

SUZAKU

Hardware Manual

Version 1.0.3

2004 年 6 月 16 日

株式会社アットマークテクノ
<http://www.atmark-techno.com/>

目次

1. はじめに	1
2. 注意事項	2
2.1. 安全に関する注意事項	2
2.2. 取り扱い上の注意事項	2
2.3. FPGA 使用に関する注意事項	3
2.4. ソフトウェア使用に関する注意事項	3
3. 作業の前に	4
3.1. 準備するもの	4
4. 概要	5
4.1. SUZAKU の特徴	5
4.2. 仕様	6
4.3. 全体ブロック図	7
4.4. 機能	8
4.4.1. プロセッサ	8
4.4.2. バス	8
4.4.3. メモリ	9
4.4.4. 割り込み	9
4.4.5. タイマ	9
4.4.6. シリアルコンソール	9
4.4.7. LAN	9
4.4.8. 外部 I/O	10
4.4.9. FPGA コンフィグレーション	10
4.4.10. ソフトウェアリセット機能	12
4.4.11. JTAG	12
4.4.12. 設定用ジャンパ	13
4.4.13. LED	13
4.4.14. 電源入力+3.3V	13
4.4.15. 内部ロジック用電源出力+3.3V	14
4.4.16. 内部電源シーケンス	14
5. メモリマップ	15
5.1. SUZAKU メモリマップ	15
6. FPGA ピンアサイン	16
7. 各種インターフェース仕様	22
7.1. 各種インターフェースの配置	22
7.2. CON2 外部 I/O、FPGA プログラム用 JTAG コネクタ	23
7.3. CON3 外部 I/O コネクタ	24
7.4. CON4 外部 I/O コネクタ	25
7.5. CON5 外部 I/O コネクタ	25
7.6. CON7 FPGA JTAG コネクタ	26
7.7. CON1 RS232C コネクタ	26
7.8. JP1 起動モード ジャンパ	27
7.9. JP2 FPGA プログラム用ジャンパ	27
7.10. D3 パワーオン LED	28
7.11. D1 ユーザコントロール LED	28
7.12. CON6 電源入力+3.3V コネクタ	28
7.13. Ethernet 10/100 Base-T	29
8. 基板形状図	30
9. FPGA プロジェクト構築方法(例 UART の追加方法)	31

9.1.	添付 CD-ROM の fpga_proj のプロジェクトのコピー	31
9.2.	新規プロジェクトの作成	31
9.3.	Platform Studio の設定	35
9.4.	Platform Studio コンパイルの実行	41
9.5.	プロジェクトのトップファイル編集	42
9.6.	Platform Studio コンパイルの実行	47
10.	FPGA コンフィグレーション	48
10.1.	BIT ファイルから MCS ファイルへの変換	48
10.2.	SUZAKU へコンフィグレーションデータのプログラム	53

表目次

表 4-1 SUZAKU 仕様	6
表 5-1 SUZAKU メモリマップ	15
表 6-1 FPGA ピンアサイン 外部 I/O 系 (1/3)	16
表 6-2 FPGA ピンアサイン 外部 I/O 系 (2/3)	17
表 6-3 FPGA ピンアサイン 外部 I/O 系 (3/3)	18
表 6-4 FPGA ピンアサイン 内部デバイス系(1/3)	19
表 6-5 FPGA ピンアサイン 内部デバイス系(2/3)	20
表 6-6 FPGA ピンアサイン 内部デバイス系(3/3)	21
表 6-7 FPGA ピンアサイン JTAG、コンフィグレーション系	21
表 7-1 各種インターフェースの内容	22
表 7-2 CON2 外部 I/O、FPGA プログラム用コネクタ	23
表 7-3 外部 I/O コネクタ	24
表 7-4 CON4 外部 I/O コネクタ	25
表 7-5 CON5 外部 I/O コネクタ	25
表 7-6 CON7 Spartan-3 用 JTAG コネクタ	26
表 7-7 CON1 RS232C コネクタ	26
表 7-8 JP1 起動モード ジャンパ	27
表 7-9 JP2 FPGA プログラム用ジャンパ	27
表 7-10 D1 ユーザコントロール LED	28
表 7-11 CON6 電源入力+3.3V コネクタ	28
表 7-12 Ethernet 10/100 Base-T	29

図目次

図 4-1 SUZAKU ブロック図	7
図 4-2 SUZAKU バス構成	8
図 4-3 FPGA コンフィグレーション	11
図 7-1 各種インターフェースの配置	22
図 8-1 SUZAKU の基板形状	30

1.はじめに

このたびは SUZAKU をお求めいただき、ありがとうございます。

本マニュアルは、SUZAKU のハードウェアの仕様や使用方法について書かかれています。

SUZAKU の機能を最大限引き出すために、ご活用いただければ幸いです。

2. 注意事項

2.1. 安全に関する注意事項

SUZAKU を安全にご使用いただくために、特に以下の点にご注意くださいますようお願いいたします。



本製品には一般電子機器用(OA機器・通信機器・計測機器・工作機械等)に製造された半導体部品を使用していますので、その誤作動や故障が直接生命を脅かしたり、身体・財産等に危害を及ぼす恐れのある装置(医療機器・交通機器・燃焼制御・安全装置等)に組み込んで使用したりしないでください。また、半導体部品を使用した製品は、外来ノイズやサージにより誤作動したり故障したりする可能性があります。ご使用になる場合は万一誤作動、故障した場合においても生命・身体・財産等が侵害されることのないよう、装置としての安全設計(リミットスイッチやヒューズ・ブレーカ等の保護回路の設置、装置の多重化等)に万全を期されますようお願い申し上げます。

2.2. 取り扱い上の注意事項

劣化、破損、誤動作、発煙、発火の原因となることがあります。取り扱い時には以下のような点にご注意ください。

- **入力電源**
3.3V+5%以上の電圧を入力しないでください。
極性を間違わないでください。
- **インターフェース**
各インターフェース(外部 I/O、RS232C、Ethernet、JTAG)には規定以外の信号を接続しないでください。
信号の極性を間違わないでください。
信号の入出力方向を間違わないでください。
- **改造**
外部 I/O コネクタ及び JTAG コネクタ(CON2、CON3、CON4、CON5、CON7)にコネクタ等を増設する以外の改造は行わないでください。
- **FPGA プログラム**
周辺回路(ボード上の部品も含む)と信号の衝突(同じ信号に2つのデバイスから出力する)を起こすようなFPGAプログラムを行わないでください。
FPGA のプログラムを間違わないでください。
- **電源の投入**
本ボードや周辺回路に電源が入っている状態では絶対に FPGA I/O、JTAG 用コネクタの着脱を行わないでください。
- **静電気**
本ボードには CMOS デバイスを使用していますので、ご使用になるまでは帯電防止対策のされている、出荷時のパッケージ等にて保管してください。

- **ラッチアップ**

電源および入出力からの過大なノイズやサージ、電源電圧の急激な変動等で使用している CMOS デバイスがラッチアップを起こす可能性があります。いったんラッチアップ状態となると、電源を切断しないかぎりこの状態が維持されるため、デバイスの破損につながる可能性があります。ノイズの影響を受けやすい入出力ラインには保護回路を入れることや、ノイズ源となる装置と共通の電源を使用しない等の対策をとることをお勧めします。

- **衝撃、振動**

落下や衝突などの強い衝撃を与えないでください。

振動部や回転部などへの搭載はしないでください。強い振動や遠心力を与えないでください。

- **高温低温、多湿**

極度に高温や低温になる環境や、湿度が高い環境では使用はしないでください。

- **塵埃**

塵埃の多い環境では使用はしないでください。

2.3. FPGA 使用に関する注意事項

- **本製品に含まれる FPGA プロジェクトについて**

本製品に含まれる FPGA プロジェクト(付属のドキュメント等も含みます)は、現状のまま(AS IS)提供されるものであり、特定の目的に適合することや、その信頼性、正確性を保証するものではありません。また、本製品の使用による結果についてもなんら保証するものではありません。

本製品は、ベンダのツール(Xilinx 製 EDK、ISE やその他ベンダツール)やベンダの IP コアを利用し、FPGA プロジェクトの構築、コンパイル、コンフィグレーションデータの生成を行っておりますが、これらツールに関する販売、サポート、保証等は行っておりません。

2.4. ソフトウェア使用に関する注意事項

- **本製品に含まれるソフトウェアについて**

本製品に含まれるソフトウェア(付属のドキュメント等も含みます)は、現状のまま(AS IS)提供されるものであり、特定の目的に適合することや、その信頼性、正確性を保証するものではありません。また、本製品の使用による結果についてもなんら保証するものではありません。

3. 作業の前に

3.1. 準備するもの

SUZAKU を使用する前に、次のものを準備して下さい。

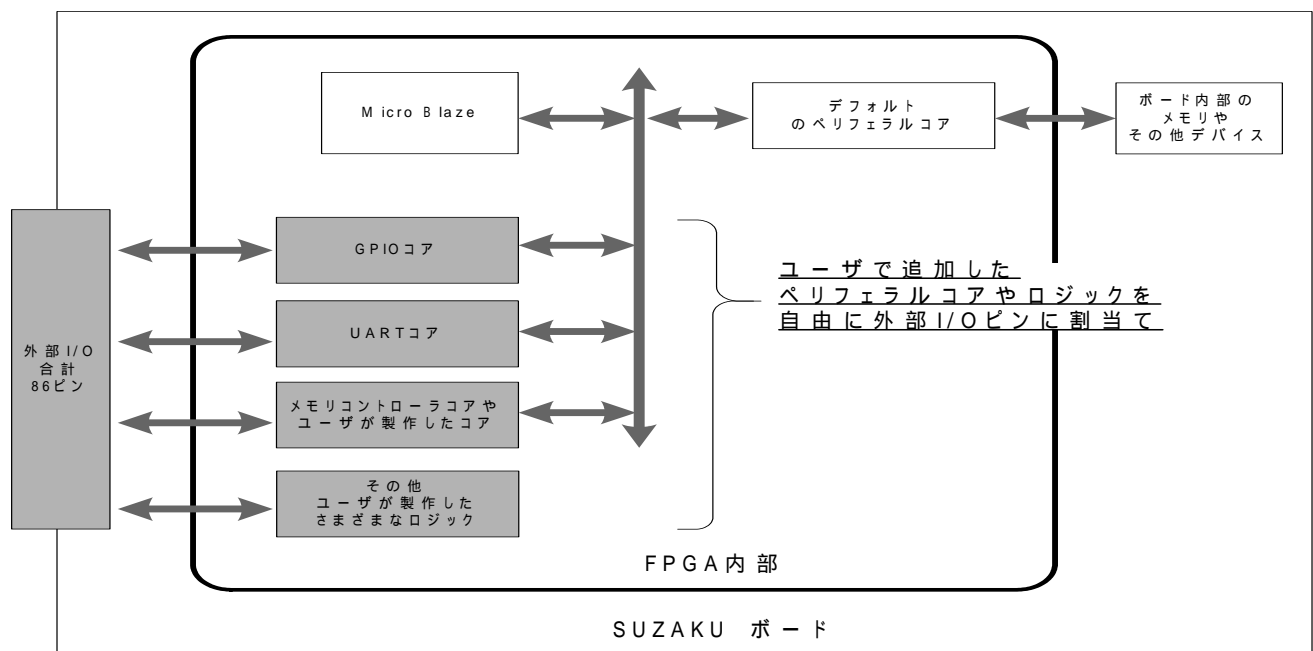
- **作業用 PC**
ハードウェア開発用として、Windows2000 または、WindowsXP が動作し、シリアルポート(1 ポート)、及びパラレルポート(1 ポート)を持つ PC を用意してください。
ソフトウェア開発用として、Linux が動作し、シリアルポート(1 ポート)を持つ PC を用意してください。
ソフトウェア開発の詳細については、Software Manual を参照ください。
- **D-Sub9 ピンクロスケーブル**
D-Sub9 ピン(メス - メス)の「クロス接続用」のケーブルを用意してください。
- **D-Sub9 ピン-10 ピン変換ケーブル**
D-Sub9 ピンと本ボードのピンヘッダ(10 ピン)を接続するための、D-Sub9 ピン-10 ピン変換ケーブルを用意してください。
- **開発キット付属 CD-ROM(以降、「付属 CD」)**
SUZAKU に関する各種マニュアルやソースコードが収納されています。
- **シリアル通信用ソフト**
minicom や Tera Term などのシリアル通信用ソフトが必要です。(Linux 用のソフトは付属 CD の「tools」ディレクトリにあります。)
- **DC3.3V 電源**
DC3.3V 出力の電源を用意してください。
- **Xilinx ISE**
Xilinx ISE を用意してください。
詳しくは Xilinx 代理店にお問い合わせください。
- **Xilinx EDK**
Xilinx EDK を用意してください。
詳しくは Xilinx 代理店にお問い合わせください。
- **Xilinx Parallel Cable または相当**
Parallel Cable を用意してください。
詳しくは Xilinx 代理店にお問い合わせください。

4. 概要

4.1. SUZAKU の特徴

SUZAKU(朱雀)は Xilinx の FPGA「Spartan-3」をベースとしたボードコンピュータです。
FPGA 上にソフトプロセッサ「MicroBlaze」と周辺ペリフェラルコアを構成し、オペレーティングシステムとして Linux(uClinux)を採用しています。

- **ソフトプロセッサと周辺ペリフェラルコアの構築**
MicroBlaze や周辺ペリフェラルコアの構築は、Xilinx 社 EDK(Embedded Development Kit)を使用します。EDK は、GUI 環境下で MicroBlaze や周辺ペリフェラルコアの各種設定が行え、その設定情報から自動的にネットリストを生成するツールです。
- **カスタマイズ**
FPGA の中は、ユーザによってカスタマイズが可能です。
また、基板外周にユーザが自由に使える外部 I/O を 86 ピン実装しています。
例えば、PIO や UART の数を増やし、外部 I/O ピンに割り当てるなどのカスタマイズが簡単に行えます。



* FPGA のカスタマイズには Xilinx 社の EDK、ISE が必要です。Xilinx 社または、Xilinx 代理店より入手してください。

- **LAN**
LAN(10Base-T/100Base-Tx)を実装しています。市販の LAN ケーブル(UTP)が接続できます。
- **オペレーティングシステム**
μClinux を標準のオペレーティングシステムとして採用しておりますので、アプリケーションソフトウェアの開発には GNU のアセンブラや C コンパイラ等を使用することができます。
また、LAN コントローラデバイスドライバ、各種プロトコルが最初から用意されていますので、簡単にネットワークに接続できます。
オペレーティングシステムの詳細については、Software Manual を参照ください。

