

# XC6SLX16 FPGA 開発ボード 快速入門ガイド

株式会社日昇テクノロジー

http://www.csun.co.jp 🛌

info@csun.co.jp 🥖

作成・更新日 2014/04/13



### copyright@2014





# • 修正履歴

-	1		
NO	バージョ	修正内容	修正日
	ン		
1	Ver1.0	新規作成	2014/04/13

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。 最新版は弊社ホームページからご参照ください。「http://www.csun.co.jp」 ※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に 禁じられています。



# 目录

1 初めての FPGA プロジェクト	5
1.1 開発ボード構造図	5
1.2 プログラム設計	6
1.2.1 設計分析	6
1.2.2 ソースコード LED_TEST. vhd	6
1.3 実験手順	7
2 初めての組込 CPU プロジェクト(Microblaze)	15
2.1 MicroBlaze ソフトコア作成	15
2.1.1 ソフトコアでウィザード作成	15
2.1.2 Platform Studio 操作画面	
2.1.3 UCF ピン制約ファイルを追加	23
2.1.4 コンパイルとダウンロード	25
2.2 Hello Worldアプリケープログラム開発	28
3 FPGA のコンフィーグ方法と書込み手順	33
3.1 序文	33
3.2 事前準備	33
3.3 SPI FLASHの書き込み	34
3.3.1 iMPACT 起動	34
3.3.2 mcs ファイル作成	35
3.3.3 MCS ファイルを SPI FLASH に書込む	37



本マニュアルは初心者向けです。

多くの初心者はどこから始めるかよく迷うでしょう?本マニュアルは本開発ボードを最速に使いこなせる 様に、下記3つの部分と分けて分かりやすく説明します:

- 初めての FPGA プロジェクト
   ISE 開発環境で新しいプロジェクトの作成から最後の開発ボードにダウンロードし、プログラムをデバッグするプロセスを紹介する。
- 初めての組込 CPU プロジェクト(MicroBlaze)
   ISE 開発環境で MicroBlaze ソフトウェア作成、プロジェクト開発のプロセスを紹介する。
- 3. FPGA のコンフィーグ方法とダウンロード手順





# 1 初めての FPGA プロジェクト

本章では LED 順次点灯実験を例とし、ISE 開発環境で新しいプロジェクトの作成から最後の開発ボードにダウンロードし、プログラムをデバッグするプロセスを紹介する。

LED 順次点灯実験でピンロックとビットシフト演算子の使用方法も紹介する。

実験準備:

- 開発ボード
- Xilinx ダウンロードケーブル
- PC 側 Xilinx ISEDesign Suite 13.1 開発ソフトウェア

#### 1.1 開発ボード構造図

Siga-S16 開発ボードでは6つのブルーLED 灯がある。1つは FPGA INT、1つは FPGA Done、この2つの LED はハードウェア設定ステータスを表示する、ユーザーから制御できません。他の4つの LED1<sup>~</sup>LED4 を本実験で使用する。



LED ピン対応は下記の通り:

LED1-----PIN:T10 LED2-----PIN:N10 LED3-----PIN:T12 LED4-----PIN:R10 対応ピンが´0´でLED 点灯、´1`でLED 消灯。 本実験ではクロックカウントが必要で、開発ボードクロックの対応ピンは: FPGA\_GCLK1-----PIN:V10



1.2 プログラム設計

#### 1.2.1 設計分析

4 つの LED の順次点灯効果を実現するには、点灯の速度が速すぎでも、遅すぎでもよくないです。本設計のタイミングは 0.5 秒に設定し、0.5 秒毎に一つの LED を点灯する。LED1 点灯、0.5 秒後 LED2 点灯(LED1、LED2 同時点灯)、順次に点灯し、4 つの LED 全て点灯後、全部消し;一回動作終了後、サイクル再開する…

FPGA 出力のデータは 1110、1 秒後 1100……0000、まで変化する、出力の変化により、順次点灯動作を実 現できる。



counter のレジスタを使用し、カウントする。0.5 毎 LED は左へ1 ビットシフトし、最低位(一番右)は0 補足。LED 全て点灯後、もう一回 0.5 秒カウントし、全滅する。



#### 1.3 実験手順

1. ISE Project Navigator 開発環境起動(スタートメニュー: Xilinx ISE Design Suite13.1->ISE Design Tools->Project Navigator).



- 2. 新しいプロジェクト作成(File-New Project)、ポップアップしたのダイアログボックス中プロジェクトの名称と保存ディレクトリと動作ディレクトリを設定する。
  - `Next` クリック。



Greate Her Project location and type.         Enter a name, locations, and comment for the project         Ngme:       IED_TEST         Location:       C:\Kilinx\LED_TEST\LED_TEST         Working Directory:       C:\Kilinx\LED_TEST\LED_TEST         Description:          Select the type of top-level source for the project         Top-level source type:         MUL	New Project Viz	ard	×
Enter a name, locations, and comment for the project          Ngme:       LED_TEST         Location:       C:\Xilinx\LED_TEST\LED_TEST         Working Directory:       C:\Xilinx\LED_TEST\LED_TEST         Description:          Select the type of top-level source for the project         Top-level source type:         MDL         More Info         Mext	<b>Create New Projec</b> Specify project	t location and type.	
Ngme:       LED_TEST         Location:       C:\Xilinx\LED_TEST\LED_TEST         Working Directory:       C:\Xilinx\LED_TEST\LED_TEST         Description:          Select the type of top-level source for the project         Top-level source type:         MDL         More Info	-Enter a name, loc	ations, and comment for the project	1
Location: C:\Xilinx\LED_TEST\LED_TEST working Directory: C:\Xilinx\LED_TEST\LED_TEST  Description: Select the type of top-level source for the project Top-level source type: HDL More Info Mext Cancel	N <u>a</u> me:	LED_TEST	
Morking Directory:       C:\%ilinx\LED_TEST\LED_TEST         Description:          Select the type of top-level source for the project         Top-level source type:         HDL         More Info         Mext	Location:	C:\Xilinx\LED_TEST\LED_TEST	
Description: Select the type of top-level source for the project Top-level source type: HDL More Info Next Cancel	Working Directory	C:\Xilinx\LED_TEST\LED_TEST	
Select the type of top-level source for the project Top-level source type: HDL More Info	Description:		
Select the type of top-level source for the project Top-level source type: MDL More Info Next Cancel			
Select the type of top-level source for the project Top-level source type: HUL More Info Mext. Cancel			
Select the type of top-level source for the project Top-level source type: HDL More Info			
Select the type of top-level source for the project Top-level source type: HUL More Info Next Cancel			
Select the type of top-level source for the project Top-level source type: HDL More Info Next Cancel			
Select the type of top-level source for the project Top-level source type: HDL More Info Rext Cancel			
Top-level source type: HDL More Info Next Cancel	Select the type o	f top-level source for the project	
HDL More Info	<u>T</u> op-level source	type:	
More Info	HDL		
More Info			1
	More Info	Next	

