



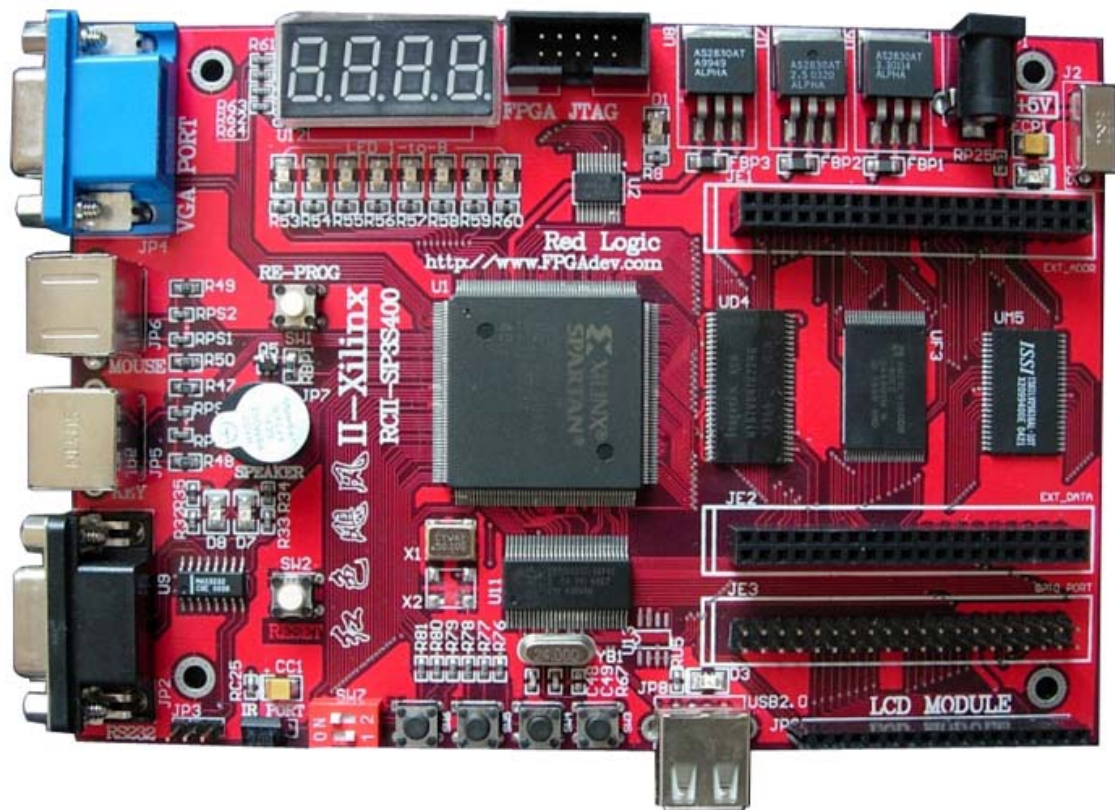
# Xilinx Spartan3 シリーズ XC3S400 開発ボードマニュアル

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。

「<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。



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.....	10
2.6 VGAインタフェース.....	11
2.7 PS/2 インタフェース.....	12
2.8 RS232 シリアルポートインタフェース.....	13
2.9 LCD液晶インタフェース.....	14
2.10 USBインタフェース.....	17
2.11 高速非同期SRAM.....	18
2.12 高速同期SDRAM.....	20
2.13 大容量、快速Flash.....	22
2.14 拡張アドレスバス.....	24
2.15 拡張データバス.....	25
2.16 拡張ユーザIO.....	27
2.17 JTAGインタフェース.....	28
2.18 電源回路.....	29
2.19 リセット回路.....	29
2.20 クロック.....	29
第三章 初体験.....	31
3.1 プロジェクトの起動.....	31
3.2 ボードに書き込み.....	31
3.2.1 JTAGモードでダウンロード.....	31
3.2.2 PROMモードで書き込む.....	36
第四章 開発基本手順.....	45
4.1 Project Navigatorの起動.....	45
4.2 プロジェクトの新規作成.....	46
4.3 新規作成プロジェクトにソースファイルの追加.....	49
4.4 合成及びシミュレーション.....	53



---

4.4.1 合成.....	53
4.4.2 シミュレーション .....	55
4.5 Constraints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

## 第一章 SP3S400 ボードの概要

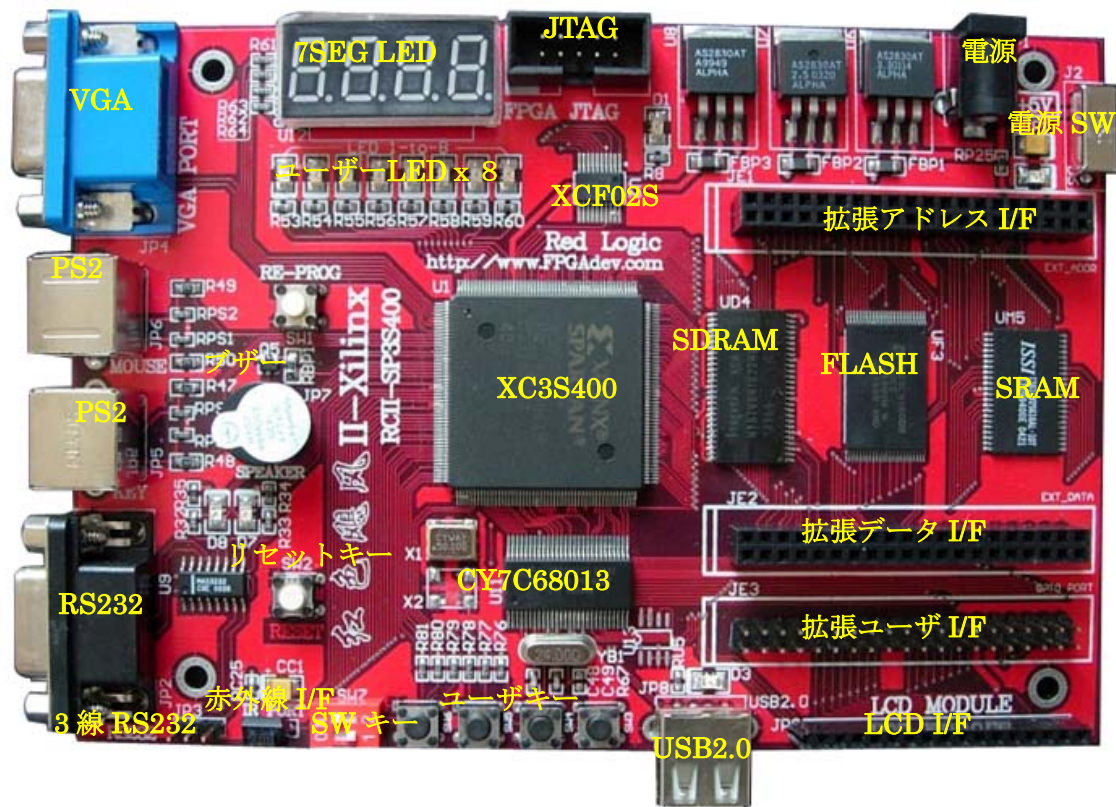
Xilinx 社最先端の 90nm プロセスを使用した Spartan ソリューションである Spartan-3 FPGA プラットフォーム、40 万システムゲートの XC3S400 マイコンを採用しております。

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 の高速の USB2.0 データインターフェースも提供しております。
- 3、多機能のインターフェース、拡張インター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)。

- (1) 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พุ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
Distributed RAM Bits:	56K
Block RAM Bits:	288K
Dedicated Multipliers:	16
DCMs:	4
Maximum User I/Os:	141

2、XCFO2S、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 x 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、7SegLED x 4、ユーザ LED x 8

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。

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 x 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



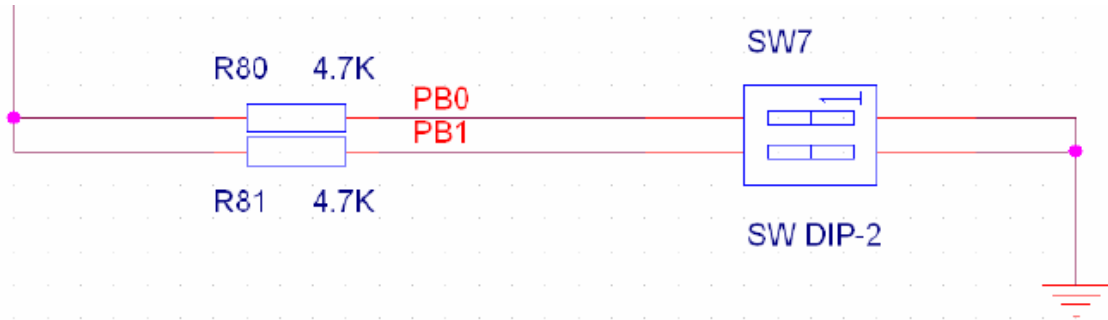
- 4、5V/1A 電源
- 5、シリアルケーブル
- 6、USB ケーブル

## 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：



ピンマップ：

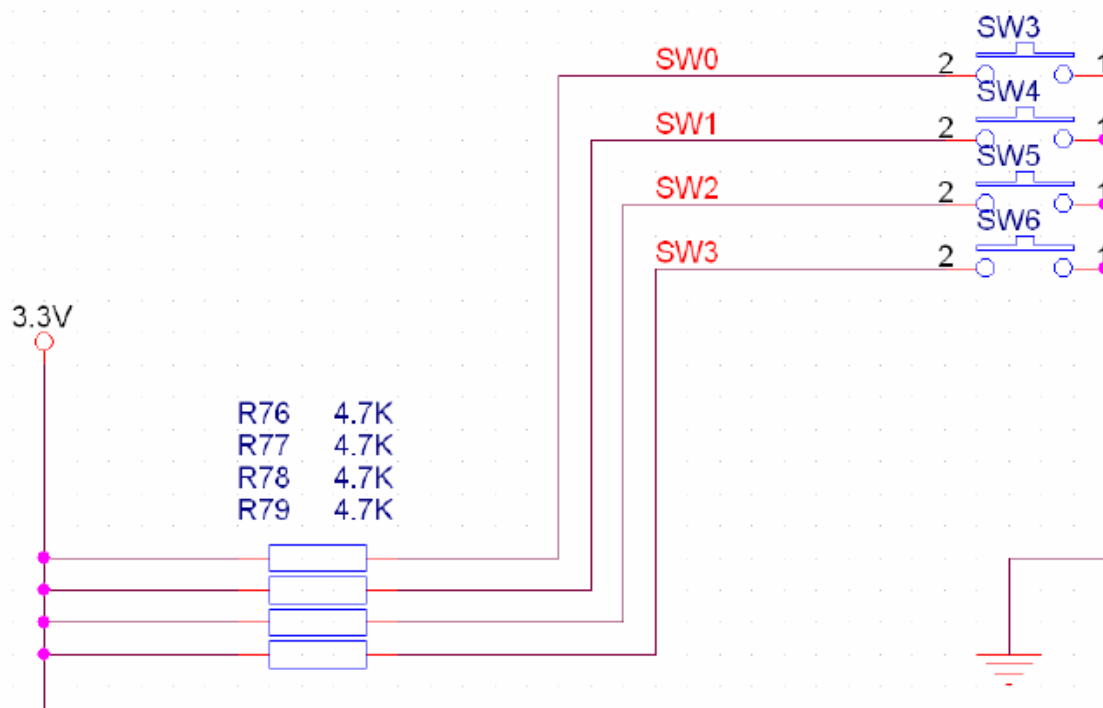
Switch	PB0 (下)	PB1 (上)
FPGA Pin	Pin 50	Pin 48

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：

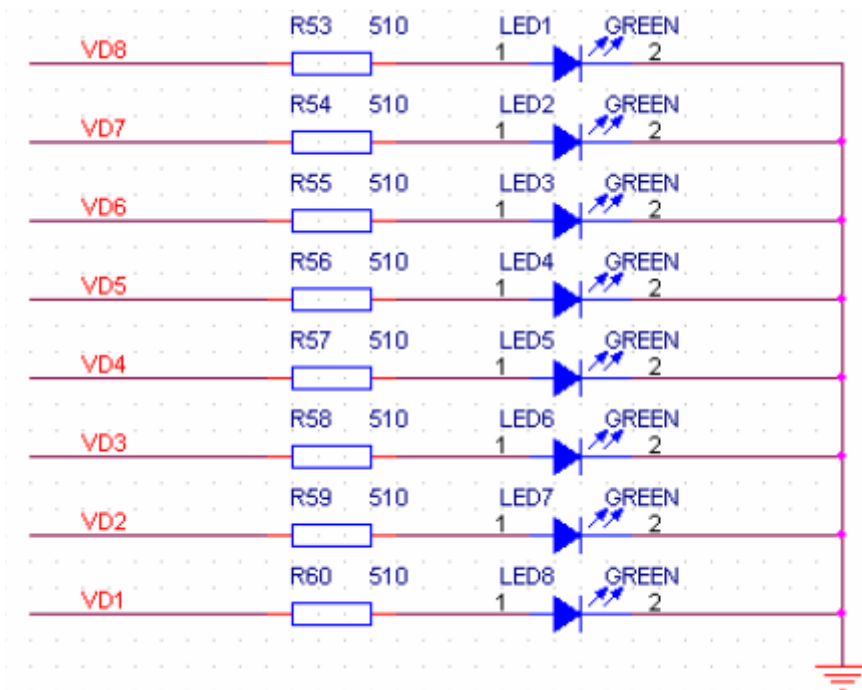


Push Button	SW6	SW5	SW4	SW3
FPGA Pin	Pin 51	Pin 52	Pin 57	Pin 58

### 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：



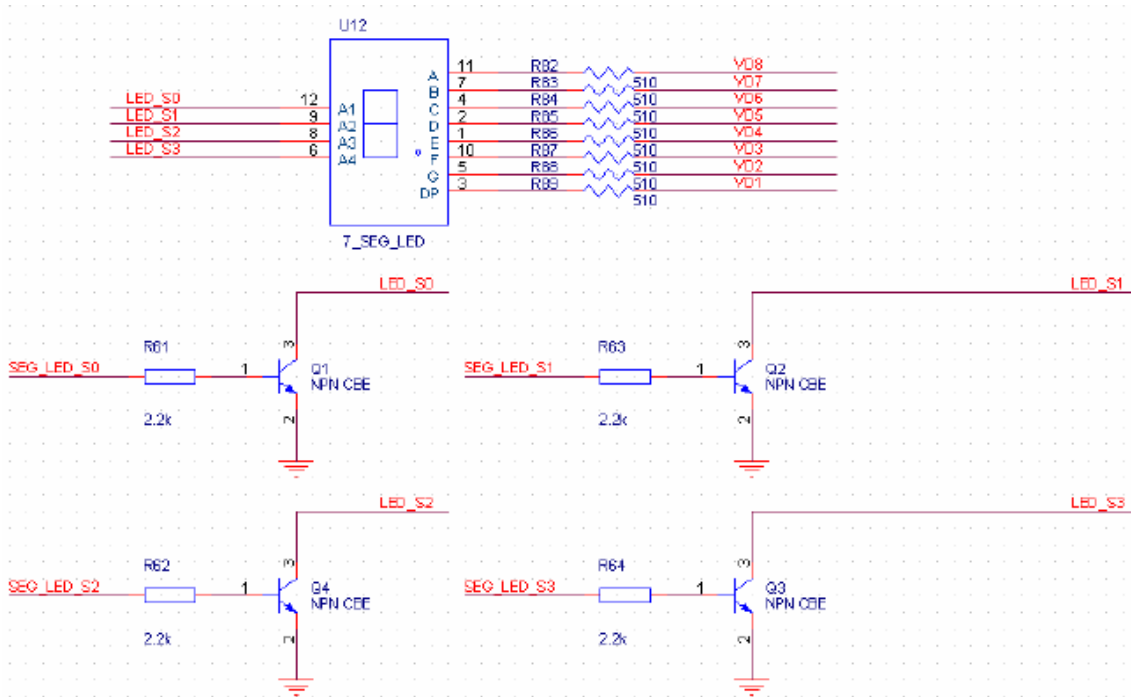
ピンマップ：

デバイス	信号名称	FPGAピン番号
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：



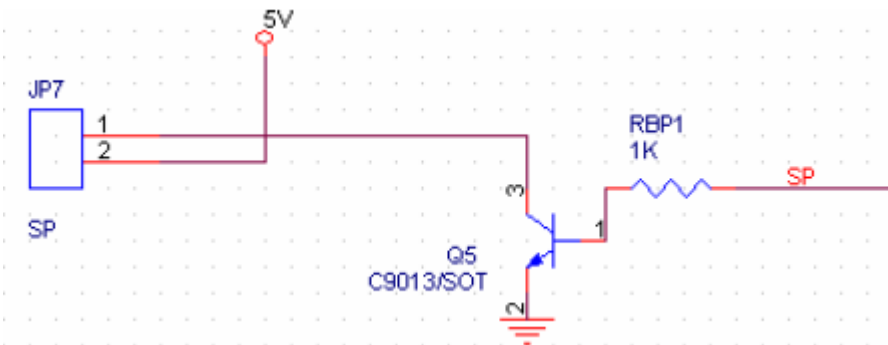
ピンマップ：

信号名称	FPGAピン番号	信号名称	FPGAピン番号
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：

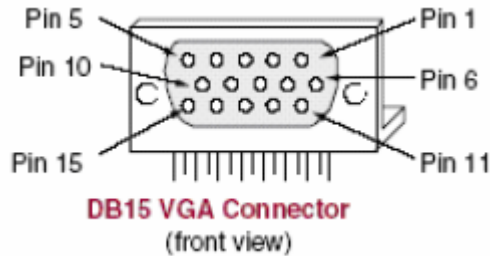


ピンマップ：

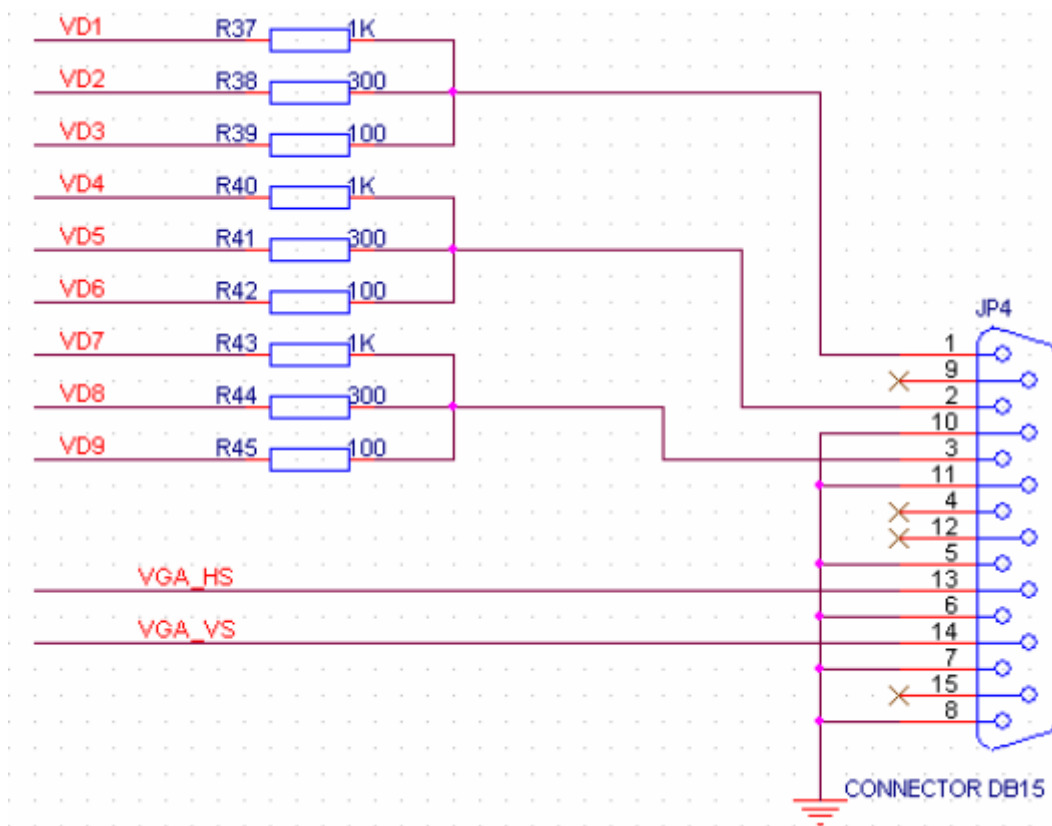
信号名称  
SP

FPGAピン番号  
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。

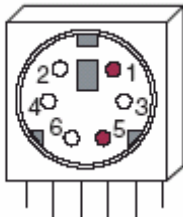
VGA\_HSとVGA\_VS信号はTTLレベルでVGAの水平と縦のスキャン信号を制御する。

ピンマップ：

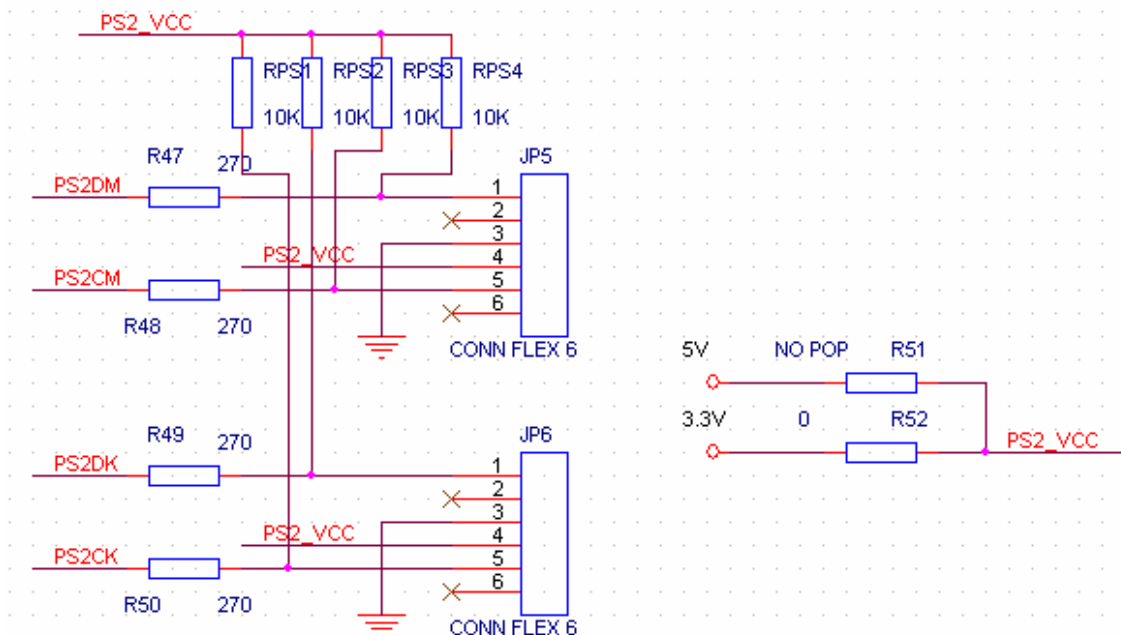
Signal	FPGA Pin
VD1	Pin 194
VD2	Pin 196
VD3	Pin 197
VD4	Pin 198
VD5	Pin 199
VD6	Pin 200
VD7	Pin 203
VD8	Pin 204
VD9	Pin 205
VGA_HS	Pin 7
VGA_VS	Pin 9