4.2. 仕様

本ボードの主な仕様を表 4 - 1 に示します。

表 4-1 SUZAKU 仕様

FPGA	Xilinx Spartan-3 (XC3S400 FT256)
ソフトプロセッサ	MicroBlaze
水晶発振器周波数	3.6864MHz(FPGA の内部 DCM により逡倍して使用)
メモリ	BRAM 8Kbyte FLASH メモリ 4Mbyte SDRAM 16Mbyte
コンフィグレーション	FLASH メモリ上に記憶、コントローラ TE7720
JTAG	2 ポート(FPGA 用、TE7720 用)
Ethernet	10Base-T / 100Base-Tx
シリアル	UART 115.2kbps
タイマ	2ch(1ch は OS で使用)
フリーI/O ピン	86 ピン
リセット機能	ソフトウェアリセット
電源	電圧: 3.3V ± 3% 消費電流: 350mA typ(プロセッサ動作時)
基板サイズ	72 × 47mm

4.3. 全体ブロック図

SUZAKU の全体ブロック図を図 4-1 に示します。
本構成は、uCLinux を動作させる最小構成です。

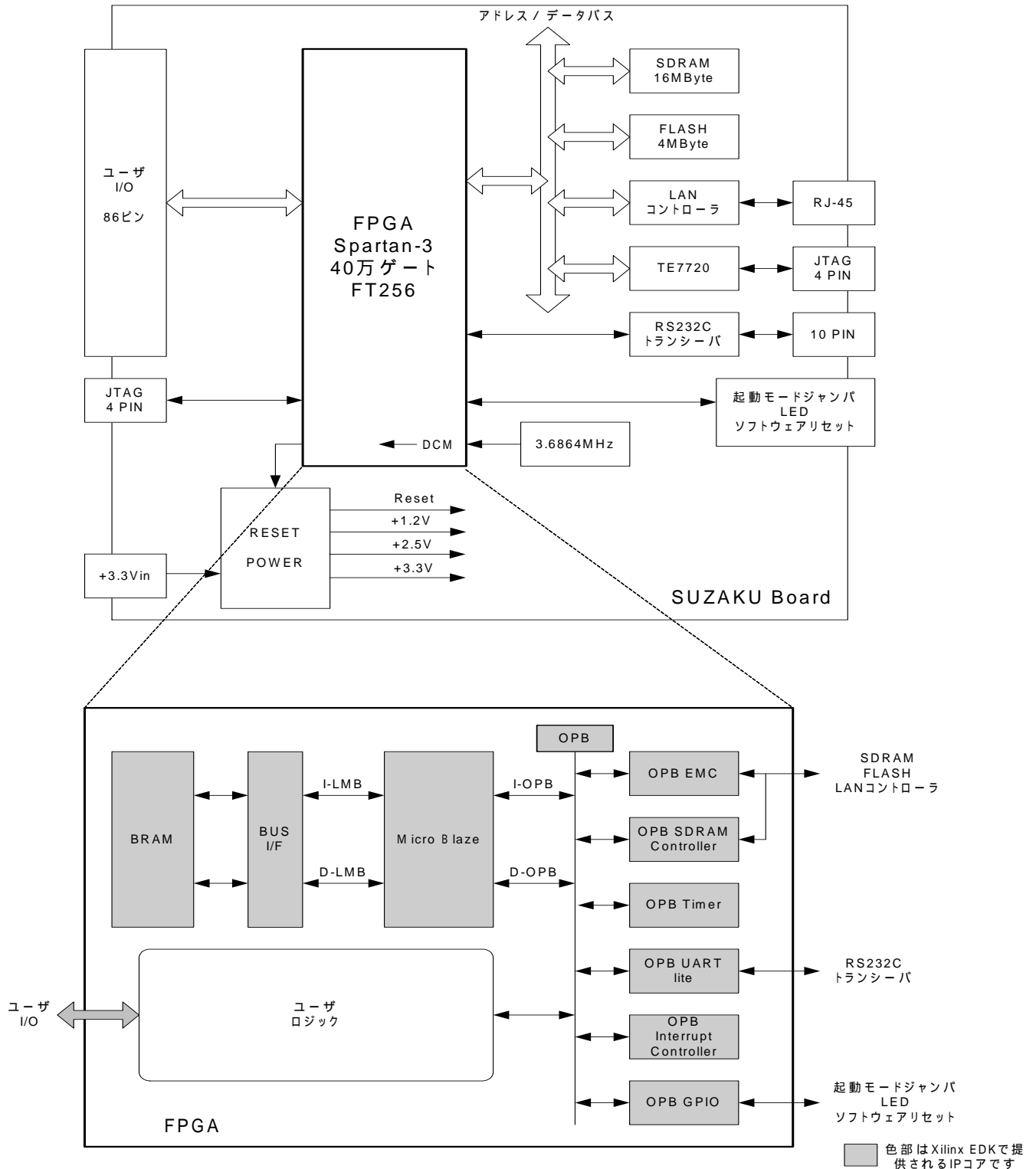


図 4-1 SUZAKU ブロック図

4.4. 機能

4.4.1. プロセッサ

FPGA 内部で MicroBlaze を使用しています。MicroBlaze の概要を以下に示します。

- ・32 ビット RISC プロセッサ
- ・32 ビット固定長命令
- ・32 個の汎用 32bit レジスタ
- ・3 ステージパイプライン
- ・命令キャッシュとデータキャッシュ
- ・ハードウェア乗算器
- ・ハードウェアデバッグロジック対応

4.4.2. バス

3 種類のバスで構成しています。

- ・FPGA 内部 LMB
MicroBlaze と BRAM(FPGA 内部メモリ)を接続する専用バス
- ・FPGA 内部 OPB
複数のペリフェラル IP コアを接続するバス
カスタマイズを行う時は、本バスにペリフェラルコアを追加していきます。
- ・FPGA 外部バス
OPB EMC 及び、OPB SDRAM を介し、外部メモリデバイスなどを接続するバス。

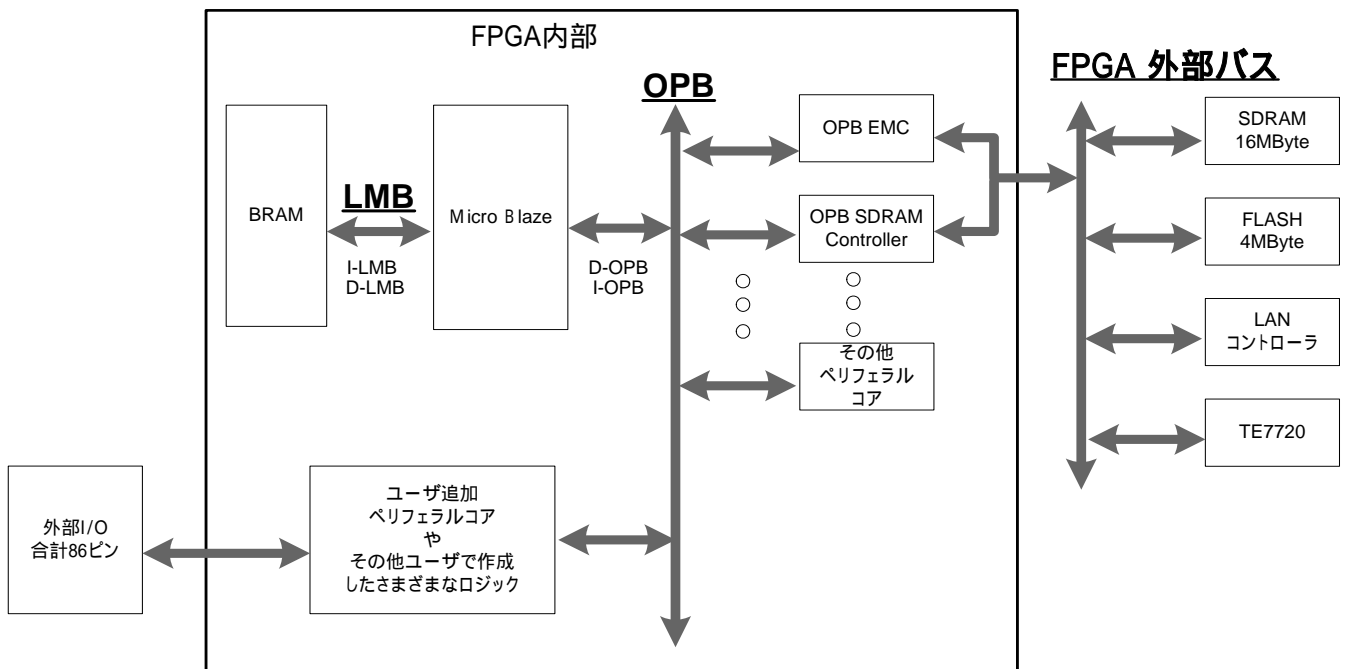


図 4-2 SUZAKU バス構成

4.4.3. メモリ

3種類のメモリで構成しています。

- ・FPGA 内部 BRAM (デフォルト 8KByte)
ブートプログラム用として使用しています。
起動完了後は、先頭の 32Byte(割り込みベクタ領域)以外であれば、ユーザプログラムで使用することもできます。
- ・FPGA 外部 FLASH メモリ 4MByte
高機能ブートローダや Linux システム、FPGA コンフィグデータなどのデータ保存に使用しています。
OPB EMC を使用し、OPB と接続しています。
- ・FPGA 外部 SDRAM 16MByte
Linux のメインメモリとして使用しています。
OPB SDRAM を使用し、OPB と接続しています。

4.4.4. 割り込み

OS 用割り込みコントローラに、FPGA 内部で OPB INTC を使用しています。

4.4.5. タイマ

OS 用タイマに、FPGA 内部で OPB Timer を使用しています。

4.4.6. シリアルコンソール

OS 用シリアルコンソールに、FPGA 内部で OPB UART Lite を使用しています。

OPB UART Lite は RS232C トランシーバを介し、コネクタ(CON1)に接続しています。

また、RS232C トランシーバは、4 チャンネルタイプのものであり、このうち 2 チャンネルを OS 用シリアルコンソールで使用し、残り 2 チャンネルは未使用となっています。これらの未使用の信号に GPIO やユーザロジックを接続してフロー制御をしたり、別の OPB UART Lite を接続して 2 ポート目の UART とすることも可能です。

・シリアルコンソールの設定

転送レート	115.2kbps
データ	8bit
ストップ bit	1bit
フロー制御	なし

4.4.7. LAN

LAN コントローラに、FPGA 外部に SMSC 社の LAN91C113 を実装しています。

LAN91C113 は、OPB EMC を使用し、OPB と接続しています。

また、RJ-45 コネクタを実装しており、市販の LAN ケーブル(UTP)が接続できます。

4.4.8. 外部 I/O

ユーザが自由に使用できる外部 I/O を 86 ピン実装しています (CON2、CON3、CON4、CON5)。

(コネクタは実装されていません)

外部 I/O は、全て FPGA のフリー I/O ピンと直接接続しています。

FPGA の I/O 用電源 (VCCO) は、全て内部ロジック用電源 +3.3V から供給しています。

I/O 電圧や駆動電流などの規定値については、Spartan-3 のデータシートをご参照ください。

内部ロジック用電源 +3.3V は、シーケンス回路及びディレー回路により立ち上がりに最大 20msec の時間がかかります。よって外部 I/O と接続するデバイスは、ラッチアップ等を起こさないために、本ボードの内部ロジック用電源 +3.3V 出力を使用するか (4.4.15 内部ロジック用電源出力 +3.3V を参照ください)、またはバッファデバイス等が必要になります。

4.4.9. FPGA コンフィグレーション

FPGA コンフィグレーション IC に TE7720 (東京エレクトロンデバイス製) を実装しています。

TE7720 は、JTAG (CON2) から送られてくるデータを FLASH メモリにプログラムし、再起動時に FLASH メモリからデータを読み込み、FPGA をコンフィグレーションする IC です。 (図 4.3)

FLASH メモリの全エリアは、プロセッサからも読み書き可能です。

LAN や RS232C から、Linux システムや FPGA コンフィグレーションデータを受信し、FLASH メモリに対して書き込んで、再起動すると全く新しい機能をもったボードとして動作させることができます。

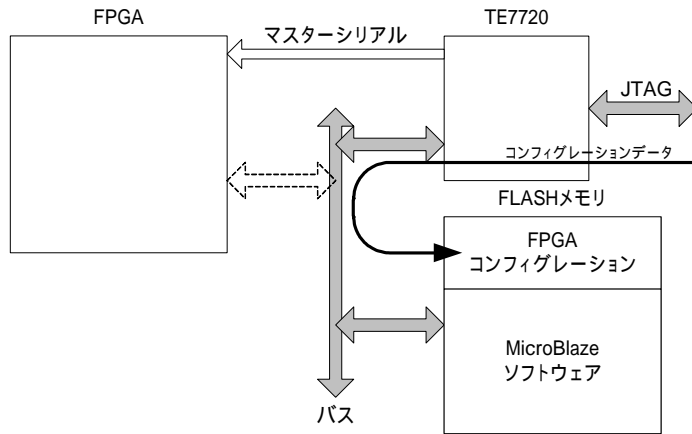
また、SUZAKU にはソフトウェアからコントロールできるリセット回路が入っていますので、遠隔地からの再コンフィグレーションも可能です。

JTAG (CON2) から TE7720 にデータを転送するためのソフトウェア (LBPLAY2.EXE) は、東京エレクトロンデバイスのホームページから無料でダウンロードできます。(付属 CD にも収録されています)

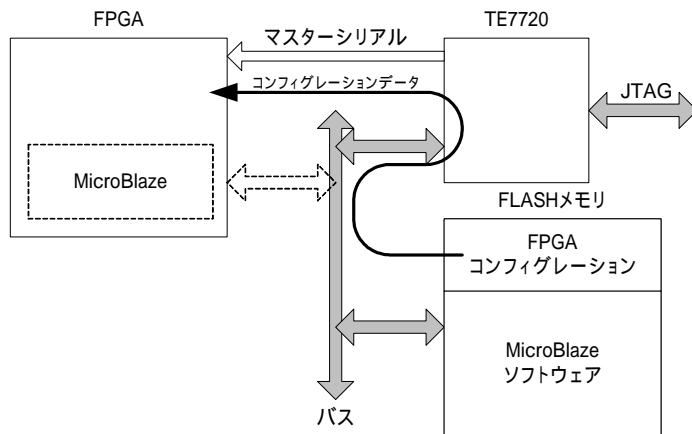
コンフィグレーション方法については、10 . FPGA コンフィグレーション を参照ください。

FPGA に間違っただデータをプログラムしたり、またプログラム中に何かの原因で、エラーを起こした場合は SUZAKU を動作させないでください。FPGA 外部回路部品 (ボード上の部品も含む) と信号の衝突や異常動作により発熱、劣化、破損する可能性がありますので、一度電源を切断し、'JP2' をショートし、再プログラミングを行ってください。