3. 次のダイアログボックスで FPGA ディバイス選択、設定する。結果は下記の通り:

~
~
~
~
~
~
~
~
~
~
~
~
~



`Next゛クリック。

4. <sup>\*</sup> Finish<sup>\*</sup> クリック、設定終了。

New Project Wixard	
<b>Project Summary</b> Project Savigator will create a new project with the following specifications.	
Project:	
Project Name: LED_TEST	
Project Path: C:\Xilinx\LED_TEST\LED_TEST	
Working Directory: C:\Xilinx\LED_TEST\LED_TEST	
Description:	
Top level Source Type: HDL	
Device:	
Device Family: Spartan6	
Device: xc6s1x16	
Package: csg324	
Speed: -3	
Top-Level Source Type: HDL	
Synthesis Tool: XST (VHDL/Verilog)	
Simulator: ISim (VHDL/Verilog)	
Preferred Language: Verilog	
Property Specification in Project File: Store all values	
Manual Compile Order: false	
VHDL Source Analysis Standard: VHDL-93	
Message Filtering: disabled	
More Info	Cancel

5. LED\_TEST ソースファイル作成(メニュー: Project->New Source)、ポップアップしたの New Source Wizard ダイアログボックス中 VHDL Module 選択、そしてファイル名欄に LED\_TEST 入力。

Mew Source Vizard	
Select Source Type Select source type, file name and its local	ti m.
<ul> <li>HWW File</li> <li>ChipScope Definition and Connection File</li> <li>Inplementation Constraints File</li> <li>IP CORE Generator &amp; Architecture Nicord)</li> <li>NEW File</li> <li>Schwatic</li> <li>Ver Document</li> <li>Verilog Module</li> <li>Verilog Tett Fixture</li> <li>VOLL Rochase</li> <li>VHDL Fost Bench.</li> <li>Enbedded Froeesson</li> </ul>	<pre> File name: LED_TEST Legation: C:\Yiline\LED_TEST\LED_TEST  # Add to project</pre>
Yore Info	Mext > Cancel

6. ポート定義のダイアログボックスでLEDと clockのポート名:





🏧 New Source Vizard

Define ∎odule

×

Specify ports for module. Module name LED\_TEST LSB Port Name Direction Bus MSB LED ✓ 3 output 0 FPGA GCLK1 \* input ¥ 🗌 input ¥ input ¥ 🗌 input ¥ input More Info < <u>B</u>ack Next > Cancel <sup>\*</sup> Finish<sup>\*</sup> クリック、設定終了。 7. 🔤 New Source Vizard Summary Project Navigator will create a new skeleton source with the following specifications. Add to Project: Yes Source Directory: C:\Xilinx\LED\_TEST\LED\_TEST Source Type: VHDL Module Source Name: LED\_TEST.vhd Entity name: LED\_TEST Architecture name: Behavioral Port Definitions: LED Bus: out FPGA\_GCLK1 Pin in CPU\_RESET Pin in More Info  $\leq \underline{B}ack$ Finish Cancel

8. Design の Hierarchy ウィンドウで LED\_TEST. vhd ファイルは既にプロジェクトに追加した。





VHDL ファイルをオーペンし、LED のテストソースコードを編集できる。または添付 DVD にある LED\_TEST. vhd のコードを直接 vhd ファイルにコピーする。

9. 次は FPGA のピン制約ファイル.ucf を作成する。空白ファイルを作成 (File->New)、ピン定義を追加 する。

#########LED Pin define###################################	###### svs clk nin•			
TIMESPEC TS sys clk nin = PERIOD sys	clk nin 50000 kHz:			
TIMOTO IS_SYD_OTK_PIN TERTOS SYD_	pin 00000 km2,			
########LED Pin define###################################				
NET LED <o></o>	LOC = T10   IO	STANDARD = "LVCM	10533": #	## 2 on DS11 LED
NET LED<1>	LOC = N10   IO	STANDARD = "LVCM	10533": #	## 2 on DS12 LED
NET LED<2>	LOC =_T12   IO	STANDARD = "LVCN	IOS33"; #	## 2 on DS13 LED
NET LED<3>	LOC = R10   IO	STANDARD = "LVCM	IOS33"; #	## 2 on DS14 LED
	<b>Y</b>			
完成後、LED_TEST_PIN. ucf ファイルと	して保存、プロジ	ェクトに追加、	結果は下記図の	ひ通り:
Design	++ 🗖 🖥 ×			
📑 View: 💿 🔯 Implementation 🔿 🧱	Simulation			
Ji er ar chy				
E LED_TEST				
1 xc6s1x16=3cs/324				
a in the IED TEST - Babari and	(IED TEST what)			
	(LED_TEST. viid)			
ELED_IEST_FIR.uct				
×				

10. プロジェクト保存してコンパイル開始: Generate Programming File クリック、ソフトウェアは自動的に bit ファイルを生成し、FPGA の設定に使用する。





Configure Target Device

Analyze Design Using ChipScope

Đ - 🧏

コンパイル成功後、Console ウィンドウで成功情報を表示する:



12. iMPACT 環境で Boundary Scan をダブルクリック、JTAG リンクをスキャン。

File Edit View Operations Du	tput Debug	<u>W</u> indow	<u>K</u> elp	
NPACT Flows ■ Bounder / Scan SystemACE Create PRJM File (PROM File WebTalk Data	Formstter)		↔ □ & ×	
				8
NPACT Processes			⇔⊡₽×	