## 2.7 PS/2 インタフェース

PS/2 DINインタフェース：



回路図：



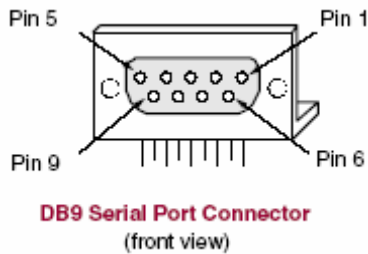
ピンマップ：

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

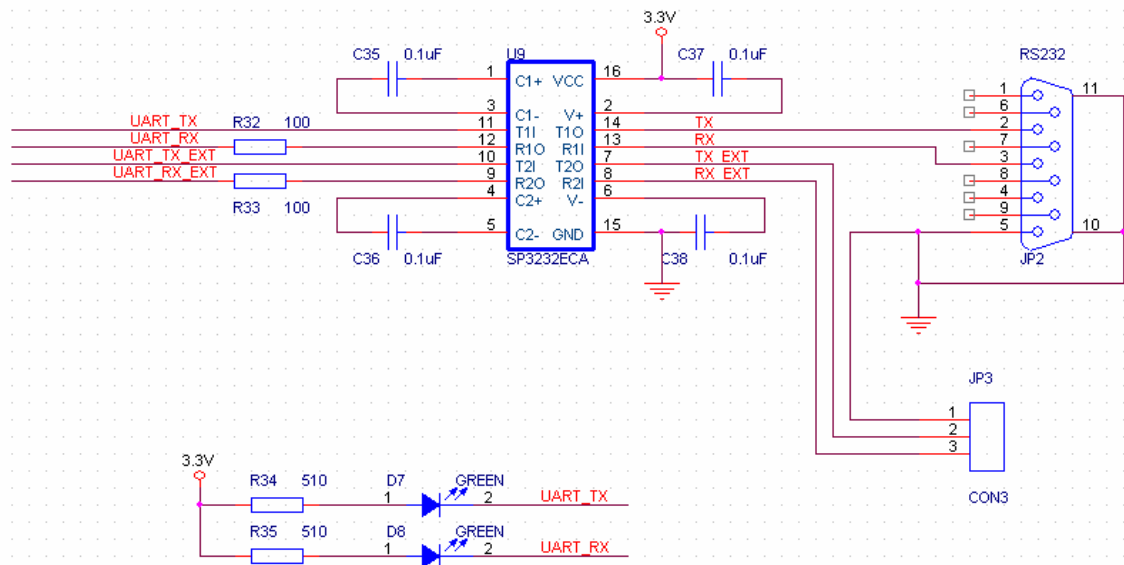
## 2.8 RS232 シリアルポートインタフェース

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：



回路図：



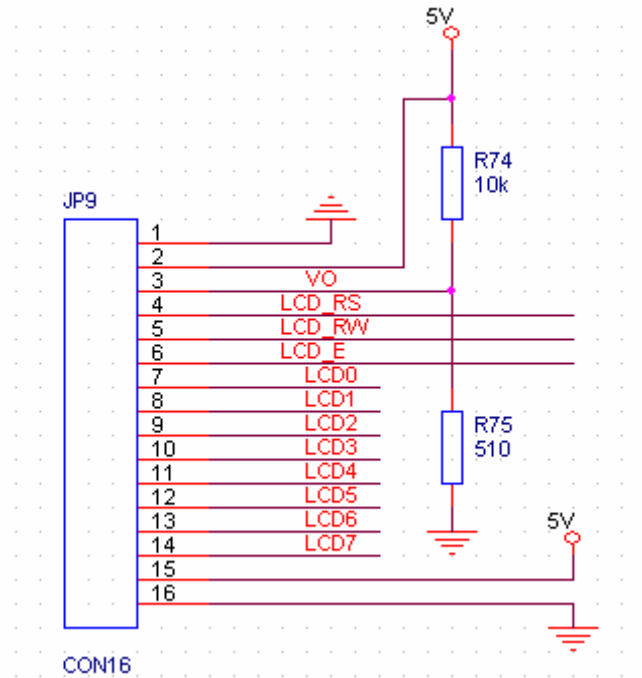
ピンマップ：

Device	Signal	FPGA Pin
U1	UART_TX	Pin 18
U1	UART_RX	Pin 16
U1	UART_TX_EXT	Pin 20
U1	UART_RX_EXT	Pin 19

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	LCD0	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。

高 位 低 位	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111	
××××0000	CGRAM (1)		0	a	P	\	p		-	タ	三	a	P	
××××0001	(2)	!	1	A	Q	a	q	□	ア	チ	ム	ä	q	
××××0010	(3)	"	2	B	R	b	r	□	イ	川	ノ	β	θ	
××××0011	(4)	#	3	C	S	c	s	□	ウ	ラ	モ	e	∞	
××××0100	(5)	\$	4	D	T	d	t	\	エ	ト	セ	μ	Ω	
××××0101	(6)	%	5	E	U	e	u	□	オ	ナ	ユ	B	0	
××××0110	(7)	&	6	F	V	f	v	□	カ	ニ	ヨ	P	Σ	
××××0111	(8)	>	7	G	W	g	w	□	ア	キ	ヌ	ラ	g	κ
××××1000	(1)	(	8	H	X	h	x	□	イ	ク	ネ	リ	f	X
××××1001	(2)	)	9	I	Y	i	y	□	ウ	ケ	】	ル	-1	y
××××1010	(3)	*	:	J	Z	j	z	□	エ	コ	リ	レ	j	千
××××1011	(4)	+	;	K	[	k	{	□	オ	サ	ヒ	ロ	x	万
××××1100	(5)	フ	<	L	¥	l		□	セ	シ	フ	ワ	¢	円
××××1101	(6)	-	=	M	]	m	}	□	ユ	ス	へ	ソ	モ	+
××××1110	(7)	.	>	N	^	n	-	□	ヨ	セ	ホ	ハ	ñ	
××××1111	(8)	/	?	O	-	o	←	□	ツ	ソ	マ	ロ	Ö	

コマンド表：



Instruction	Code										Description	Execution Time (max) (when $f_{op}$ or $f_{osc}$ is 270 kHz)	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Clear display	0	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	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	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 $\mu$ s
Display on/off control	0	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 $\mu$ s
Cursor or display shift	0	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 $\mu$ s
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 $\mu$ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 $\mu$ s
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 $\mu$ s
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 $\mu$ s
Write data to CG or DDRAM	1	0	Write data									Writes data into DDRAM or CGRAM.	37 $\mu$ s $t_{ADD} = 4 \mu\text{s}^*$
Read data from CG or DDRAM	1	1	Read data									Reads data from DDRAM or CGRAM.	37 $\mu$ s $t_{ADD} = 4 \mu\text{s}^*$
			I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8 bits, DL = 0: 4 bits N = 1: 2 lines, N = 0: 1 line F = 1: 5 $\times$ 10 dots, F = 0: 5 $\times$ 8 dots BF = 1: Internally operating BF = 0: Instructions acceptable									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 $f_{op}$ or $f_{osc}$ is 250 kHz, $37 \mu\text{s} \times \frac{270}{250} = 40 \mu\text{s}$

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,  $t_{ADD}$  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：

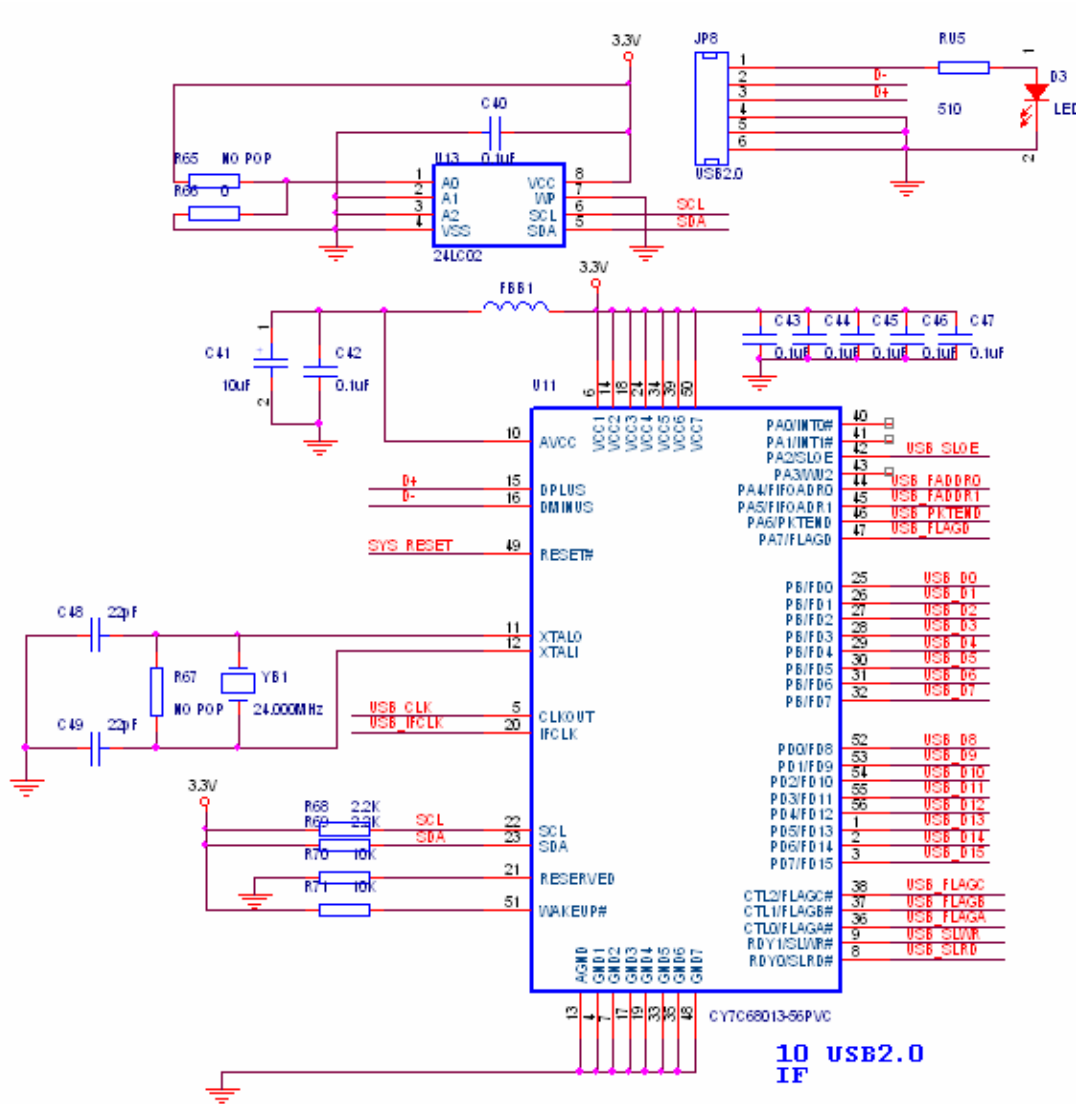
<http://www.dragonwake.com/download/LCD/hd44780u.pdf>



## 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：

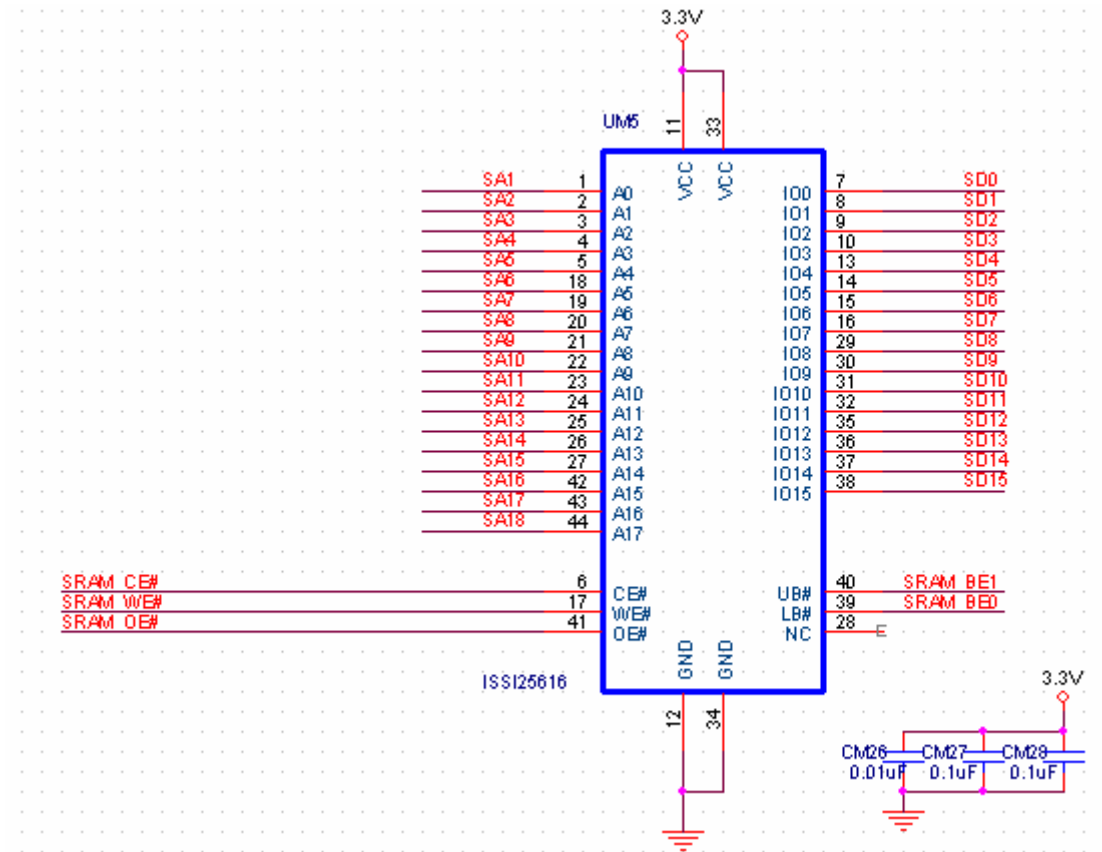
CY7C18013 Pin	Signal	FPGA Pin
RDY0/SLRD	USB_SLRD	Pin 86
RDY1/SLWR	USB_SLWR	Pin 85
CTL0/FLAGA (pf半分)	USB_FLAGA	Pin 95
CTL1/FLAGB (full)	USB_FLAGB	Pin 94
CTL2/FLAGC (empty)	USB_FLAGC	Pin 78
PA7/FLAGD	USB_FLAGD	Pin 67
PA2/SLOE	USB_SLOE	Pin 74
PA6/PKTEND	USB_PKTEND	Pin 68
PA4/FIFOADR0	USB_FADDR0	Pin 72
PA5/FIFOADR1	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
A0	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



A16	Pin 166	SA17
A17	Pin 165	SA18

**データ信号**

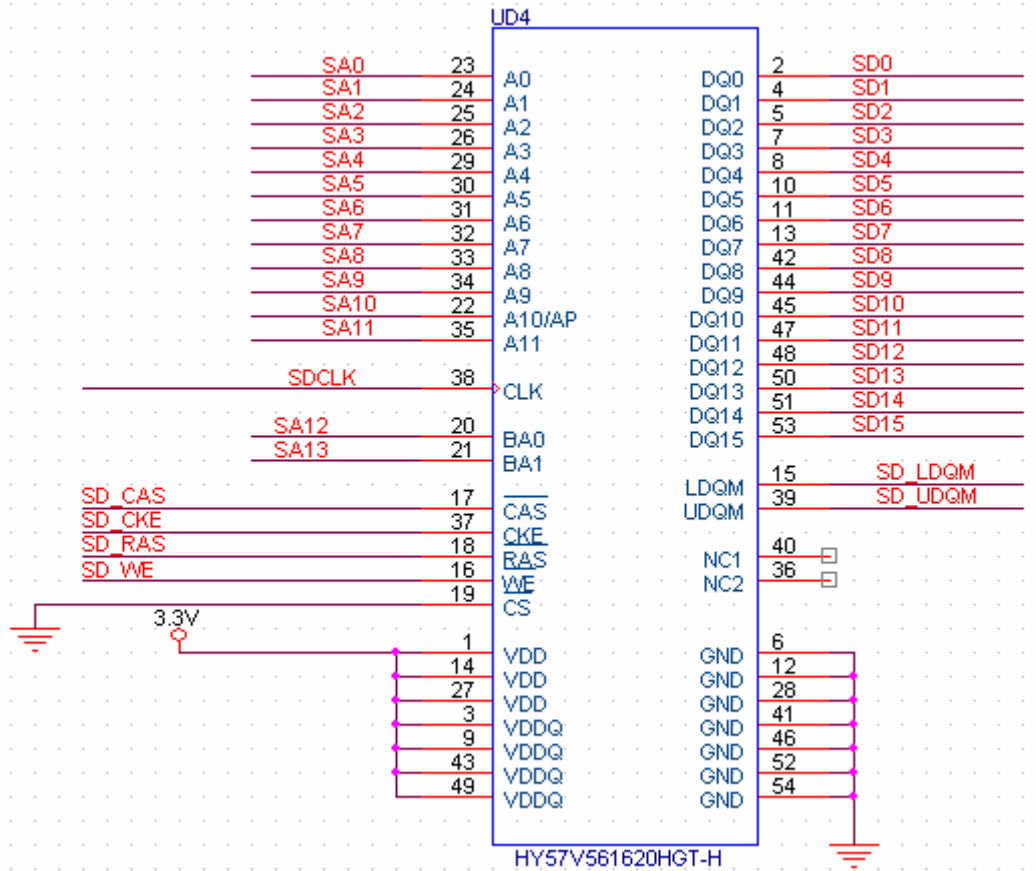
SRAM Pin	FPGA Pin	External Bus Signal
sram_db[0]	Pin 102	SD0
sram_db[1]	Pin 106	SD1
sram_db[2]	Pin 107	SD2
sram_db[3]	Pin 108	SD3
sram_db[4]	Pin 109	SD4
sram_db[5]	Pin 111	SD5
sram_db[6]	Pin 122	SD6
sram_db[7]	Pin 123	SD7
sram_db[8]	Pin 124	SD8
sram_db[9]	Pin 125	SD9
sram_db[10]	Pin 126	SD10
sram_db[11]	Pin 128	SD11
sram_db[12]	Pin 130	SD12
sram_db[13]	Pin 131	SD13
sram_db[14]	Pin 132	SD14
sram_db[15]	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
BE0	SRAM_ BE0	Pin 137
BE1	SRAM_ BE1	Pin 138

**2.12 高速同期SDRAM**

回路図：



### アドレス信号：

SDRAM Pin	FPGA Pin	External Bus Signal
A0	Pin 191	SA0
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
BA0	Pin 172	SA12
BA1	Pin 171	SA13

### データ信号



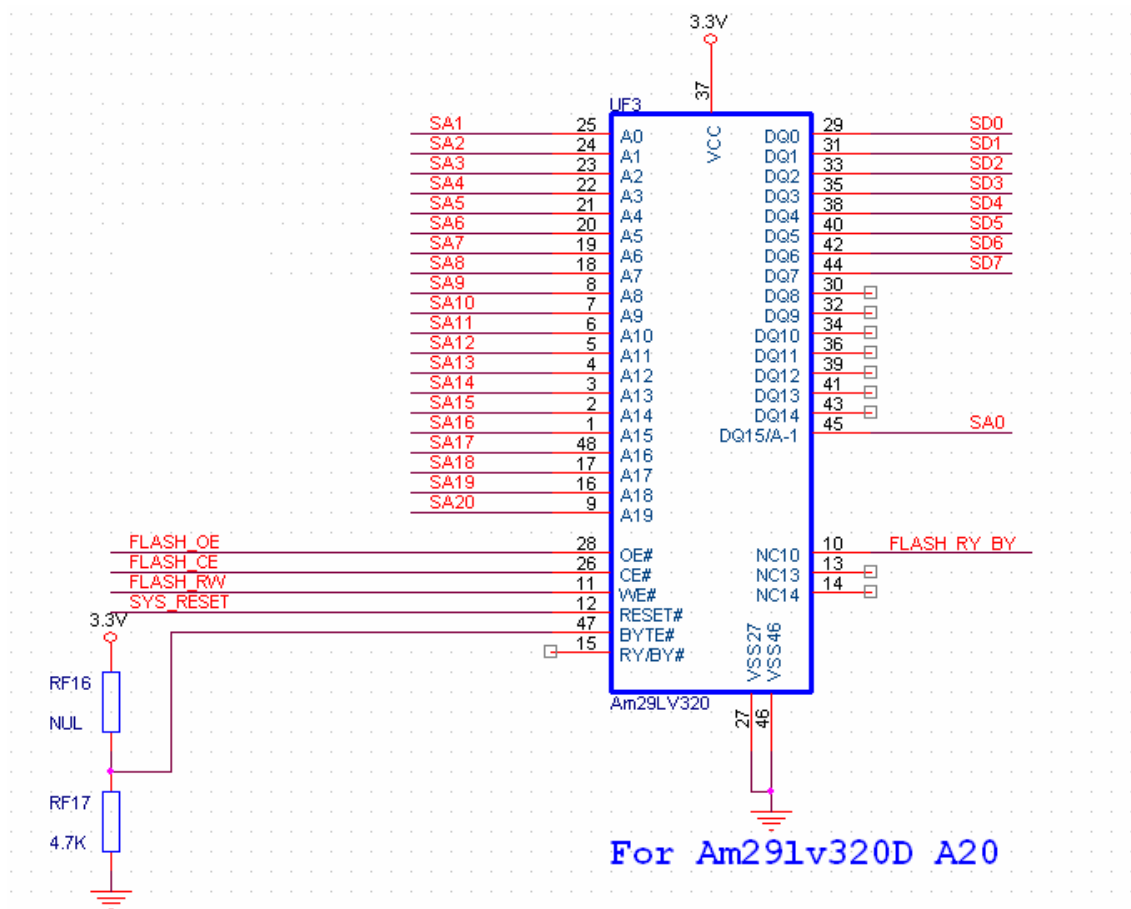
SDRAM Pin	FPGA Pin	External Bus Signal
DQ0	Pin 102	SD0
DQ1	Pin 106	SD1
DQ2	Pin 107	SD2
DQ3	Pin 108	SD3
DQ4	Pin 109	SD4
DQ5	Pin 111	SD5
DQ6	Pin 122	SD6
DQ7	Pin 123	SD7
DQ8	Pin 124	SD8
DQ9	Pin 125	SD9
DQ10	Pin 126	SD10
DQ11	Pin 128	SD11
DQ12	Pin 130	SD12
DQ13	Pin 131	SD13
DQ14	Pin 132	SD14
DQ15	Pin 133	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：**