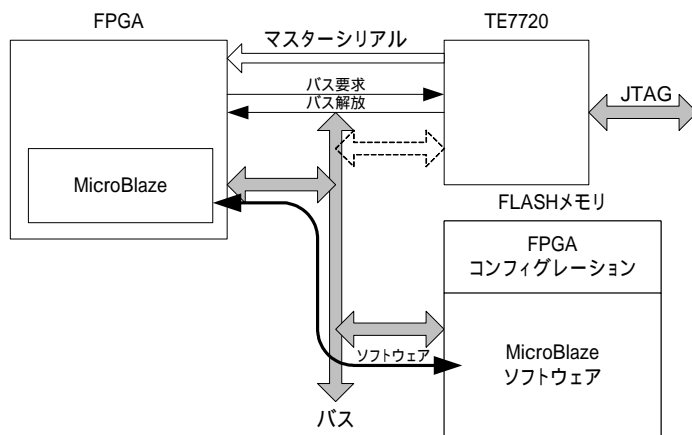
SUZAKU は、電源再投入時 'JP2' をショートすると、FPGA に対しコンフィグレーションを停止させることができ、その間にプログラムすることができます。



JTAGからTE7720経由でFLASHメモリに書き込み



電源投入時FLASHメモリからTE7720経由でFPGAをコンフィグレーション



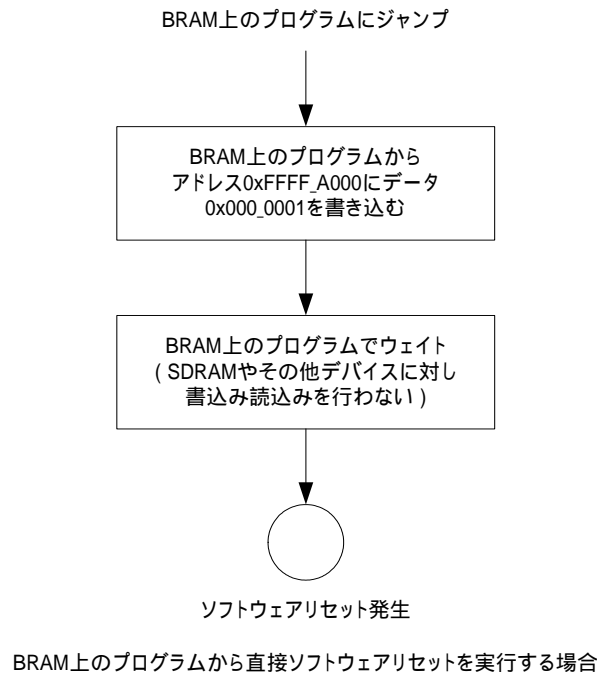
FPGAのコンフィグレーション完了後、MicroBlazeがFLASHメモリを使用

図 4-3 FPGA コンフィグレーション

4.4.10. ソフトウェアリセット機能

ソフトウェアリセットを実行すると、FLASH メモリからコンフィグレーションデータの再読み込み及び、FPGA のコンフィグレーションを実行、各デバイス IC へリセットを出力します。

ソフトウェアリセットは、Linux の reboot コマンドを使用するか、または、BRAM 上のプログラムから、直接アドレス 0xFFFF_A000 にデータ 0x000_0001 を書き込むことにより実行できます。BRAM 上のプログラムから直接ソフトウェアリセットを実行する場合は、SDRAM やその他デバイスに対し書き込み読み込み（プログラムの実行を含む）を行わないでください。



4.4.11. JTAG

JTAG には、以下の 2 種類があります。

・FPGA プログラム用 JTAG コネクタ(CON2)

FPGA のコンフィグレーションデータを Flash メモリにプログラムする時に使用する JTAG コネクタです。
(コネクタは実装されていません)

本コネクタに Xilinx 製 Parallel Cable 等の JTAG ケーブルを接続し、専用のソフト(LBPLAY2.EXE)を使用してプログラムを行います。

本 JTAG の I/O 電圧は+3.3V です。+3.3V に対応した JTAG ケーブルをご使用ください。

また、TMS、TDI、TCK は、本ボード内で 4.7k を介し+3.3V にプルアップされています。

コンフィグレーション方法については、10 . FPGA コンフィグレーション を参照ください。

・FPGA 用 JTAG コネクタ(CON7)

FPGA 用 JTAG コネクタです。(コネクタは実装されていません)

FPGA の JTAG ピンと直接接続されています。

本 JTAG の I/O 電圧は+2.5V です。+2.5V に対応した JTAG ケーブルをご使用ください。

また、TMS、TDI、TCK は、本ボード内で 4.7k を介し+2.5V にプルアップされています。

4.4.12. 設定用ジャンパ

設定用ジャンパには、以下の 2 種類があります。

・起動モードジャンパ (JP1)

起動モードを切り替えるジャンパです。

オープンでオートブートします。

ショートでブートローダモードになります。

(起動モードについての詳細はソフトウェアマニュアルを参照してください)

・FPGA プログラム用ジャンパ(JP2 , Spartan-3 の F3 と接続しています)

FPGA プログラム用 JTAG からコンフィグレーションデータを Flash メモリにプログラムする時に使用するジャンパです。

オープンでノーマルブートします。

ショートで FPGA コンフィグレーションデータを Flash メモリにプログラムできます。

コンフィグレーション方法については、10 . FPGA コンフィグレーション を参照ください。

(電源再投入時、本ジャンパをショートすると、FPGA に対しコンフィグレーションを停止させることができ、その時に Flash メモリにプログラムできます)

4.4.13. LED

LED には、以下の 2 種類があります。

・パワーオン LED 緑 (D3)

本ボードに 3.3V が供給されると点灯します。

・ユーザコントロール LED 赤 (D1 , Spartan-3 の G5 と接続しています)

ユーザコントロール可能な LED です。

‘ LO. ’ レベルで点灯します。

FPGA と接続しています。

4.4.14. 電源入力+3.3V

CON2、CON3 及び CON6 の ‘ 電源入力+3.3V ’ から、本ボードへの電源供給が可能です。

+3.3V は、精度 ±3% で、単調増加としてください。

極度に短い間隔でのオン/オフ繰り返しは行わないでください。

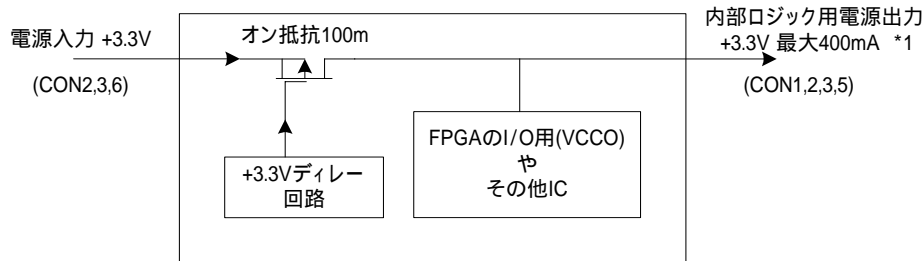
また、入力には積層セラミックコンデンサ 10 μF を実装しています。

4.4.15. 内部ロジック用電源出力+3.3V

内部ロジック用電源+3.3Vは、FPGAのI/O用(VCCO)やその他ICに供給している電源です。

CON1、CON2、CON3、CON5から、外部のデバイスに合計最大400mA *1の電源供給が可能です。

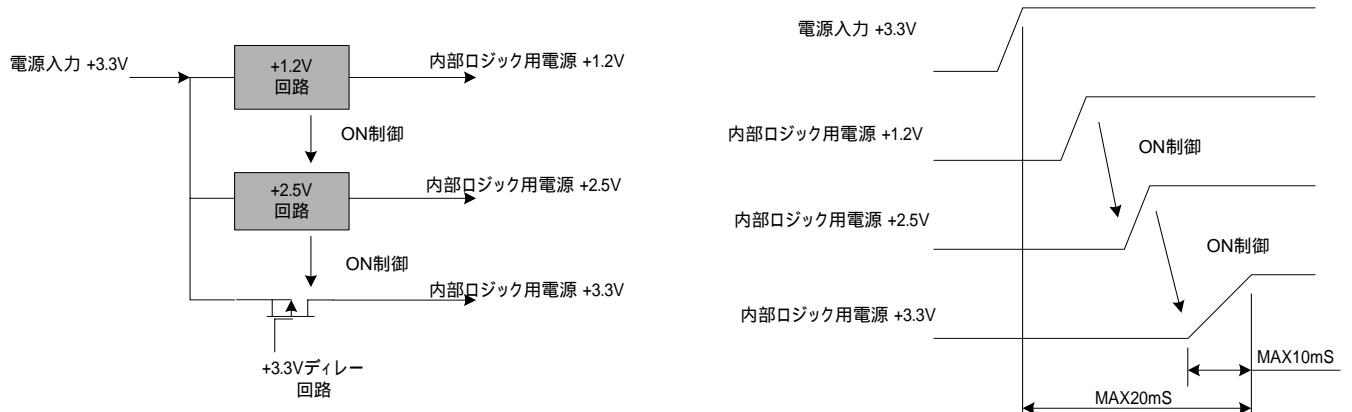
ただし、外部のデバイスの負荷変動が大きい場合、電源入力+3.3Vの応答によっては、電圧変動が発生することがあります。



- *1 外部 I/O から信号を出力する場合は、
合計最大電流 = 400mA - 外部 I/O 信号の出力電流
となります。

4.4.16. 内部電源シーケンス

内部電源は、以下のようなシーケンスで立ち上がります。



5. メモリマップ

5.1. SUZAKU メモリマップ

本ボードのメモリマップは次の通りです。
本構成は、uCLinux を動作させる最小構成です。

表 5-1 SUZAKU メモリマップ

Start Address	End Address	ペリフェラル	デバイス
0x0000 0000	0x0000 1FFF	BRAM	
0x0000 1000	0x7FFF FFFF	Reserved	
0x8000 0000	0x80FF FFFF	OPB-SDRAM Controller	SDRAM 16MByte
0x8100 0000	0xFEFF FFFF	Free	
0xFF00 0000	0xFF7F FFFF	OPB-EMC	FLASH メモリ 4MByte
0xFF80 0000	0xFFCF FFFF	Free	
0xFFE0 0000	0xFFEF FFFF	OPB-EMC	LAN コントローラ
0xFFFF 0000	0xFFFF 0FFF	Free	
0xFFFF 1000	0xFFFF 10FF	OPB-Timer	
0xFFFF 1100	0xFFFF 1FFF	Free	
0xFFFF 2000	0xFFFF 20FF	OPB-UART Lite	RS232C
0xFFFF 2100	0xFFFF 2FFF	Free	
0xFFFF 3000	0xFFFF 30FF	OPB-Interrupt Controller	
0xFFFF 3100	0xFFFF 9FFF	Free	
0xFFFF A000	0xFFFF A0FF	OPB-GPIO	ブートモードジャンパ LED ソフトウェアリセット
0xFFFF A100	0xFFFF FFFF	Free	

6.FPGA ピンアサイン

FPGA(Xilinx Spartan-3 XC3S400 FT256)の全ピンアサインを示します。

表 6-1 FPGA ピンアサイン 外部 I/O 系 (1/3)

番号	バンク	信号名	I/O	用途	接続先
A5	0	IOa_0	I/O	外部 I/O	CON2 (7 項参照)
A7	0	IOb_0	I/O	"	"
A3	0	IO/VREF0P	I/O	"	"
D5	0	IO/VREF0N	I/O	"	"
B4	0	01N_0/VRP_0	I/O	"	"
A4	0	01P_0/VRN_0	I/O	"	"
C5	0	25N_0	I/O	"	"
B5	0	25P_0	I/O	"	"
E6	0	27N_0	I/O	"	"
D6	0	27P_0	I/O	"	"
C6	0	28N_0	I/O	"	"
B6	0	28P_0	I/O	"	"
E7	0	29N_0	I/O	"	"
D7	0	29P_0	I/O	"	"
C7	0	30N_0	I/O	"	"
B7	0	30P_0	I/O	"	"
D8	0	31N_0	I/O	"	"
C8	0	31P_0/VREF0	I/O	"	"
B8	0	32N_0/GCLK7	I/O	"	"
A8	0	32P_0/GCLK6	I/O	"	"
A9	1	IOa_1	I/O	"	"
A12	1	IOb_1	I/O	"	"
C10	1	IOc_1	I/O	"	"
D12	1	IO/VREF1	I/O	"	"
A14	1	01N_1/VRP_1	I/O	"	"
B14	1	01P_1/VRN_1	I/O	"	"
A13	1	10N_1/VREF1	I/O	"	"
B13	1	10P_1	I/O	"	"
B12	1	27N_1	I/O	"	"
C12	1	27P_1	I/O	"	"
D11	1	28N_1	I/O	"	"
E11	1	28P_1	I/O	"	"

表 6-2 FPGA ピンアサイン 外部 I/O 系(2/3)

番号	バンク	信号名	I/O	機能	接続先
B11	1	29N_1	I/O	外部 I/O	CON3 (7 項参照)
C11	1	29P_1	I/O	"	"
D10	1	30N_1	I/O	"	"
E10	1	30P_1	I/O	"	"
A10	1	31N_1/VREF1	I/O	"	"
B10	1	31P_1	I/O	"	"
C9	1	32N_1/GCLK5	I/O	"	"
D9	1	32P_1/GCLK4	I/O	"	"
G16	2	Oa_2	I/O	"	"
B16	2	16P_2	I/O	"	"
C16	2	17N_2	I/O	"	"
C15	2	17P_2/VREF2	I/O	"	"
D14	2	19N_2	I/O	"	"
D15	2	19P_2	I/O	"	"
D16	2	20N_2	I/O	"	"
E13	2	20P_2	I/O	"	"
E14	2	21N_2	I/O	"	"
E15	2	21P_2	I/O	"	"
E16	2	20P_2	I/O	"	"
F12	2	21N_2	I/O	"	"
F13	2	21P_2	I/O	"	"
F14	2	22N_2	I/O	"	"
F15	2	22P_2	I/O	"	"
G12	2	23N_2/VREF2	I/O	"	"
G13	2	23P_2	I/O	"	"
G14	2	24N_2	I/O	"	"
G15	2	24P_2	I/O	"	"
H13	2	39N_2	I/O	"	"
H14	2	39P_2	I/O	"	"
H15	2	40N_2	I/O	"	"
H16	2	40P_2/VREF2	I/O	"	"
K15	3	IOa_3	I/O	"	"
P16	3	01N_3/VRP_3	I/O	"	"
R16	3	01P_3/VRN_3	I/O	"	"

表 6-3 FPGA ピンアサイン 外部 I/O 系 (3/3)

