

Xilinx Spartan3 シリーズ

XC3S400 開発ボードマニュアル

株式会社日昇テクノロジー

http://www.csun.co.jp

info@csun.co.jp

2010/10/30



copyright@2010



• 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2010/10/30

- ※ 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。
 「<u>http://www.csun.co.jp</u>」
- ※(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。





第一章 SP3S400 ボードの概要	5
1.1 仕様概要	5
1.2 外形寸法	6
1.3 商品内容	6
第二章 ハードウェア仕様	8
2.1 SWキー	8
2.2 ユーザーキー	8
2.3 ユーザーLED	9
2.4 7SegLED	9
2.5 ブザー1	0
2.6 VGAインタフェース1	1
2.7 PS/2 インタフェース1	2
2.8 RS232 シリアルポートインタフェース1	3
2.9 LCD液晶インタフェース1	4
2.10 USBインタフェース1	7
2.11 高速非同期SRAM1	8
2.12 高速同期SDRAM2	0
2.13 大容量、快速Flash 2	2
2.14 拡張アドレスバス2	4
2.15 拡張データバス2	5
2.16 拡張ユーザI02	7
2.17 JTAGインタフェース2	8
2.18 電源回路2	9
2.19 リセット回路2	9
2.20 クロック2	9
第三章 初体験	1
3.1 プロジェクトの起動	1
3.2 ボードに書き込み	1
3.2.1 JTAGモードでダウンロード3	1
3.2.2 PROMモードで書き込む3	6
第四章 開発基本手順 4	5
4.1 Project Navigatorの起動4	5
4.2 プロジェクトの新規作成 4	6
4.3 新規作成プロジェクトにソースファイルの追加 4	9
4.4 合成及びシミュレーション5	3



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

4.	4.1	合成	53
4.	4.2	シミュレーション	55
4.5	Const	traintsの編集	61
4.6	実行.		62
4.7	ボー	ドにダウンロードして実行	62

株式会社日昇テクノロジー



第一章 SP3S400 ボードの概要

Xilinx 社最先端の 90nm プロセスを使用した Spartan ソリューションである Spartan-3 FPGA プラットフォーム、40 万システムゲートの XC3S400 マイコンを採用しております。 本ボードの主な特徴:

- 1、Xilinx 開発プラットフォーム、MicroBlaze ソフトプロセッサを使用することもで きます。
- 2、480MbpsのハイスピードのUSB2.0データインタフェースも提供しております。

3、多機能のインタフェース、拡張インタフェースを提供しております。

Xilinx FPGAのスタートキット、設計研究キットとしてお勧めです。

1.1 仕様概要



1、Xilinx Spartan3 シリーズ XC3S400

下記五つのモジュールで構成されています:CLB(コンフィギュラブル ロジック ブロック)、 IOB (SelectIO) 、BlockRAM、Multiplier (乗算器)、DCM (デジタル クロック マネジメン ト)。

CLB: RAM 基づいた LUT を含めて、ロジックとメモリを実現する。一つの CLB で4つの連動の Slice を含めている。

- (2) IOB:外部 I/O ポートとデバイス内部ロジック間のデータフローを制御する。三つの 信号通路がある:インプット通路、アウトプット通路、トライステート通路。
- (3) BlockRAM:完全デュアル ポートの 18Kb ブロック RAM
- (4) Multiplier:二つの 18 ビットのバイナリデータを入力可、16 ビットの結果を計算 して出力する。
- (5) DCM: 遅延ロックループ、デジタル周波数合成、位相シフト、ステータスロジック、四つの機能がある。

本ボード採用しているマイコン XC3S400 は PQ208 パッケージで、デバイス特性は下記:

System Gates :	400K
Equivalent Logic Cells:	8,064

Jquivarent	LUSIC	00110	•	0,00

Distributed RAM Bits: 56K

Block RAM Bits: 288K

Dedicated Multipliers: 16

DCMs:

- Maximum User I/Os: 141
- 2、XCF02S、コンフィグデバイス、PROM モードのダウンロード、JTAG モードもサポートする。
- 3、512KB 高速 SRAM、256K x 16Bit
- 4、8MB 高速 SDRAM、4M x 16Bit、最高 166MHz のスピード
- 5、2MB FLASH、2M×8Bit、アクセススピードは90ns
- 6、RS-232 シリアルポート、DB9 メス
- **7、VGA インターフェース**、512 色
- 8、7SegLEDx4、ユーザLEDx8
- 9、1602 LCD 液晶インタフェース
- 10、PS/2インターフェース、3.3vと5vのデバイスをサポートする。

4

- 11、USB2.0 高速データインタフェース、CY68013 で実装している
- 12、ユーザシリアルインタフェース
- 13、拡張インタフェース、Video、AUSB_Dio、Wireless などの実現

1.2 外形寸法

153×106(mm) ※突起物は除く

1.3 商品内容

- 1、SP3S400 開発ボード
- 2、LPT ダウンロードケーブル
- 3、1602LCD 液晶モジュール





- 4、5V/1A 電源
- 5、シリアルケーブル
- 6、USB ケーブル



第二章 ハードウェア仕様

2.1 SWキー

二つの SW キーを提供している (SW7)。

回路図:

	R80 4.7K	PB0 PB1	SW7	
	R81 4.7K		SW DIP-2	
				_
			.	
ピンマップ:				
Switch		PB0 (下)	PB1 (上)	
FPGA Pin		Pin 50	Pin 48	

ONの位置にすると、PB信号はLowになる。

2.2 ユーザーキー



四つのユーザーキーを提供している (SW3~SW6)。 回路図:

ピンマップ:

不可能への挑戦	株式会社日	昇テクノロシ	・ 、 日昇デ	低価格、高品質が不可能? クノロジーなら可能にする
Push Button	SW6	SW5	SW4	SW3
FPGA Pin	Pin 51	Pin 52	Pin 57	Pin 58

2.3 ユーザーLED

8つのユーザーLEDを提供している (LED1~LED8)。

回路図:

VD8	R53 510	LED1	GREEN
VD7	R54 510	LED2	GREEN
VD6	R55 510	LED3	GREEN
VD5	R56 510	LED4	GREEN
VD4	R57 510	LED5	GREEN
VD3	R58 510	LED6	GREEN
VD2	R59 510	LED7	GREEN
VD1	R60 510	LED8	GREEN
			!

ピンマップ:

デバイス	信号名称	FPGAビン番号
LED1	VD8	Pin 204
LED2	VD7	Pin 203
LED3	VD6	Pin 200
LED4	VD5	Pin 199
LED5	VD4	Pin 198
LED6	VD3	Pin 197
LED7	VD2	Pin 196
LED8	VD1	Pin 194

2.4 7SegLED

4 つの 7SegLED を提供している。 回路図:



ピンマップ:

不可能への挑戦

信号名称	FPGAビン番号	信号名称	FPGAビン番号
VD1	Pin 194	VD7	Pin 203
VD2	Pin 196	VD8	Pin 204
VD3	Pin 197	SEG_LED_S0	Pin 5
VD4	Pin 198	SEG_LED_S1	Pin 4
VD5	Pin 199	SEG_LED_S2	Pin 3
VD6	Pin 200	SEG_LED_S3	Pin 2

2.5 ブザー

一つのブザーを提供している。クロックの周波数の変更で各種の音楽を生成できる。 回路図:



高品質が不可能



信号	名称
信号	名称

FPGAビン番号

SP

Pin 15

2.6 VGAインタフェース



回路図:



制御信号:VD1(Red_L), VD2 (Red_M), VD3(Red_H), VD4(Green_L), VD5(Green_M), VD6(Green_H), VD7(Blue_L), VD8(Blue_M), VD9(Blue_H), VGA_HS(Horizontal Sync), VGA_VS(Vertical Sync)。

各色の線は、高、中、低の制御線があって、100、300、1Kの抵抗を接続してFPGAのIOと繋 がっている。VGAインタフェースは赤、緑、青三色線があるので、9つの制御線がある。論 理的に29=512色が表示できる。

VGA_HSとVGA_VS信号はTTLレベルでVGAの水平と縦のスキャン信号を制御する。





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

ピンマップ:	
Signal	FPGA Pin
VD1	Pin 194
VD2	Pin 196
VD3	Pin 197
VD4	Pin 198
VD5	Pin 199
VD6	Pin 200
VD7	Pin 203
VD8	Pin 204
VD9	Pin 205
VGA_HS	Pin 7
VGA_VS	Pin 9

2.7 PS/2 インタフェース

PS/2 DINインタフェース:



回路図:





ピンマップ:

Signal	FPGA Pin
PS2DK	Pin 12
PS2CK	Pin 13
PS2DM	Pin 10
PS2CM	Pin 11

2.8 RS232 シリアルポートインタフェース

二つのRS232インタフェースを提供している (DB9と3Pin拡張ピン)。同時に送信、受信できる。

DB9インタフェース図:



(front view)

回路図:





データ転送時、対応しているLEDが点灯する。

2.9 LCD液晶インタフェース

回路図:



ピンマップ:

Device	Signal	FPGA Pin
JP9-4	LCD_RS	Pin 93
JP9-5	LCD_RW	Pin 90
JP9-6	LCD_E	Pin 87
JP9-7	LCDO	Pin 22
JP9-8	LCD1	Pin 24
JP9-9	LCD2	Pin 26
JP9-10	LCD3	Pin 27
JP9-11	LCD4	Pin 28
JP9-12	LCD5	Pin 29
JP9-13	LCD6	Pin 31
JP9-14	LCD7	Pin 33
説明:		