Flash Pin	FPGA Pin	External Bus Signal
A0	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
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
DQ0	Pin 102	SD0
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	SA0

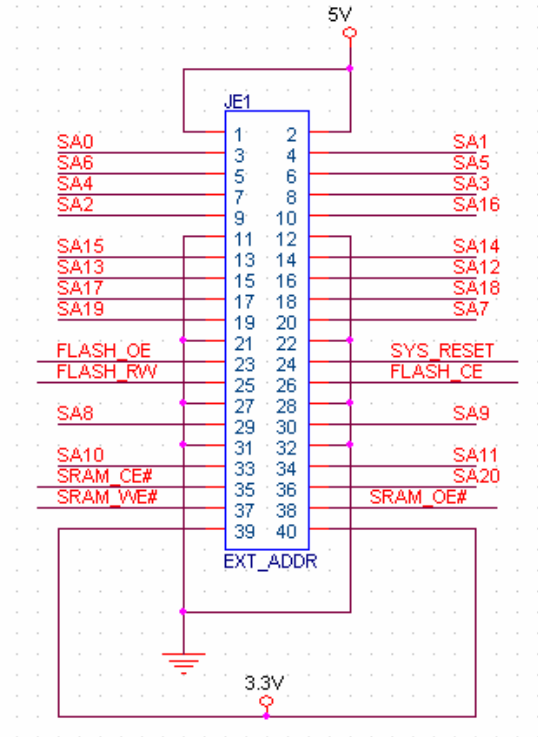
#### コントロール信号

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：



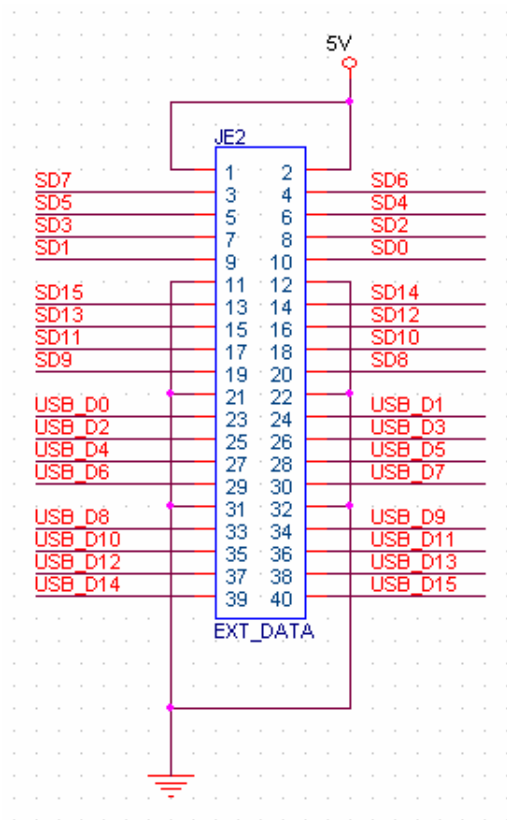


ピンマップ：

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：

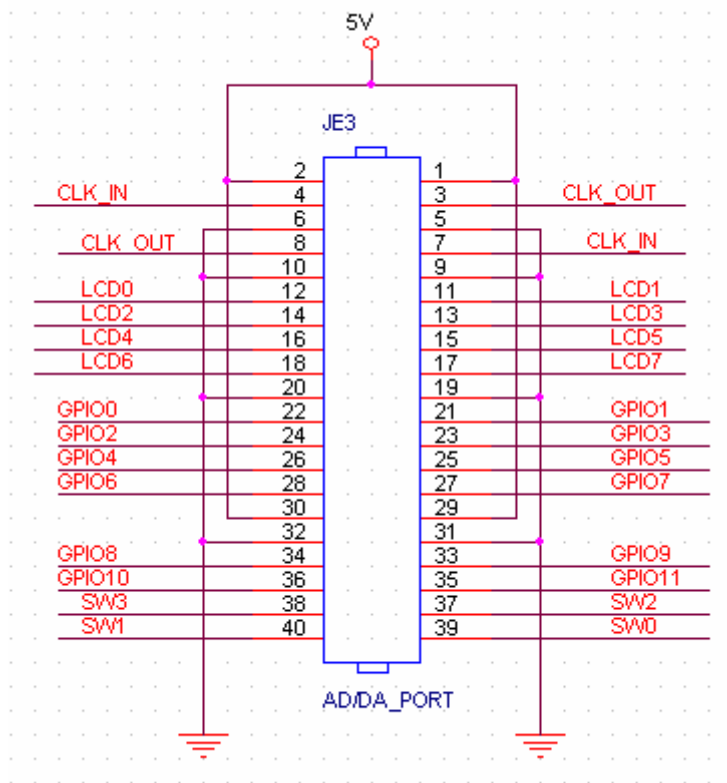


ピンマップ：

External Bus Signal	FPGA Pin	External Bus Signal	FPGA Pin
SD0	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：



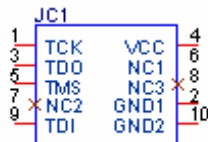
ピンマップ：

External Bus Signal	FPGA Pin	External Bus Signal	FPGA Pin
CLK_IN	Pin 79	CLK_OUT	Pin 81
LCD0	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
SW0	Pin 58	SW1	Pin 47
SW2	Pin 52	SW3	Pin 51

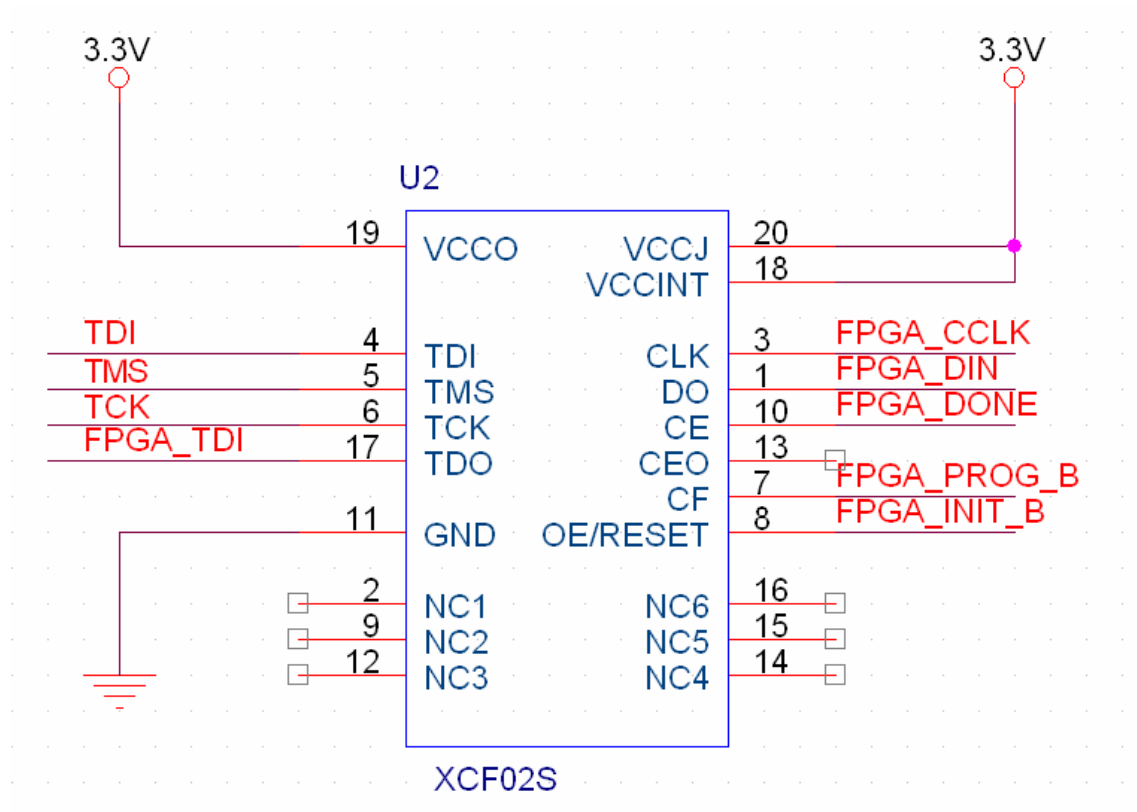
## 2.17 JTAGインターフェース

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。

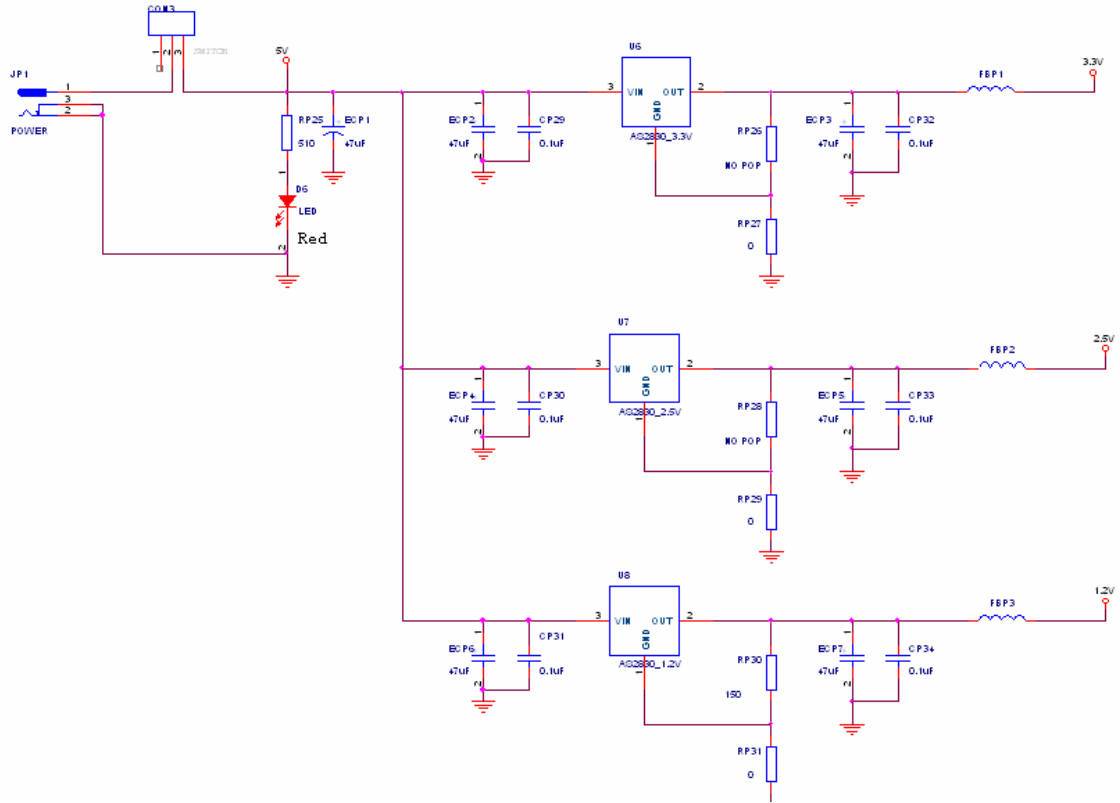
回路図：



XCF02S を利用する場合は PROM モードでダウンロードする。

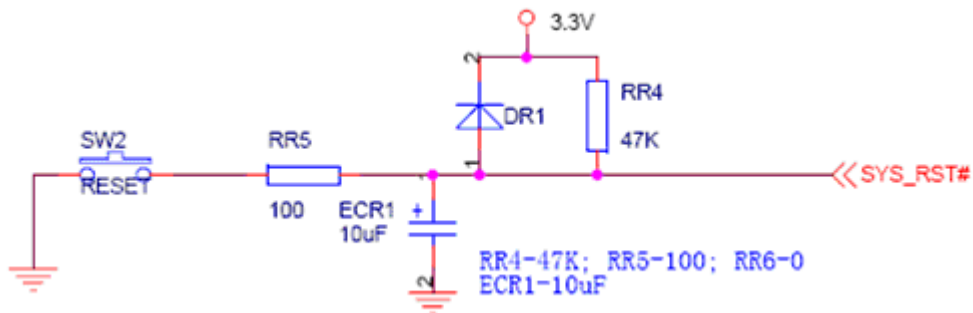


## 2.18 電源回路



## 2.19 リセット回路

回路図：

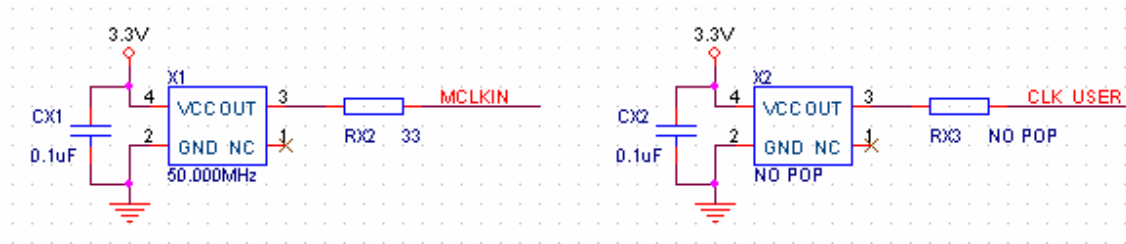


ピンマップ：

Device	Signal	FPGA Pin
UR1-7	SYS_RESET	Pin 152

## 2.20 クロック

回路図：



ピンマップ：

Clock	Signal	FPGA Pin
50.000MHz	MCLKIN	Pin 76
ユーザオプション	GPI011	Pin 77

説明：

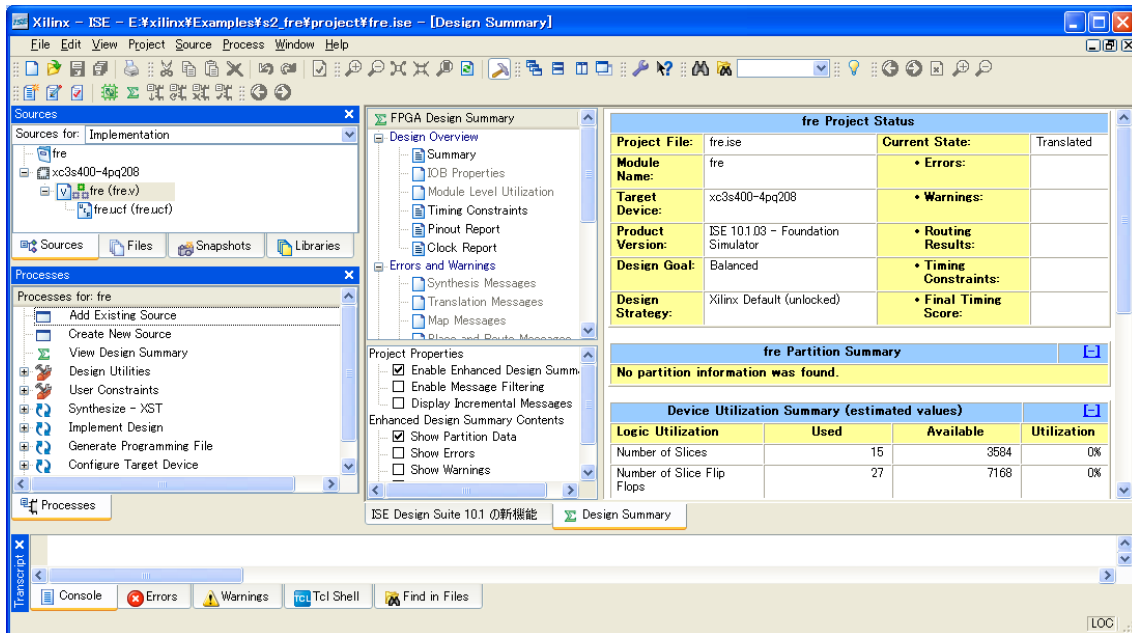
FPGA 内部は DLL があって、50MHz のクロックをニーズに応じて必要なクロックを分頻できる。特殊なクロックは 50MHz で生成できない場合はユーザオプションを利用ください。

## 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。

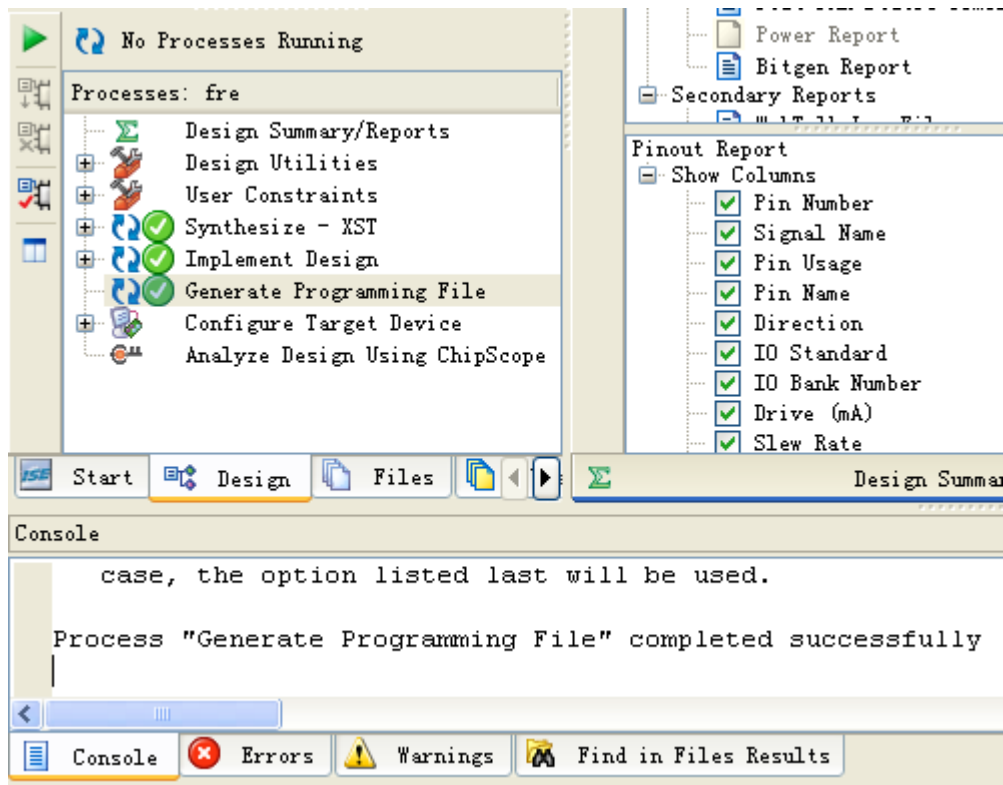


### 3.2 ボードに書き込み

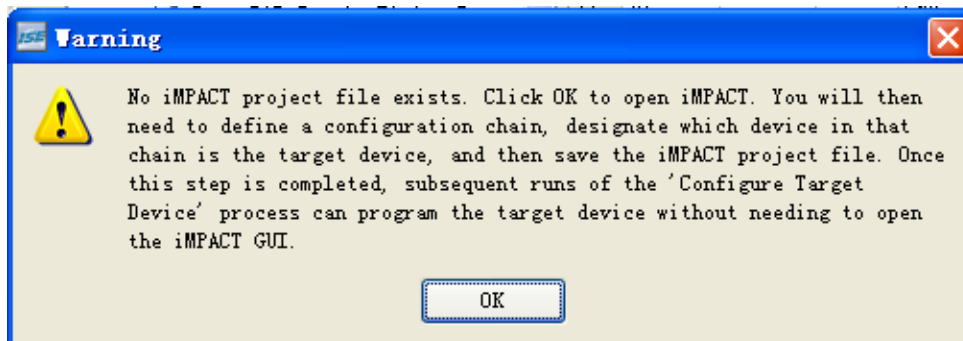
2種類の方法があります：JTAGモードとPROMモード。JTAGモードでダウンロードする場合は電源切ると情報が保存されない。PROMモードの場合はメモリに書き込まれて、電源入ると自動的にロードして実行する。