番号	バンク	信号名	I/O	機能	接続先
P15	3	16N_3	I/O	外部 I/O	CON5 (7 項参照)
P14	3	16P_3	I/O	"	"
N16	3	17N_3	I/O	"	"
N15	3	17P_3/VREF3	I/O	"	"
M14	3	19N_3	I/O	"	"
N14	3	19P_3	I/O	"	"
M16	3	20N_3	I/O	"	"
M15	3	20P_3	I/O	"	"
L13	3	21N_3	I/O	"	"
M13	3	21P_3	I/O	"	"
L15	3	22N_3	I/O	"	"
L14	3	22P_3	I/O	外部 I/O	CON4 (7 項参照)
K12	3	23N_3	I/O	"	"
L12	3	23P_3/VREF3	I/O	"	"
K14	3	24N_3	I/O	"	"
K13	3	24P_3	I/O	"	"
J14	3	39N_3	I/O	"	"
J13	3	39P_3	I/O	"	"
J16	3	40N_3/VREF3	I/O	"	"
K16	3	40P_3	I/O	"	"

表 6-4 FPGA ピンアサイン 内部デバイス系(1/3)

番号	バンク	信号名	I/O	機能	接続先
T12	4	LA(22)	O	FPGA 外部アドレスバス	SDRAM、FLASH メモリ、 LAN コントローラ
T14	4	LA(21)	O	"	"
N12	4	LA(20)	O	"	"
P13	4	LA(19)	O	"	"
T10	4	LA(18)	O	"	"
R13	4	LA(17)	O	"	"
T13	4	LA(16)	O	"	"
P12	4	LA(15)	O	"	"
R12	4	LA(14)	O	"	"
M11	4	CFG_DATA	I	コンフィグレーション DATA	TE7720
N11	4	LA(13)	O	FPGA 外部アドレスバス	SDRAM,FLASH メモリ、 LAN コントローラ
P11	4	LA(12)	O	"	"
R11	4	LA(11)	O	"	"
M10	4	LA(10)	O	"	"
N10	4	LA(9)	O	"	"
P10	4	LA(8)	O	"	"
R10	4	SYS_CLK_OUT	O	SDRAM へのクロック出力	SDRAM
N9	4	CFG_INIT*	I	コンフィグレーション INIT	TE7720、JP2
P9	4			空き	
R9	4	RAM_CLK	I	SDRAM のクロック DCM フォードバック用入力	SDRAM
T9	4	SYS_CLK_IN	I	システムクロック入力	発振器 3.6864MHz
N5	5	LA(7)	O	FPGA 外部アドレスバス	SDRAM,FLASH メモリ、 LAN コントローラ
P7	5	LA(6)	O	"	"
T5	5	LA(5)	O	"	"
T8	5	LA(4)	O	"	"
T3	5	LA(3)	O	"	"
R3	5	LA(2)	O	"	"
T4	5	LA(1)	O	"	"
R4	5	LA(0)	O	"	"
R5	5	LD(15)	I/O	FPGA 外部データバス	SDRAM,FLASH メモリ、 LAN コントローラ
P5	5	LD(14)	I/O	"	"
N6	5	LD(13)	I/O	"	"
M6	5	LD(12)	I/O	"	"
R6	5	LD(11)	I/O	"	"

表 6-5 FPGA ピンアサイン 内部デバイス系(2/3)

番号	バンク	信号名	I/O	機能	接続先
P6	5	LD(10)	I/O	"	"
N7	5	LD(9)	I/O	"	"
M7	5	LD(8)	I/O	"	"
T7	5	LD(7)	I/O	"	"
R7	5	LD(6)	I/O	"	"
P8	5			空き	
N8	5			空き	
K1	6	LDA(5)	I/O	FPGA 外部データバス	SDRAM,FLASH メモリ、 LAN コントローラ
R1	6	LD(4)	I/O	"	"
P1	6	LD(3)	I/O	"	"
P2	6	LD(2)	I/O	"	"
N3	6	LD(1)	I/O	"	"
N2	6	LD(0)	I/O	"	"
N1	6	BUS_REQ	O	バスリクエスト	TE7720
M4	6	BUS_REL	I	バス獲得	TE7720
M3	6	RAM_CS*	O	SDRAM CS	SDRAM
M2	6	RAM_RAS*	O	SDRAM RAS	"
M1	6	RAM_CAS*	O	SDRAM CAS	"
L5	6	RAM_WE*	O	SDRAM WE	"
L4	6	RAM_CKE	O	SDRAM CKE	"
L3	6	RAM_UQDM	O	SDRAM UQDM	"
L2	6	RAM_LQDM	O	SDRAM LQDM	"
K5	6	RAM_BS1	O	SDRAM BS	"
K4	6	RAM_BS0	O	SDRAM BS	"
K3	6	FLASH_CE*	O	FLASH メモリ CE	FLASH メモリ
K2	6	FLASH_OE*	O	FLASH メモリ OE	"
J4	6	FLASH_WE*	O	FLASH メモリ WE	"
J3	6	FLASH_BYTE*	O	FLASH メモリ BYTE*	"
J2	6	FLASH_R_B	I	FLASH メモリ R/B	"
J1	6	MAC_BE1*	O	LAN コントローラ BE1	LAN コントローラ
G2	7	MAC_BE0*	O	LAN コントローラ BE0	"
C1	7	MAC_AEN	O	LAN コントローラ AEN	"
B1	7	MAC_RD*	O	LAN コントローラ RD	"
C2	7	MAC_WR*	O	LAN コントローラ WR	"
C3	7	MAC_ARDY	I	LAN コントローラ ARDY	"
D1	7	MAC_ADS*	O	LAN コントローラ ADS	"
D2	7	MAC_INTR	I	LAN コントローラ INTR	"
E3	7			空き	
D3	7			空き	
E1	7	CNSL_CTS*	I	コンソール CTS	RS232C トランシーバ =>CON1 (7 項参照)
E2	7	CNSL_RXD	I	コンソール RXD	"
F4	7	CNSL_RTS	O	コンソール RTS	"
E4	7	CNSL_TXD	O	コンソール TXD	"

表 6-6 FPGA ピンアサイン 内部デバイス系(3/3)

番号	バンク	信号名	I/O	機能	接続先
F2	7	FPGA_RESET_EN	O	自己リセット出力	リセット回路
F3	7	BOOTMODE	I	ブートモード検出	JP1 (7 項参照)
G5	7	LED*	O	ユーザコントロール LED	D1 (7 項参照)
F5	7	SYS_RST_IN	I	システムリセット入力	リセット回路
G3	7			空き	
G4	7			空き	
H3	7			空き	
H4	7			空き	
H1	7			空き	
G1	7			空き	

表 6-7 FPGA ピンアサイン JTAG、コンフィグレーション系

番号	バンク	信号名	I/O	機能	接続先
C14		TCK	I	JTAG	CON7 (7 項参照)
A2		TDI	I	JTAG	"
A15		TDO	O	JTAG	"
C13		TMS	I	JTAG	"
T15		CFG_CLK	O	コンフィグレーション CLK	TE7720
B3		PROG_B	I	コンフィグレーション PROG_B	リセット回路
R14		CFG_DONE	O	コンフィグレーション DONE	TE7720
C4		HSWAP_EN		オープン	
P3		M0	I	コンフィグレーションモード	グラウンド
T2		M1	I	コンフィグレーションモード	グラウンド
P4		M2	I	コンフィグレーションモード	グラウンド

7. 各種インターフェース仕様

7.1. 各種インターフェースの配置

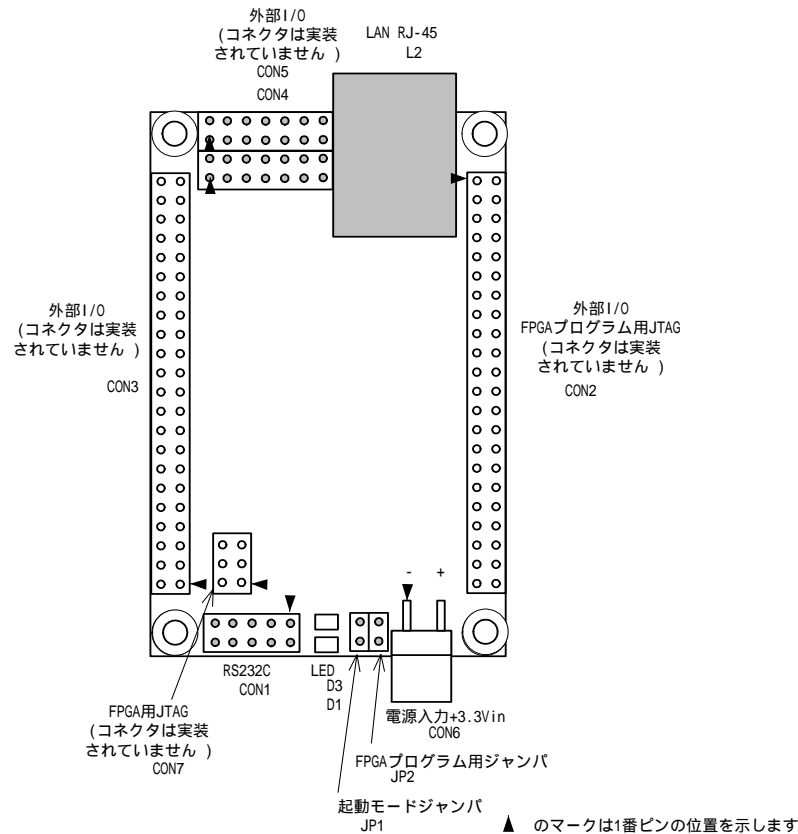


図 7-1 各種インターフェースの配置

表 7-1 各種インターフェースの内容

部品番号	説明
CON2	外部 I/O、FPGA プログラム用 JTAG コネクタ Total I/Os 32PIN
CON3	外部 I/O コネクタ Total I/Os 34PIN
CON4	外部 I/O コネクタ Total I/Os 10PIN
CON5	外部 I/O コネクタ Total I/Os 10PIN
L2	Ethernet 10/100 Base-T コネクタ
CON7	FPGA JTAG コネクタ
CON1	RS232C コネクタ
D3	パワーオン LED 緑
D1	ユーザコントロール LED 赤
JP1	起動モードジャンパ
JP2	FPGA プログラム用ジャンパ
CON6	電源入力+3.3V コネクタ

7.2.CON2 外部 I/O、FPGA プログラム用 JTAG コネクタ

外部 I/O 及び FPGA プログラム用 JTAG コネクタです。(コネクタは実装されていません)

表 7-2 CON2 外部 I/O、FPGA プログラム用コネクタ

番号	信号名	I/O	機 能
1	GND		グラウンド
2	+3.3VOUT		内部ロジック用電源出力 +3.3V
3	CFG_TCK		FPGA プログラム用 JTAG TCK
4	CFG_TDI		" TDI
5	CFG_TDO		" TDO
6	CFG_TMS		" TMS
7	IOa_0	外部 I/O	Spartan-3 接続ピン番号 A5
8	IOb_0		" A7
9	IO/VREF0P		" A3
10	IO/VREF0N		" D5
11	01N_0/VRP_0		" B4
12	01P_0/VRN_0		" A4
13	25N_0		" C5
14	25P_0		" B5
15	27N_0		" E6
16	27P_0		" D6
17	28N_0		" C6
18	28P_0		" B6
19	GND		グラウンド
20	32P_0/GCLK6	外部 I/O	Spartan-3 接続ピン番号 A8
21	GND		グラウンド
22	32N_0/GCLK7	外部 I/O	Spartan-3 接続ピン番号 B8
23	29N_0		" E7
24	29P_0		" D7
25	30N_0		" C7
26	30P_0		" B7
27	31N_0		" D8
28	31P_0/VREF0		" C8
29	IOa_1		" A9
30	IOb_1		" A12
31	IOc_1		" C10
32	IO/VREF1		" D12
33	01N_1/VRP_1		" A14
34	01P_1/VRN_1		" B14
35	10N_1/VREF1		" A13
36	10P_1		" B13
37	27N_1		" B12
38	27P_1		" C12
39	28N_1		" D11
40	28P_1		" E11
41	GND		グラウンド
42	GND		グラウンド
43	+3.3VIN		電源入力 +3.3V
44	+3.3VIN		電源入力 +3.3V

7.3. CON3 外部 I/O コネクタ

外部 I/O 及び TE7720 用 JTAG コネクタです。(コネクタは実装されていません)

表 7-3 外部 I/O コネクタ

番号	信号名	I/O	機能
1	+3.3VIN		電源入力 +3.3V
2	+3.3VIN		電源入力 +3.3V
3	GND		グラウンド
4	GND		グラウンド
5	29N_1		外部 I/O Spartan-3 接続ピン番号 B11
6	29P_1		" C11
7	30N_1		" D10
8	30P_1		" E10
9	31N_1/VREF1		" A10
10	31P_1		" B10
11	01N_2/VRP_2		" B16
12	01P_2/VRN_2		" C16
13	16N_2		" C15
14	16P_2		" D14
15	17N_2		" D15
16	17P_2/VREF2		" D16
17	19N_2		" E13
18	19P_2		" E14
19	20N_2		" E15
20	20P_2		" E16
21	21N_2		" F12
22	21P_2		" F13
23	32N_1/GCLK5		" C9
24	GND		グラウンド
25	32P_1/GCLK4		外部 I/O Spartan-3 接続ピン番号 D9
26	GND		グラウンド
27	22N_2		外部 I/O Spartan-3 接続ピン番号 F14
28	22P_2		" F15
29	23N_2/VREF2		" G12
30	23P_2		" G13
31	24N_2		" G14
32	24P_2		" G15
33	39N_2		" H13
34	39P_2		" H14
35	40N_2		" H15
36	40P_2/VREF2		" H16
37	01N_3/VRP_3		" P16
38	01P_3/VRN_3		" R16
39	IOa_3		" K15
40	IOa_2		" G16
41			空き
42	EXRESET*		未接続 注意. 必ず未接続とし、信号を入力しないでください。
43	+3.3VOUT		内部ロジック用電源出力 +3.3V
44	GND		グラウンド

7.4. CON4 外部 I/O コネクタ

外部 I/O コネクタです。(コネクタは実装されていません)

表 7-4 CON4 外部 I/O コネクタ

番号	信号名	I/O	機 能
1			空き
2			空き
3	22N_3		外部 I/O Spartan-3 接続ピン番号 L15
4	22P_3		L14
5	23N_3		K12
6	23P_3/VREF3		L12
7	24N_3		K14
8	24P_3		K13
9	39N_3		J14
10	39P_3		J13
11	40N_3/VREF3		J16
12	40P_3		K16

7.5. CON5 外部 I/O コネクタ

外部 I/O コネクタです。(コネクタは実装されていません)