Pin1、GND

Pin2、VSS 5V電源

Pin3、VOは液晶のコントラスト調整、電源入れる時一番弱くてGNDと接続する時は一番高い。



Pin4、RSはResister Select、Highレベルの時はデータレジスト、Lowレベル時は指令レジ ストを選択する。

Pin5、RWはRead/Write信号ライン、Highレベル時はRead操作、Lowレベル時はWrite操作。 Pin6、EはEnable信号、HighからLowに変換する時、液晶モジュールがコマンドを実行する。 Pin7~14、D0~D7、8bit双方向データライン。

1602液晶モジュール内部のCGROMに160個のドットマトリックスの文字やグラフィックを保存している。

廣位	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
××××0000	CGRAM (1)		0	ə	Р	1	р		-	9	Ξ	a	Р
××××0001	(2)	!	1	A	Q	a	q	П	7	7	4	ä	q
××××0010	(3)		2	В	R	b	r	Г	1	川	1	β	θ
×××××0011	(4)	#	3	C	S	c	5	J	ウ	7	ŧ	E	00
×××××0100	(5)	\$	4	D	Т	d	t	1	r	h.	セ	μ	0
×××××0101	(6)	%	5	E	U	e	u	D	*	+	2	В	0
××××0110	(7)	8.	6	F	v	ſ	v	Ŧ	7	-	3	Р	Σ
×××××0111	(8)	>	7	G	W	g	w	7	*	X	7	g	π
××××1000	(1)	(8	Н	x	h	x	1	7	*	IJ	1	X
××××1001	(2))	9	I	Y	i	у	7	4	10 23	N	-1	y2
××××1010	(3)		1	J	Z	j	z	I	3	1)	V	· j ·	千
××××1011	(4)	+	1	K]	k	1	*	#	E	D	x	万
××××1100	(5)	7	<	L	¥	1	I	セ	2	7	7	e	A
××××1101	(6)	-	-	M]	m)	2	ス	2	148	ŧ	+
××××1110	(7)		>	N	-	n		3	セ	ホ	~	n	
××××1111	(8)	1	?	0	-	0	+	"	y	7	D	ö	10000

コマンド表:



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

Instruction	DE	DAW	DP7	DRC	C	DRA	DP2	DP2	DP4	DRO	- Description	Execution Time (max) (when fop or
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	108C IS 270 KHZ)
Return home	0	0	0	0	0	0	0	0	1	_	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 µs
Display on/off control	0	0	0	0	0	0	1	D	С	В	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 µs
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	-	_	Moves cursor and shifts display without changing DDRAM contents.	37 µs
Function set	0	0	0	0	1	DL	N	F	-	_	Sets interface data length (DL), number of display lines (N), and character font (F).	37 µs
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 µs
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 µs
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μs
Write data to CG or DDRAM	1	0	Write	e data							Writes data into DDRAM or CGRAM.	37 µs t _{ADD} = 4 µs*
Read data from CG or DDRAM	data 1 1 Read data i CG or () M						Reads data from DDRAM or CGRAM.	37 μs t _{ADD} = 4 μs*				
$\begin{tabular}{lllllllllllllllllllllllllllllllllll$							DDRAM: Display data RAM CGRAM: Character generator RAM ACG: CGRAM address ADD: DDRAM address (corresponds to cursor address) AC: Address counter used for both DD and CGRAM addresses	Execution time changes when frequency changes Example: When f_{co} or f_{osc} is 250 kHZ, $37 \ \mu s \times \frac{270}{250} = 40 \ \mu s$				

Note: -indicates no effect.

> * After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, $t_{\mbox{\tiny ADD}}$ is the time elapsed after the busy flag turns off until the address counter is updated.

もっと詳しい内容は下記資料をご参照ください:

http://www.dragonwake.com/download/LCD/hd44780u.pdf



2.10 USBインタフェース

Cypress 社の CY68013 マイコンを採用している。 回路図:



データ信号:

CY7C18013 Pin	Signal	FPGA Pin
PB/FD0	USB_D0	Pin 116
PB/FD1	USB_D1	Pin 115
PB/FD2	USB_D2	Pin 114
PB/FD3	USB_D3	Pin 113
PB/FD4	USB_D4	Pin 101
PB/FD5	USB_D5	Pin 100
PB/FD6	USB_D6	Pin 97



PB/FD7	USB_D7	Pin 96
PB/FD8	USB_D8	Pin 65
PB/FD9	USB_D9	Pin 64
PB/FD10	USB_D10	Pin 63
PB/FD11	USB_D11	Pin 62
PB/FD12	USB_D12	Pin 61
PB/FD13	USB_D13	Pin 120
PB/FD14	USB_D14	Pin 119
PB/FD15	USB_D15	Pin 117
コントロール信号:		
CY7C18013 Pin	Signal	FPGA Pin
RDY0/SLRD	USB_SLRD	Pin 86
RDY1/SLWR	USB_SLWR	Pin 85
CTLO/FLAGA(pf半分)	USB_FLAGA	Pin 95
CTL1/FLAGB(full)	USB_FLAGB	Pin 94
CTL2/FLAGC(empty)	USB_FLAGC	Pin 78
PA7/FLAGD	USB_FLAGD	Pin 67
PA2/SLOE	USB_SLOE	Pin 74
PA6/PKTEND	USB_PKTEND	Pin 68
PA4/FIFOADR0	USB_FADDR0	Pin 72
PA5/FIFOADR1	USB_FADDR1	Pin 71
CLKOUT	USB_CLK	Pin 184
IFCLK	USB_IFCLK	Pin 80
もっと詳しい内容はUSBの	データシートをご	参照ください。

2.11 高速非同期SRAM

ISSI社の高速非同期SRAMを採用している。容量は256K×16Bit。Enable信号はByte毎に独立 なので、Byte毎に操作できる。 回路図:



アドレス信号:

SRAM Pin	FPGA Pin	External Bus Signal
AO	Pin 190	SA1
A1	Pin 189	SA2
A2	Pin 187	SA3
A3	Pin 185	SA4
A4	Pin 183	SA5
A5	Pin 182	SA6
A6	Pin 181	SA7
A7	Pin 180	SA8
A8	Pin 178	SA9
A9	Pin 176	SA10
A10	Pin 175	SA11
A11	Pin 172	SA12
A12	Pin 171	SA13
A13	Pin 169	SA14
A14	Pin 168	SA15
A15	Pin 167	SA16



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

A16	Pin 166	SA17
A17	Pin 165	SA18
データ信号		
SRAM Pin	FPGA Pin	External Bus Signal
sram_db[0]	Pin 102	SD0
sram_db[1]	Pin 106	SD1
sram_db[2]	Pin 107	SD2
sram_db[3]	Pin 108	SD3
sram_db[4]	Pin 109	SD4
<pre>sram_db[5]</pre>	Pin 111	SD5
sram_db[6]	Pin 122	SD6
sram_db[7]	Pin 123	SD7
sram_db[8]	Pin 124	SD8
<pre>sram_db[9]</pre>	Pin 125	SD9
<pre>sram_db[10]</pre>	Pin 126	SD10
<pre>sram_db[11]</pre>	Pin 128	SD11
<pre>sram_db[12]</pre>	Pin 130	SD12
sram_db[13]	Pin 131	SD13
<pre>sram_db[14]</pre>	Pin 132	SD14
<pre>sram_db[15]</pre>	Pin 133	SD15
コントロール信号		
SRAM Pin	FPGA Pin	External Bus Signal
CE#	SRAM_ CE#	Pin 140
OE#	SRAM_ OE#	Pin 139
WE#	SRAM_ WE#	Pin 141
BEO	SRAM_ BEO	Pin 137
BE1	SRAM_ BE1	Pin 138

2.12 高速同期SDRAM

回路図:



		UD4
S 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	οż.	
	23	A0 \sim
	24	A1 a a a a a a DQ1 4 SD2
CA2	20	A2 * * * * * * DQ2 7 SD3
	20	A3 A A A A A A A DQ3 A SD4
	28	A4 DQ4 0 SD5
	30	AS A A A A A A DQS 44 SDB
	31	A6 DQ6 42 SD7
	32	A7 DQ7 42 SD9
	33	A8 DQ8 42 SD0
	34	A9 DQ9 44 SD9
SA10	-22	A10/AP
	35	A11 DQ11 47 SD11
and a second second second second second	A	DQ12 50 SD12
SUCLK	38	CLK 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	·	DQ14 51 SD14
<u></u>	20	BA0 0 0 0 0015 53 SUIS
<u></u>	21	BA1 CONTRACTOR AND
	·	DOM 15 SULDUM
SUCAS	17	CAS CAS CONTRACTOR LIDOM 39 SULUDOM CAS
SUCKE	37	
SU_RAS	18	RAS NC1 40
SD_WE	16	1AF
	19	
3.3V		
.u∓	1	
	14	
	27	
	3	VDDO COND 41
	9	VDDQ
	43	
	49	VDDQ GND 54
		HY57V561620HGT-H

アドレス信号:

SDRAM Pin	FPGA Pin	External Bus Signal
AO	Pin 191	SAO
A1	Pin 190	SA1
A2	Pin 189	SA2
A3	Pin 187	SA3
A4	Pin 185	SA4
A5	Pin 183	SA5
A6	Pin 182	SA6
A7	Pin 181	SA7
A8	Pin 180	SA8
A9	Pin 178	SA9
A10	Pin 176	SA10
A11	Pin 175	SA11
BAO	Pin 172	SA12
BA1	Pin 171	SA13
データ信号		