#### 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。

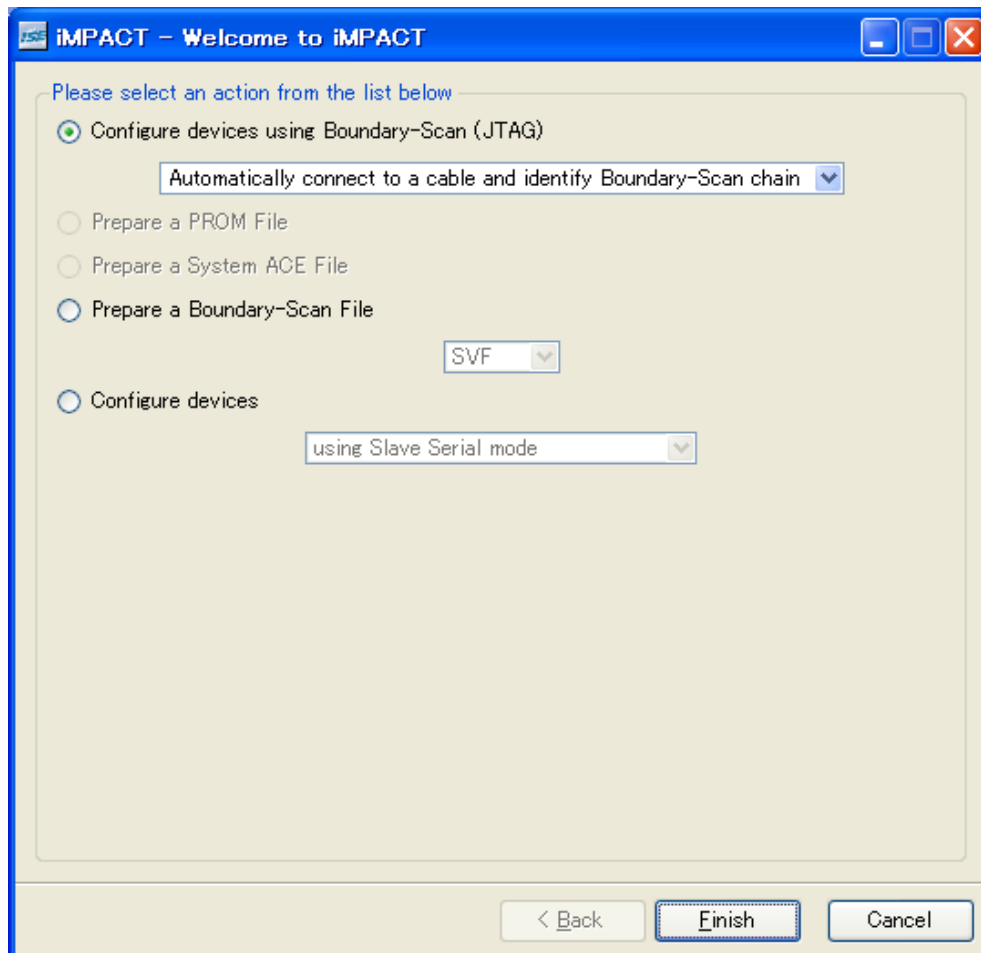
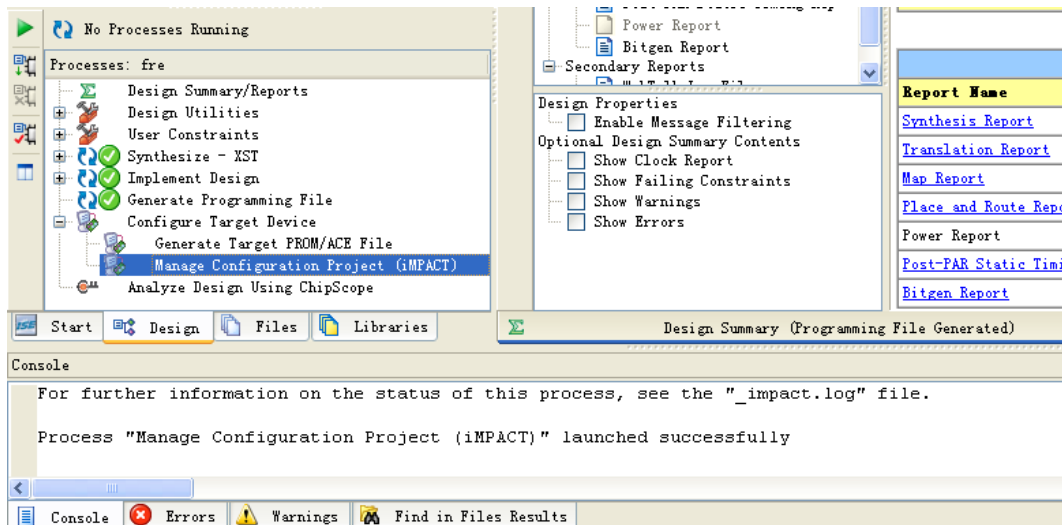


Configure Target Device をダブルクリックする。

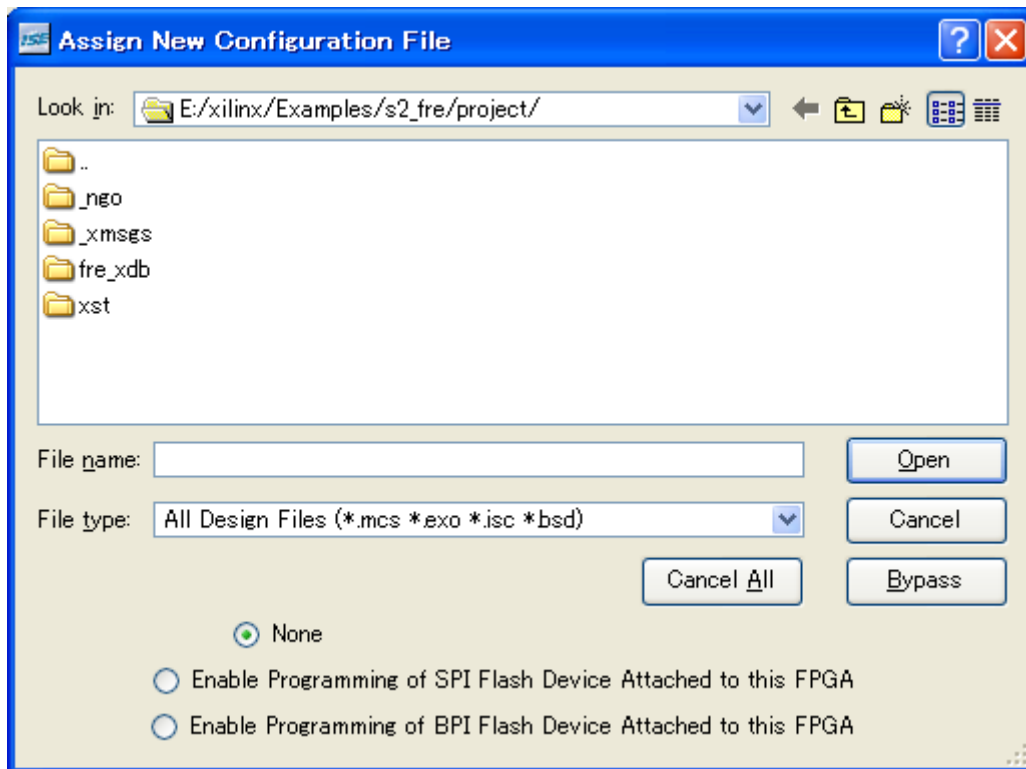


OK を押す。或いは Configure Target Device 下にある Manage Configuration Project (iMPACT) をダブルクリックする。

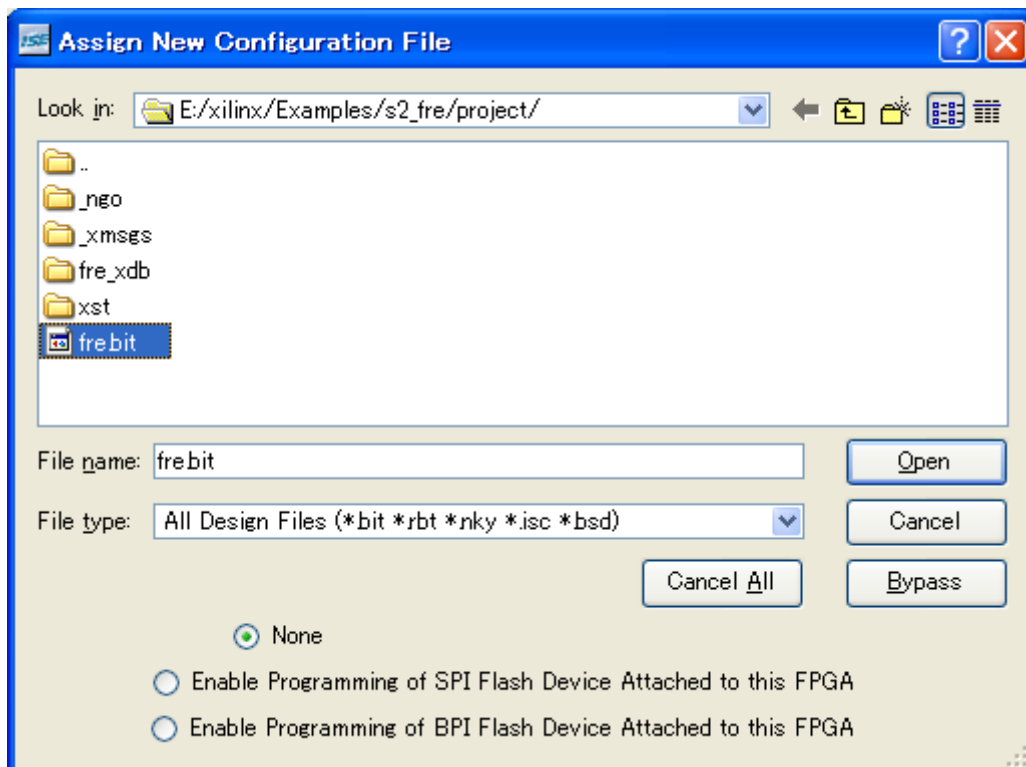




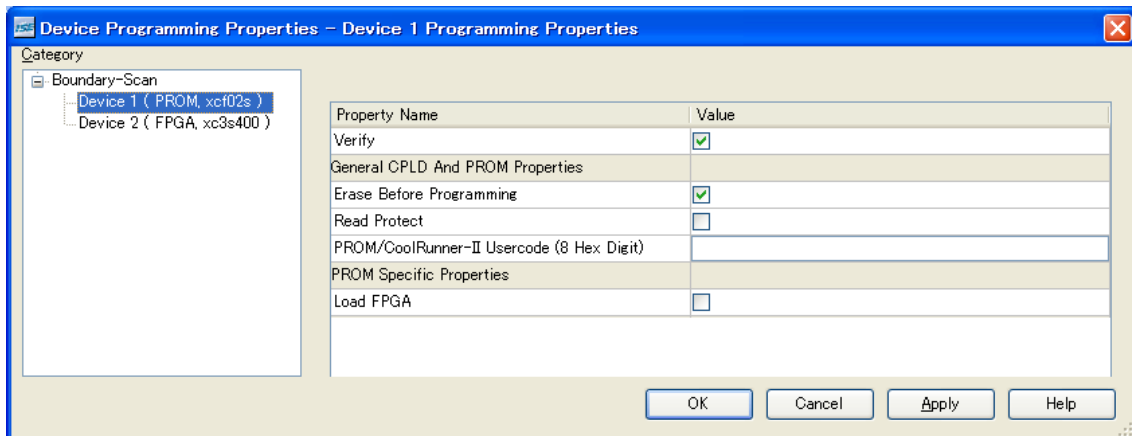
JTAG モードを選択して、“Automatically connect to a cable and identify Boundary-Scan chain” のデフォルトの設定で Finish を押す。



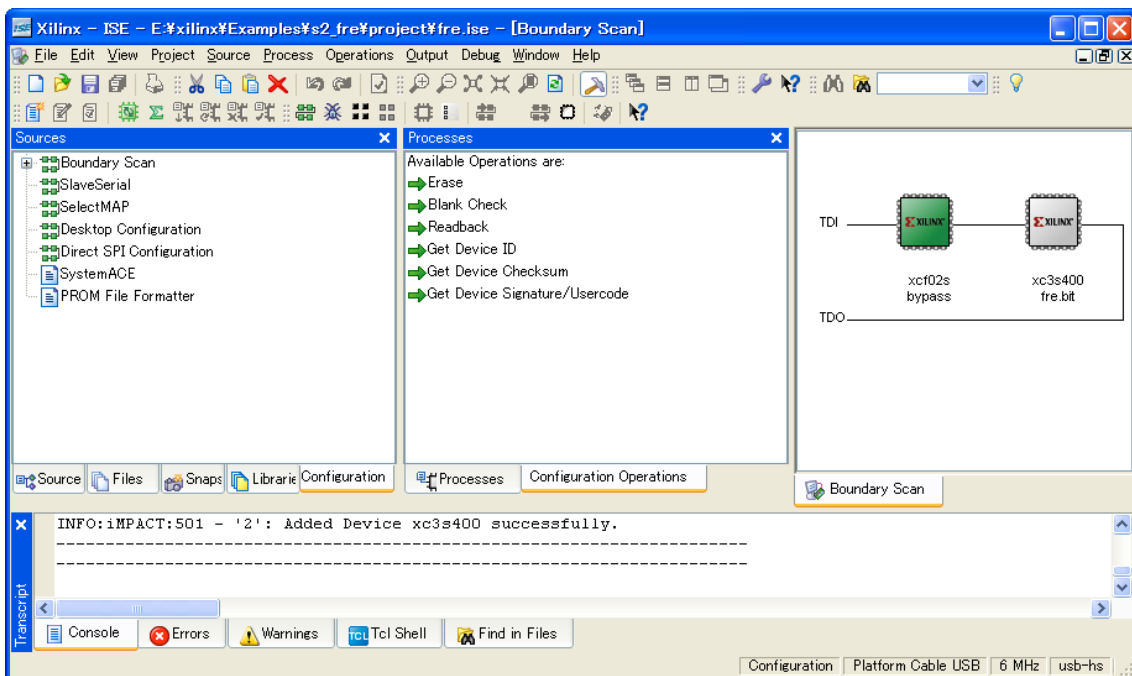
Bypass を押す。



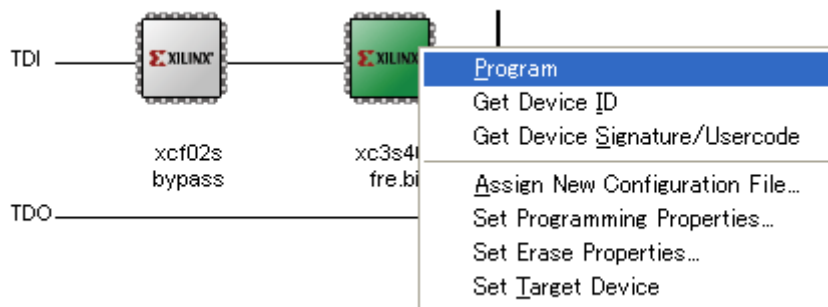
Open を押す。

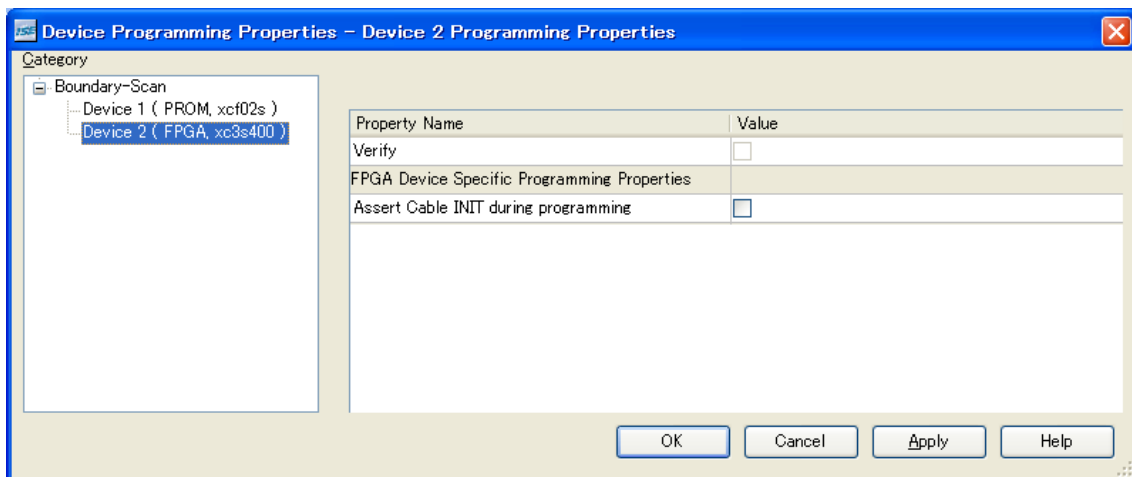


デフォルトの設定でOKを押す。



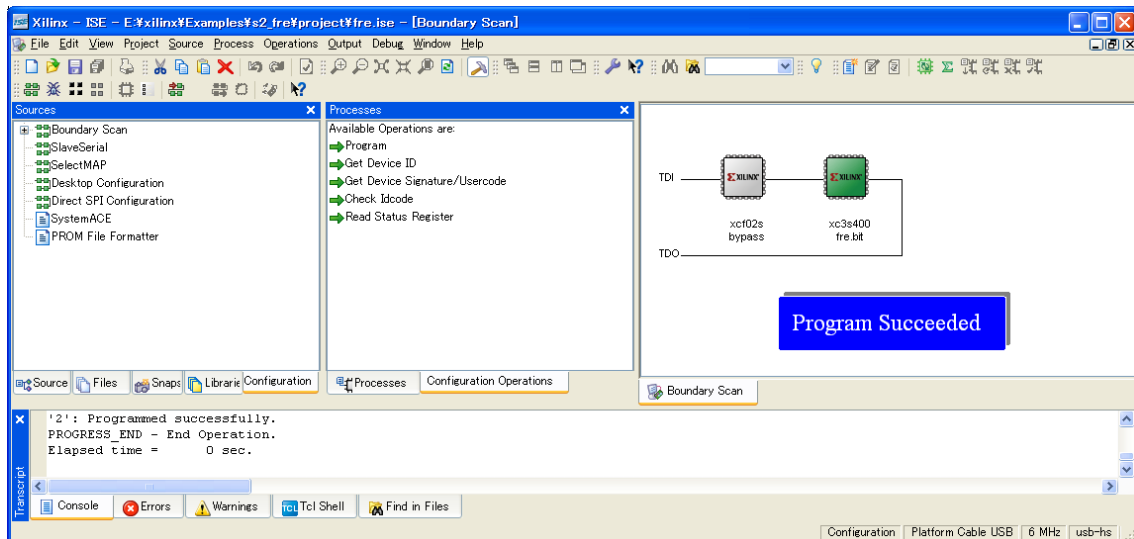
FPGA を選択して右クリックする。





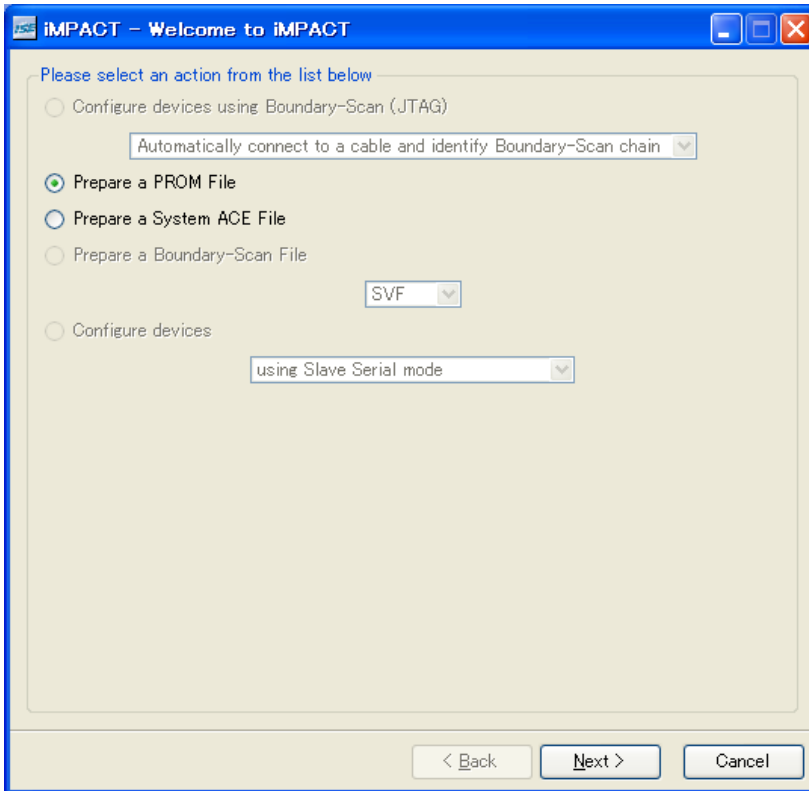
OK を押す。

ダウンロード完了後の画面：

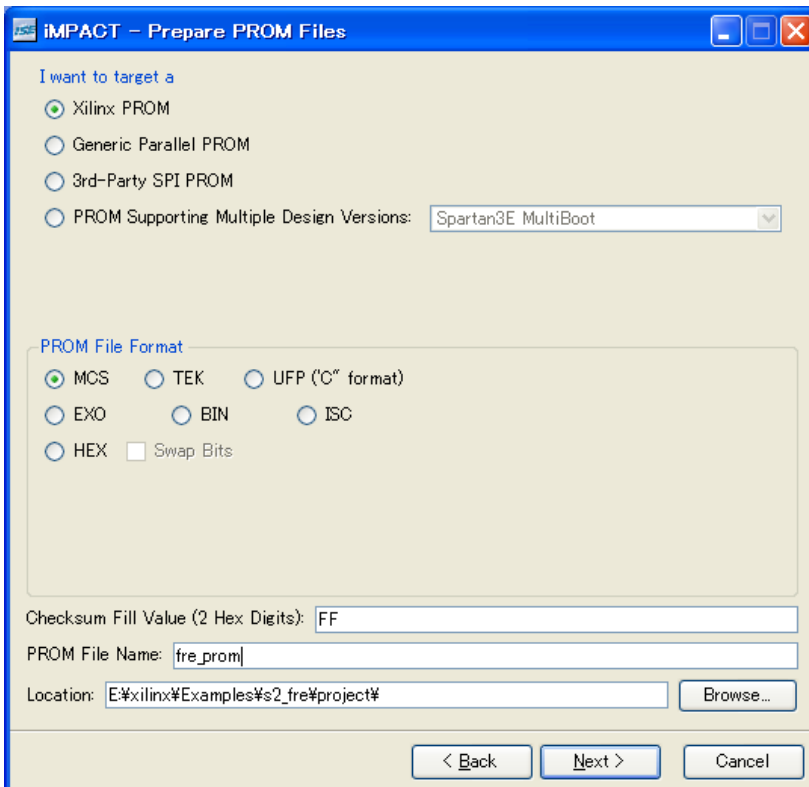


### 3.2.2 PROMモードで書き込む

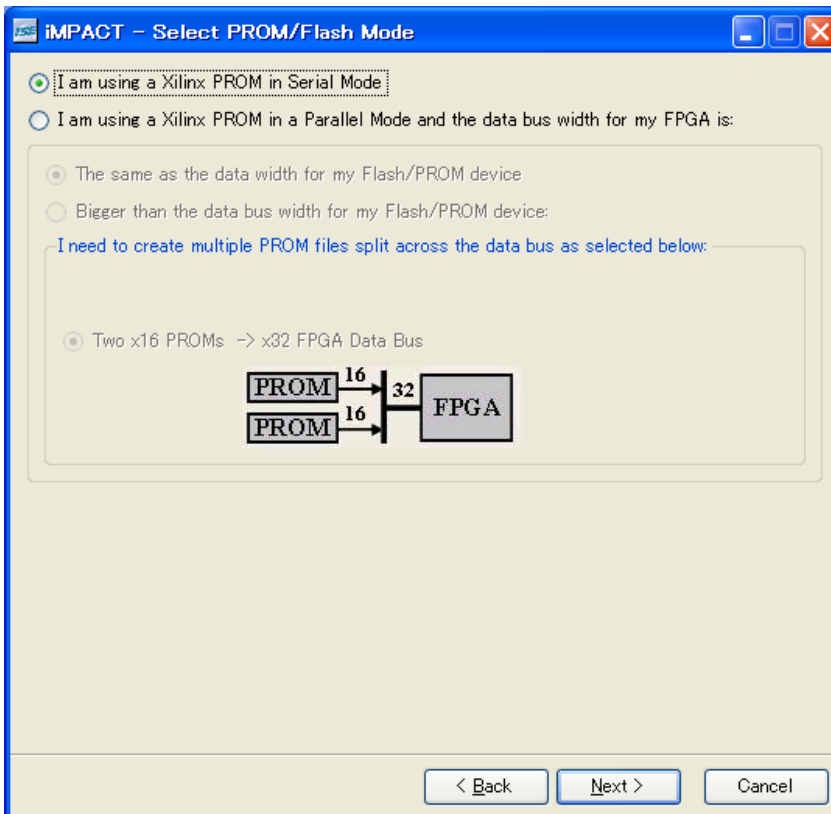
Processes 欄の Generate Target PROM/ACE File をダブルクリックする。