表 7-5 CON5 外部 I/O コネクタ

番号	信号名	I/O	機 能
1	GND		グラウンド
2	+3.3VOUT		内部ロジック用電源出力 +3.3V
3	16N_3		外部 I/O Spartan-3 接続ピン番号 P15
4	16P_3		P14
5	17N_3		N16
6	17P_3/VREF3		N15
7	19N_3		M14
8	19P_3		N14
9	20N_3		M16
10	20P_3		M15
11	21N_3		L13
12	21P_3		M13

7.6. CON7 FPGA JTAG コネクタ

FPGA 用 JTAG コネクタです。(コネクタは実装されていません)。

本 JTAG の I/O 電圧は+2.5V です。+2.5V に対応した JTAG ケーブルをご使用ください。

表 7-6 CON7 Spartan-3 用 JTAG コネクタ

番号	信号名	I/O	機能
1	GND		グランド
2	+2.5VOUT		内部ロジック用電源出力 +2.5V
3	TCK	I	JTAG
4	TDI	I	JTAG
5	TDO	O	JTAG
6	TMS	I	JTAG

7.7. CON1 RS232C コネクタ

RS232C コネクタです。レベルバッファを介して FPGA と接続されています。

ボード側で使用しているコネクタ型式/メーカーは、A1-10PA-2.54DSA/ヒロセ(相当品)です。

- ・ シリアルコンソールの設定
- ・ 転送レート 115.2kbps
- ・ データ 8bit
- ・ ストップ bit 1bit
- ・ フロー制御 なし

表 7-7 CON1 RS232C コネクタ

番号	信号名	I/O	機能
1			空き
2			空き
3	RXD	I	Spartan-3 接続ピン番号 E2 (シリアルコンソール用)
4	RTS	O	" F4
5	TXD	O	" E4 (シリアルコンソール用)
6	CTS	I	" E1
7			空き
8			空き
9	GND		グランド
10	+3.3VOUT		内部ロジック用電源出力 +3.3V

7.8. JP1 起動モード ジャンパ

起動モードを切り替えるジャンパです。

オープンでオートブートします。

ショートでブートローダモードになります。

FPGA と接続されています。

(起動モードについての詳細はソフトウェアマニュアルを参照してください)

表 7-8 JP1 起動モード ジャンパ

番号	信号名	I/O	機 能
1	DLOAD		オープン : オートブート ショート : ブートローダモード Spartan-3 接続ピン番号 F3
2	GND		グラウンド

7.9. JP2 FPGA プログラム用ジャンパ

FPGA プログラム用 JTAG からコンフィグレーションデータを Flash メモリにプログラムする時に使用するジャンパです。

コンフィグレーション方法については、10. FPGA コンフィグレーション を参照ください。

表 7-9 JP2 FPGA プログラム用ジャンパ

番号	信号名	I/O	機 能
1	TE77PRG		オープン : ノーマルブート ショート : コンフィグレーションデータプログラム
2	GND		グラウンド

7.10. D3 パワーオン LED

本ボードに 3.3V が供給されると点灯(緑色)します。

7.11. D1 ユーザコントロール LED

ユーザコントロール可能な LED です。

‘ LO. ’ レベルで点灯(赤)します。

FPGA と接続されています。

表 7-10 D1 ユーザコントロール LED

番号	信号名	I/O	機 能
	LED0		LO.レベル : 点灯 HI.レベル : 消灯 Spartan-3 接続ピン番号 G5

7.12. CON6 電源入力+3.3V コネクタ

電源入力コネクタです。電源入力+3.3V は、 $+3.3V \pm 3\%$ で、単調増加としてください。

CON2、CON3 の ‘ 電源入力+3.3V ’ とボード内部で接続されています。

ボード側で使用しているコネクタ型式/メーカーは、B2PS-VH/日本圧着端子(相当品)です。

ケーブル側のコネクタ型式/メーカーは、

ハウジング VHR-2N/日本圧着端子(相当品)、

コンタクト BVH-21T-P1.1/日本圧着端子(相当品) または、BVH-41T-P1.1/日本圧着端子(相当品)

が使用できます。

表 7-11 CON6 電源入力+3.3V コネクタ

番号	信号名	I/O	機 能
1	GND		グラウンド
2	+3.3VIN		電源入力 +3.3V

7.13. Ethernet 10/100 Base-T

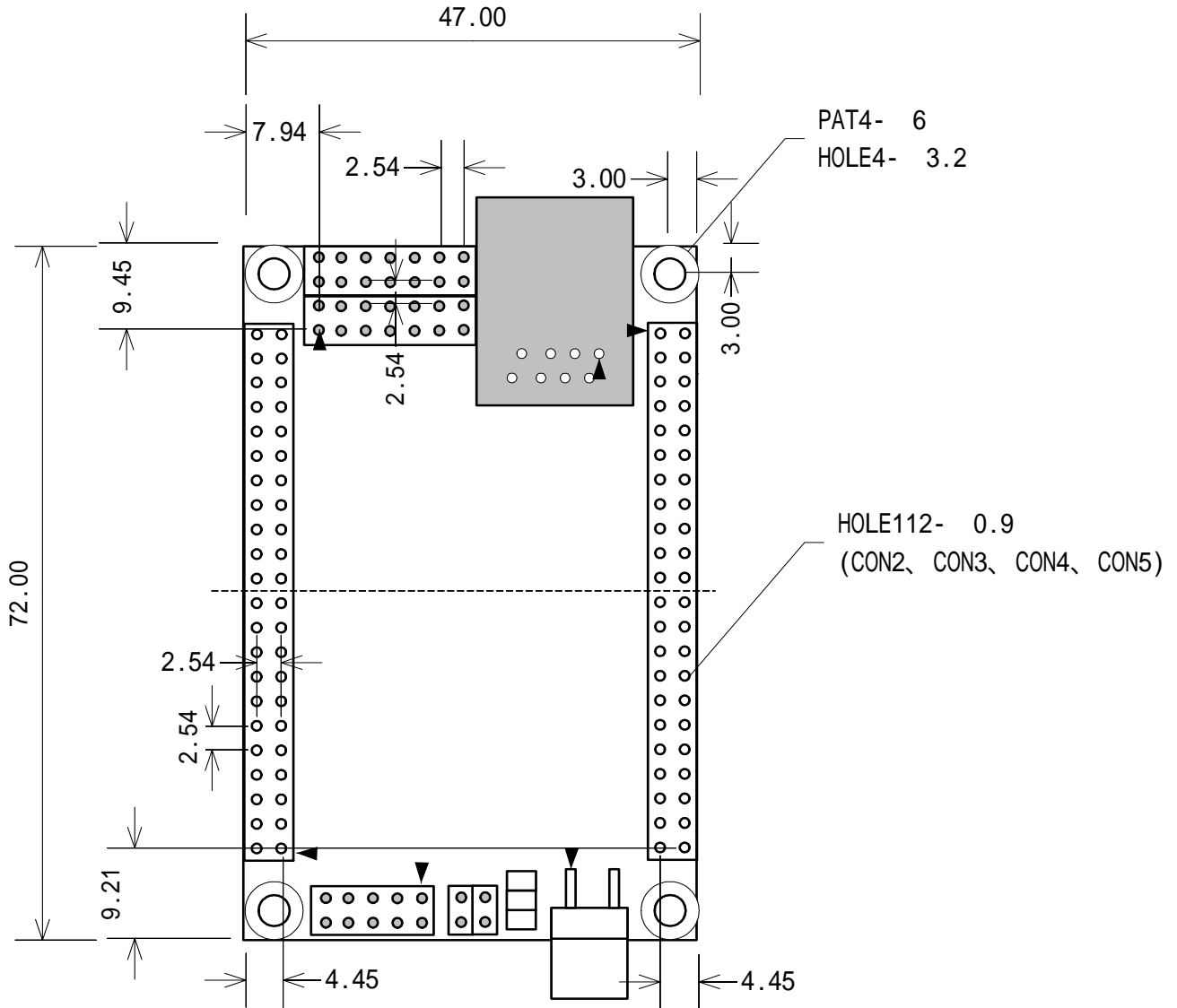
ボード側で使用しているコネクタ型式/メーカーは、J0026D21B/PULSE です。

表 7-12 Ethernet 10/100 Base-T

番号	信号名	I/O	機 能
1	TX+		差動ツイストペア出力+
2	TX-		差動ツイストペア出力-
3	RX+		差動ツイストペア入力+
4			75 終端 (4 番ピンと 5 番ピンはショートしています)
5			75 終端 (4 番ピンと 5 番ピンはショートしています)
6	RX-		差動ツイストペア入力-
7			75 終端 (7 番ピンと 8 番ピンはショートしています)
8			75 終端 (7 番ピンと 8 番ピンはショートしています)

8. 基板形状図

本ボードの基板形状図を図 8-1 に示します。



(単位:mm)

図 8-1 SUZAKU の基板形状

9. FPGA プロジェクト構築方法(例 UART の追加方法)

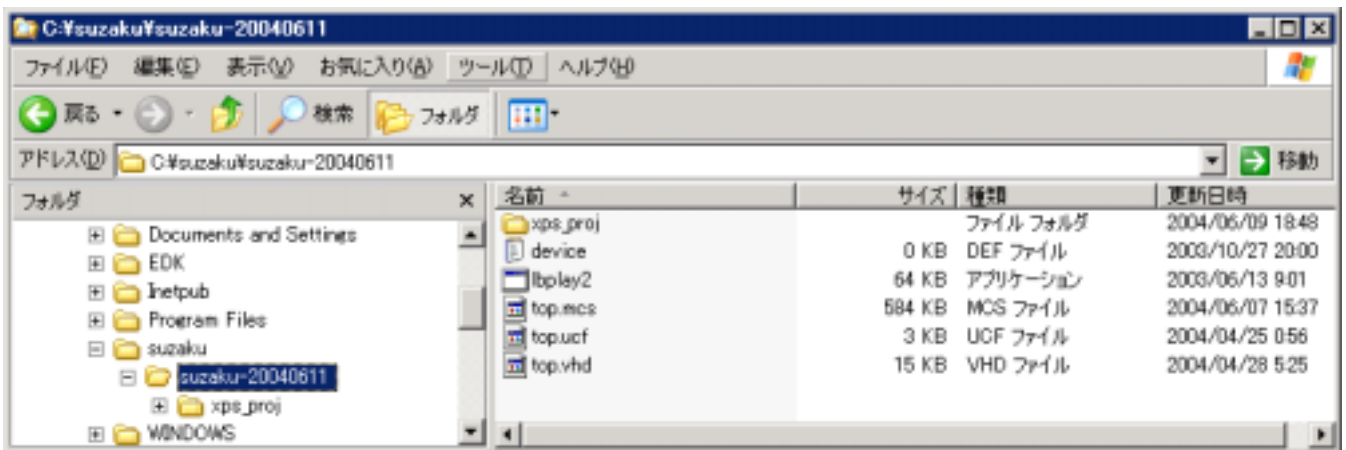
例として添付 CD-ROM のデフォルトの FPGA プロジェクトに、UART を追加する方法を説明します。

9.1. 添付 CD-ROM の fpga_proj プロジェクトの展開

添付 CD-ROM の fpga_proj の 圧縮ファイル 'suzaku-*****.zip' (*は更新日)をハードディスクに展開します。

展開後のフォルダ構成は以下のようになっています。

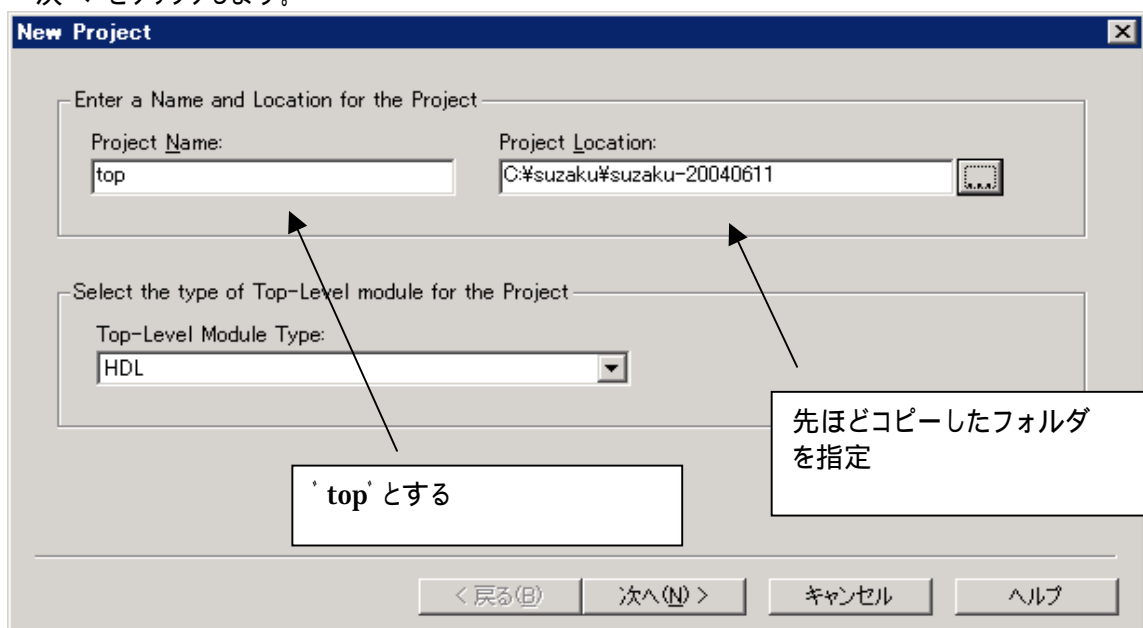
xps_proj	: EDK のプロジェクトフォルダ
top.vhd	: プロジェクトトップファイル。'xps_proj.xmp' を呼び出しています。
top.ucf	: FPGA ピンアサイン情報ファイル。
top.mcs	: FPGA コンフィグレーションデータファイル。
lbplay2.exe	: FPGA プログラム用ファイル(コンフィグレーションデバイス TE7720 用)
device.def	: FPGA プログラム用ファイル(コンフィグレーションデバイス TE7720 用)