13. Boundary Scan ウインドウで、右クリックして Initialize Chain を選択。ソフトウェアは自動的に xc6slx16 チップを検索。



Right click device to select operations (SP1)BP1 TDI Exume xc6slx16 bypass TDO
Auto Assign Configuration Files Query Dialog Do you want to continue and assign configuration files(s)? Don't show this message again, save the setting in preference. <u>Yes</u> No ICCENTRY Succeeded
Boundary Scan       * Yes * 選択、ISE Project Navigator でLed_test.bit ファイル生成、オーペン。
Assign New Configuration File
Ny Computer     Image: Internet of the i
Eile nemei lad test hit
Cancel
Files of type: All Design Files (*. bit *. rbt *. nky *. isc *. bsd)

次のダイアログボックスで No 選択、Flash PROM の書き込みは別の実験で説明する。





次ダイアログボックスで、OK、選択。

🐉 Device Programming Properties -	Device 1 Pro	gramin	g Properties	×
Category □ Toundary Scan □ Device 1 (FFGA xe8sls16 )	Property Name	Value		
	Yerify			
		0E	Cancel Apply Help	

14. xc6s1x16 チップ選択(右クリック)、Programは FPGA のダウンロードを開始する。



ダウンロード完成後、下記の情報を表示する:

### Program Succeeded

開発ボード上の LED は順次点灯動作を開始する。同じ原理で、LED の点滅順序、時間は自由に設定できる。



# 2 初めての組込 CPU プロジェクト (Microblaze)

多くの FPGA プロジェクトソリューションでは、FPGA+CPU がよく使われる。例えば通信デバイスでは FPGA はデータ処理、CPU は管理処理を担当。コストを考慮し、ソフト CPU を FPGA 内に組み込んで、FPGA だけで FPGA+CPU のソリューションを完成する事でコストを下げる。

Xilinx 社改善した MicroBlaze 組み込みソフトコアは FPGA システムに組み込まれる RISC プロセッサ・ソフトコアです。実行スピードが速い、リソース使用量が少ないなどのメリットがあります。

本章では ISE で Microblaze ソフトコアを作成し、C 言語で HelloWorld のプロジェクトを実現する。

#### 実験準備:

開発ボード、開発環境 ISE13.1。

下記2ステップで実現する: 1、MicroBlaze ソフトコア作成

2、 Hello World プログラム開発

### 2.1 MicroBlaze ソフトコア作成

MircoBlaze ソフトコアは ISE の EDK (Xilinx Platform Studio) 開発環境で作成する。

# 2.1.1 ソフトコアでウィザード作成

Xilinx Platform Studio ソフトウェアをオーペン、ポップアップしたのダイアログボックスにある Base System Builder wizard を 選択、ソフトコア作成ウィザードに入る。

🗢 Xilinx Platform Studio 🛛 🛛 🔀
Create new or open-existing project BSB D Base System Builder wizard (recommended) C Blank XPS project C Open a recent project
Browse for More Projects
Browse EDK examples (projects) on the web <u>here</u> OK Cancel Help

プロジェクトの保存ディレクトリを設定:



New project	
roject file	
D:\Xilinx\Workspace\system.xmp	<u>B</u> rowse
Advanced options - optional (press	s F1 key for help) ory Search Path
	Browse

PLB system 選択。PLB は旧バージョンのソフトコアシステム、操作は比較的に簡単です。AXI は新バージョンソフトコアシステム。



ウィザードに従い、Next クリック:





FPGA モデルを設定する、Siga-S16の場合、設定は下記の通り:

🔶 Base System	Builder						? 🔀
Welcome	Board	System	F	rocessor	Peripheral	Cache	Summary
<b>Board Selection</b> Select a target dev	velopment board.						
Board	•			-			
I would like to	create a system for th	ne following o	levelopme	nt board			
Board Vendor	Xilinx						×
Board Name	Spartan-3A DSP 1600A S	Starter Board					
Board Revision	1						~
⊙ I would like to	create a system for a	custom board					
-Board Information	Device			Packaga		Speed Grade	
spartan6	xc6slx16		~	csg324		-3	~
Use Stepping							~
Reset Polarity Act	ive High						
P-lated T-franction							
This option allows Using this option, board. Supported du IIC, and SPI. The a hardware, you will	you to rapidly and eas you must specify the F wices include DDR and generated system can be have to add the FPGA p	ily create a GGA device yo SDRAM memory used to run in location c	base or s u will be controlle simulatic onstraint	starter desi e using and ers, 10/100 uns. If you is into the	gn that does not r external memories : Ethernet, GFIO, an would like to down generated UCF file.	equire a specific and I/O devices th 1 serial devices s load this system o	target board. at are on your uch as VARTs, nto your
More Info					< <u>B</u> ac	k <u>N</u> ext >	Cancel



Single-Processor を選択 (Single-Processor System)、Next をクリック

Yelcone	Board	System	Processor	Feripheral	Cache
System Configuratio	n				
onrigure your system.	ingle-Processor S	System		(7) Dual-Process	er System
Single-Processor System Salart this option to create a design with a single processor. This Wizard will let you configure the processor, the peripheral set and some major configuration parameters for the peripherals.			Salast this processors processors, peripherals	option to create a c This Virard will let peripherals unique t s shared by the proces	esign with two you select the typ o each processor, a surs.
Processor 1	_	,	Proce	ssor 1 Processo	r 1 Peripherals GPIC (mmm)
	Processor 1 F	Peripherals	[	Shared P Mailbox	er cherais Nutex
			ecor9	ssor 2 Processo DDR	r 2 Peripherals EMAC

ハードウェアプラットフォームのメインクロックに基づき、対応の水晶発振器の周波数を選択(Siga-S16 開発ボード上のクロックは 50Mhz)。システムクロックの範囲は水晶発振器の2倍以内にする必要。メモリの デフォルトは 8KB。

🔶 Base System Bu	ilder					? 🛛
Melcone	Board	Systen	Processor	Per i pher el	Cache	Sunnary
Processor Configure Configure the process	ation or (s).					
Reference Clock Frequ Processor 1 Config	uency 50.00					VHz
Processor Type	Ni croBlaza					~
System Flock Freque	ncy 66.67					Y NHz
local Menory	8 338					~
Debug Interface	On-Chip HW De	ebug Modale				~
Enable Floating	Point Unit					
Vor= Info				< <u>B</u> ack	<u>H</u> est >	Cancel