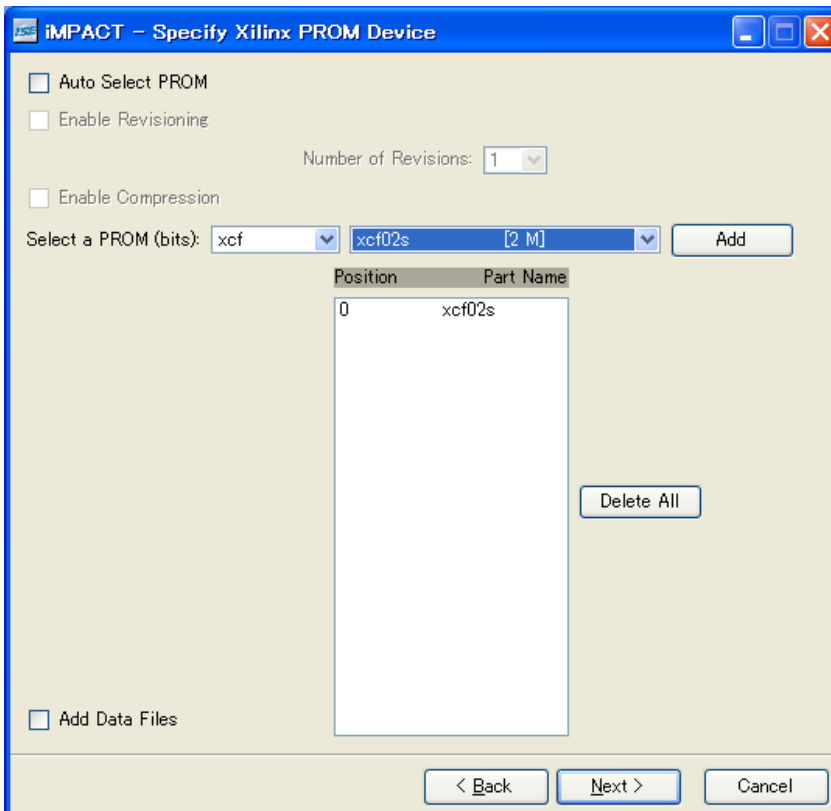
Next を押す。



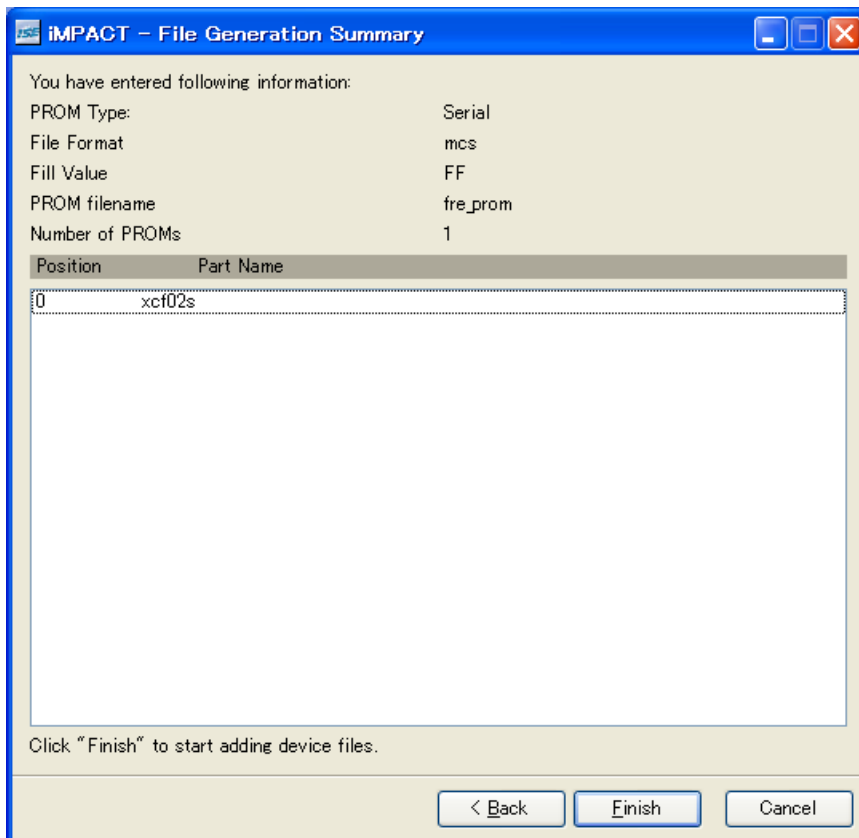
File Name を入力する。その他はデフォルトで Next を押す。



Next を押す。



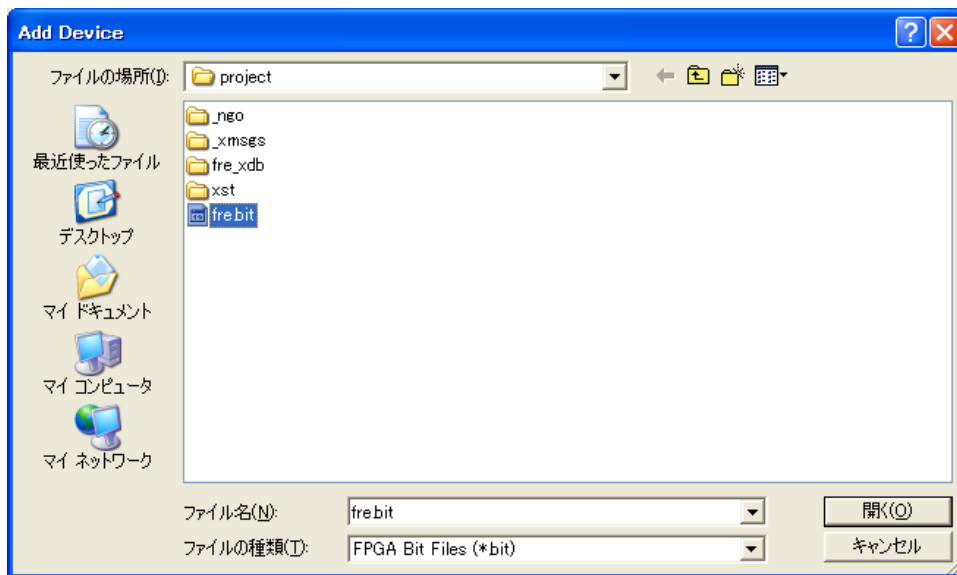
Xcf02s を選択して Add を押す。Next を押す。



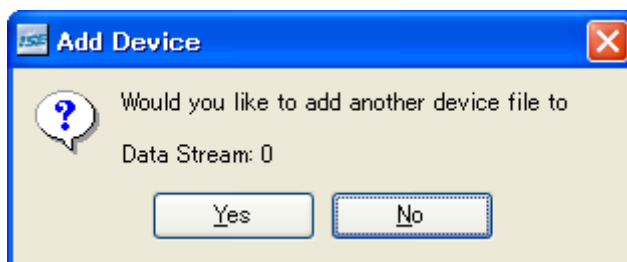
Finish を押す。



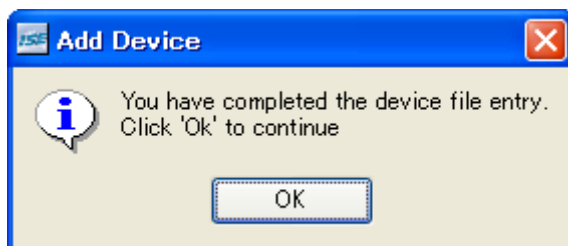
OK を押す。



FPGA Bit ファイルを選択して“開く”を押す。

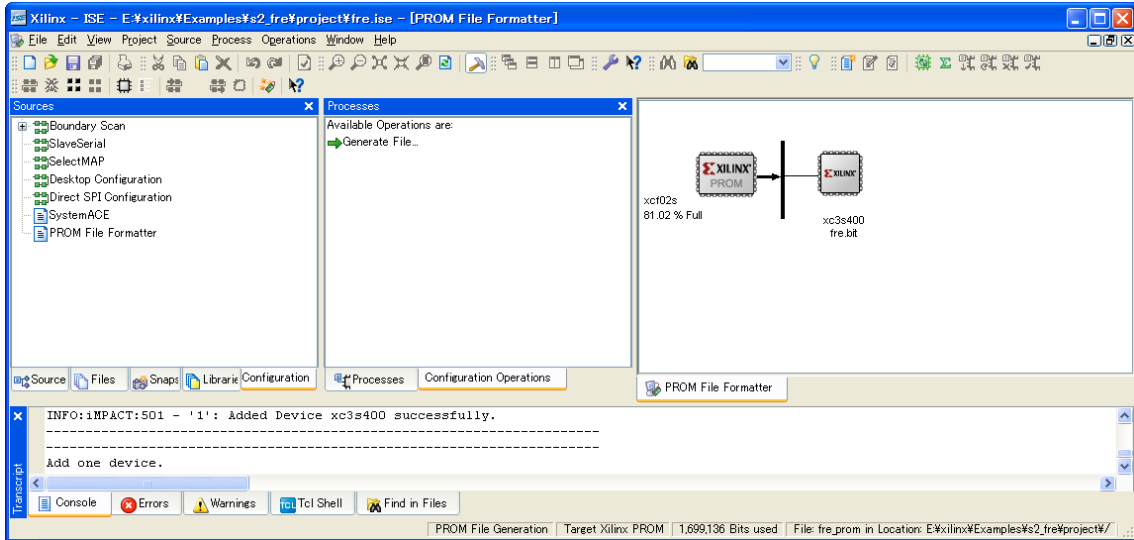


No を押す。

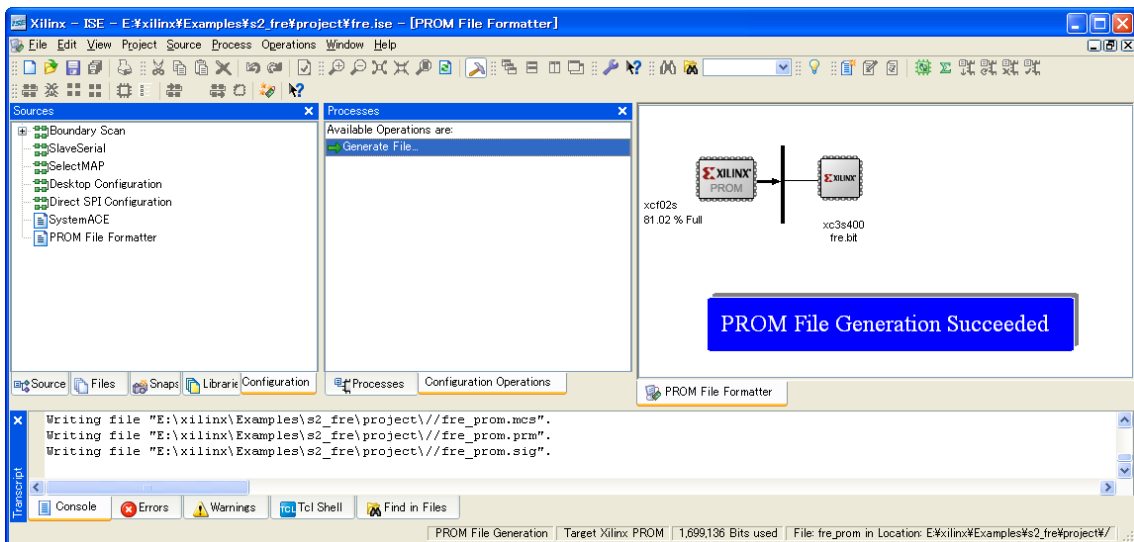


OK を押す。





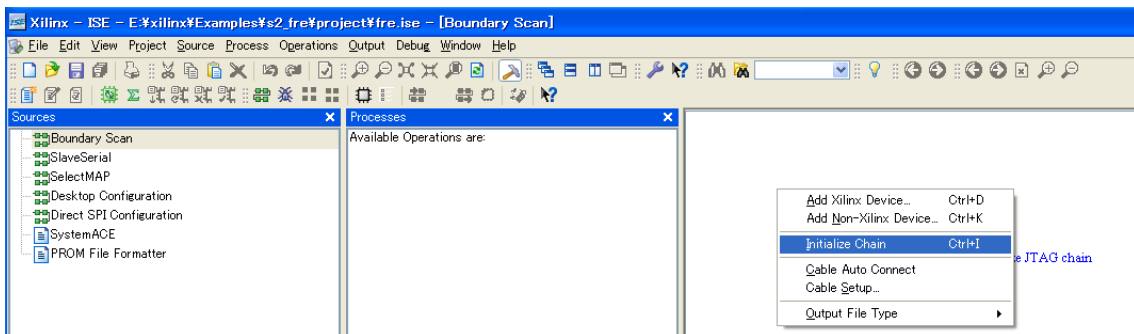
Processes 中の Generate File... をダブルクリックする。