9.2. 新規プロジェクトの作成

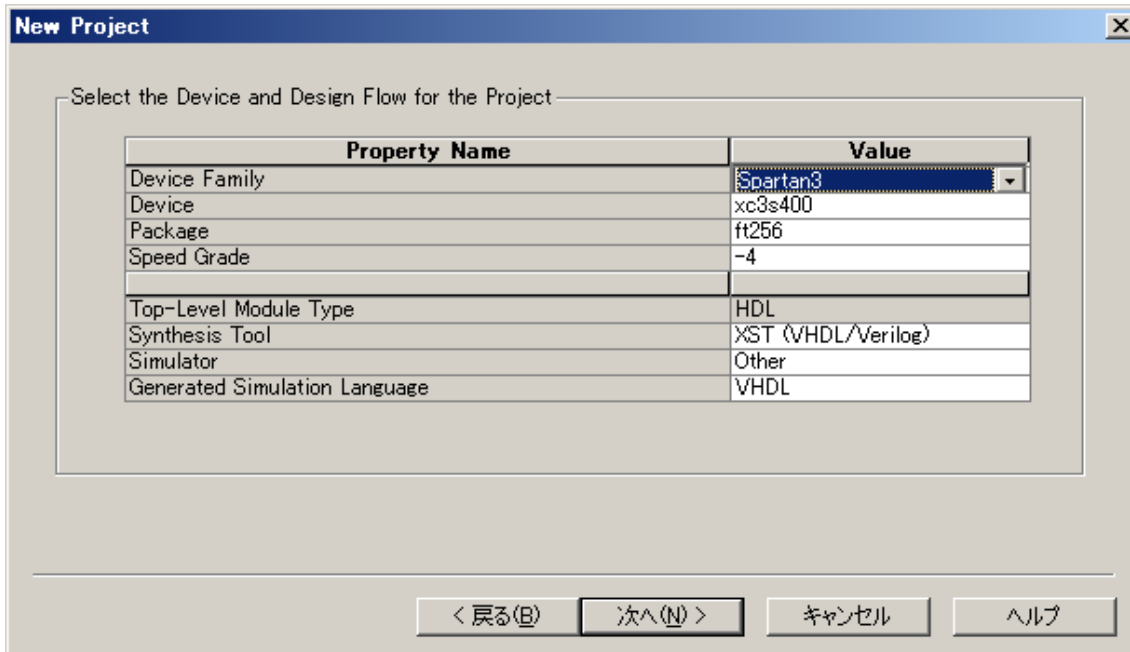
Xilinx' Project Navigator' を起動し、' File' メニューから ' New Project' をクリックします。

Project Name を ' top' とし、Project Location を先ほどプロジェクトを展開したフォルダを指定します。最後に ' 次へ' をクリックします。

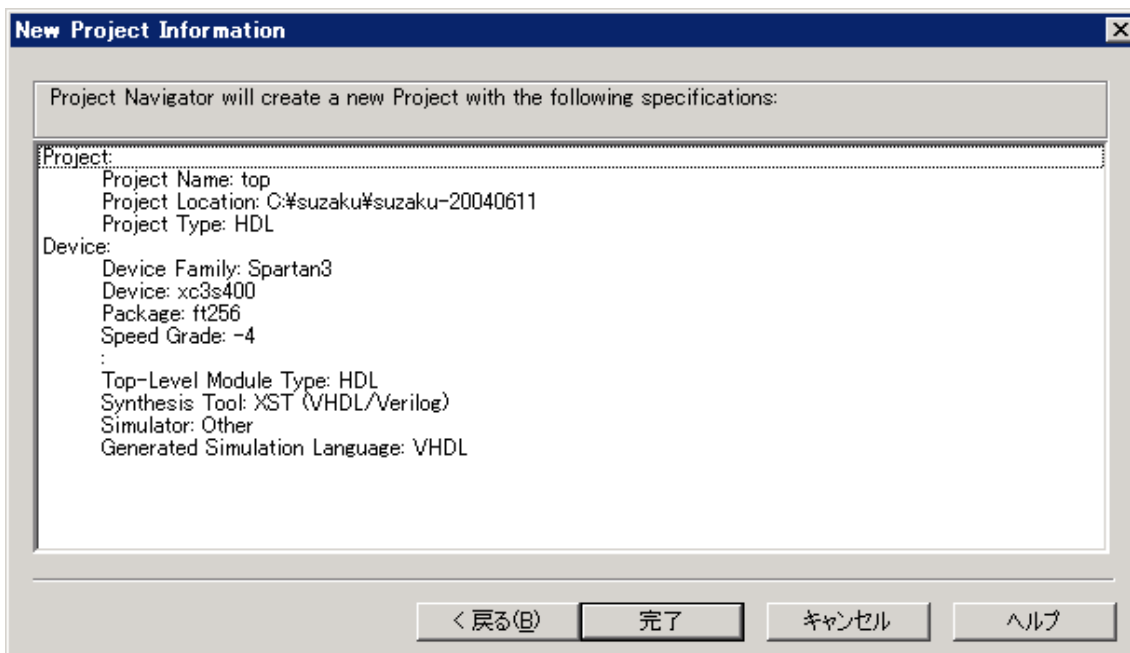


Select the Device and Design Flow for Project を以下の設定にします。

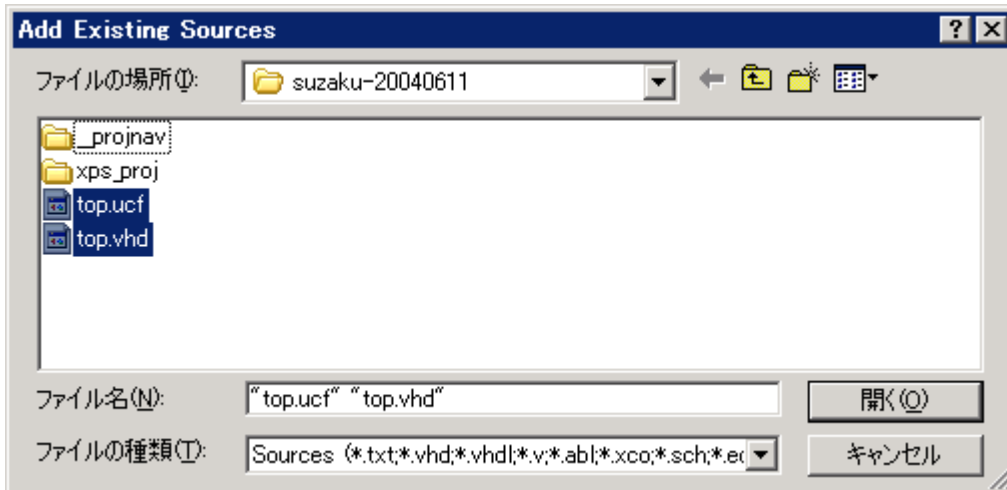
Device Family ----- Spartan-3
 Device ----- xc3s400
 Package ----- ft256
 Speed Grade ----- -4
 Synthesis Tool ----- XST(VHDL/Verilog)



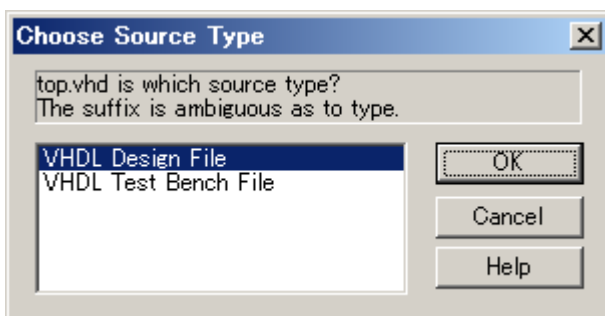
以下の画面が出るまで '次へ' をクリックし、最後に '完了' をクリックします。



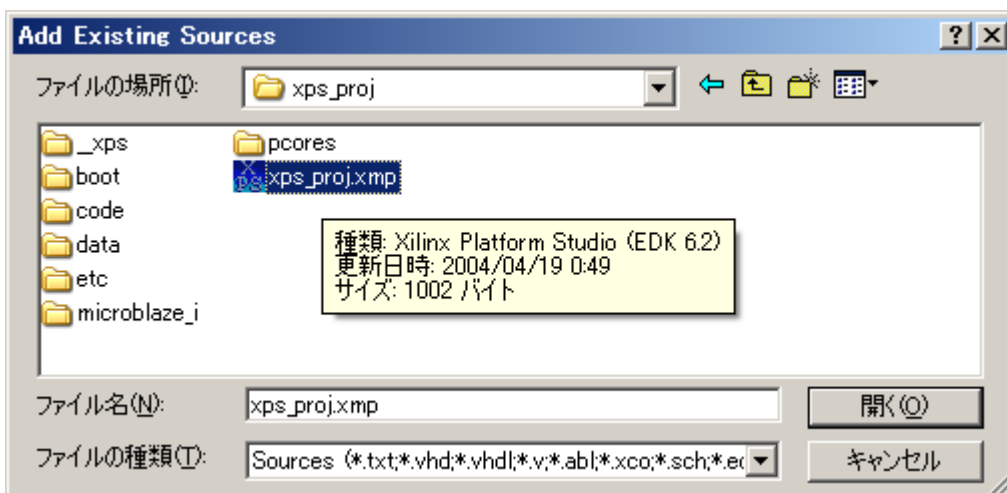
- ・ 'Project' メニューから 'Add Sources' をクリックします。
- ・ 'top.ucf' と 'top.vhd' を選択し、'開く' をクリックします。



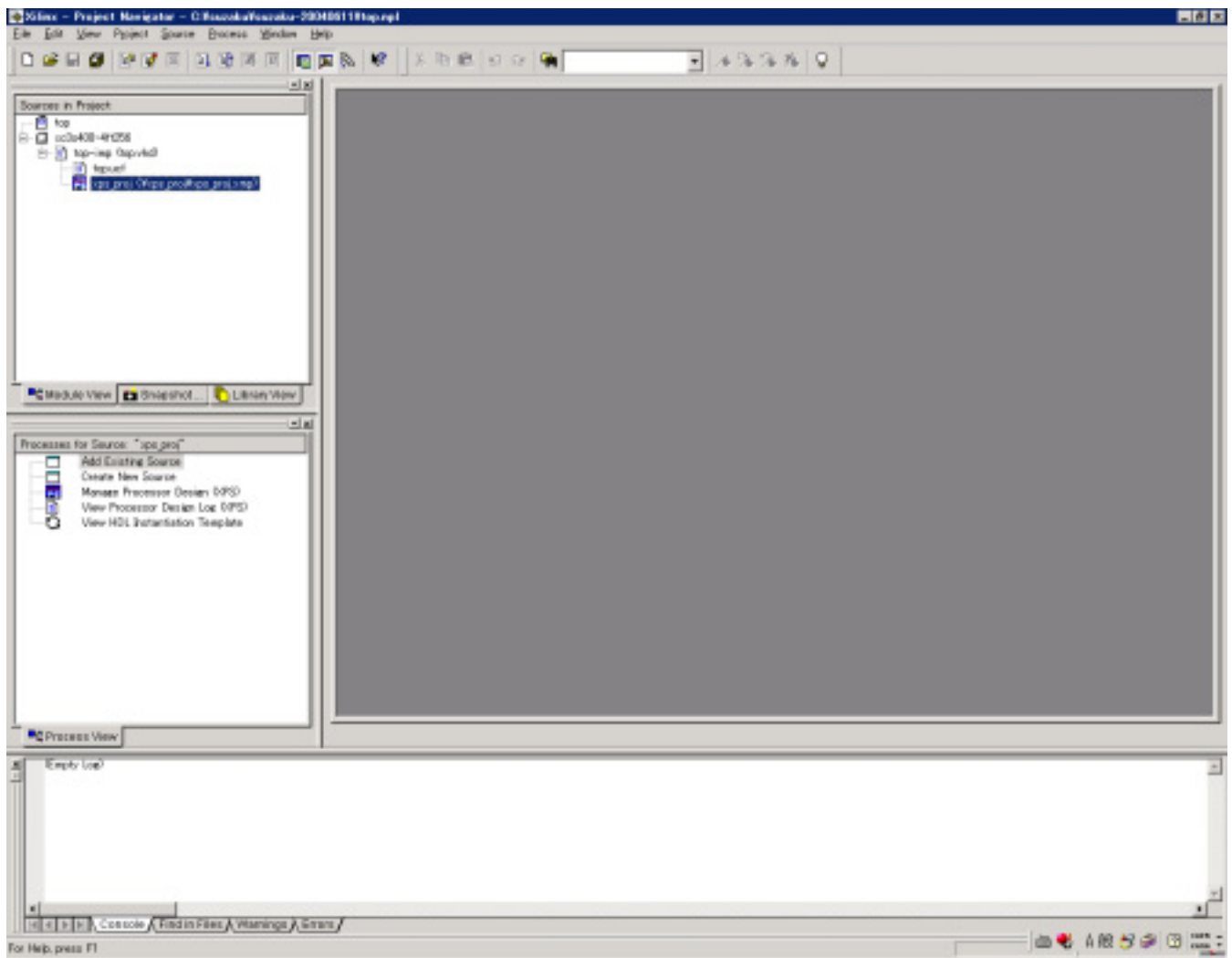
- ・ 'VHDL Design File' を選択し 'OK' をクリックします。