これはシステムデフォルトで一番簡単なカーネルである。



Add Device で MCB\_DDR2 外部デバイスを追加する。他のデバイスも MicroBlaze 生成後追加も可能。



ニーズによってプロセッサの指令とデータ cache のサイズを設定する、ここでは 1K バイドにする。



🔶 Base System Bui	lder					? 🛛
fielcome	Board	System	Fracessor	Feripheral	Cache	Sumary
Cache Configuration Select cache size and d	cache nemory for	processor (s).				
-Trocessor 1 (MicroB) In MicroBlace, cache RAMS for large sized	sre) Eache s are optional a caches. he	and configurable. Cache	es are implement	ted using BPGA LU7x fo	or small caches or	Block
Instruction Cache S	ize 1kB	~	Data Cache	Sice 1kB		×
Instruction Cache W	encary		Data Cache	Nenory		
					=	
					=)	
Nore Info			•	Lock	Kext >	Cancel

EDK で自動的にシステムカーネルのアドレスを設定する、Finish。

🗢 Base System Bu	nilder					? 🗵
Yelcone	Board	System	Processor	l'eripher al	Cache	Summary
Summary Below is the summary	of the system you	are creating.				
System Sammery						
Core Name	Instance Name	Base Address	Migh Address			
Eile Leastin	niersblar. 0 MCB_DDR2 (Ir dab_confic (Ir ilsb_confic	0±1800000 050000000 050002000	92497787787 CadCool1787 Oa000017979			
Over all D: Uiline Work D: Uiline Work D: Uiline Work D: Uiline Work D: Uiline Work D: Uiline Work	npacelayalan sap nbacelayatan sap nbacelayatan mis nyaa datelayatan napacelatelayatan napacelatelayan an nyacelatelbitgan ut	uc£ cnd				
✓ Save Base System B D:\Xilinx\Vorkspace	uilder (.bsb) Setti ksysten.bsb	ngs File				
More Info				C Lock	- Einish	Cancel

# 2.1.2 Platform Studio 操作画面

ウィザードでソフトコア作成後、Platform Studio-カーネル開発環境に入る。下記図の通り:



低価格、高品質が不可能?

日昇テクノロジーなら可能にする

Xilinx Platform Studio (EDK_0.40)	d) - D:\Xilinx\Vork	space\system.xmp	- [System	Assembly View	d		
📥 File Edit View Project Hardware De	vice C <u>o</u> nfiguration <u>D</u> ebu;	g <u>S</u> imulation <u>W</u> indow	Help				_ 8 ×
: 🔄 🗟 💣 : 🗶 🔍 🧇 : 🗤 🎥	3 3 🔁 🔀 🗄						
Project ↔ 🗆 🗗 🗙	LLP	Bus Interfaces	Ports Ad	lresses		Bus Interface Filters	
Platform Project Files		Name	Bus Name	IP Type 👉 1mb_v10	IP Version 2.00. a	By Connection     V Connected     V Unconnected	
		ilmb mb plb		☆ 1mb_v10 ☆ nlb v46	2.00.a 1.05.a	- By Bus Standard	
<ul> <li>iMPACT Command File: etc/download.cmd</li> <li>Implementation Options File: etc/fast_r</li> </ul>		microblaze_0		microblaze	8.10. a	✓ PLBV46 → ✓ Xilinx Point To Point	
□ bitgen Options File. etc/bitgen.ut □ Elf Files □ microblaze 0		+ dlmb_ontlr		☆ lmb_bram	3.00. a	✓ XIL_BRAM	
- Project Options Device: xc6slx16csg324-3		ilmb_ontlr     imb_DDR2     MCB_DDR2		☆ lmb_bram ☆ mpmc	3.00.a 6.03.a	XIL_MDDEDUG3	
Netlist: TopLevel	<u>.</u>	0_mbm ⊞		🚖 mdm	2.00.Ъ	- By Interface Type	
- HDL: VHDL		clock_generator_U	,	T clock_gen	4.01.a 3.00.o	✓ Slaves	
Sim Model: BEHAVIORAL				M broc_sys	5.00. a	Master Slaves	
						✓ Monitors	
		-				initiators	
	< >	<			>		
<	Legend Master Slave Master Production License Superseded ODisco	er/Slave ⊫Target ≺Init (paid) ĝLicense (eva ontinued	iator Oconne al)	cted OUnconnected ÄPre Production <sup>B</sup>	M Monitor Beta ﷺDevelor	pment	
🔶 Project 🕏 IP Catalog	🗢 Start Up Page	🗵 🗵 Design	Summary [	🔄 📀 Block D	iagram 🗵 🔇	System Assembly View 🔀	
Console	Console ↔ □ ♂ ×						
Zoom In/Out = ALT + (Mouse + Lef Pan = SHIFT + (Mouse + Left Butt	t Button) or ARROW on) or ARROW UP/DOW	UP/DOWN. N/LEFT/RIGHT.					<
							>
📃 Uonsole 🐴 Warnings 💟 Errors							<b>)</b> .::

Ports 項目で、MCB\_DDR2 を右クリック、Configure IP を選択、DDR2 コントローラーをコンフィーグする。



Base Configuration ページでは変更なし。



XPS Core Config - DDH2\_SDHAM - mpmc\_w6\_03\_a HU 📜 😂 Base Configuration. Henory Interface Port Configuration. Advanced Port Type Configuration MPMC/Spartan6 Memory Controller Port0 B32 Port1 Port2 Port3 Port4 832 832 832 832 NA B32 NA B32 B32 MCS Location HENC3 ¥ INSCITVE 💌 INFCLIAE M INACTIVE 🔽 INACTIVE 🕑 INACTIVE 🖓 non Addresses 0x18000000 Slift Register Base Address DIFFEFEFE Base Address The pinset of MPMC must be compatible with MIG Clemory Interface Generator), please see MPMC data about for more information. To create a functional MPMC, use these tabs: Base Configuration: select the port type for each port and remove unused ports between active ports Memory Interface : select the memory part and configure the memory settings Port Configuration: set the required port parameters The Advanced tab contains additional MPMC configuration options and is intended for advanced users only