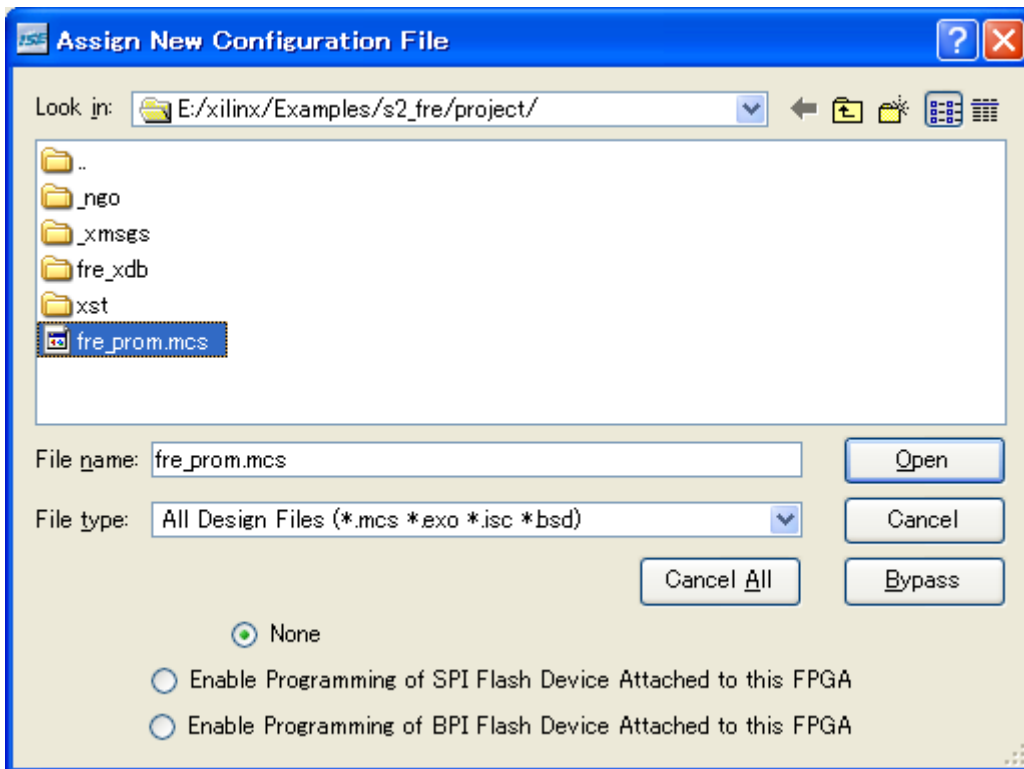


これで PROM モードで書き込み用の .mcs ファイルが生成される。

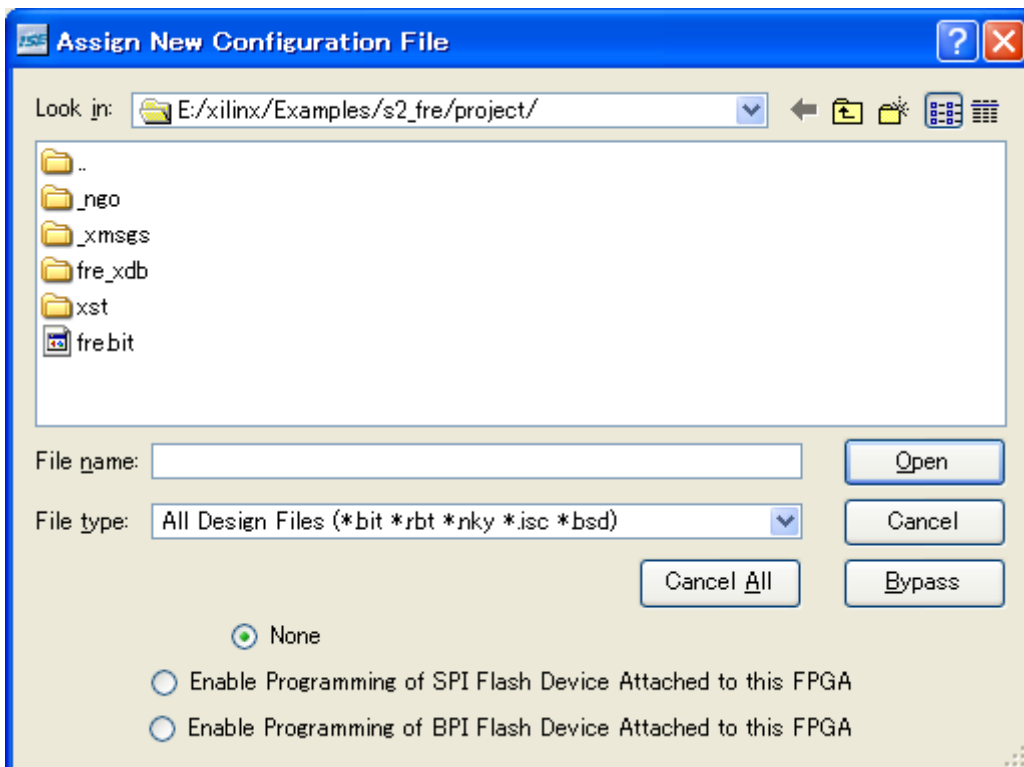
Boundary Scan をクリックする。

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

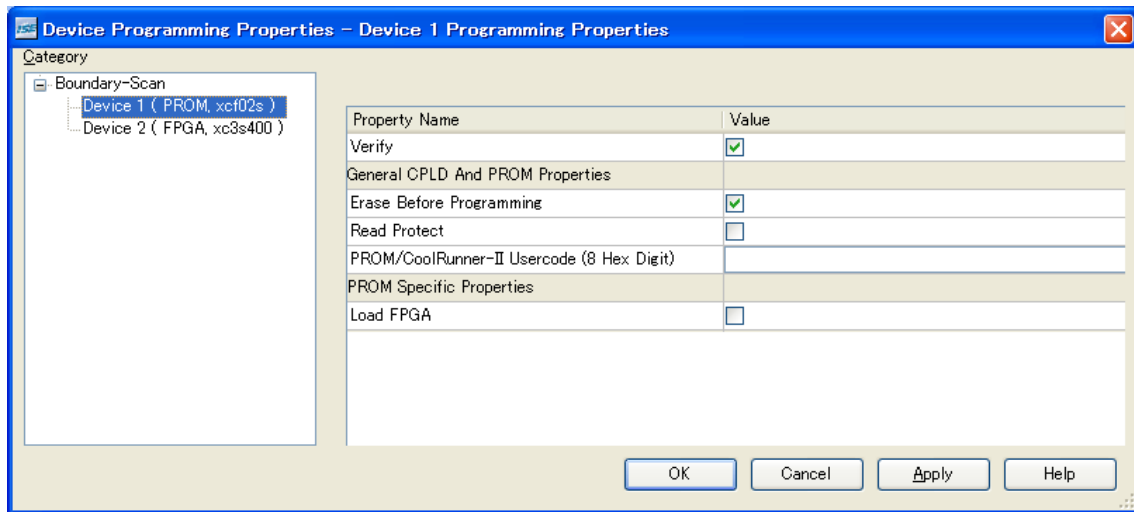




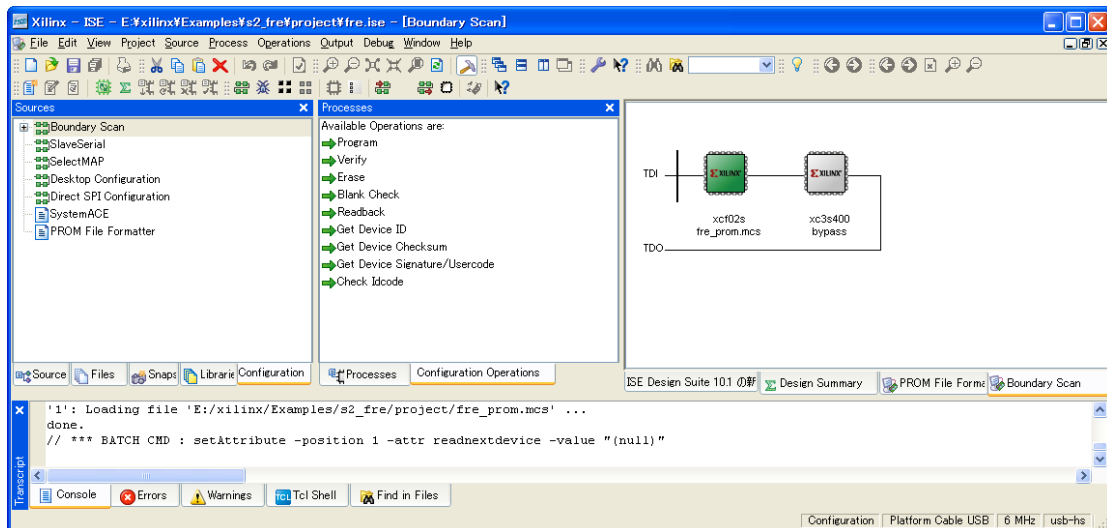
先ほど生成したファイルを指定して Open を押す。



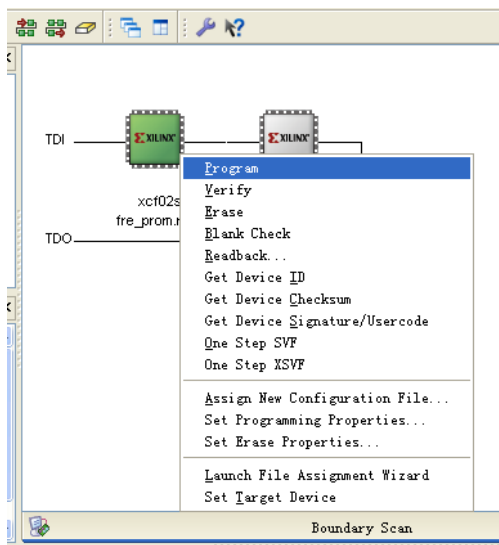
Bypass を押す。

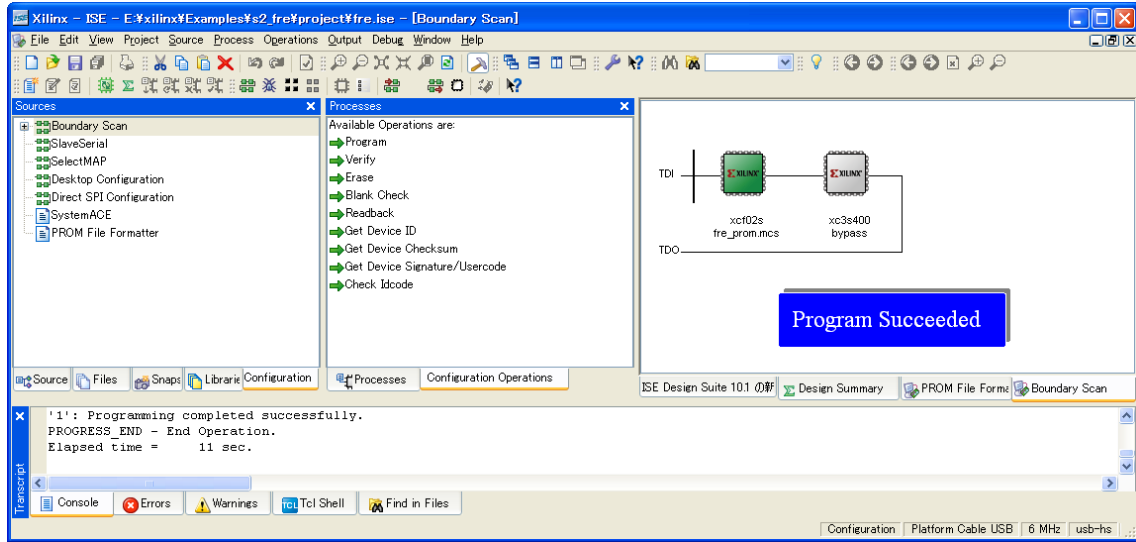


OK を押す。



Xcf02s を右クリックして Program を押す。





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。

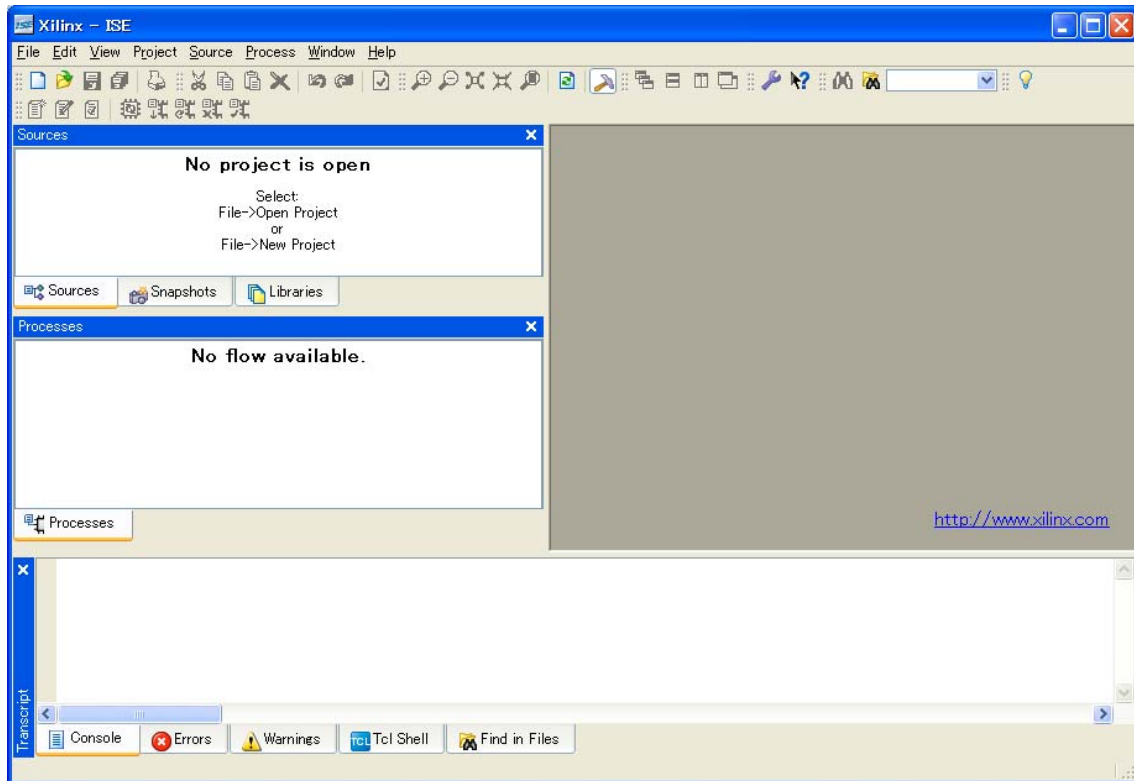
実現機能：ボード上のキーで LED を下記ロジックで制御する。

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
S3	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
S5	0	0	1	1	0	0	0	1	0	0	0	0
S6	0	1	1	0	0	0	1	0	0	0	0	0
S7	1	1	0	0	0	1	0	0	0	0	0	0
S8	1	0	0	1	1	0	0	0	0	0	0	0
Default					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。

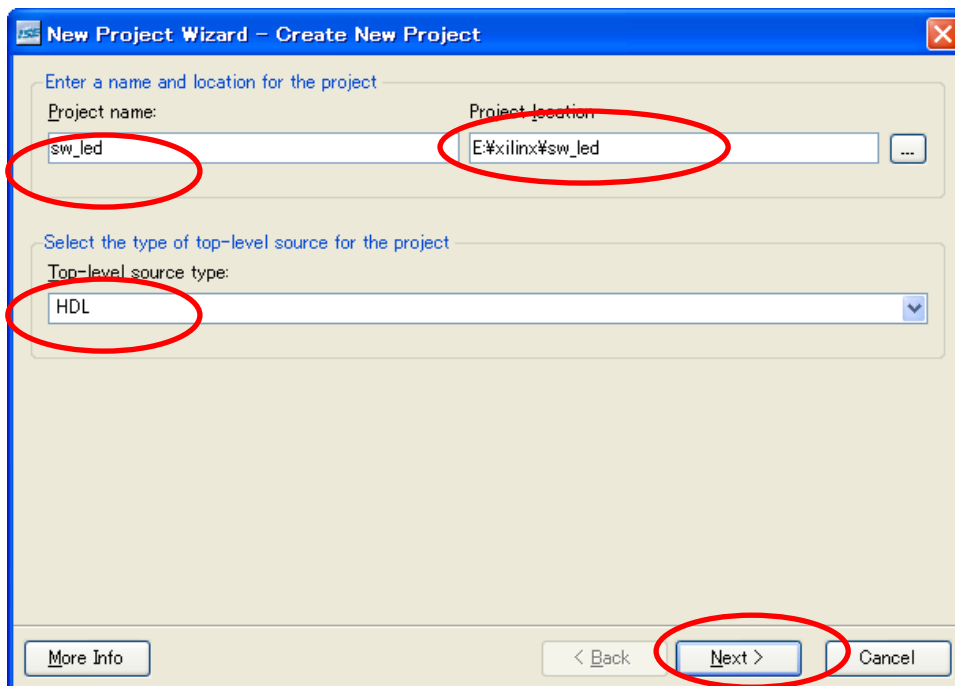
スタート→すべてのプログラム→Xilinx ISE 10.1i→Project Navigator をクリックする。



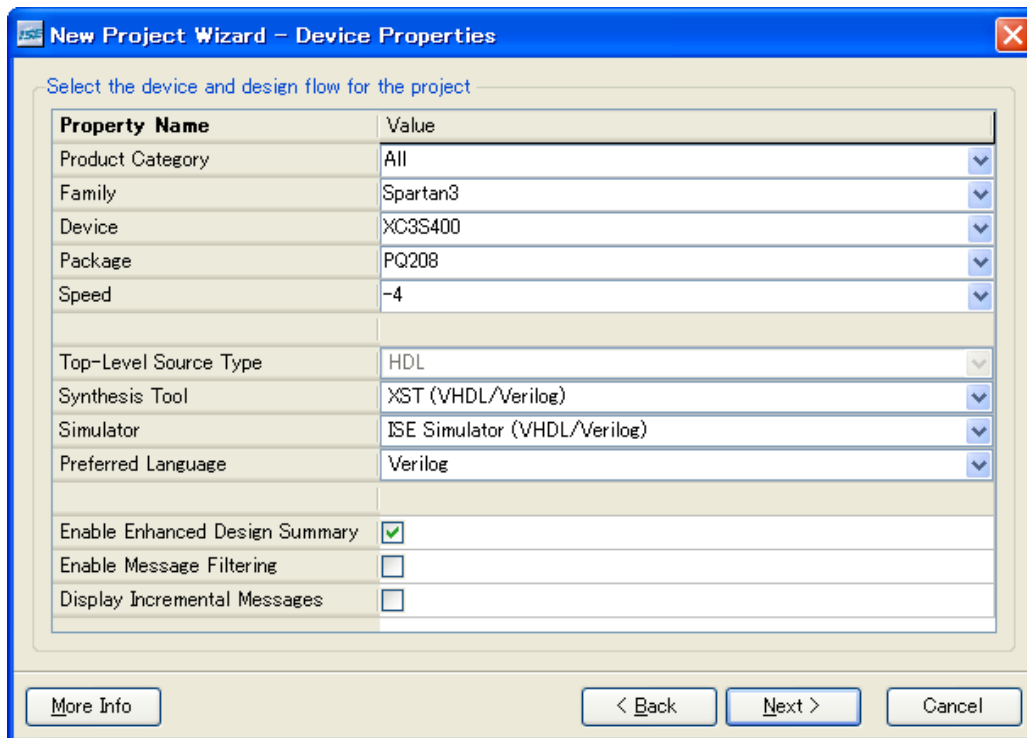
## 4.2 プロジェクトの新規作成

新規作成する場合の手順は下記通り：

- 1、File→New Project

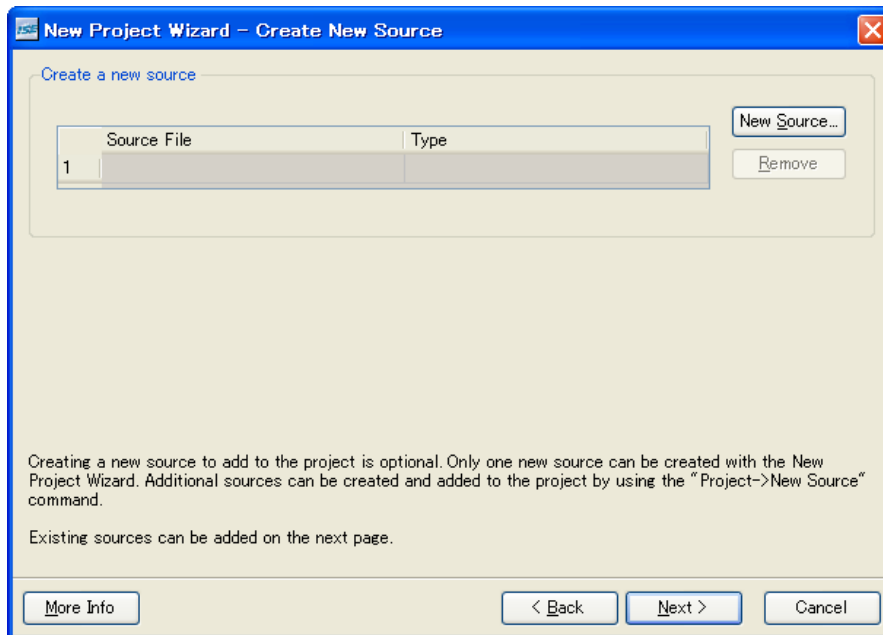


2、Project Name 欄に名称を入力する。Project Location 欄に保存場所を選択する。  
Top-Level Module Type は主に二種類を使います。  
HDL (Hardware Design Language) ハードウェア設計言語モードと Schematic 原理図モード。  
この試験では HDL を選択する。Next を押す。



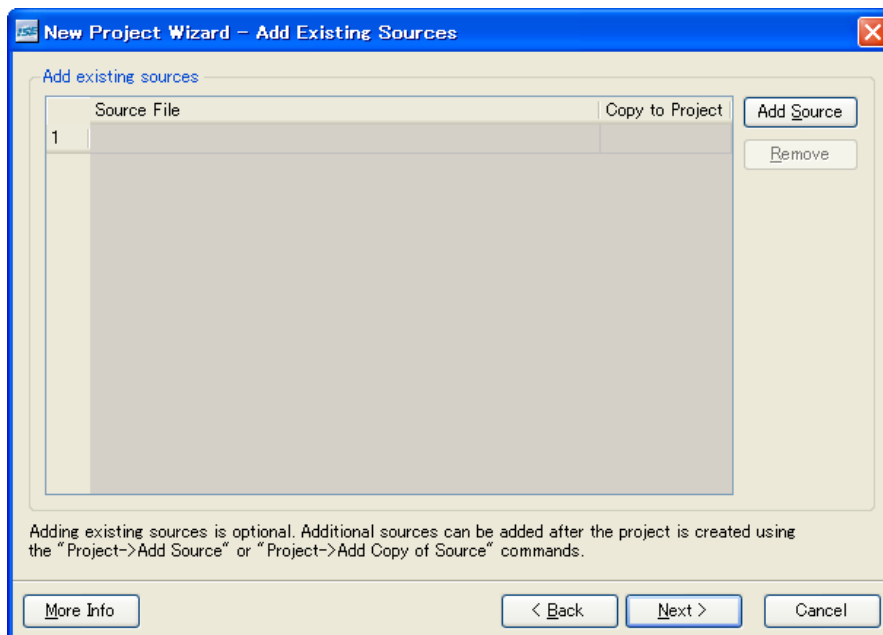
本試験は合成ツールとシミュレーションツールは全部 ISE を利用します。デフォルトのまま Next を押す。

3、ソースを作成する。



後で作成してもいいので、ここはこのまま Next を押す。

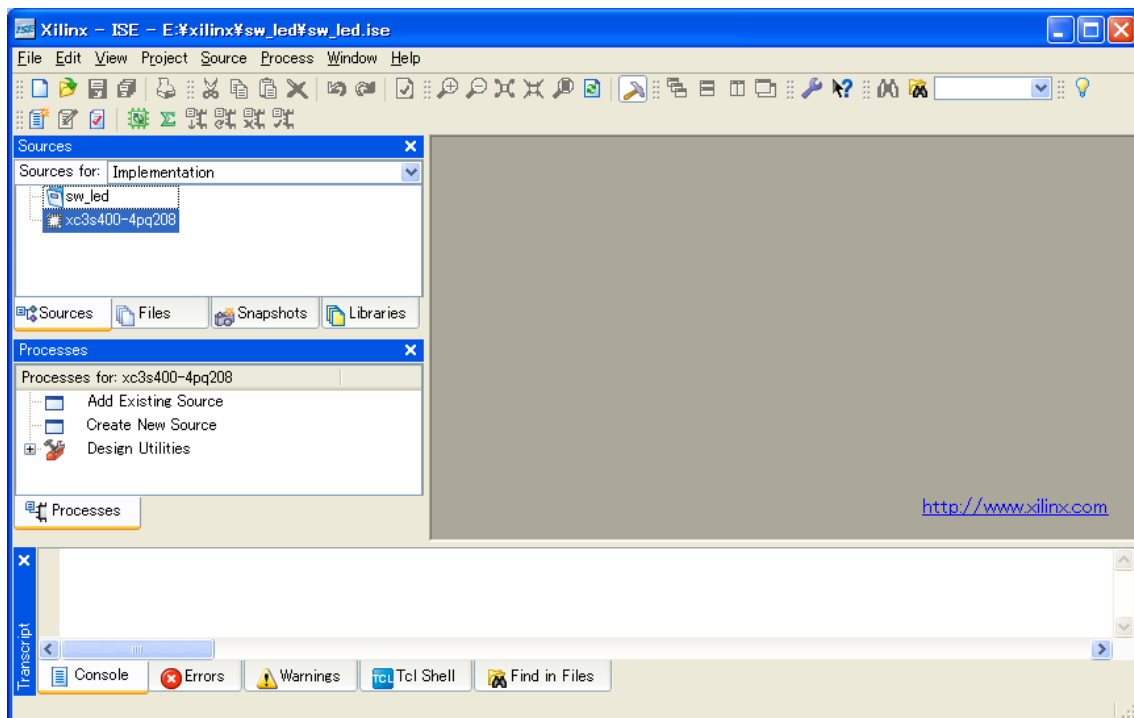
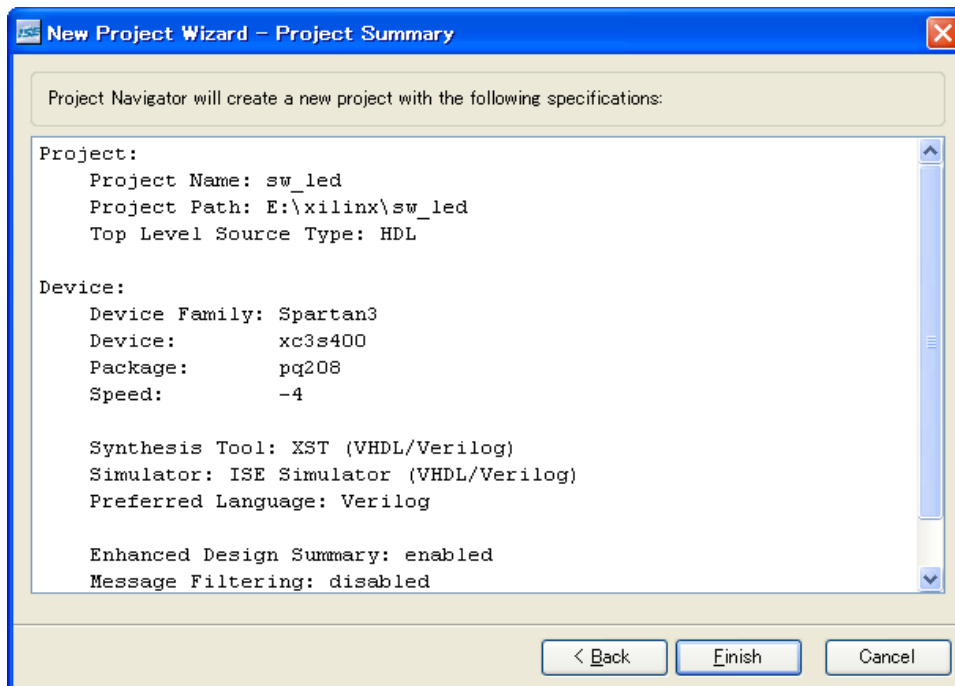
4、既存ソースの追加。