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

SDRAM Pin	FPGA Pin	External Bus Signal
DQQ	Pin 102	SD0
DQ1	Din 102	CD1
DQI	Pin 106	5D1
DQ2	Pin 107	SD2
DQ3	Pin 108	SD3
DQ4	Pin 109	SD4
DQ5	Pin 111	SD5
DQ6	Pin 122	SD6
DQ7	Pin 123	SD7
DQ8	Pin 124	SD8
DQ9	Pin 125	SD9
DQ10	Pin 126	SD10
DQ11	Pin 128	SD11
DQ12	Pin 130	SD12
DQ13	Pin 131	SD13
DQ14	Pin 132	SD14
DQ15	Pin 133	SD15

コントロール信号

Signal	FPGA Pin
SD_CAS	Pin 148
SD_CKE	Pin 156
SD_RAS	Pin 149
SD_WE	Pin 146
SD_LDQM	Pin 143
SD_USB_DQM	Pin 144
SDCLK	Pin 147
2.13 大容量、	快速Flash
回路図:	



アドレス信号:

不可能への挑戦

Flash Pin	FPGA Pin	External Bus Signal
AO	Pin 190	SA1
A1	Pin 189	SA2
A2	Pin 187	SA3
A3	Pin 185	SA4
A4	Pin 183	SA5
A5	Pin 182	SA6
A6	Pin 181	SA7
A7	Pin 180	SA8
A8	Pin 178	SA9
A9	Pin 176	SA10
A10	Pin 175	SA11
A11	Pin 172	SA12
A12	Pin 171	SA13
A13	Pin 169	SA14
A14	Pin 168	SA15

低価格、高品質が不可能?

the state	不可能への挑戦

A15	Pin 167	SA16
A16	Pin 166	SA17
A17	Pin 165	SA18
A18	Pin 162	SA19
A19	Pin 161	SA20
データ信号		
Flash Pin	FPGA Pin	External Bus Signal
DQO	Pin 102	SDO
DQ1	Pin 106	SD1
DQ2	Pin 107	SD2
DQ3	Pin 108	SD3
DQ4	Pin 109	SD4
DQ5	Pin 111	SD5
DQ6	Pin 122	SD6
DQ7	Pin 123	SD7
DQ15/A-1	Pin 191	SAO

コントロール信号

Flash Pin	FPGA Pin
FLASH_CE	Pin 155
FLASH_OE	Pin 154
FLASH_RW	Pin 150
FLASH_RY_BY	Pin 135

2.14 拡張アドレスバス

回路図:







ピンマップ:

External Bus Signal	FPGA Pin	External Bus Signal	FPGA Pin
SA0	Pin 191	SA10	Pin 176
SA1	Pin 190	SA11	Pin 175
SA2	Pin 189	SA12	Pin 172
SA3	Pin 187	SA13	Pin 171
SA4	Pin 185	SA14	Pin 169
SA5	Pin 183	SA15	Pin 168
SA6	Pin 182	SA16	Pin 167
SA7	Pin 181	SA17	Pin 166
SA8	Pin 180	SA18	Pin 165
SA9	Pin 178	SA19	Pin 162
SA20	Pin 161	SYS_RESET	Pin 152
FLASH_RW	Pin 150	SRAM_CE#	Pin 140
FLASH_CE	Pin 155	SRAM_WE#	Pin 141
FLASH_OE	Pin 154	SRAM_OE#	Pin 139

2.15 拡張データバス

回路図:







ピンマップ:

External Bus Signal	FPGA Pin	External Bus Signal	FPGA Pin
SDO	Pin 102	SD8	Pin 124
SD1	Pin 106	SD9	Pin 125
SD2	Pin 107	SD10	Pin 126
SD3	Pin 108	SD11	Pin 128
SD4	Pin 109	SD12	Pin 130
SD5	Pin 111	SD13	Pin 131
SD6	Pin 122	SD14	Pin 132
SD7	Pin 123	SD15	Pin 133
USB_D0	Pin 116	USB_D8	Pin 65
USB_D1	Pin 115	USB_D9	Pin 64
USB_D2	Pin 114	USB_D10	Pin 63
USB_D3	Pin 113	USB_D11	Pin 62
USB_D4	Pin 101	USB_D12	Pin 61
USB_D5	Pin 100	USB_D13	Pin 120
USB_D6	Pin 97	USB_D14	Pin 119
USB_D7	Pin 96	USB_D15	Pin 117



2.16 拡張ユーザIO

回路図:



ピンマップ:

External Bus Signal	FPGA Pin	External Bus Signal	FPGA Pin
CLK_IN	Pin 79	CLK_OUT	Pin 81
LCDO	Pin 22	LCD4	Pin 28
LCD1	Pin 24	LCD5	Pin 29
LCD2	Pin 26	LCD6	Pin 31
LCD3	Pin 27	LCD7	Pin 33
GPI00	Pin 35	GPI01	Pin 34
GPI02	Pin 37	GPI03	Pin 36
GPI04	Pin 40	GPI05	Pin 39
GPI06	Pin 43	GPI07	Pin 42
GPI08	Pin 45	GPI09	Pin 44
GPI010	Pin 46	GPI011	Pin 77
SWO	Pin 58	SW1	Pin 47
SW2	Pin 52	SW3	Pin 51



2.17 JTAGインタフェース

JTAG インタフェースを通じて FPGA ダウンロードする。ChipSCOP もサポートする。デバッ グ段階では JTAG モードを利用する。

回路図:



XCF02S を利用する場合は PROM モードでダウンロードする。





2.18 電源回路



2.19 リセット回路

回路図:



ピンマップ:

Device	Signal	FPGA Pin
UR1-7	SYS_RESET	Pin 152

2.20 クロック

回路図:

₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩	式会社日昇	テクノロジー	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
3.3V X1 X1 X1 X1 VCC OUT 3 CX1 2 GND NC 4 50,000MHz	RX2 33	3.3V X2 X2 X2 X2 CX2 Q Q Q Q Q Q Q Q Q Q Q Q Q	CLK USER RX3 NO POP
ピンマップ:			
Clock	Signal	FPGA Pin	
50.000MHz	MCLKIN	Pin 76	

説明:

FPGA 内部は DLL があって、50MHz のクロックをニーズに応じて必要なクロックを分頻できる。特殊なクロックは 50MHz で生成できない場合はユーザオプションを利用ください。

ユーザオプション GPI011 Pin 77



第三章 初体験

Examples¥s2_freを例として操作手順を説明します。

3.1 プロジェクトの起動

既存プロジェクトを起動する場合は File→Open Project で開く。或いは. ise ファイルをダ ブルクリックで開く。

🔤 Xilinx - ISE - E3¥xilinx¥Examples¥s2_fre¥project¥fre.ise - [Design Summary]						
<u>File E</u> dit <u>V</u> iew P <u>r</u> oject <u>S</u> ource <u>P</u> rocess <u>W</u> indow <u>H</u> elp						
8 🗋 🖻 🗑 🕼 8 🍇 🖻 Ĝ 🗙 🕪 🔎 10 8 8) 🔎 X X 🔎 🖻 📘 🔲 🗖	1 🌽 😽 🖉	û 😹 🛛 🖌 🕅	/ =00 🛛 🗩 🗩 👘		
●●●■						
Sources X	📡 FPGA Design Summary		fre Proi	ect Status	~	
Sources for: Implementation	📄 Design Overview	Project File:	fre ise	Current State:	Translated	
🤤 fre	📄 Summary	Module	fre	+ Frrors:		
☐ xc3s400-4pq208	🗋 IOB Properties	Name:		Lindio		
- V - fre (fre.v)	🗋 Module Level Utilization	Target	xc3s400-4pq208	Warnings:		
treuct (treuct)	Timing Constraints	Device:				
📭 Sources 👔 Files 📸 Snapshots 👔 Libraries	Clock Report	Product Version:	ISE 10.1.03 - Foundation Simulator	Routing Results:		
Processes X	Errors and Warnings	Design Goal:	Balanced	Timing Constraints:		
Processes for: fre	Translation Messages	Design	Xilinx Default (unlocked)	Final Timing		
Add Existing Source	Map Messages	Strategy:		Score:		
	Blann and Routo Managara					
🛛 🚬 View Design Summary	Project Properties		fre Partition S	ummary	E	
E 2 Design Utilities	Enable Enhanced Design Summ	No partition i	nformation was found.			
H W User Constraints	Display Incremental Messages					
Oynthesize - ASI Jundement Design	Enhanced Design Summary Contents	Devic	e Utilization Summary (estimated values)		
Generate Programming File	- 🗹 Show Partition Data	Logic Utilizat	ion Used	Available	Utilization	
Configure Target Device	Show Errors	Number of Slice	s	15 3584	0%	
	Show warnings	Number of Slice	Flip	27 7168	0%	
"" Processes		1 lops			I	
ISE Design Suite 101.00新棚能 🕎 Design Summary						
×					~	
Console 🛛 🕰 Warnings 🛛 🚾 Ict Shell 🖉 🦝 Find in Files						
					LOC	

3.2 ボードに書き込み

2 種類の方法があります: JTAG モードと PROM モード。JTAG モードでダウンロードする場合 は電源切ると情報が保存されない。PROM モードの場合はメモリに書き込まれて、電源入れ ると自動的にロードして実行する。