- ・ 再度 'Add Sources' で 'xps_prj\¥xps_proj.xmp' を選択し、'開く' をクリックします。

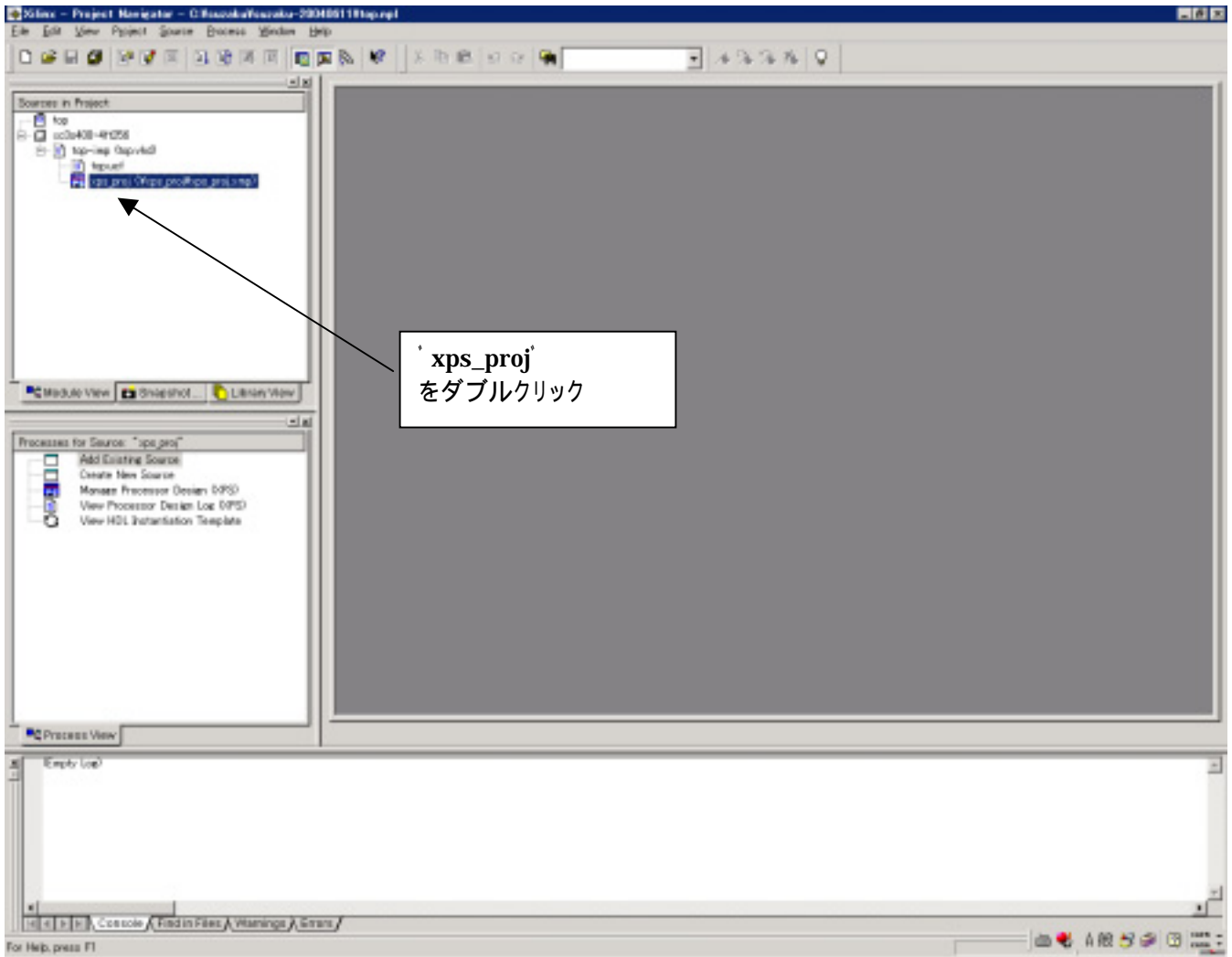


以上で新規プロジェクトができました。
下記のような表示になります。

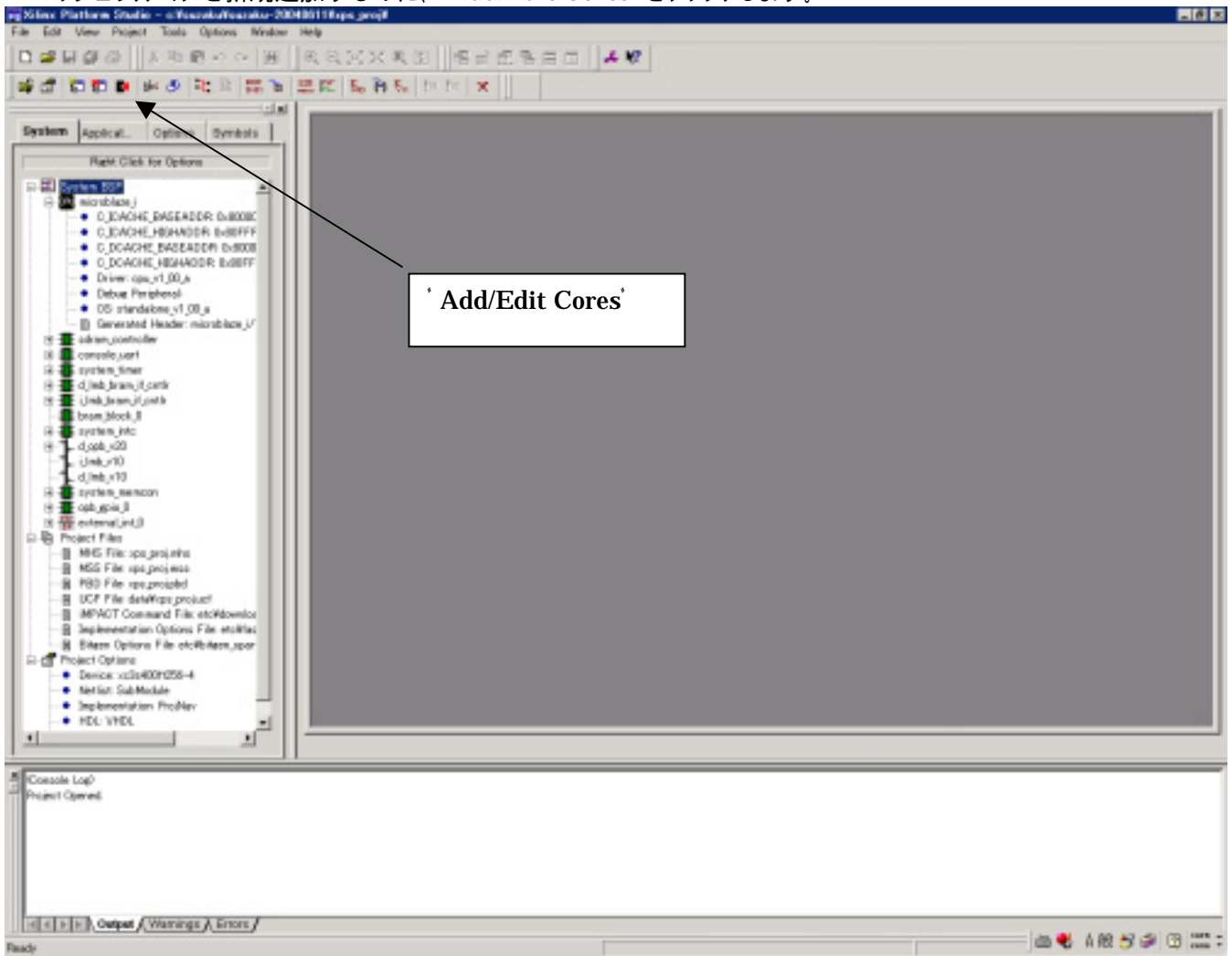


9.3. Platform Studio の設定

まず、'Project Navigator' から 'xps_proj' をダブルクリックします。
'Platform Studio' が自動的に開きます。



'Platform Studio' が自動的に開きます。
ペリフェラルコアを新規追加するのに、'Add/Edit Cores' をクリックします。

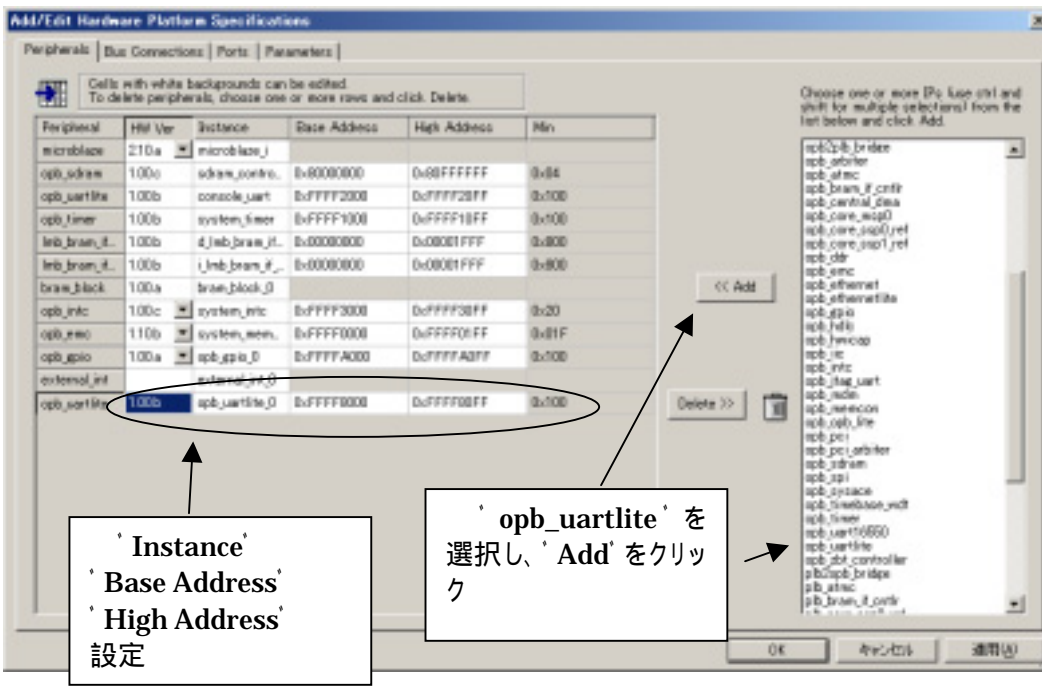


‘Peripherals’ タブで右側のペリフェラルコア一覧から ‘opb_uartlite’ を選択し、‘Add’ で追加、‘Instance’、‘Base Address’、‘High Address’ を指定します。

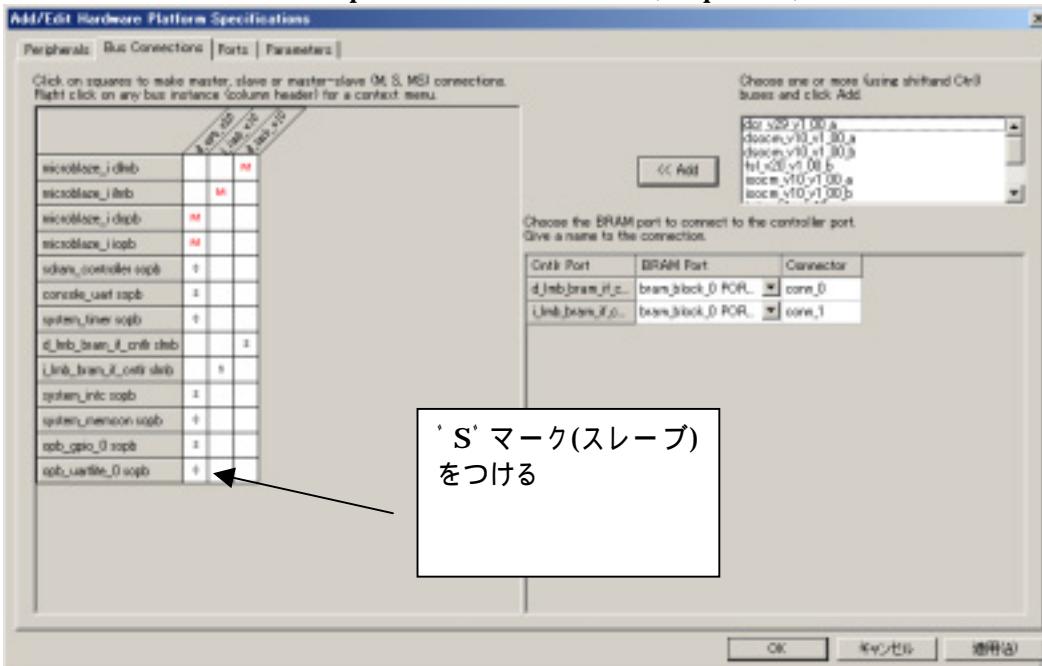
今回は、以下の設定としました。

Instance opb_uartlite_0
 Base Address 0xFFFF8000
 High Address 0xFFFF80FF

その他のデフォルトのペリフェラルコア(microblaze~external_int)構成は、uCLinux を動作させる最小構成です。



Bus Connections タブで opb_uartlite の OPB バス(d_opb_v20)をクリックして ‘S’ マーク(スレーブ)をつけます。



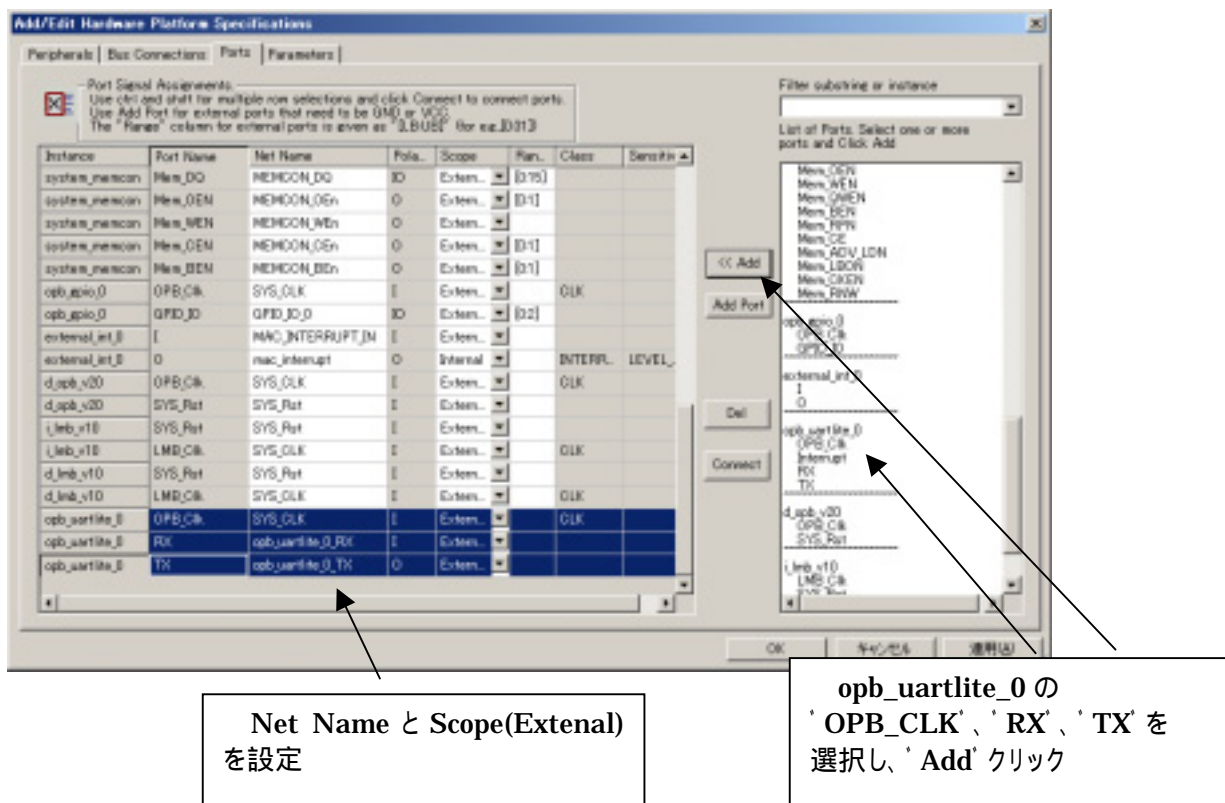
Ports タブで、右側の信号一覧の中から、' opb_uartlite_0' の ' OPB_CLK'、' RX'、' TX' を選択し、' Add' をクリックします。