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 という名称のプロジェクトが作成される。

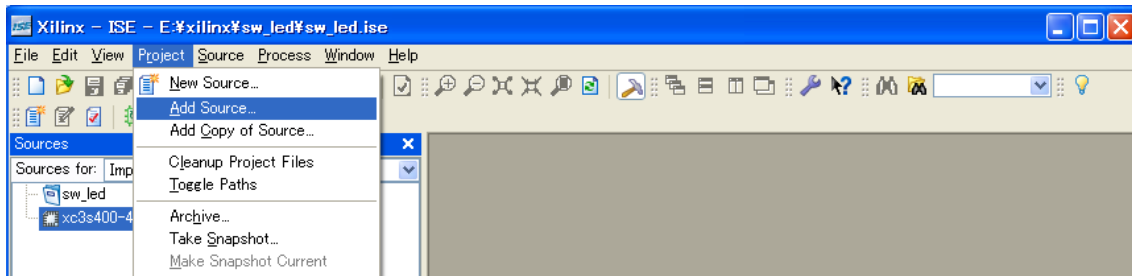




### 4.3 新規作成プロジェクトにソースファイルの追加

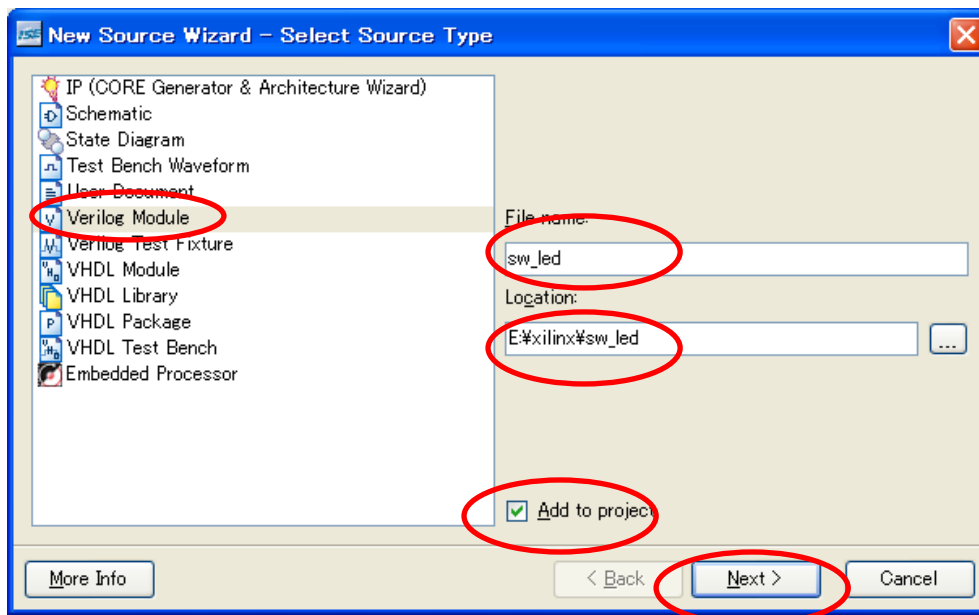
#### 1、既存ソースファイルの追加

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。

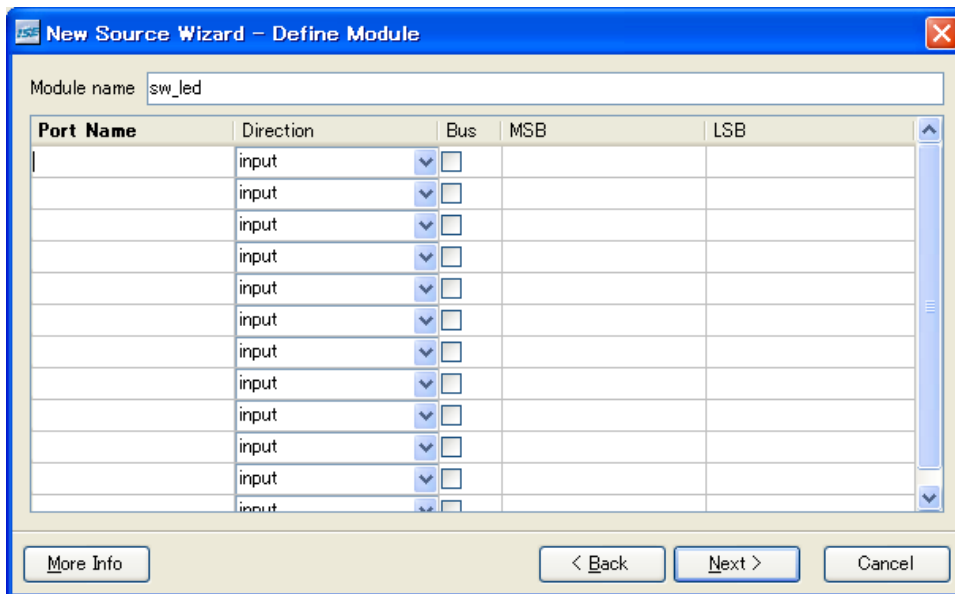


## 2、ソースファイルの新規作成

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



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。

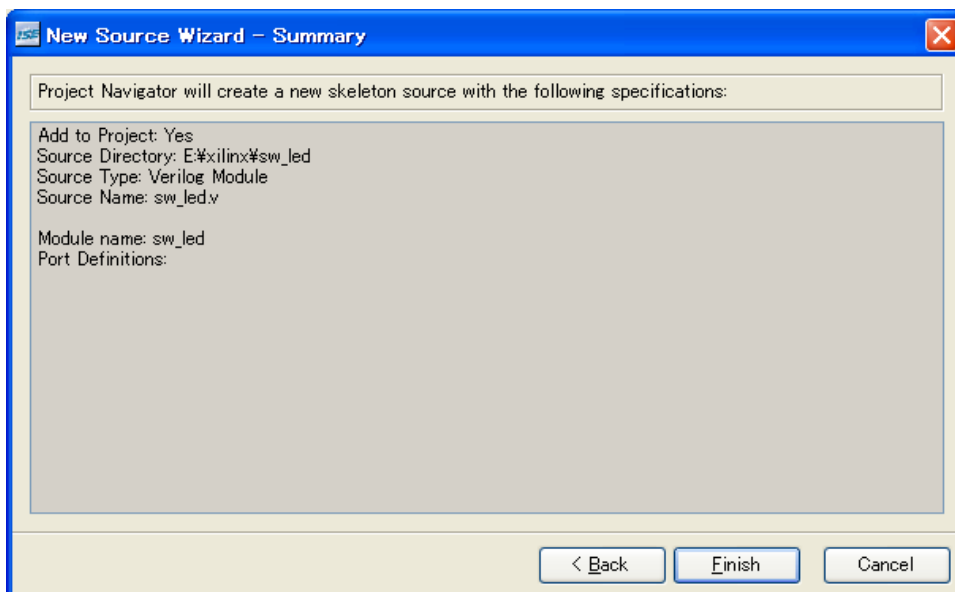


The dialog box 'New Source Wizard - Define Module' shows the 'Module name' as 'sw\_led'. Below is a table for defining ports:

Port Name	Direction	Bus	MSB	LSB
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		

Buttons at the bottom: More Info, < Back, Next >, 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。

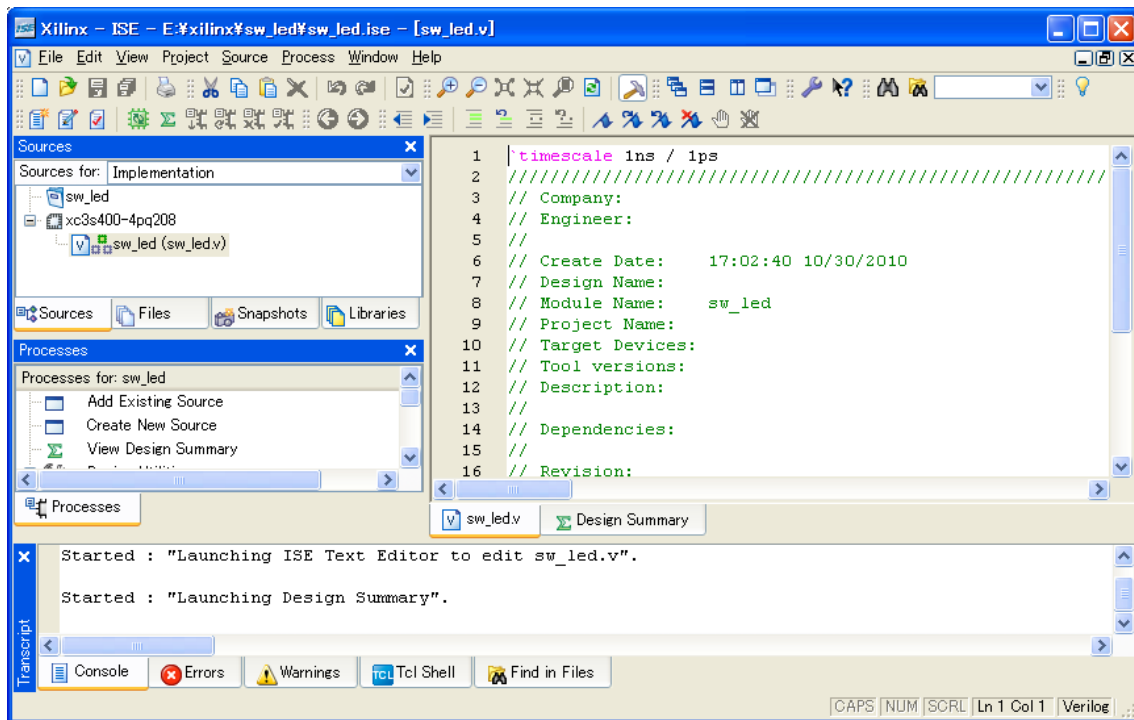


The dialog box 'New Source Wizard - Summary' displays the following specifications:

```
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:
```

Buttons at the bottom: < Back, Finish, 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 とい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;
```



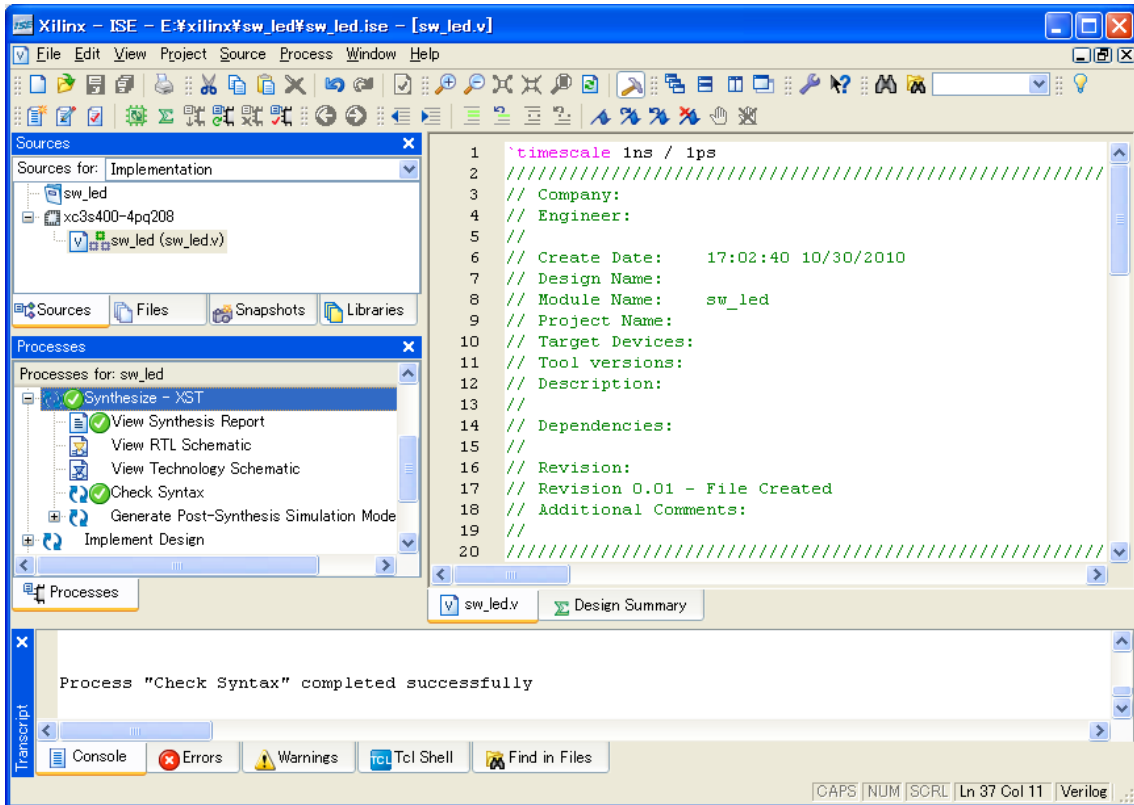
```
wire led4;
wire led5;
wire led6;
wire led7;
wire [3 :0] sw;
reg [7 :0] led;
assign sw = {sw0, sw1, sw2, sw3};
assign {led0, led1, led2, led3, led4, led5, led6, led7} = led;
always @ ( posedge clk or negedge rst)
begin
if(!rst)
led<='b0;
else
case ( sw )
4'b0001: led <= 8'b0000_0001;
4'b0010: led <= 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

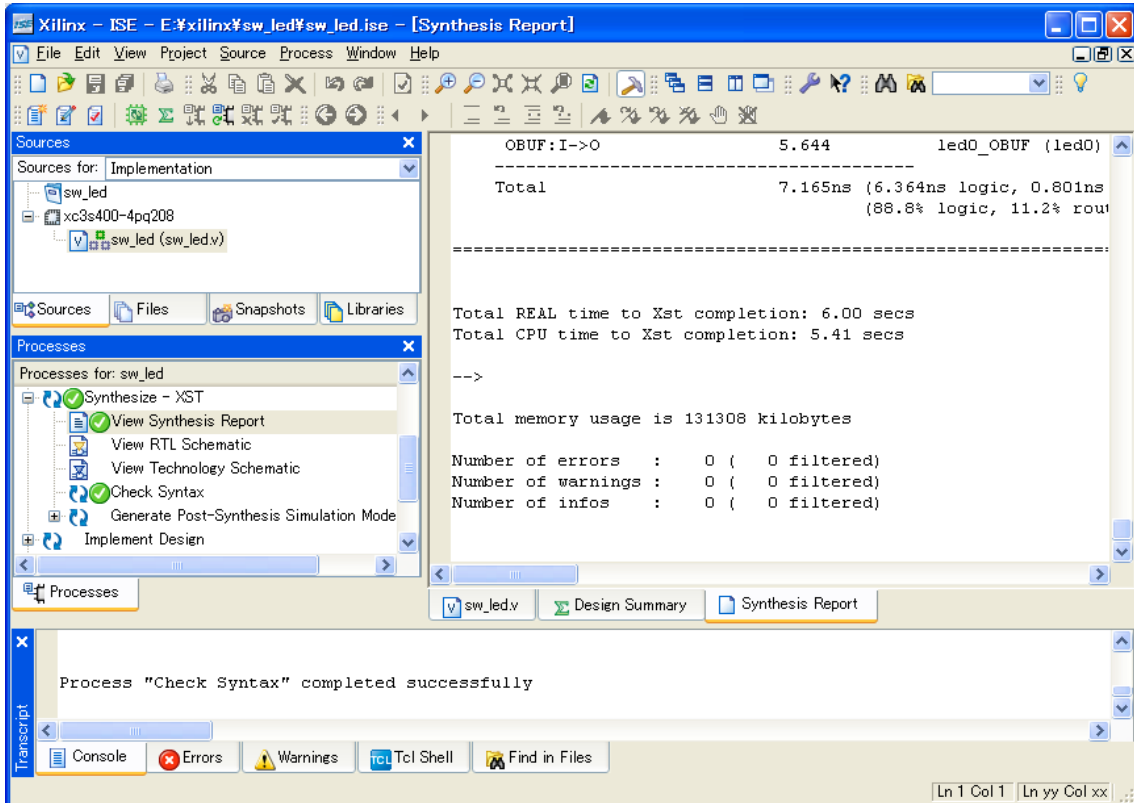
### 4.4.1 合成

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。



View Synthesis Reportをダブルクリックして結果を見る。



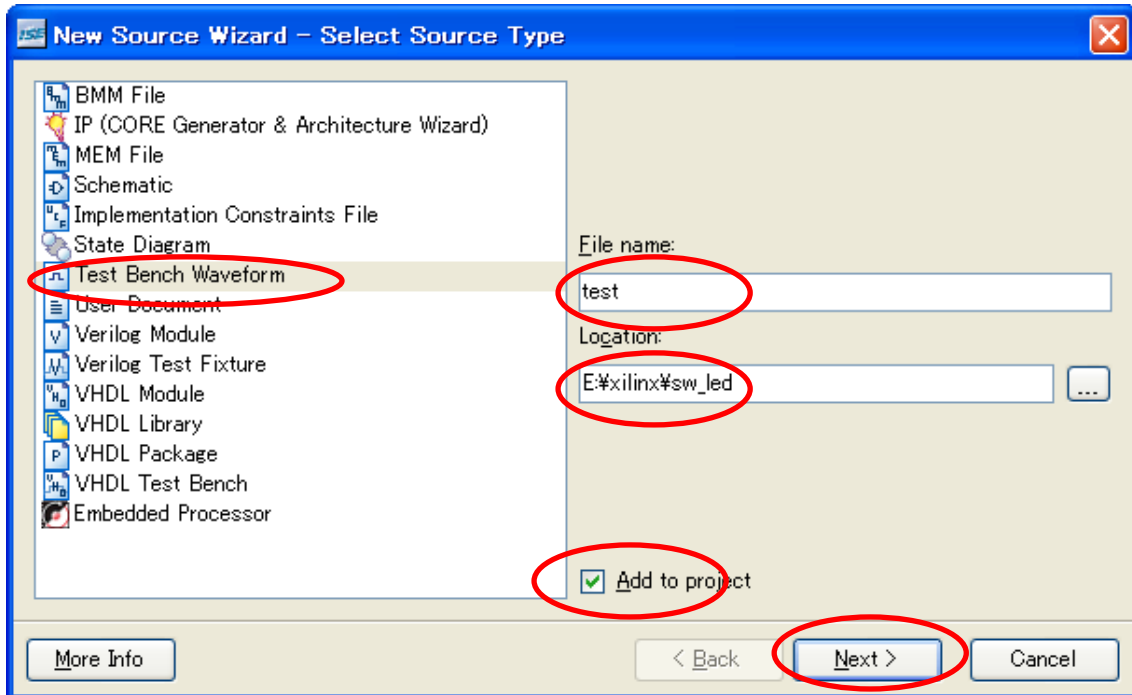
ここでエラー発生した場合は Error ポップが表示される。提示内容によって修正して再度

合成すればいいです。

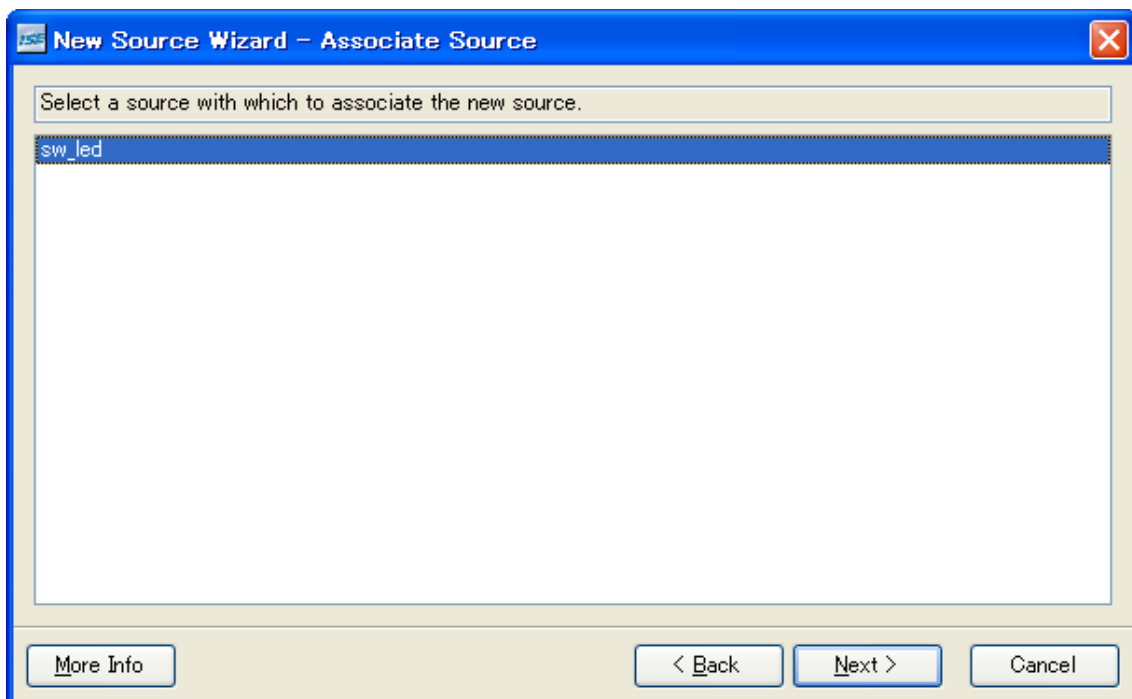
#### 4.4.2 シミュレーション

##### 1、シミュレーションファイルの作成

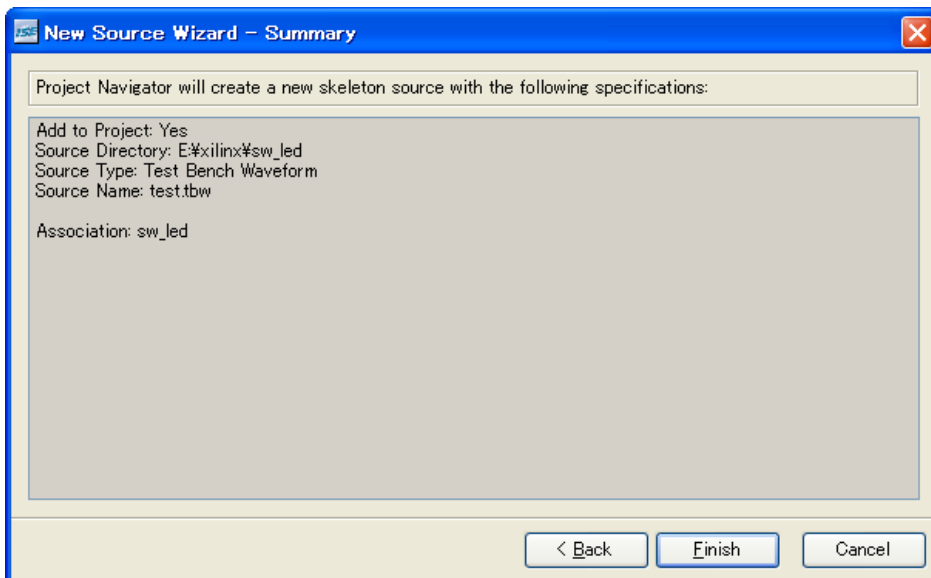
Project→New Source をクリックして、Test Bench Waveform ファイルを生成する。