`Memory Interface`ページで、DDR2の型番を MT47H64M16XX-25Eに設定する。

Memory Part Selector Type DDR2 V Manufac	:turer <b>* V</b> Style <b>*</b>	Densi †	ty \star 💙 Width ★	Fart No.	7H64M16XX-25E
Selected Memory Info					
Part No. MT47H5-	4M16XX-25E Size 128	ИВ Туре	DDR2 Base Address	0x48000000 High Address	0x4FFFFFF
Memory/DIMM Settings Memory	Part Settings   MIG Settings   MCB				
Settings			Configuration	Info	
Number of DIMMs 1	Dynamic Write ODT Setting	OFF 🗸 🗸	X CE Width 1	Memory DM Width	2
Memory Data Width <b>16</b>	Partial Array Salf Refresh	FULL 🗸	X ODT Width 1	Memory Address Width	13
ODT Setting Reseved/50 O	Jum 🔽 Auto Self Refresh	ENABLED 🗸	X Clock Width	Memory Bank Address Width	3
Enable DQSN in DDR2	V High Temp Self Refresh	NORMAL 🔽	X CSn Width 🚺 🗸	Memory DQS Width	2
Reduced Drive Output <b>FULL</b>	Memory Clock Period (ps)	5,000	X No. of Ranks 1	CAS Write Latency	5 👻
Enable Write Leveling			X Registered Memory		

Memory Interface の MCB ページで、MCB ZIO Pin Location は L6、MCB RZQ PinLocation は C2 と設定 する。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする

se Configuration Memory Interface Po	ort Configuration Advan	red				HDL 🎾
lemory Part Selector						
Type DDR2 🛛 Manufacturer *	🖌 Style 🖈	🖌 Density	\star 💉 Width	* 🚩 Part No	Select a	part (22)
Selected Memory Info						
Part No. MT47H64M16XX-25E	Size 128M	B Type	DDR2 Base Addre	ss 0x4800000	High Address	0x4FFFFFFF
Memory/DIMM Settings Memory Part Settin	ngs MIG Settings MCB					
MCB ZIO Pin Location					L6	~
MCB RZQ Fin Location					C2	*
Enable Soft Calibration Logic						TRUE 🗸
Skip Input Termination Calibration						
Skip Dynamic Calibration						
Skip Jynamic Input Termination						
		•••••••				

# 2.1.3 UCF ピン制約ファイルを追加

PortsのExternal Portsに外部信号名をリストし、ソフトウェアは既に自動的にDDR2、ClockとResetの外部信号を設定している、下記図の通り:





低価格、高品質が不可能? 日昇テクノロジーなら可能にする

Xilinx Platform Studio (EDK_0.40d)	- D:\Xilinx\Work	space\s	ystem.xmp - [System Assen	bly View]		
🐟 <u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject H <u>a</u> rdware Device	e C <u>o</u> nfiguration <u>D</u> ebug	; <u>S</u> imulat	ion <u>W</u> indow <u>H</u> elp			_ 8 ×
: 📷 📷 🖬 🖄 💌 🧇 : 🗛 🖬 🖬 🗰 : 🕼	3 🚝 🗄 🛙 🖀 🛃					
IP Catalog ↔ □ 중 ×	😸 Bus Interfaces	Ports	Addresses	(E) 🐢	Port Filters	
	Name		Net Direction	Range Class	By Interface	
Description	😑 External Ports				BUS	
- FDK Trstell	fpga_0_MCB_DD	R2_mebx_ ··	· fpga_O_MCB_DDR… ₩ 0	V [12:0] NONE	By Connection	
+ analog	fpga_0_MCB_DD	R2_mebx_ ··	· fpga_0_MCB_DDR… ₩ 0	✓ [2:0] NONE	Defaults	
+ Bus and Bridge	fpga_0_MCB_DD	0R2_mebx_ **	· fpga_O_MCB_DDR… 🔽 O	NONE	Connected	
+ Clock Reset and Interrunt	fpga_0_MCB_DD	R2_mebx	· fpga_O_MCB_DDR… 💟 O	NONE NONE	- Vnconnected	
+ Communication High=Sneed	fpga_0_MCB_DD	R2_mebx_ ··	· fpga_O_MCB_DDR… 💟 O	NONE NONE	By Class	
+ Communication Low-Sneed	fpga_0_MCB_DD	R2_mebx_ ··	· fpga_O_MCB_DDR… 🔽 O	NONE	Clocks Only	
- DMA and Timer	Epga_O_MCB_DD	R2_mebx	· fpga_0_MCB_DDR… ∨ IO	✓ [15:0] NONE	Clocks	
🚽 🛧 ÁVI Control DMÁ	Epga_0_MCB_DD	R2_mebx_ ··	· fpga_O_MCB_DDR… 💟 IO	NONE NONE	Resets Only	
ART CERTAI DER	tpga_0_MCB_DD	R2_mebx_ ··	· fpga_U_MCB_DDR… V IU	NUNE	Resets	
🔭 🏋 AXI DMA Engine	Epga_0_MCB_DD	R2_mebx_··	· fpga_O_MCB_DDR… V O	NONE NONE	Interrupts Only	
🚽 🙀 AXI FIFO Memory Mapped To Streamij	Epga_0_MCB_DD	R2_mebx_ ··	· fpga_O_MCB_DDR… ▼ O	NONE NONE	- V Interrupts	
🚽 📩 AXI Watchdog Timer	fpga_0_MCB_DD	R2_mebx_··	· fpga_U_MCB_DDR··· V U	NUNE	🗸 🗸 🗸 🗸 🗸 🗸	
- 👉 AXI Timer/Counter	fpga_0_MLB_DD	K2_mebx_ ··	· fpga_U_MCB_DDR···· V U	NUNE	😑 By Direction	
	tpga_0_MLB_DD	KZ_mebx_ ··	· fpga_U_MCB_DDR···· V IU	NUNE	🖌 🔽 Inputs	
AT AIGO DWY	fpga_U_MLB_DD	W2_mebx	· fpga_U_MCB_DDK··· V 10	NUNE	🖉 🗹 Outputs	
🕆 🏋 Fixed Interval Timer	fpga_0_MLB_DD	KZ_mebx_··	· ipga_U_MCB_DDK··· V U	NUNE	🖾 🗹 InOuts	
🚽 🧙 XPS Central DMA Controller	tpga_U_oik_1_	SAZ GTK	· LTTV 2	ULK DOW		
🕂 🕁 XPS Watchdog Timer	I Ipga U rst 1	sys_rst_"	· sys_rst_s 🛛 💌 1	NSI J		
🔶 XPS Timer/Counter	+ armp					
H- Debug	I at all					
FPG4 Reconfiguration	· · · · · · · · · · · · · · · · · · ·					
+ General Purpose IO	Tub base_U					
+ IO Modules						
+ Interprocessor Communication				×		
Memory and Memory Controller	<			>		>
+ PCI	Legend					
+ Peripheral Controller	Master Slave M	acter/Slav	e 🕨 Target 🤇 Initiator 🍋 Connected	Olloconnected M Monitor		
+ Processor V	Due duction Olicor	ascer, sidv		Bus dustion BRots Broweler	amant	
	Production Dilicer	ise (paid)	Ucense (eval) QLocal AP	re production #3 Deta pa Develo	prineric	
	Superseded OD	Iscontinue				
🗇 Project 🛷 IP Catalog	😎 Start Up Page	<u> </u>	Design Summary 🔝 🧇 Bl	ock Diegram 🔛 📀 System A	ssembly View 🔀	
Console						⇔□₽×
						~
						~
<						>
Uonsole						<b>)</b> .:

手動で DDR2 の rzq と zio 信号を外部信号と設定、MCB\_DDR2 の示すのドロップダウンメニューで Make External を選択:

😹 Bus Inter	rfaces Ports	Addresses			
Bane		Net		Direction	n Range 🗠
- nc	:bx_dran_ba	fpra_0_MCB_DD	2 ncbz cran ba pin	✓ 0	[C_NEW_BANEA]
no	abx dram ras n	fpga O MCB 10	2 nobz dran ras n p	in 💙 0	
no	obz_dran_cas_n	fpgs_0_MCB_EDS	82_nobx_dran_cas_n_p	in 🔽 O	
nc	bx_dram_ve_n	fpga_0_MCB_DD	😢 ncbx_dran_ve_n_pi	n 🔽 O	
- nc	:bx_dran_cke	fpga 0 MCB 10	🔼 nebz_dran_eke_piz	. 🗸 0	
no.	bx_dram_clk	fpga 0 MCB DDB	2 nobz dran olk piz	. 💙 0	
no	ebx_dran_elk_n	fp ga_0_MCB_DDB	82 ncbx_dran_elk_n_p	in 🔽 O	
nc	bz_dran_dq	fpgs_0_MCB_DDS	32_ <mark>nc</mark> bx_dran_dq_pin	🔽 I0	[C_NEW_DATA_I
nc	:bx_dran_dqs	fpra 0_MCB_DDB	82_ <mark>mc</mark> bz_dran_dqs_pin	. 🗸 10	
no.	sbx_dran_dgs_n	tpgs 0 MCB IDE	82 nobz dran dgs n p	in 💙 10	
no	zbw_dwam_udqs 🚽	fp ga 0 MCB_DDB	RZ_nebx_dran_udqs_pi	n 💙 IO	
nc	n_zpbr_mewb_xds	fpgs_0_WCB_DDB	32_nebx_dran_udqs_n_	pin 🛛 🔽 IO	
nc	:bx_dran_udn	fpcs_0_MCB_DDB	82_ncbz_dran_udm_piz	✓ 0	
no.	:bx_dran_lon	fpra O MCB IDE	82_nobz_dran_ldm_piz	. 💙 0	
no	ebx_dran_odt	fpga_0_MCB_DDB	RZ_nebx_dran_odt_piz	. 🛛 👽 🛛	
- rz	24	IDR2_SDRAM_YIC	1	V IO	
- zi	0	DUR2_SDRAM_zic	0	V I0	
🕒 Debug ac	attaire				
🗄 - Jaterrep	t				
🗄 - I therne t	10_100				-
🔒 - Buttons_	48it				
🕀 LEDS_4Bi	t				
🕒 🛛 adi 🤊 🖽	र 📃				
⊞- 352					
<u>₽-50_5P</u>					~
<	1				>
a second s					

DDR2\_SDRAM\_zio と DDR2\_SDRAM\_rzq は自動的に External Ports リストに追加される:



Bus Interfaces Forts Addresses					
la-	H=1	I	Direction	Range	
- Tyternal Ports					
DDR2 SARAM rie	DDB2_SDBAN_zio	- v 1	0	×	
- ANR SDRAM reg	DDR2_SDRAW_rrq	~ I	D C	N I	
type O M.S. DORA acht armo addr pin	ipga U_VCB_DUR2_nebs_dram_addr_pin	V 1	1	V 112:07	
<ul> <li>Epsa_0_MCB_DBR2_orebx_draw_ba_prin</li> </ul>	fpga_0_WCB_DDR2_webx_dram_ba_pin	× 0	)	\$2:03	
- fora_C_MCS_DOR2_order_area_era_aria	fpga_0_WCB_DDR2_wcba_draw_cas_a_pia	~ 0	1	*	
- Epgs_O_MCB_DDE2_acbs_drue_cke_pis	fpgs_0_VCB_DDR2_nebs_dram_cks_pin	· · · 0	1	~	
- Loga O M.B. MAR2 webx draw alk a sin	fpga_0_WCB_BDR2_webx_draw_clk_s_pis.	× (	)	V	
- Lora C. M.B. DORZ ache drow alle pin	fpgs_0_VCB_DDR2_nebx_dran_clk_pic	~ 0	1	~	
- Epgs O MCB DDE2 achs drug dy pin	fpga 0 WCB DDR2 mebs dram dq pin.	V 1	10	V [15:0]	
- Lora O M.B DOR2 webx dree dos a sin	fpga 0 WCB DDR2 meba dran des a pia	V 1	0	V	
- Loro O M.S. DOR? ache area dos pis	fpga_0_VCB_DDR2_acba_dran_dgs_pin	~ I	10	*	
- Epgs O MCB DDEP acht drug 1 do pis	fpga 0 WCB DDR2 mebs dram 1 de pin	¥ 0	1	~	
- Loga O MCB DOR2 webx draw odt pix	fpgu 0 WCB DDR2 meba dram odt pin	~ 0	1	×	
- fare 0 MCB DDR2 webs drew res a pin	fpga 0 WCB DDR2 ocbs dran ras a pin	~ 0	1	~	
- Epps O MCB DDE2 achs drug ado pis	fpga 0 WCB DDR2 mebs dram uds pin	¥ 0	1	×	
- fora O MCB DOR2 webx draw udes a pin	foga 0 WCB DDR2 nebu dran udes a pia	× 1	0	×	
form 0 MCB DDR2 mehr drew udgs pin	fors 0 WCB DDR2 ochr dran oder pin	~ I	0	×	
Fors O MCB DDE2 acbs drup av a pin	fors 0 WCB BDR2 mebr dram we n pin	~ 0	1	×	
- forg 0 elk 1 sys elk pip	CLR S	× 1		V	