3.2.1 JTAGモードでダウンロード

Processes 欄の Generate Programming File をダブルクリックすると JTAG モードでダウン ロードする用の bit ファイルが生成される。



Configure Target Device をダブルクリックする。

🎫 Tarı	ning 🔀
⚠	No iMPACT project file exists. Click OK to open iMPACT. You will then need to define a configuration chain, designate which device in that chain is the target device, and then save the iMPACT project file. Once this step is completed, subsequent runs of the 'Configure Target Device' process can program the target device without needing to open the iMPACT GUI.
	OK

OK を押す。或いは Configure Target Device 下にある Manage Configuration Project (iMPACT)をダブルクリックする。



	No Processes Running Processes: fre Design Summary/Reports Design Villities User Constraints Office Office Office Office Office Office Design Villities Design Villities <		Power Report For Properties Content of the second	Report Name Synthesis Report Translation Report Map Report Place and Route Repor Power Report Power Report Post-PAR Static Timin Bitgen Report			
ISE	Start 💷 Design 🖺 Files 🚺 Libraries	r	Design Summary (Programming	g File Generated)			
Cons	sole	_					
1	For further information on the status of this process, see the "_impact.log" file. Process "Manage Configuration Project (iMPACT)" launched successfully						
<							
	🗐 Console 😰 Errors 🔔 Warnings 🙀 Find in Files Results						

🖾 iMPACT - Welcome to iMPACT
Please select an action from the list below Onfigure devices using Boundary-Scan (JTAG)
Automatically connect to a cable and identify Boundary-Scan chain 💌
O Prepare a PROM File
O Prepare a System ACE File
○ Prepare a Boundary-Scan File
SVF 👻
O Configure devices
using Slave Serial mode
< <u>B</u> ack <u>F</u> inish Cancel

JTAGモードを選択して、"Automatically connect to a cable and identify Boundary-Scan chain"のデフォルトの設定でFinishを押す。



🔤 Assign	New Configuration File		? 🛛
Look <u>i</u> n: R	🔄 E:/xilinx/Examples/s2_fre/project/ 💽	+ E) 💣 🏢 🏛
a.			
	3		
fre_xdb)		
- 200			
File <u>n</u> ame:			<u>O</u> pen
File <u>t</u> ype:	All Design Files (*.mcs *.exo *.isc *.bsd)	• (Cancel
	Cancel <u>A</u> ll		<u>B</u> ypass
	 None 		
	 Enable Programming of SPI Flash Device Attached to this 	s FPGA	
	O Enable Programming of BPI Flash Device Attached to this	s FPGA	

Bypass を押す。

Assign New Configuration File	? 🔀
Look in: 🔄 E:/xilinx/Examples/s2_fre/project/ 💌 🗲	🗈 💣 🖽 🏢
i suite	
📼 frebit	
File name: fre bit	Open
File type: All Design Files (* bit * rbt * nky * isc * bsd)	Cancel
Cancel <u>A</u> ll	<u>B</u> ypass
 None 	
C Enable Programming of SPI Flash Device Attached to this FPG	iA
C Enable Programming of BPI Flash Device Attached to this FPG	iA

Open を押す。



Bevice Programming Propertie	es – Device 1 Programming Properties	X
Category ⊡-Boundary-Scan 		
Device 2 (FPGA, xc3s400)	Property Name	Value
	Verify	
	General CPLD And PROM Properties	
	Erase Before Programming	
	Read Protect	
	PROM/CoolRunner-II Usercode (8 Hex Digit)	
	PROM Specific Properties	
	Load FPGA	
		OK Cancel <u>Apply</u> Help

デフォルトの設定でOKを押す。

Xilinx - ISE - E:¥xilinx¥Examples¥s2_fre¥pro	ject¥fre.ise – [Boundary Scan]	
🕵 <u>F</u> ile <u>E</u> dit <u>V</u> iew P <u>r</u> oject <u>S</u> ource <u>P</u> rocess Operations	<u>O</u> utput Debug <u>W</u> indow <u>H</u> elp	
8 🗅 🖻 🗊 🕼 8 🖁 🔓 🖌 8 🖬 🕑	🔎 🔎 🗙 🗶 🖻 📄 🔜 🖉 🖉 🖉 🖉 🖉 🖉	? 🛙 🖉 🗖 📉 💌 🗑 🖗
:[] 2 2 禁 Σ 乳乳がお :: ::: ※ ※ :: :::	1 🗄 🖬 🚔 🛱 🗗 🛷 😽	
Sources ×	Processes X	
🖻 📸Boundary Scan	Available Operations are:	
SlaveSerial	➡ Erase	000000 000000
SelectMAP	Blank Check	
Desktop Configuration	Readback	
Direct SPI Configuration	Get Device ID Get Device Checkeum	
	Get Device Signature/Usercode	xcf02s xc3s400
		TDO Dypass Tre.bit
		100
Source 🖺 Files 👩 Snaps 🖺 Librarie Configuration	Processes Configuration Operations	퉳 Boundary Scan
X INFO: iMPACT: 501 - '2': Added Device	e xc3s400 successfully.	<u>^</u>
		_
		×
	Shall 🕞 Eind in Eilea	
	Configu	ıration 🛛 Platform Cable USB 🛛 6 MHz 🛛 usb-hs 🔛

FPGA を選択して右クリックする。





🔤 Device Programming Propertie	es – Device 2 Programming Properties	×
Category ☐-Boundary-Scan ☐-Device 1 (PROM, xcf02s) ☐Device 2 (FPGA, xc3s400)	Property Name Verify FPGA Device Specific Programming Properties Assert Cable INIT during programming	
	ОК	Cancel <u>A</u> pply Help

OK を押す。

ダウンロード完了後の画面:

Zilinx – ISE – E:¥xilinx¥Examples¥s2_fre¥pro	iject¥fre.ise - [Boundary Scan]		
Bile Edit View Project Source Process Operations	: <u>O</u> utput Debu <u>e</u> <u>W</u> indow <u>H</u> elp		
8 🗅 🖻 🗐 🖓 8 🐰 🖻 🗳 🔽 🖄 🕬 💷 🖸	8 P X X P 🖻 🔊 🗟 🗆 🗅 8 / /	N? 8 M 😹 🛛 🗹 🗹 🖉 8 🖬 🗱 💥 🧏	C
Sources X	Processes ×		
	Available Operations are:		
- PSIaveSerial	-Program		
SelectMAP	Get Device ID		
- Desktop Configuration	Get Device Signature/Usercode		
	Check Idcode	"" ""	
SystemACE	Read Status Register	xcf02s xc3s400	
PROM File Formatter		bypass fre.bit	
		Program Succeeded	
		8	
📑 Source 🍙 Files 🛛 👩 Snaps 🍙 Librarie Configuration	Processes Configuration Operations	Parameters Seen	
		Boundary Scan	
× '2': Programmed successfully.			^
PROGRESS_END - End Operation.			
ETapseu Cime - 0 Sec.			
÷			>
5 Console Console	Shell 🙀 Find in Files		<u> </u>
		Configuration Platform Cable USB 6	MHz usb-hs

3.2.2 PROMモードで書き込む

Processes 欄の Generate Target PROM/ACE File をダブルクリックする。



iMPACT - Welcome 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 Prepare a Boundary-Scan File	
SVF 💌	
O Configure devices	
using Slave Serial mode	
< <u>B</u> ack <u>N</u> ext > C	ancel



MPACT - Prepare PROM Files		
I want to target a		
⊙ Xilinx PROM		
🔘 Generic Parallel PROM		
🔿 3rd-Party SPI PROM		
O PROM Supporting Multiple Design Versions:	Spartan3E MultiBoot	~
- PROM File Format		
MCS ○ TEK ○ UFP ('C" format)		
◯ EXO 🔵 BIN 🔵 ISC		
O HEX Swap Bits		
Checksum Fill Value (2 Hex Digits): FF		
PROM File Name: fre_prom		
Location: E:¥xilinx¥Examples¥s2_fre¥project¥		Browse
[< <u>B</u> ack <u>N</u> ext >	Cancel

File Name を入力する。その他はデフォルトで Next を押す。



🔤 iMPACT – Select PROM/Flash Mode				
⊙ Iam using a Xilinx PROM in Serial Mode				
○ Iam using a Xilinx PROM in a Parallel Mode and the data bus width for my FPGA is:				
The same as the data width for my Flash/PROM device				
O Bigger than the data bus width for my Flash/PROM device:				
I need to create multiple PROM files split across the data bus as selected below:				
Two x16 PROMs -> x32 FPGA Data Bus				
PROM 16 32 FPGA				
< <u>Back</u> Next > Cancel				
Next を押す。				
Auto Select PROM				
Number of Revisions: 1				
Enable Compression				
Select a PROM (bits): xcf 💌 xcf02s [2 M] 💙 Add				
Position Part Name				
U xctU2s				
Delete All				
K Back Next > Cancel				

Xcf02sを選択して Add を押す。Next を押す。



MPACT - File Generation Summary		
You have entered following information:		
PROM Type:	Serial	
File Format	mcs	
Fill Value	FF	
PROM filename	fre_prom	
Number of PROMs	1	
Position Part Name		
0 xcf02s		
Click "Finish" to start adding device files.		
	Kara Kara Kara Kara Kara Kara Kara Kara	ancel