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。

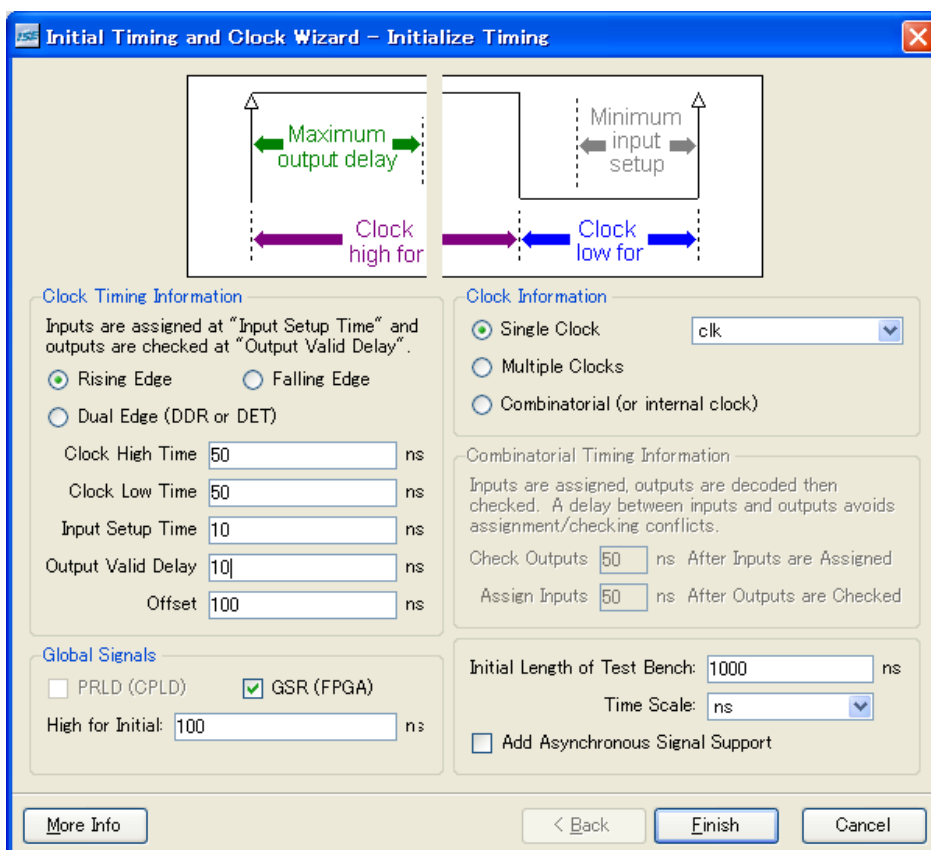


対象ファイル sw\_led を選択して、Next を押す。



情報を確認して Finish を押す。

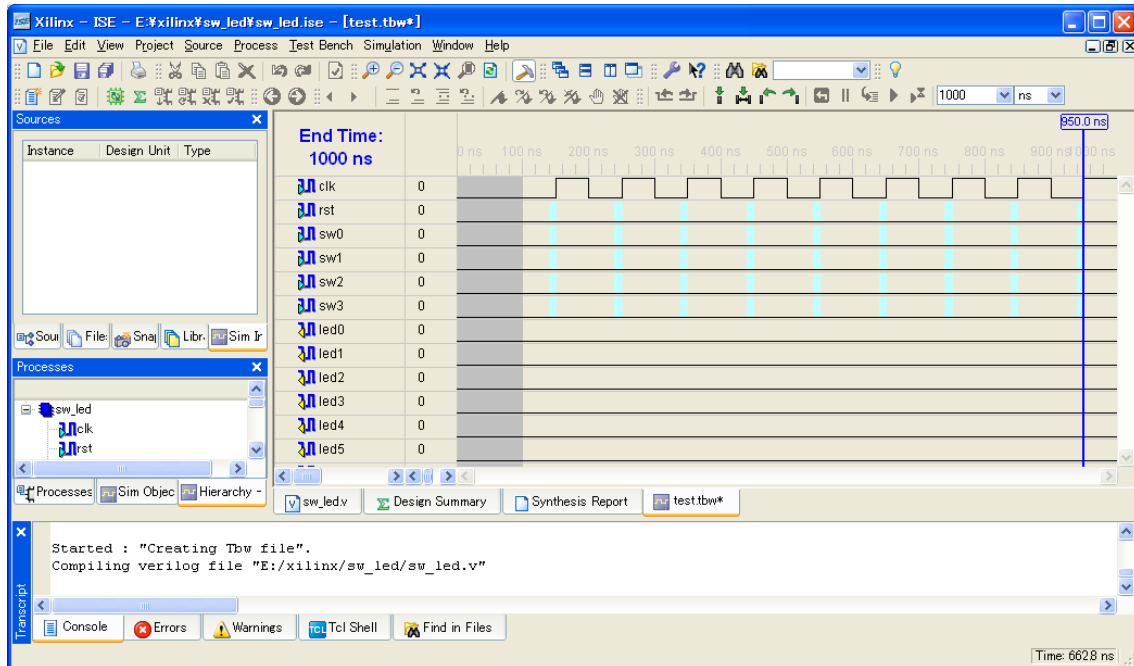
test.tbw ファイルが作成される同時に下記画面が表示され、タイミングについてのパラメータを設定する。



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



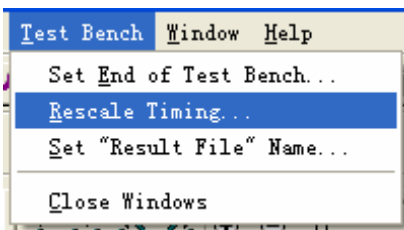
示される。

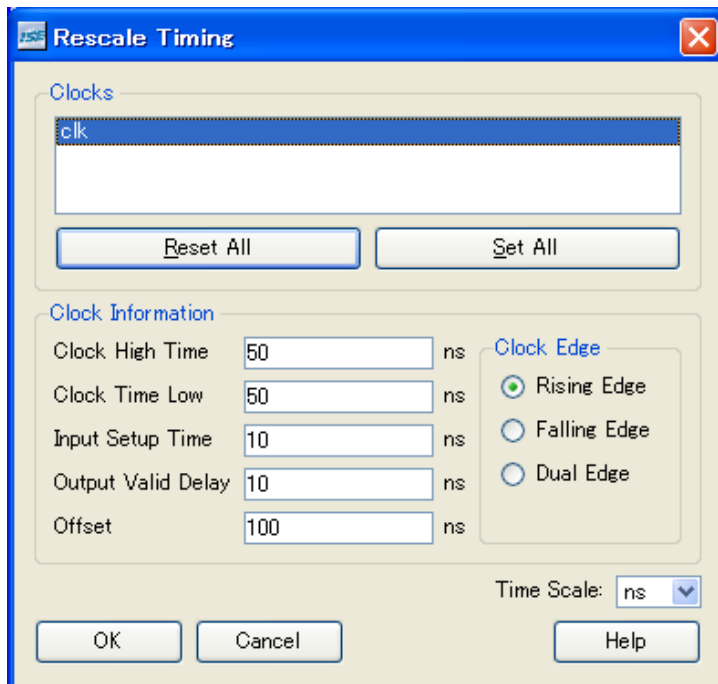


## 2、パラメータの修正

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。

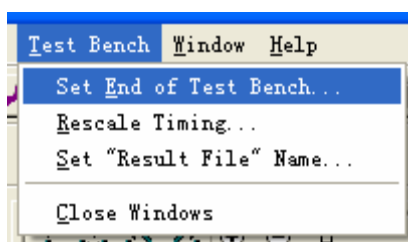
メニューの Test Bench → Rescale Timing…





修正して OK を押す。

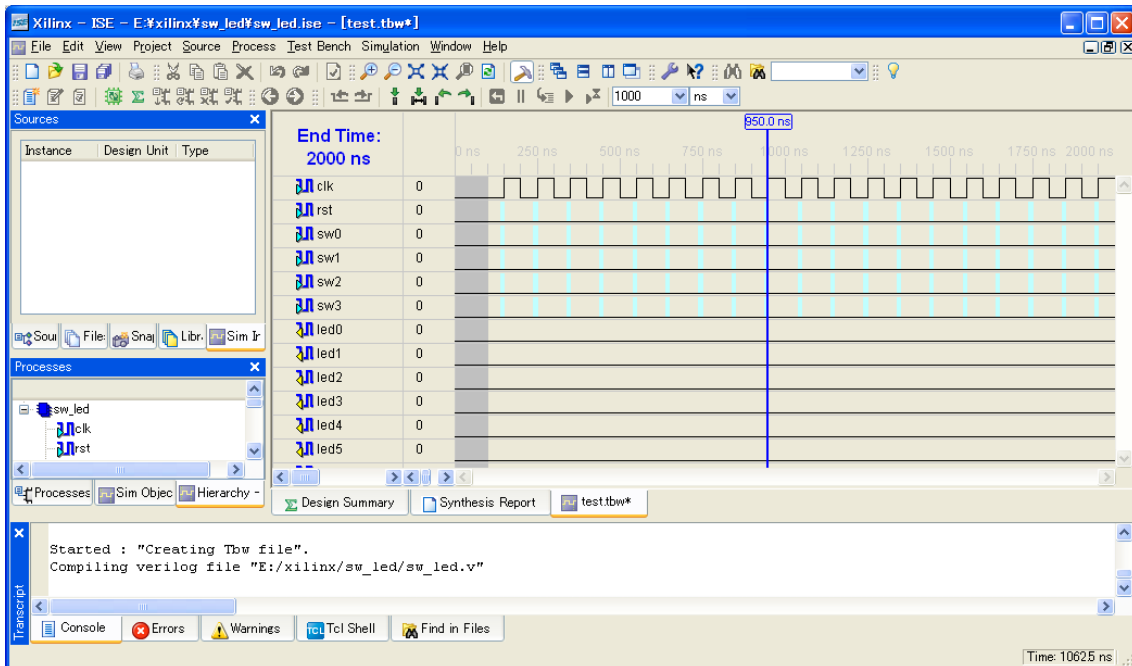
或いは Test Bench → Set End of Test Bench...



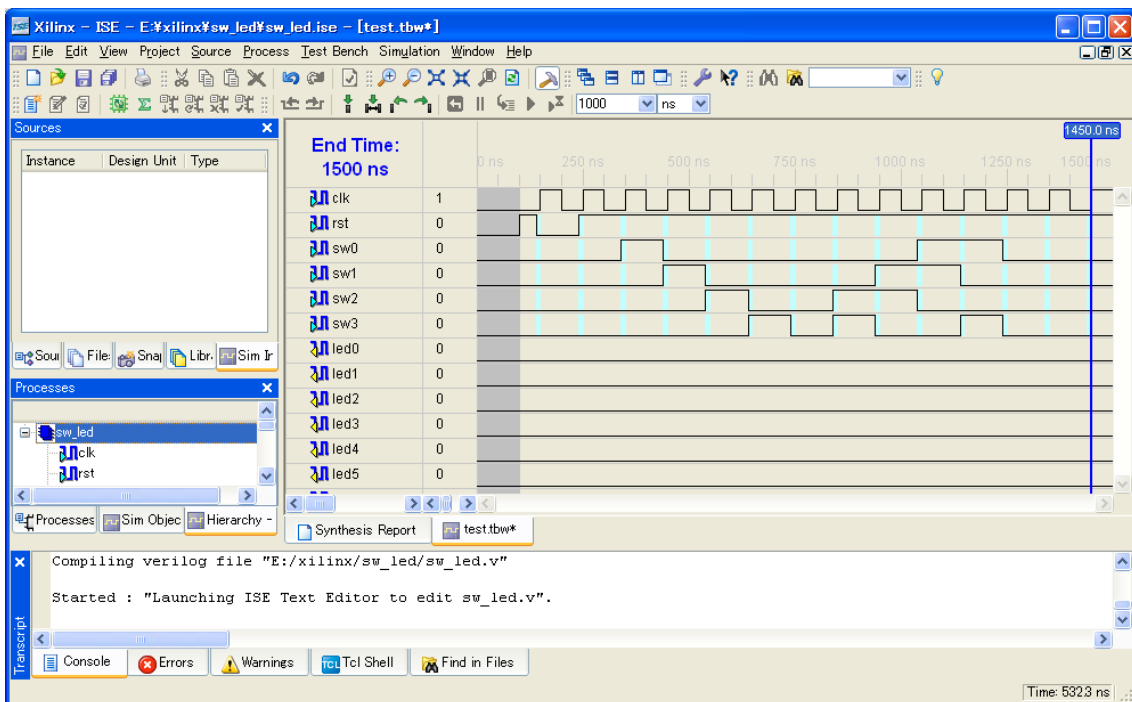
終了タイミングを設定できます。



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

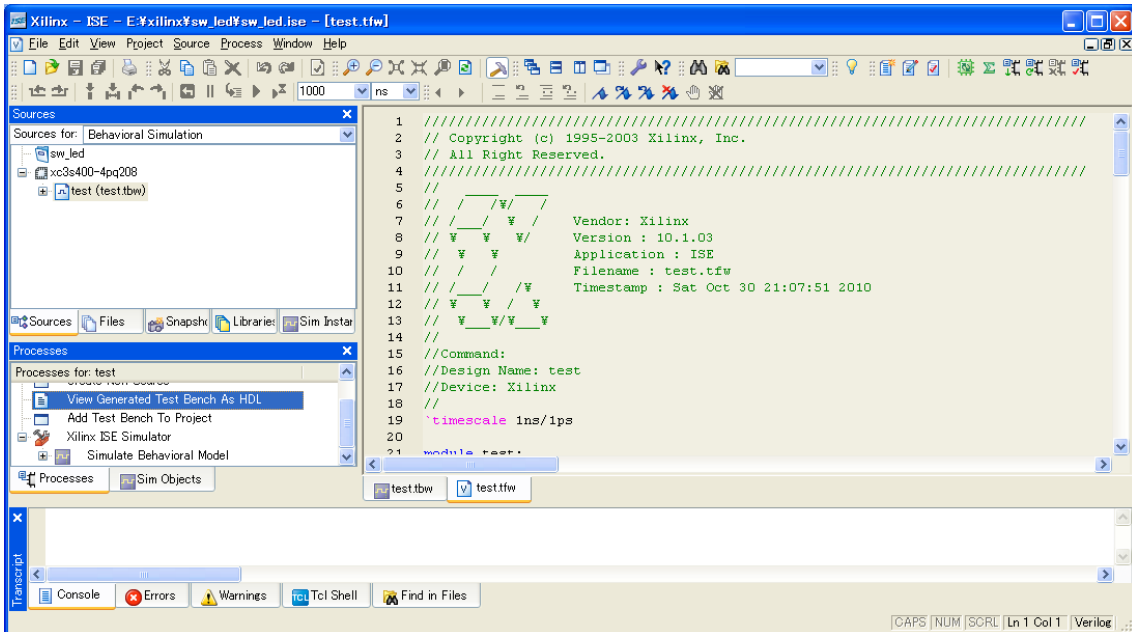


波形図の青色のバーをクリックすると High、Low のレベルを設定できる。

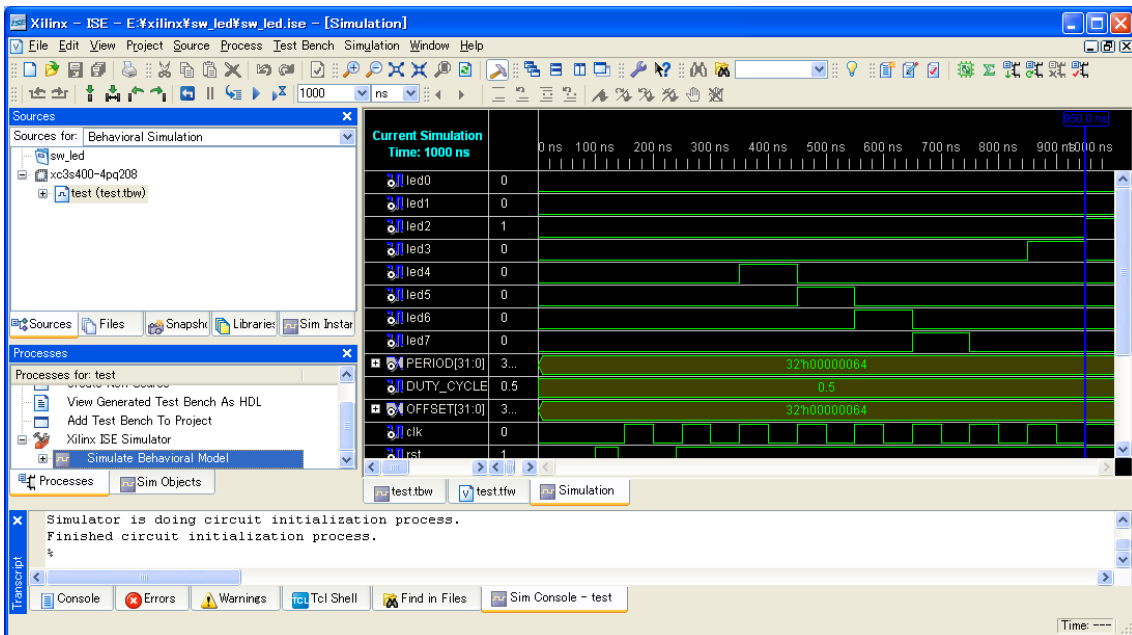


### 3、シミュレーション

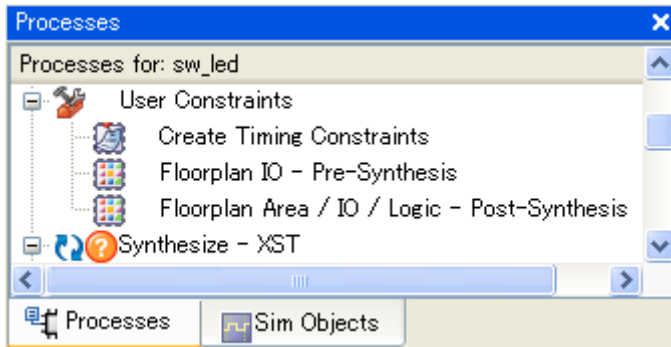
test.tbw を選択して、Process View から View Generated Test Bench As HDL をダブルクリックする。



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。



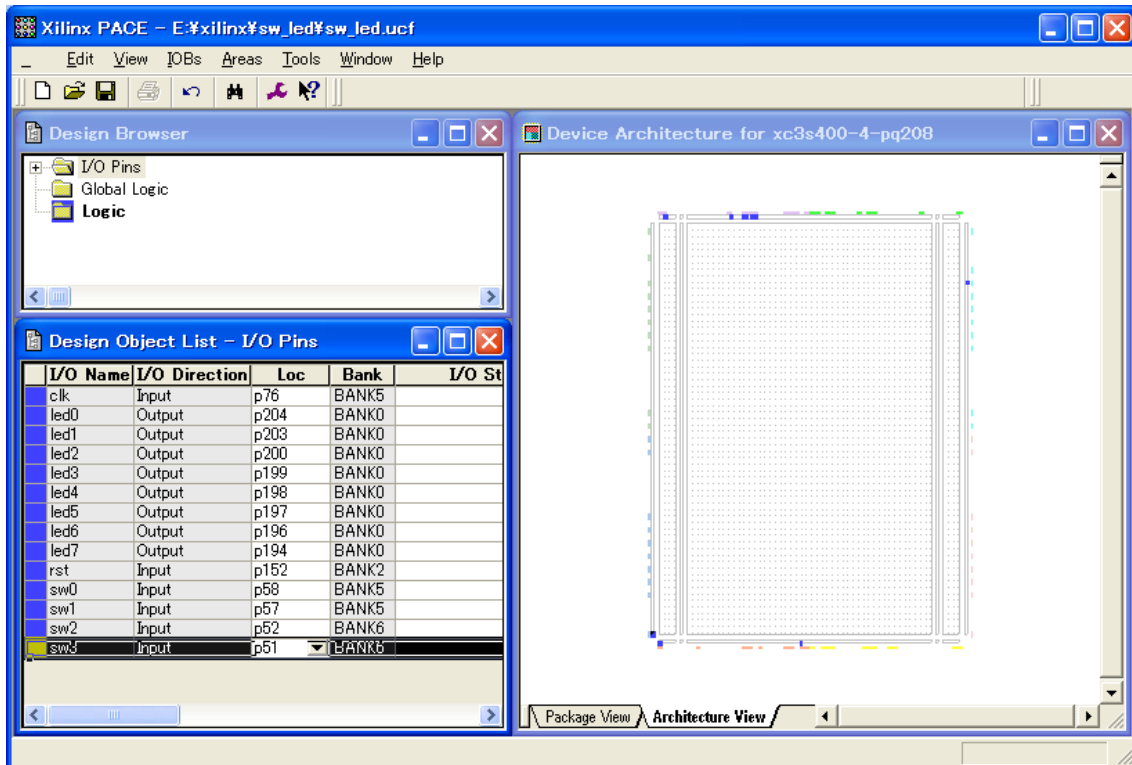
#### 4.5 Constraintsの編集



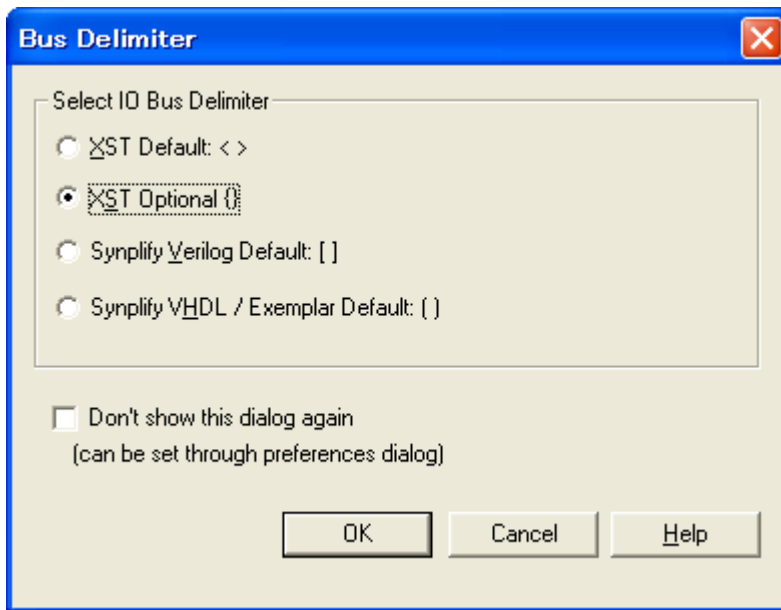
Timing Constrains、IO-Pre-Synthesis、Area/IO/Logic-Post-Synthesisがある。

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。



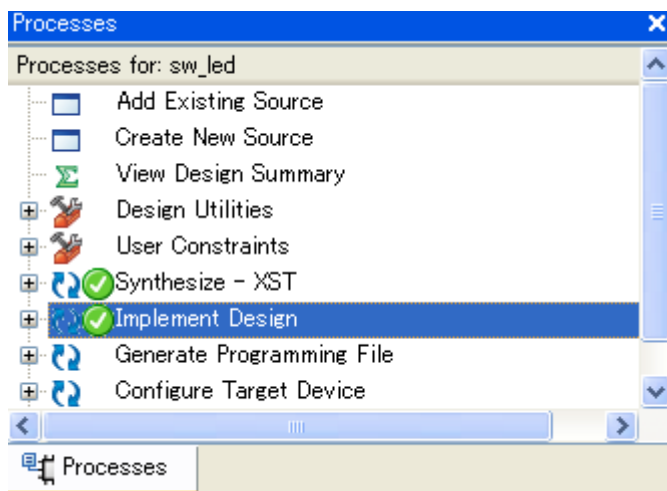
Loc 欄を pxxx のフォーマットで入力して保存する。



XST Optional を選択して OK。Xilinx 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。



完了後緑色のチェ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。