Project 中の UCF ファイルをダブルクリックし、ピンと制約を定義する:

🔶 Xilinx Platform Studio (EDK_0.40d) -	D:\Xilir
📄 Eile Edit View Project Hardware Device	Configuratio
i 🗊 🗑 🗗 i 🖄 🛛 🧇 i 🗛 🔛 i 🛤 🏭	<u>第</u> 1 王 1
Project ↔ 🗆 🗗 🗙	23
Platform	24
🖃 Project Files	25
MHS File: system nbs	26
- IMFAUT Command File: etc/downLoad.cmd	
Implementation Options File: etc/fast_runti Bitzer Options File: etc/hitzer ut	29
■ Elf Files	30
⊕ microblaze_0	31
Device: xc6slx18csg324-3	32
Netlist: TopLevel	34
- Implementation: XFS (Xflow) - HDL: VHDL	35
Sim Nodel: BEHAVIORAL	🗏 36
- Design Summary	37
UCF ファイルに下記の信号完美を追加し、保存す	ろ DDR2 の他ピンけ Spartan6 シリーズでけ固定された
ため、完美ナス以西ね1	
<pre>1 ## 2 Net fpga_0_clk_1_sys_clk_pin LOC = V10   TNM_1</pre>	NET = sys_clk_pin;
<pre>3 TIMESPEC TS sys_clk_pin = PERIOD sys_clk_pin ! 4 ##</pre>	50000 kHz;
5 NET fpgs_0_rst_1 sys_rst_pin	LOC = N4   IOSTANDARD = "LVCMOS18"; ## SW2 pushbutton
6 ##	
8 ########DDR2 Pin define###################################	
9 NET MCB_DDR2_rzq 10 NET MCB_DDR2_zio	LOC = C2   IOSTANDARD = SSTL18_II ; ## rzq: C2 on U2 LOC = L6   IOSTANDARD = SSTL18 II ; ## zio: L6 on U2
-	—

### 2.1.4 コンパイルとダウンロード

メニュー Device Configuration -> Download Bitstream クリック、プログラムは bit ファイル 生成し、FPGA にダウンロードする。



低価格、高品質が不可能? 日昇テクノロジーなら可能にする



SDK 起動。EDK で Project→Export Hardware Design to SDK…選択

🙆 Errors

🚺 Warnings

Console





SDK ソフトウェアでダイアログボックス起動、Workspaceの保存フォルダを入力する:

🖤 Torkspace Launcher	
Select a workspace	
Xilinx SDK stores your projects in a folder called a workspace. Choose a vorkspace folder to use for this session.	
Workspace: D. \Xilinx\Workspace\SDK	Brovse
☐ ∐se this as the default and do not ask again	
	OK Cancel



次の章 Hello World アプリケーションプログラムの開発に入る。

# 2.2 Hello World アプリケープログラム開発

SDK ウィザードを使用し、Hello World プロジェクトを作成、順次に File→New→Xilinx C Project 選択。



Hello World テンプレート使用、プロジェクト名を Hello\_world に変更



🚳 New Project		
New Xilinx C Project Create a managed mak	e application project. Choose from one of the sample applications.	
<u>P</u> roject name: hello	world	
✓ Use default loca	tion	
Location: D:\Xilinx	Workspace\SDK\hello_world Browse	
Choose fil	e system: default 💌	
Target Hardware —		
Processor:	microblaze 0	
Select Project Tem Dhrystone Empty Application Hello World IwIF Echo Server Memory Tests Peripheral Tests SREC Bootloader	2 Description Let's say 'Hello World' in C.	
?	< Back Next > Einish Cancel	

初めてプロジェクト作成なので、同時に前章で作成したソフトコアを基にライブラリファイルを作成する 必要。名前は siga\_bsp。

🐵 New Project 📃 🗖 🔀
New Xilinx C Project Create a managed make application project. Choose from one of the sample applications.
⊙Create a new Board Support Package project
The template provided by application Hello World' will be used to configure the project.
Project name: siga_bsp
Vse default location
Location: D:\Xilinx\Wothspace\SDK\siga_bsp Browse Showse file symptom: default 💌
OTarget an existing Board Support Package Available Board Support Packages:
No Board Support Packages found
(?) ( <u>Back</u> Mext > <u>Finish</u> Cancel

プロジェクト作成完成後の SDK 操作画面:



低価格、高品質が不可能?

日昇テクノロジーなら可能にする



SDK ソフトウェアはプロジェクト保存の時自動的コンパイルする。プロジェクトを右クリックし、Build Project 選択でもコンパイルできる。コンパイル完成後、console ウィンドウで下記の情報が表示する:



Hello Worldアプリのデバッグを開始する。Run→Debug Configurations.. 選択、設定ウィンドウに入る:



🐵 Debug Configurations	
Create, manage, and run cor	nfigurations
type filter text C/C++ Application C hello_world Debug C C/C++ Attach to Appli hello_world Debug C C/C++ Postmortem Debu; Launch Group Xilinx C/C++ ELF	Configure launch settings from this dialog: - Press the 'New' button to create a configuration of the selected type. - Press the 'Duplicate' button to copy the selected configuration. - Press the 'Delete' button to remove the selected configuration. - Press the 'Itter' button to configure filtering options. - Edit or view an existing configuration by selecting it. Configure launch perspective settings from the <u>Perspectives</u> preference page.
Filter matched 7 of 7 items	
?	Debug Close

Xilinx C/C++ ELF をダブルクリック、Xilinx C/C++ ELF ディレクトリ下で hello\_world Debug ファイ ルが表示する:



🐵 Debug Configurations					
Create, manage, and run configura	ntions				Ť
type filter text C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger Launch Group Xilinx C/C++ ELF Kilinx C/C++ ELF Kello_world Debug	Name: hello_world Debug	itialization 🗞 STDIO Con	nnection Remote Deb	ug 😰 Debugger Optio	ns Common Browse Browse
Filter matched 6 of 6 items				Apply	Re <u>v</u> ert
0				Debug	Close

STDIO Connection タブで JTAG UART を STDIOの入力出力インタフェースに設定:

Debug Configurations		
Create, manage, and run configura	tions	To-
type filter text C/C++ Application C hello_world Debug C C/C++ Attach to Application C hello_world Debug (1) C C/C++ Postmortem Debugger Launch Group Xilinx C/C++ ELF hello_world Debug (2)	Hame i hello world Debug (2) (2) Main & Source & Device Initialization & S Connect STDIO to Console Port: TTAG UART BAUD Rate: 9800	TDIO Connection 🕱 Remote Debug 🕱 Debugger Options 🔲 Common
Filter matched 8 of 8 items		Apply Reyert
?		Lebug Close



Debug をクリックしてデバッグ開始。SDK 開発環境で FPGA に bitstream ファイルダウンロードしてない状態を提示するダイアログボックスが表示する。EDK で既に bitstream ファイルをダウンロードしたため、Yes をクリックし、デバッグ操作画面が表示する。



メニューRun->Resume で実行する。成功すると、Console ウィンドウで Hello World が表示する。初めての Microblaze のプログラムが成功。



#### 3.1 序文

Xilinx Spartan6はSPIのコンフィーグモードをサポートする。SPIコンフィーグモードはハードウェア を簡略化され、コストを削減する同時にSPI チップの容量が大い、コンフィーグファイルを保存する以外 に他のデータ保存にも使われ、ユーザーアプリケーションの範囲を拡張に利用できる。 Siga-S16 開発ボード上に 8Mbit の SPI FLASH を搭載している、型番は M25P80。

#### 3.2 事前準備

▶ ISE13.1 バージョンまたはこれ以上のバージョン、本実験は ISE13.1 で行う。



▶ Siga-S16 開発ボードと USB JTAG エミュレータ

▶ Siga-S16 開発ボードの DIP スイッチの設定は MO:ON、 M1:Off、MASTER SPI MODE

#### 3.3 SPI FLASHの書き込み

#### 3.3.1 iMPACT 起動

スタート->全てのプログラム->Xilinx ISE Design Suite 13.1->ISE->Accessories->iMPACT、次の画面が 表示される:





Telcome to iMPACT	
Please select an action from the list below	
○ Configure devices using Boundary-Scan (JTAG)	
Automatically connect to a cable and identify Boundary-Scan chain 👻	
Prepare a PROM File	
O Prepare a System ACE File	
O Frepare a Boundary-Scan File	
24L	

# 3.3.2 mcs ファイル作成

mcs ファイル生成する前に、SPI FLASH の型番と容量を確認する必要がある。PROM File Formatter インタフェースで下記図の通りに設定する:

🐉 PROM File Formatter							×
Step 1. Select Storage Target	Step 2. Add St	orage Device(s)		Step 3.	E	nter	Data
Storage Device Type :	Storage Dewice (hits)	81		heral File Deta	Valu	e	
Xilinx Flash/PROM Non-Volatile FPGA	Add Storage Device R	emove Storage Device		Checksum Fill Value	FF		
Spartan3AN	8		(	Output File Name	Test		
Configure Single FPGA				Output File Location	D:\Xilinx\		Þ
Configure Single FPGA Configure MultiBoot FPGA			<u> </u>	Flash/PROM Fil	Le Property	Value	
Configure from Paralleled PROMs			-	File Format	MCS		~
				Add Non-Configura	ation Data Files No		*
	Auto Select PROM						
Description:							
<ul> <li>In this step, you will enter information to assist :</li> <li>Checksum Fill Value: When data is insuffichecksum of the unused portions.</li> <li>Output File Hame: This allows you to specion utput File Location: This allows you to File Format: PROM files can be generated in</li> </ul>	in setting up and generat zient to fill the entire fy the base name of the s specify the directory in any number of industry :	ing a PROM file for the nemory of a PROM, the w file to which your PROM which the file named at standard formats. Depen	e targe alue s data u oove wi ding oj	ted storage devi pecified here is will be written 11 be created n the PROM file OK	ce and mode. : used to calculate format your PROM p Cancel	e the rogramm Ho	elp

OK をクリック、Add Device ファイルの情報が表示する、直接 OK をクリック。





次のの Add Device ファイルのウィンドウで前の ISE で生成した led\_test.bit ファイルを選択、"Open" をクリック。次に表示される Add Device ファイルのウィンドウで、No を選択。





# Generate Succeeded

### 3.3.3 MCS ファイルを SPI FLASH に書込む

iMPACT Flows 操作画面の Boundary Scan 項目をダブルクリック:

iMPACT Flows	
Boundary Scan	
SystemACE	
🚽 🖃 📄 Create PROM File (PROM File Formatter)	
🗄 🗎 WebTalk Data	
Boundary Scan 画面で右クリックし、 Initializ Right click to Add Device or Initialize JTAG chain	ze Chain を選択:
Add Xilinx Device Ctrl+D Add Non-Xilinx Device Ctrl+K	
Initialize Chain Ctrl+T	
Cable Setup	
Qutput File Type	
ITAG チェン上の FPGA チップを検出した場合、下 Right click device to select operations	No を選択。
Stute Assign Configuration Hiles Overs	lialog 🛛
Whatto Assign Configuration Files query i	
Do you want to continue and assign configurat	ion files(s)?
Tez Wo	
	Identify Succeeded



次に下記画面で OK をクリック。



SPI/BPI インタフェースをダブルクリック(赤い枠の部分)、SPI FLASH のコンパイルファイルを追加。



ポップアップした Add PROM File ウィンドウで先ほど生成した Test.mcs ファイルを選択、オーペン。次 にポップアップした Select Attached SPI/BPI インタフェースで正確な SPI FLASH 型番 M25P80 を選択、OK をクリック。





FPGA の SPI インタフェースで FLASH の標示が表示する:



次の Device Programming Properties ウィンドウで直接 OK クリック、ソフトウェアは自動的に SPI FLASH に書込む。





書き込み完了:





