и Stack находятся в локальной памяти BRAM.

Generate a linker script			,				
Generate linker script			10				
Control your application's memory map.							
Project: mb_app Output Script:		Basic Advanced Place Code Sections in:	al_memory_ilmb_bram_if_cntlr_Mem_microblaze_0_local_memory_dlmb_bram_if_cntlr_Mem_				
C:\hdl.projects\vitis_mb\workspace\mb_app\src\lscript.ld	Browse	Place Data Sections in:	$al\_memory\_llmb\_bram\_if\_cntlr\_Mem\_microblaze\_0\_local\_memory\_dlmb\_bram\_if\_cntlr\_Mem ~~$				
Modify project build settings as follows:		Place Heap and Stack in:	${\it sl\_memory\_ilmb\_bram\_if\_cntlr\_Mem\_microblaze\_0\_local\_memory\_dlmb\_bram\_if\_cntlr\_Mem} ~~ \\$				
Set generated script on all project build configurations	~	Heap Size:	1 KB 0x00000400				
		Stack Size:	1 KB 0x00000400				
Memory Base Address size microbiaze_0_local_memory0000000000 64 KB pa7_ddr_0_dPQ_AXB_BASBN02000000 512 pa7_dppL[inear_0 0.4fC000000 16 MB							
Fixed Section Assignments			Generate Cancel				

Теперь мы можем собрать системный проект и загрузить его для тестирования на MicroZed. Для этого нам нужно создать новую конфигурацию отладки. Выберите System Project Debug и создайте новую конфигурацию отладки.



Tube

После того как MicroZed настроен, мы можем выполнить код по шагам или запустить процессоры в соответствии с требованиями приложения.

3 • 20 0 • 5 • 10 • 0 • 10 × 0 = N3.0.83											Quick Access	Design & Debug
R Dahun II	≪Zuna mb	derion 1 wr	anh ann	Delloworld c	R helloworld c	N lectint ld	Direction Id	* * *	09: Variabl	Po Brook	9° Evorer Nodular	III Peoint
E. System-Bologuez Joya Link zystem (System Project Debug) E. System-Bologuez Joya Link zystem (System Poisto Belog) * 2 RAU *	Klangth Udsignijum. Kimbjap @ https://wimbi.com/dia/     Skrigti ***********************************							<	Typ	e Vak	Platform D	
tophore (	S7 print S8 S9 clear 60 retur 61 } 62 Console E C TCF Debug Virtual	("Hello World up_platform() n 0; Vitis Serial Termin Terminal - ARM C	d From MicroB ); nal O Executabl Cortex-A9 MPCor	laze\n\r"); is 📄 Vitis Log 🛣 F	roblems 🗟 Debu	gger Console		20 · C	(▼ = □	XSCT Console SCT Process Info: MicroBl main() at/ St: init soth Info: A main() at/ St: init soth	Emulation Consol aze #0 (target 6) ; src/helloworld.cs Jeatform(); MR Cortex-A9 NRCor src/helloworld.cs platform();	Stopped at 0x A 55 e #0 (target 55

Конечно, в этом примере оба процессора используют один и тот же UART, поэтому я запускал один процессор за другим.



### Заключение

Это очень простой пример того, как многопроцессорная система может быть определена и разработана с использованием Vitis. Это также показывает способность разрабатывать системные решения на уровне платформы и выполнять соответствующую отладку.

По мере развития нашего приложения Vitis, я уверен, что мы будем использовать этот подход в более сложных приложениях и проектах в будущем!

Понравилась статья? Не забудьте поддержать автора перевода 🕺

Хотите предложить материал для публикации? Напишите нам на fpga-systems@yandex.ru