Finish を押す。



OK を押す。



Add Device					? 🗙
ファイルの場所(1):	🗀 project		•	= 🗈 💣 🎟•	
最近使ったファイル	Co_ngo Co_xmsgs Cofre xdb				
デスクトップ	misst frebit				
21 K#15/h					
71 J)21-9					
र्ग इन रूग-१					
	ファイル:名(<u>N</u>): ファイルの種類(<u>T</u>):	frebit FPGA Bit Files (*bit)		•	開((<u>0</u>) キャンセル

FPGA Bit ファイルを選択して"開く"を押す。

🔤 Add	Device	X
?	Would you like to add another device file to	
~	Data Stream: 0	
	Yes No	

No を押す。



OK を押す。



Xilinx - ISE - E:¥xilinx¥Examples¥s2_fre¥pro	ject¥fre.ise - [PROM File Formatter]		
🥵 Eile Edit ⊻iew Project Source Process Operations	<u>Window</u> Help		
🗅 🖻 🛃 🕼 🖧 🎽 🛍 🏠 🗙 🗠 🔎 🚺	🗩 🔎 🗶 🗶 🖉 🖉 🖉 🗶 🗶 🔍 🔍	i i 🤌 😽 i DA 🙀 🔜 💌 i 🖓 i i 🗊 🗭	▶ 第 第 第 第 2 章
井淡詳計			
Sources X	Processes	×	
🖶 😁 Boundary Scan	Available Operations are:		
	Generate File	,0000000, ,0000000,	
		E XILINX E XILINX	
		PROM	
		xct02s 81.02 % Full	
PROM File Formatter		fre.bit	
📭 Source 🏠 Files 👩 Snaps 🏠 Librarie Configuration	Processes Configuration Operations	ROM File Formatter	
<pre>X INFO:iMPACT:501 - '1': Added Device</pre>	≥ xc3s400 successfully.		<u>^</u>
add one device.			~
			>
🛓 🔲 Console (😮 Errors 🔥 Warnings 🔂 Tcl	Shell 🙀 Find in Files		
	PROM File Generation Tar	get Xilinx PROM 1,699,136 Bits used File: fre prom in L	ocation: E:¥xilinx¥Examples¥s2 fre¥project¥/ .:

Processes 中の Generate File...をダブルクリックする。

Xilinx - ISE - E¥xilinx¥Examples¥s2_fre¥pro	oject¥fre.ise – [PROM File Formatter]		
Bile Edit View Project Source Process Operations	: <u>W</u> indow <u>H</u> elp		
8 🗋 🖻 🐻 🖓 8 🖓 8 🖄 🖻 🖄 💌 🖾 📝	# # X X # B 🛛 🗡 🗛 🗛 🛛 🗆 # / /	n 12 12 12 12 12 12 12 12 12 12 12 12 12	
森 淡 計 計 尊 三 壽 二 幕 〇 😺 🕅			
Sources ×	Processes X		
🗊 😁 Boundary Scan	Available Operations are:		
	➡ Generate File	0000000,	
- BSelectMAP			
- PhDirect SPI Configuration			
SystemACE		81.02 % Full vo2e400	
PROM File Formatter		fre.bit	
		PROM File Generation Succeeded	
			_ '
Source Tiles Snaps Librarie Configuration	Processes Configuration Operations	B PROM File Formatter	
Vriting file "E:\xilinx\Examples\s	2 fre\project\//fre prom.mcs".		~
Writing file "E:\xilinx\Examples\s	2_fre\project\//fre_prom.prm".		
Writing file "E:\xilinx\Examples\s	2_fre\project\//fre_prom.sig".		-
			>
Console Console	Shell 🙀 Find in Files		
			v · .v/
	FROM File Generation Target Allinx	FROM 1,099,100 bits used File ire_prom in Location: Ef#xilinx#Examples#s2_fre	#project#/

これで PROM モードで書き込み用の.mcs ファイルが生成される。

Boundary Scan をクリックする。

右側の空白欄で右クリックして"Initialize Chain"をクリックする。

🔤 Xilinx - ISE - E:¥xilinx¥Examples¥s2_fre¥pro	ject¥fre.ise - [Boundary Scan]
🕵 <u>F</u> ile <u>E</u> dit <u>V</u> iew P <u>r</u> oject <u>S</u> ource <u>P</u> rocess Operations	: Qutput Debug Window Help
8 🗅 🖻 🗟 🕼 🕹 8 X 🖻 🖺 🗙 🕫 🍽 🖸	8 P P X X P B 🔊 8 着 🗉 🗖 🗗 8 🖊 🧌 M 8 🕅 🗖 🚽 8 🖉 8 G O 8 G O 🛙 P P
[] 【 2 ◎ Σ 乳 乳 乳 乳 ● ※ = = ※ = = ※ = = = = = = = =	尊 町 森 - 森 ロ 禄 祝
Sources X	Processes X
Boundary Scan	Available Operations are:
- BassaveSerial	
- BB Desktop Configuration	Add Xilinx Device Ctrl+D
- BDirect SPI Configuration	Add <u>Non-Xilinx</u> Device Ctrl+K
- SystemACE	Initialize Chain CitrleT
E PROM File Formatter	# Indiance on an Other state of the state of
	<u>Cable Auto Connect</u>
	Cable Setup
	Qutput File Type ►



🔤 Assign	New Configuration File	? 🔀
Look <u>i</u> n: 🥳	E:/xilinx/Examples/s2_fre/project/	• 🗈 💣 🔡 🏢
🗀		
xmsgs		
🛅 fre_xdb		
🚞 xst		
🔤 fre_pro	m.mcs	
File <u>n</u> ame:	fre_prom.mcs	<u>O</u> pen
File <u>t</u> ype:	All Design Files (*.mcs *.exo *.isc *.bsd) 💌	Cancel
	Cancel <u>A</u> II	<u>B</u> ypass
	 None 	
	C Enable Programming of SPI Flash Device Attached to this F	PGA
	C Enable Programming of BPI Flash Device Attached to this F	PGA .:

先ほど生成したファイルを指定して Open を押す。

🚾 Assign New Configuration File	? 🔀
Look in: 🔄 E:/xilinx/Examples/s2_fre/project/ 💌 🗲	🗈 💣 🖽 🏢
Conses	
🔁 fre_xdb	
Carl trachit	
File <u>n</u> ame:	<u>O</u> pen
File type: All Design Files (*bit *rbt *nky *.isc *bsd)	Cancel
Cancel <u>A</u> ll	<u>B</u> ypass
 None 	
O Enable Programming of SPI Flash Device Attached to this FPG	iΑ
C Enable Programming of BPI Flash Device Attached to this FPG	iA .::

Bypass を押す。



Bevice Programming Properti	es – Device 1 Programming Properties	
Category		
Device 2 (FPGA xc3s400)	Property Name	Value
	Verify	
	General CPLD And PROM Properties	
	Erase Before Programming	
	Read Protect	
	PROM/CoolRunner-II Usercode (8 Hex Digit)	
	PROM Specific Properties	
	Load FPGA	
	ОК	Cancel <u>A</u> pply Help

OK を押す。



Xcf02sを右クリックして Programを押す。





Xilinx - ISE - E:¥xilinx¥Examples¥s2_fre¥pro	ect¥fre.ise – [Boundary Scan]	
🥪 Eile Edit ⊻iew Project Source Process Operations	Qutput Debug Window Help	
:□ > = # # ↓ :: 从 □ □ × > # ↓ : f ? 7 撇 x ? ? ? ? ? ? ? : :: :: :: :: :: :: :: ::	₽₽XX₽8 > \$==□□ ₽ ? == #===0₽ ₽	M 🕷 📃 🔍 I 🖇 I 🕲 🕲 I 🖉 🖉 🖉
Sources X	Processes X	
Boundary Scan SigClaveSerial SigClaveSerial SigClaveCerial SigClaveCerial SigClaveCerial SystemACE PROM File Formatter	Available Operations are: Program Frase Blank Check Peadback Get Device ID Get Device Signature/Usercode Check Idcode	TDI Example for the promines bypass
📾 Source 🜔 Files 👩 Snaps 🜔 Librarie Configuration	Processes Configuration Operations IS	E Design Suite 10.1 🐠 🍸 Design Summary 🛛 🔯 PROM File Form: 🔯 Boundary Scan
<pre>X '1': Programming completed success: PROGRESS_END - End Operation. Elapsed time = 11 sec. Console C Errors Warnings Tol.</pre>	ully. Shell 🛛 🙀 Find in Files	Configuration Platform Cable USB 6 MHz usb-hs .

再度電源を入れると書き込んだファイルが実行される。



第四章 開発基本手順

Examples¥s1_led¥sw_ledを例として開発基本手順を紹介します。 実現機能:ボード上のキーでLEDを下記ロジックで制御する。

State	SW0	SW1	SW2	SW3	LED0	LED1	LED2	LED3	LED4	LED5	LED6	LED7
S1	0	0	0	1	0	0	0	0	0	0	0	1
S2	0	0	1	0	0	0	0	0	0	0	1	0
S 3	0	1	0	0	0	0	0	0	0	1	0	0
S4	1	0	0	0	0	0	0	0	1	0	0	0
S 5	0	0	1	1	0	0	0	1	0	0	0	0
<u>\$6</u>	0	1	1	0	0	0	1	0	0	0	0	0
S 7	1	1	0	0	0	1	0	0	0	0	0	0
S 8	1	0	0	1	1	0	0	0	0	0	0	0
	-	D	efault		0	0	0	0	0	0	0	0

4.1 Project Navigatorの起動

Xilinx ISE 10.1i バージョンを基に説明します。

スタート→すべてのプログラム→Xilinx ISE 10.1i→Project Navigator をクリックする。



Eile Edit View Breiset Seuree Breeses Window Hele	
	S = □ □ N N N N N N
Sources X	
No project is open	
Select: File->Open Project or File->New Project	
🖹 😭 Sources 📸 Snapshots 👔 Libraries	
No flow available.	
₽ ⊈ Processes	http://www.xilinx.com
×	
<u>5</u>	~
	8
💾 🔲 Console 🛛 🐼 Errors 🛛 🔔 Warnings 🔤 Tcl Shell 🛛 🛣 Find in Files	

4.2 プロジェクトの新規作成

新規作成する場合の手順は下記通り:

1, File→New Project

🔤 New Project Wizard - Crea	nte New Project	
Enter a name and location for the Project name:	Project Project Jecution E:¥xilinx¥sw_led	
Select the type of top-level source Top-level source type: HDL	e for the project	\
More Info	< Back	t > Cancel



Project Name 欄に名称を入力する。Project Location 欄に保存場所を選択する。
 Top-Level Module Type は主に二種類を使います。

HDL(Hardware Design Language)ハードウェア設計言語モードと Schematic 原理図モード。 この試験では HDL を選択する。Next を押す。

Property Name	Value	
Product Category	All	~
Family	Spartan3	~
Device	XC3S400	~
Package	PQ208	~
Speed	-4	~
Top-Level Source Type	HDL	V
Synthesis Tool	XST (VHDL/Verilog)	¥
Simulator	ISE Simulator (VHDL/Verilog)	~
Preferred Language	Verilog	¥
Enable Enhanced Decign Summary		
Enable Message Filtering		
Display Incremental Messages		

本試験は合成ツールとシミュレーションツールは全部 ISE を利用します。デフォルトのまま Next を押す。

3、ソースを作成する。



🔤 New Project Wizard – Create New	Source	×
Oreate a new source		New Sames
Source File	Туре	<u>Remove</u>
Creating a new arrange to add to the preject in		- evented with the New
Project Wizard. Additional sources can be crea command.	s optional. Only one new source can be ated and added to the project by using	s the "Project->New Source"
Existing sources can be added on the next part	ge	
More Info	< <u>B</u> ack	Next > Cancel

後で作成してもいいので、ここはこのまま Next を押す。

4、既存ソースの追加。

🔤 New	Project Wizard - Add Existing Sources		
Add	existing sources		
1	Source File	Copy to Project	Add Source
Adding the [«] Pi	existing sources is optional. Additional sources can be oject->Add Source" or "Project->Add Copy of Source"	e added after the project is creat * commands.	ed using
<u>M</u> ore	Info	< <u>B</u> ack <u>N</u> ext >	Cancel

後で作成してもいいので、ここはこのまま Next を押す。

5、新規作成するプロジェクトの詳細情報を確認して、問題なければ Finish を押して、 sw_led と言う名称のプロジェクトが作成される。



15	New Project Wizard – Project Summary	
	Project Navigator will create a new project with the following specifications:	
	Project: Project Name: sw_led Project Path: E:\xilinx\sw_led Top Level Source Type: HDL Device:	
	Device Family: Spartan3 Device: xc3s400 Package: pq208 Speed: -4	
	Synthesis Tool: XST (VHDL/Verilog) Simulator: ISE Simulator (VHDL/Verilog) Preferred Language: Verilog	
	Enhanced Design Summary: enabled Message Filtering: disabled	
	< <u>B</u> ack <u>F</u> inish Cancel	
150	Xilinx - ISE - Ei¥xilinx¥sw_led¥sw_led.ise	
Ei	le <u>E</u> dit <u>V</u> iew P <u>r</u> oject <u>S</u> ource <u>P</u> rocess <u>Wi</u> ndow <u>H</u> elp	
		• # 9
S	ources X ources for: Implementation X Swy_led ** xc3s400-4pq208	
Ð	Sources 🐚 Files 🥳 Snapshots 🐚 Libraries	
Pr P	x rocesses x rocesses for: xc3s400-4pq208	
Ģ	t Processes	//www.xilinx.com
Transcript ×	Console Console Warnings Tol Shell K Find in Files	×

4.3 新規作成プロジェクトにソースファイルの追加

1、既存ソースファイルの追加

xc3s400-4pq208 を選択して、Project→Add Source をクリックして、既存の.v.vhd ファイ ルを追加する。



🔤 Xilinx – ISE	- E:¥xilinx¥sw_led¥sw_led.is	e	
<u>File E</u> dit <u>V</u> iew	P <u>roject S</u> ource <u>P</u> rocess <u>W</u> indow	Help	
8 🗋 🖻 🖥 🕼	<u>N</u> ew Source	🖸 :: 🕫 🗩 🗶 🔊 : 🔊 : 🗖 🗖 🗖 : 🖉 : 🖉 : 🕅 🗖	💌 🗄 💡
: 💕 🗭 🔎 🔅	<u>A</u> dd Source		
Sources	Add <u>C</u> opy of Source		
Sources for: Imp	C <u>l</u> eanup Project Files		
sw led	<u>T</u> oggle Paths		
- xc3s400-4	Arc <u>h</u> ive		
	Take <u>S</u> napshot		
	<u>M</u> ake Snapshot Current		

2、ソースファイルの新規作成

xc3s400-4pq208を選択して、Project→New Sourceをクリックして、下記画面が出る。

🔤 New Source Wizard - Select Source Type	\mathbf{X}
 IP (CORE Generator & Architecture Wizard) Schematic State Diagram Test Bench Waveform User Document Verilog Module Verilog Test Fixture VHDL Module VHDL Library VHDL Package VHDL Test Bench Embedded Processor 	File name: sw_led Logation: E:¥xilinx¥sw_led
More Info	< <u>B</u> ack <u>N</u> ext > Cancel

左側から作成するソースファイルの種類を選択して、File name と Location を入力する。 Add to project を選択する事でプロジェクトに追加される。Next を押す。



🔤 New Source W	izard – Define M	odule			×
Module name sw_led	1				
Port Name	Direction	Bus	MSB	LSB	~
	input	×			
	input	×			
	input	×			
	input	×			
	input	*			_
	input	×			
	input	×			
	input	×			
	input	×			
	input	×			
	input	×			
	ionut				
More Info			< <u>B</u> ac	k <u>N</u> ext >	Cancel

この画面で新規作成するソースファイルの入力、出力ポートを追加してもいいですが、直接 Next を押して、ソースファイルを編集する時追加してもいいです。

🚾 New Source Wizard – Summary	$\mathbf{\times}$
Project Navigator will create a new skeleton source with the following specifications:	
Add to Project: Yes Source Directory: E¥xilinx¥sw_led Source Type: Verilog Module Source Name: sw_led.v	
Module name: sw_led Port Definitions:	
<u>Eack</u> <u>Einish</u> Cancel	

新規作成するソースファイルの詳細情報を確認して、問題なければ Finish を押して、sw_led と言う名称のソースファイルのフレームが作成される。





```
ソースを編集する。
```

module sw_led(clk, rst, sw0, sw1, sw2, sw3, led0, led1, led2, led3, led4, led5, led6, led7); input clk; input rst; input sw0; input sw1; input sw2; input sw3; output led0; output led1; output led2; output led3; output led4; output led5; output led6; output led7; wire led0; wire led1; wire led2; wire led3;





```
wire led4;
wire led5;
wire led6;
wire led7;
wire [3 :0] sw;
reg [7 :0] led;
assign sw = \{sw0, sw1, sw2, sw3\};
assign {led0, led1, led2, led3, led4, led5, led6, led7} = led;
always @ ( posedge clk or negedge rst)
begin
if(!rst)
1 ed \le 0;
else
case (sw)
4'b0001: led <= 8'b0000_0001;
4'b0010: led <= 8'b0000 0010;
4'b0100: led <= 8'b0000_0100;
4'b1000: led <= 8'b0000_1000;
4'b0011: led <= 8'b0001_0000;
4'b0110: led <= 8'b0010_0000;
4'b1100: led <= 8'b0100_0000;
4'b1001: led <= 8'b1000_0000;
default: led <= 8'b0000_0000;
endcase
end
endmodule
保存する。
```

4.4 合成及びシミュレーション

4.4.1 合成

Processes 欄の Synthesize-XST をダブルクリックする。 主にはソースプログラムの構文エラーを検査する。



Xilinx - ISE - E:¥xilinx¥sw led¥sw led.ise - [s	w led v]	
Image: Second Secon		
	······································	
	·/////////////////////////////////////	
- 💽 🖸 🔄 👬 🖉 🔐 🦉 🔐 🖉 🚳 - 🖌	≝ ≝ ≝ ≝ ≌ ∧ % % % % 🥨 🕺	
Sources ×	1 `timescale 1ns / 1ps	~
Sources for: Implementation	2 /////////////////////////////////////	11
esw_led	3 // Company:	
□	4 // Engineer:	
<mark>V]</mark> ∰asw_led (sw_led.v)	5 // 6 // Create Date: 17:02:40 10/30/2010	
	7 // Design Name:	
	8 // Module Name: sw led	
Sources The sources Contraries	9 // Project Name:	
Processes ×	10 // Target Devices:	
Processes for: sw led	11 // Tool versions:	
□-83/O Synthesize - XST	12 // Description:	
View Synthesis Report	13 // 14 // Dependencies:	
View RTL Schematic	15 //	
View Technology Schematic	16 // Revision:	
Pi@Check Syntax	17 // Revision 0.01 - File Created	
🕢 🕀 🏹 Generate Post-Synthesis Simulation Mode	18 // Additional Comments:	
🗉 🔥 Implement Design	19 //	
	20 ////////////////////////////////////	<u>// 💌</u>
₽# Processes		>
1	v sw_led.v 🛛 🔊 Design Summary	
×		^
Process "Check Syntax" completed su	ccessfully	_
[基]		~
		>
📱 📋 Console 🛛 🔞 Errors 🛛 🔬 Warnings 🛛 🚾 Tcl S	Shell 🙀 Find in Files	
	CAPS NUM SCRL In 37 Col 11 Ver	rilog .:

View Synthesis Reportをダブルクリックして結果を見える。

🔤 Xilinx - ISE - E:¥xilinx¥sw_led¥sw_led.ise - [Synthesis Report]		
☑ File Edit View Project Source Process Window H	elp 🔳 🖉 🗶		
🗅 🖻 🗊 🗳 🍐 🔏 🖻 🏔 🗙 🕪 🍽 🚺	8 🕫 🔎 🗶 💥 🔎 🖻 \mid 🔊 8 🐂 💷 🖙 8 🎤 🐕 8 🎮 🕷 🔛 💌 8 🖓		
● f 🗹 🖉 🌞 🗵 跳 跳 跳 跳 🕄 🏵 🛯 •	▶ □ □ □ □ □ □ □ 1 → 34 34 34 30 32		
Sources ×	OBUF:I->O 5.644 led0_OBUF (led0) 🔨		
Sources for: Implementation	 Total 7.165ns (6.364ns logic, 0.801ns (88.8% logic, 11.2% rou		
Cources Files Snapshots Libraries	Total REAL time to Xst completion: 6.00 secs Total CPU time to Xst completion: 5.41 secs		
Processes for: sw_led	>		
Cyrothesize - XST View Synthesise Report View RTL Schematic View Technology Schematic Check Syntax Check Syntax Check Syntax Duplament Post-Synthesis Simulation Mode	Total memory usage is 131308 kilobytes Number of errors : O (O filtered) Number of warnings : O (O filtered) Number of infos : O (O filtered)		
	✓		
Processes	V sw_led.v Synthesis Report		
Process "Check Syntax" completed successfully			
E Console	Shall De Find in Files		
	Ln 1 Col 1 Ln yy Col xx		

ここでエラー発生した場合は Error ポップが表示される。提示内容によって修正して再度





合成すればいいです。

4.4.2 シミュレーション

1、シミュレーションファイルの作成

Project→New Source をクリックして、Test Bench Wavefrom ファイルを生成する。

株式会社日昇テクノロジー

📧 New Source Wizard - Select Source Type	×
 BMM File IP (CORE Generator & Architecture Wizard) MEM File Schematic Implementation Constraints File State Diagram Test Bench Waveform User Document Verilog Module Verilog Test Fixture VHDL Module VHDL Library VHDL Package VHDL Test Bench Embedded Processor 	File name: test Logation: E¥xilinx¥sw_led
More Info	< <u>B</u> ack <u>N</u> ext > Cancel

左側から作成するソースファイルの種類を選択して、File name と Location を入力する。 Add to project を選択する事でプロジェクトに追加される。Next を押す。

🔤 New Source Wizard – Associate Source	
Select a source with which to associate the new source.	
sw_led	
More Info	< <u>B</u> ack <u>N</u> ext > Cancel



対象ファイル sw_led を選択して、Next を押す。

🔤 New Source Wizard - Summary	×
Project Navigator will create a new skeleton source with the following specifications:	
Add to Project: Yes Source Directory: E¥xilinx¥sw_led Source Type: Test Bench Waveform Source Name: test.tbw	
Association: sw_led	
< <u>B</u> ack <u>F</u> inish Cancel	

情報を確認して Finish を押す。

test.tbw ファイルが作成される同時に下記画面が表示され、タイミングについてのパラメ ータを設定する。

🧱 Initial Timing and Clock Wizard – Initia	ilize Timing 🛛 🗙
Maximum output delay Clock high for	Minimum input setup Clock
Clock Timing Information Inputs are assigned at "Input Setup Time" and outputs are checked at "Output Valid Delay".	Clock Information ⊙ Single Clock
Rising Edge Falling Edge Dual Edge (DDR or DET)	 Multiple Clocks Combinatorial (or internal clock)
Clock High Time 50 ns Clock Low Time 50 ns Input Setup Time 10 ns Output Valid Delay 10 ns Offset 100 ns	Combinatorial Timing Information Inputs are assigned, outputs are decoded then checked. A delay between inputs and outputs avoids assignment/checking conflicts. Check Outputs 50 ns After Inputs are Assigned Assign Inputs 50 ns After Outputs are Checked
Global Signals PRLD (OPLD) GSR (FPGA) High for Initial: 100 ns	Initial Length of Test Bench: 1000 ns Time Scale: ns 🗸
More Info	< Back Einish Cancel

シミュレーションの必要に応じて設定して Finish を押す。新規作成した波形ファイルが表



示される。

🔤 Xilinx – ISE – E:¥xilinx¥sw_led¥sv	w_led.ise - [test.tbv	#]												
☑ File Edit View Project Source Proces	ss <u>T</u> est Bench Sim <u>u</u> la	tion <u>W</u> ine	low <u>H</u> elp											
🗋 🆻 🗒 🕼 💩 💥 🖻 🗳 🗙	🖄 🎯 🛛 📝 🗄 🔎 🔎	XX	🔎 🔁	2 8		🕒 8 🎝	<mark>₽ א?</mark> ∃	🖄 🖗	õ	~	8 💡			
f 7 9 🎕 x X X X X (3 € i + → i Ξ	<u>n</u> <u>-</u>	2 1	% %	× • • ;	8 🗄 🗠	±r †	t 📩 i	61	G II 4		1000	🗙 ns 💌	
Sources X Instance Design Unit Type	End Time: 1000 ns		Ons 1	00 ns	200 ns	300 ns	s 40(0 ns	500 ns	600 ns	700	ns 800)ns 900 n	50.0 ns 1000 ns
	III cik	0		ſ				ΪΓ						· · ·
	🛺 rst	0												
	Sw0	0												
	sw1 🕄	0												
	sw2	0												
	Sw3	0												
🖙 Sour 🗈 File: 🚙 Spar 🕞 Libr. 🗔 Sim Ir.	💦 led0	0												
	💦 led1	0												
Processes X	💦 led2	0												
State and State	💦 led3	0												
	اed4 ک <mark>ا ا</mark>	0												
🚽 🖓 🖌	💦 led5	0												
< · · · · · · · · · · · · · · · · · · ·	< >	< > >	<											>
🖳 Processes 🔤 Sim Objec 🄤 Hierarchy -	🖸 swiled.v 🛛 🗺 D	esian Sun	mary	Svntł	nesis Repo	rt 🗖	test.tbw	*						
		-		• ·										
Started : "Creating Tbw f Compiling verilog file "E	ile". :/xilinx/sw_led	/sw_led	l.v"											
		-												>
Console Console	ngs TCI Shell	K Find in	n Files										Time:	662.8 ns

2、パラメータの修正

タイミングパラメータは前のステップで設定しましたが、修正したい場合は下記方法があ ります。

 $\checkmark = = = -\mathcal{O}$ Test Bench \rightarrow Rescale Timing...

<u>T</u> est Bench <u>W</u> indow <u>H</u> elp
Set <u>E</u> nd of Test Bench
<u>R</u> escale Timing
<u>S</u> et "Result File" Name
<u>C</u> lose Windows



🚾 Rescale Timing									
Clocks									
<u>R</u> eset A	Reset All								
Clock Information —									
Clock High Time	50		ns	-Clock Edge					
Clock Time Low	50		ns	💿 Rising Edge					
Input Setup Time	10		ns	🚫 Falling Edge					
Output Valid Delay	10		ns	🔵 Dual Edge					
Offset	100		ns						
Time Scale: ns V OK Cancel Help									

修正して OK を押す。

或いはTest Bench \rightarrow Set End of Test Bench…



Set End of	Test Benc	h	
Test Bench E	nds: 2000	ns	•
ОК	Cancel	н	lelp

OK を押すと、End Time が 1000ns から 2000ns に変更される。



🔤 Xilinx – ISE – E:¥xilinx¥sw_led¥s	w_led.ise - [test.tb	bw*] 🗌 🔀
<u> E</u> ile <u>E</u> dit ⊻iew P <u>r</u> oject <u>S</u> ource <u>P</u> roce	ss <u>T</u> est Bench Sim <u>u</u> la	ulation Window Help
H 🗅 🆻 🖥 🖉 🌭 H X 🖻 🛍 🗙	🖄 🖓 🖸 🗄 🏓 🌽	P X X ₽ ◙ ≥ 8 = □ □ 8 ₽ X 8 M 🕷 🔜 🛛 8 ?
🗊 🖉 🖉 🎆 🗵 🏗 🕱 🏋 📰 🔛	G 🕑 🗄 🗠 🗠 🕴	† 🏠 作 🏫 🔲 🗐 🕨 🗡 1000 🔍 ns 👻
Sources ×	End Times	950.0 ns
Instance Design Unit Type	End time:	0 ns 250 ns 500 ns 750 ns 1000 ns 1250 ns 1500 ns 1750 ns 2000 ns
	2000 ms	
	LII cik	
	UI rst	0
	M sw0	0
	🚺 sw1	
	sw2	
	sw3	
📭 Sour 🍙 File: 🚕 Snar 🍙 Libr. 🔤 Sim Ir	. 💦 led0	0
	🚺 led1	0
A Contraction of the second se	led2 کی ا	0
🛱 💼 sw led	led3 آل	0
	led4 کی ا	0
🚽 🖓 🖬	💦 led5	0
<	< T 3	
Processes The Sim Objec The Hierarchy -	🗾 🕎 Design Summary	ry D Synthesis Report at test.tbw*
×		
Started : "Creating Thw s	file". Trívilinu/au lad	
compiling vering ine "r	1:/x111nx/sw_ieu,	20/30_1ed.V"
		>
💆 📄 Console 🛛 👩 Errors 🛛 🔥 Warni	ings 🔂 🔂 Tcl Shell	🙀 Find in Files
		Time: 10625 ns

波形図の青色のバーをクリックすると High、Low のレベルを設定できる。

🔤 Xilinx - ISE - E:¥xilinx¥sw_led¥sw	v_led.ise - [test.tbv	#]						
File Edit View Project Source Proces	ss <u>T</u> est Bench Sim <u>u</u> la	tion <u>W</u> indow <u>H</u>	<u>l</u> elp					IIX
I 🗋 🆻 🖥 🕼 I 🌭 I 🗶 🖻 🖄 🗙 I	🐚 🕬 📝 🛛 🔎 🔎	XXP	3 🔊 🖓 🖥	3 🗆 🖻 8 🗸	🍳 😽 🗄 🖉 🖗	✓ # ¹	?	
:[[2] 2] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	也也 十百个1	1 🖸 II 6	► ► 1000	💙 ns 🔍	•			
Sources X							1450	0.0 ns
Instance Design Unit Ture	End Time:						1350 po 1500	
anstance besign onter type	1500 ns							115
	🔊 cik	1						^
	🔊 rst	0						
	JI sw0	0						
	🔊 sw1	0						
	🔊 sw2	0						
	🔊 sw3	0						
📾 Sour 🗈 File: 🚙 Spar 💽 Libr. 🔤 Sim Ir	🔰 led0	0						
	🔰 led1	0						
Processes X	الم led2	0						
	الله Ied3	0						
	الله led4	0						
🛺 rst 🔍	اللہ led5	0						
<	<	< < > <						
🖳 Processes 🔤 Sim Objec 🔤 Hierarchy -	Synthesis Report	test.tbw	*					
Commiling verilog file "F	·/vilinv/ew_led	/ew led y"						
compring verifing file L	., x11111x, 5%_100,	50_1Cu.v						-
Started : "Launching ISE	Text Editor to (edit sw_led	d.v".					_
të								~
E Concela	T-I Chall	na rout o rout						2
		A ring in Files						
							Time: 532.	}ns

3、シミュレーション

test.tbw を選択して、Process View から View Generated Test Bench As HDL をダブルク リックする。



Zilinx - ISE - E:¥xilinx¥sw_led¥sw_led.ise - [test	tfw]	
<u> File Edit View Project Source Process Window Help</u>		
8 🗋 🖻 🗑 🕼 8 👗 🔓 🕼 🗶 🛤 🕪 🕗 8 🗩	PXXP2 P	
비소소 : 1 소 소 소 G 비 (e) 조 1000 · · ·	▼ns ▼84 ▶ □ 12 □ 12 □ 12 ▲ 74 74 74 79 391	i i
Sources X Sources for: Behavioral Simulation •••••••••••••••••••••••••••••	<pre> 1 ////////////////////////////////////</pre>	
Processes X Processes X Processes for: test X Add Test Bench To Project	<pre>10 // /_ / /¥ Finestamp : Sat Oct 30 21:07:51 2010 12 // ¥ ¥ / ¥ 13 // ¥ ¥/¥ 14 // 15 //Command: 16 //Design Name: test 17 //Device: Xilinx 18 // 19 `timescale ins/1ps</pre>	
🖻 🎾 Xilinx ISE Simulator	20	~
🐨 🔤 Simulate Behavioral Model 🛛 💽		>
₩ Processes Sim Objects	Test the V test tfw	
×		
Concole France Warnings Tol Shall	De Cind in Clina	
	CAPS NUM ISCRU Ln 1 Col 1 Ver	rilog .:

Simulate Behavioral Model をダブルクリックしてシミュレーションの結果を見る。

🔤 Xilinx - ISE - E:¥xilinx¥sw_led¥sw_led.ise - [Sim	ulation]											X
☑ <u>File</u> <u>Edit</u> <u>View</u> Project <u>Source</u> Process <u>T</u> est Bench Sir	m <u>u</u> lation <u>W</u> indow <u>H</u> elp											Ð×
🗋 🖻 🗐 🕼 🌭 🔏 🖻 🛍 🗶 🕫 🞯 🕗 🖉 🔎	P 🗙 🗶 🔎 🖻 🛛	2 : 4	. 8 0 🗅 8	<i>▶</i> № 8	00 🐼		💌 8 💡	' 8 🗊 🛙	1 🗹 🔅	i 🛛 🏗	R 92 92	
🗄 🗠 🍲 🕈 📥 🏫 🦘 🖸 🛯 🦕 🛌 1000 👘	🗙 ns 💌 🔺 🕨	- n	E 2 / 4 %	1 74 76	1 1 1 1							
Sources X					-						950.0 n	
Sources for: Behavioral Simulation	Current Simulation						500					
- @sw_led	Time: 1000 ns		Uns 100 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns	900 11000 1	IS
☐ xc3s400-4pq208	led0	0										^
	led1	0										
	led2	1										
	led3	0										
	led4	0	Í									
	all led5	0										
Sources 🕞 Files 🧼 Snanshr 🍙 Libraries 🗔 Sim Instar	led6	0										
	led7	0										
Processes X	PERIOD[31:0]	3				32	'h000000	34				
Processes for: test	DUTY_CYCLE	0.5					0.5					
- 📄 View Generated Test Bench As HDL	■ 🛃 OFFSET[31:0]	3					"h0000000	64				
Add Test Bench To Project	o. clk	0										
Allinx ISE Simulator Simulate Behavioral Model	All rst	1										~
		< >	<									>
	test.tbw 🔽 tes	st.tfw	w Simulation									
× Simulator is doing circuit initializat	tion process.											^
Finished circuit initialization proces	38.											
ξ. [*]												~
	l	Cim.	Oracela test	1								>
Console 🛛 Errors 🗥 Warnings 🛄 I CI Shell	A Find in Files	an sim	Console - test									
											Time:	-



4.5 Constraintsの編集



Timing Constrains、IO-Pre-Synthesis、Area/IO/Logic-Post-Synthesisがある。

本試験ではIO-Pre-Synthesisだけ利用する。

Process viewからIO-Pre-Synthesisをダブルクリックする。

🧱 Xilinx P	ACE – E¥xili	nx¥sw_led	sw_led.u	əf		
_ <u>E</u> dit	<u>V</u> iew <u>I</u> OBs <u>A</u>	<u>A</u> reas <u>T</u> ools	<u>W</u> indow	<u>H</u> elp		
			10			
		·				
📔 Design	Browser				Device Architecture for xc3s400-4-pq208	
	Pins					
Glob	al Logic					-
📕 🔚 Log	ic					
<				>		
-						
🖹 Design	Object List -	- I/O Pins				
I/O Nar	ne I/O Directi	ion Loc	Bank	1/0 St		
	Input	p76	BANK5	50.00		
	Output	p70	BANKO			
led1	Output	p203	BANKO			
led2	Output	p200	BANKO			
led3	Output	p199	BANKO			
led4	Output	p198	BANKO			
led5	Output	p197	BANKO			
led6	Output	p196	BANKO			
led7	Output	p194	BANKO			
rst	Input	p152	BANK2			
swU	Input	p58	BANK5			
SWI	Input	po /	BANK5			
SW2	Input	po2	BAINKO			
SWO		por 🖻	(BAINKO			
						_
<u> </u>				>	Veckage View Architecture View /	

Loc 欄を pxxx のフォーマットで入力して保存する。



Bus Delimiter
Select IO Bus Delimiter ○ ☆ST Default: < > • ☆ <u>ST Optional ()</u> • Synplify <u>V</u> erilog Default: [] • Synplify V <u>H</u> DL / Exemplar Default: ()
 Don't show this dialog again (can be set through preferences dialog) OK Cancel

XST Optional を選択して OK。Xilinu PACE ウィンドウを閉じる。

4.6 実行

Processes viewの Implement Design をダブルクリックする。

Processe	9S	×							
Processe	es for: sw_led	^							
	Add Existing Source								
	Create New Source								
···· Σ	View Design Summary								
🗄 🤡	Design Utilities	=							
🗄	User Constraints								
E 🔁)Synthesize – XST								
±- 000	Implement Design								
⊞ (2)	Generate Programming File								
🖻 🔁	Configure Target Device	~							
<		-							
₽(Proc	besses								

完了後緑色のチックが表示したら成功。

4.7 ボードにダウンロードして実行

「3.2 ボードに書き込み」節をご参照ください。

以上。