Net Name にそれぞれ以下の名前をつけます。

OPB_CLK SYS_CLK(大文字)
 RX opb_uartlite_0_RX
 TX opb_uartlite_0_TX

Scope を ' External' に設定します。

Net Name に指定した場合は、top.vhd(プロジェクトトップファイル)で component として呼び出す時の port 名になります。



'parameters' タブで右側の設定項目一覧から、全てのパラメータを選択し、'Add' をクリックします。次にペリフェラルのパラメータを指定します。

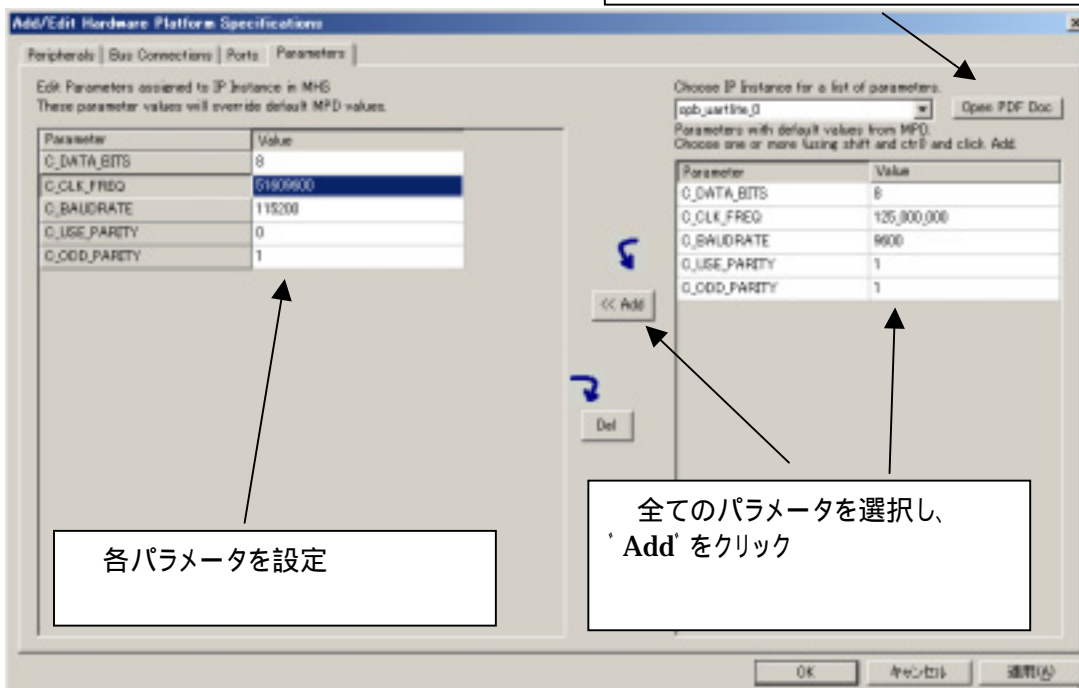
今回は以下の設定にしました。

C_DATA_BITS	8	
C_CLK_FREQ	51609600	<= SUZAKU の発振器(3.6864MHz を DCM で 14 通倍)
C_BAUDRATE	115200	
C_USEPARITY	0	

各ペリフェラルのデータシートを見る場合は、'Open PDF Doc' をクリックします

最後に 'OK' ボタンを押します。

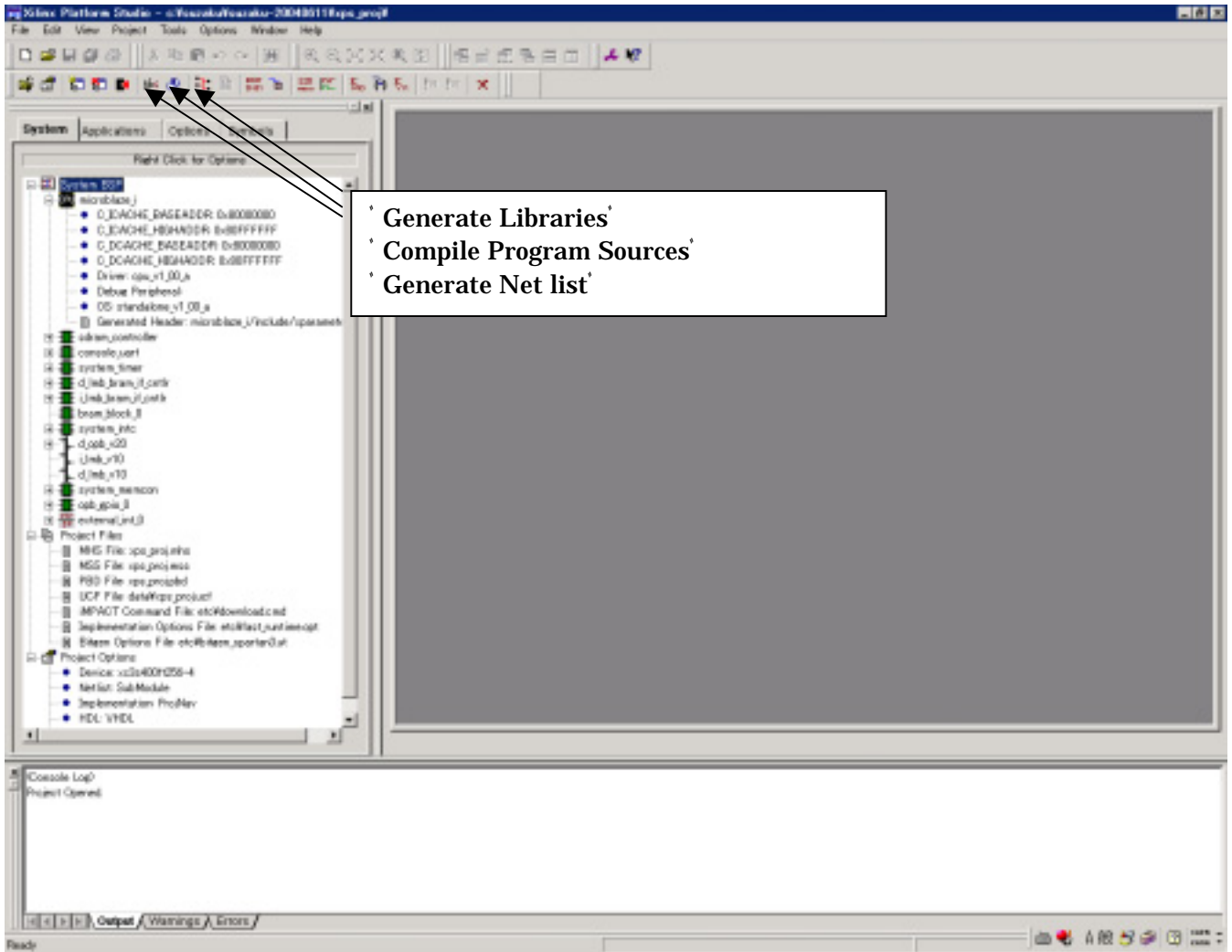
データシートを見る場合、
このボタンをクリックする



9.4. Platform Studio コンパイルの実行

'Generate Libraries'、'Compile Program Sources'、'Generate Net list' を順番にクリックし、コンパイルします。

以上で 'Platform Studio' での作業は終わりです。

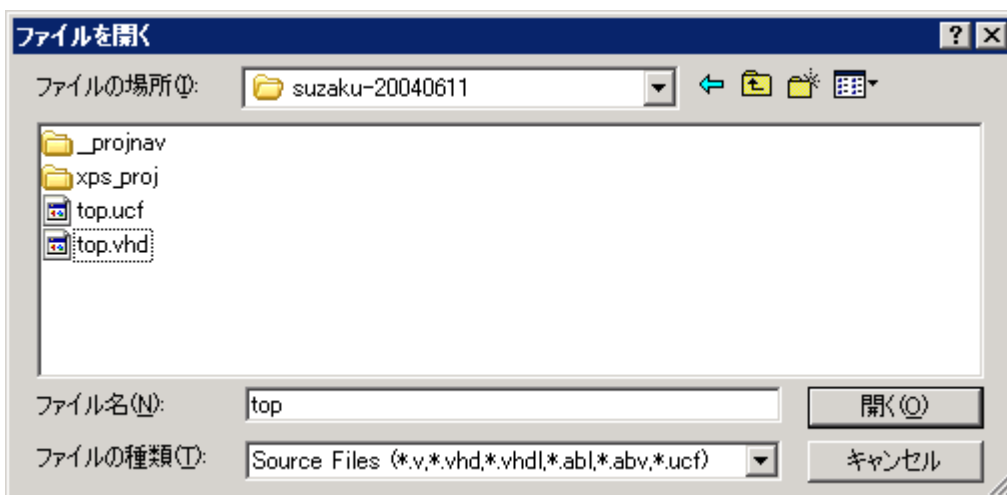
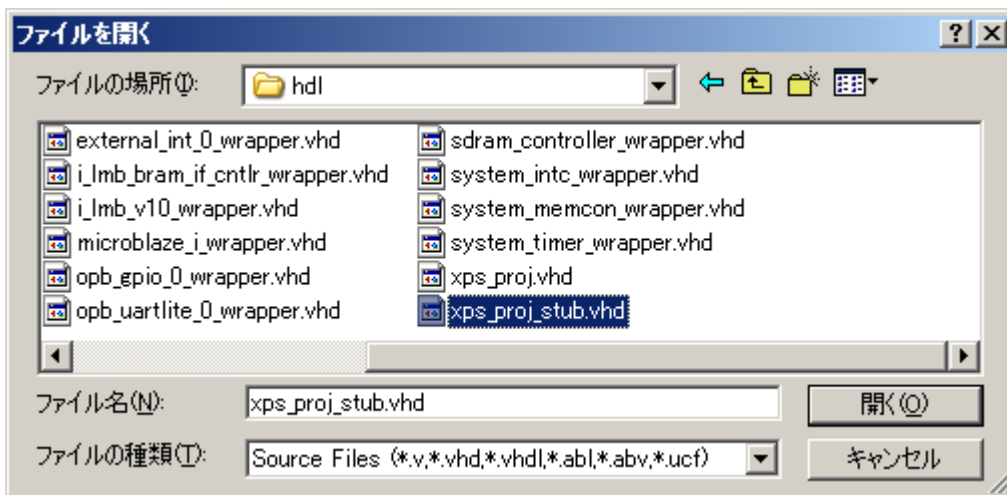


9.5. プロジェクトのトップファイル編集

'Platform Studio' で自動生成されたファイルを、プロジェクト トップファイルで呼び出すように編集します。編集するファイルは `top.vhd` と `top.ucf` の 2 つです。

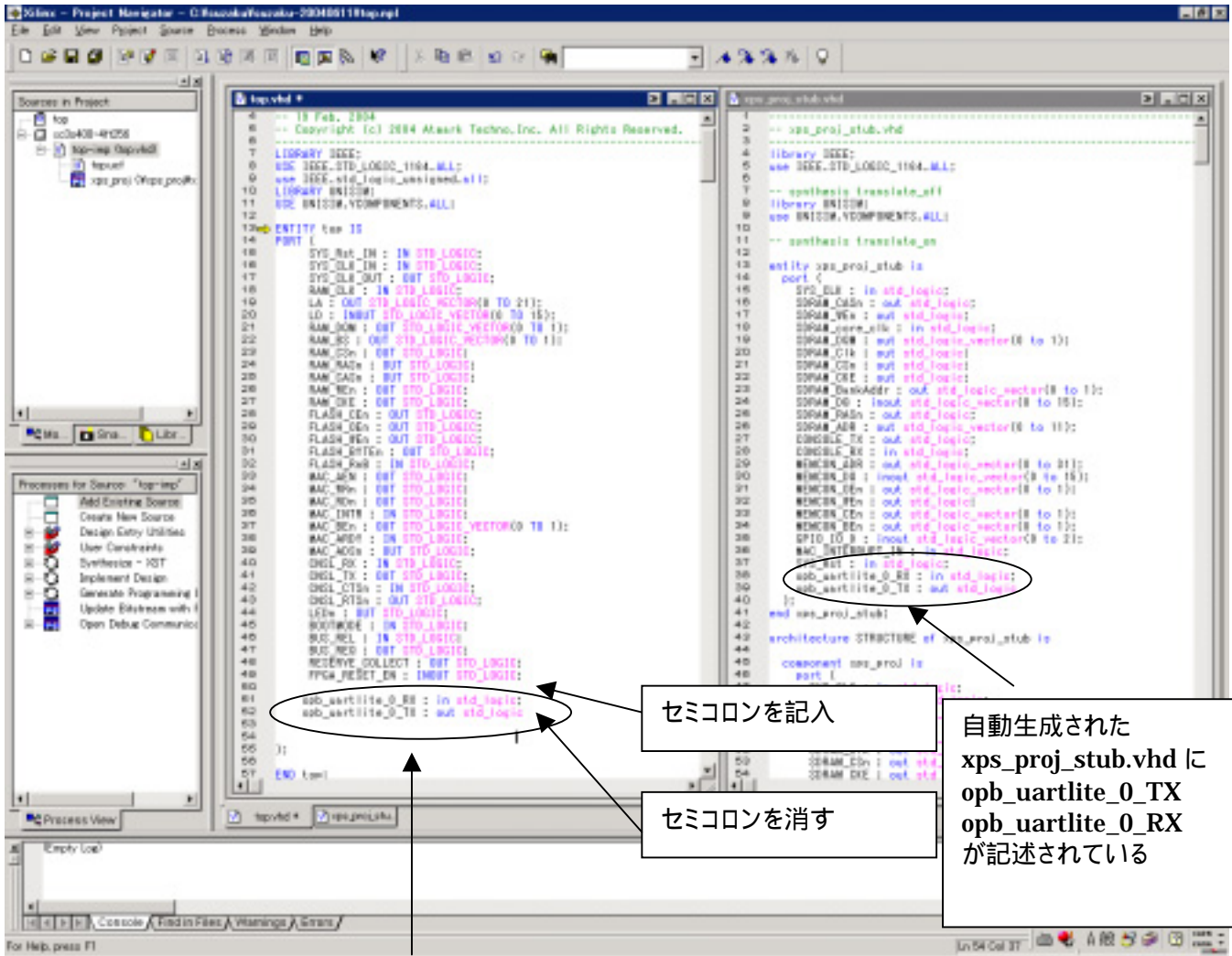
まずは、'Project Navigator' にもどり、'File' メニューの 'Open' で以下の 2 つのファイルを開きます。

`xps_proj\hdl\xps_proj_stub.vhd`
`top.vhd`



'xps_proj_stub.vhd' は、XPS で MicroBlaze と周辺ペリフェラルコアを生成させた時に自動的に作成されるファイルです。

このファイルに記述されている Port 名や Component 名を top.vhd に追記していきます。



top.vhd の ENTITY top IS の port に
 opb_uartlite_0_RX : in std_logic;
 opb_uartlite_0_TX : out std_logic
 を追記

The screenshot shows the Xilinx Project Navigator interface with two VHDL files open: `top.vhd` and `xps_proj_stub.vhd`.

Annotations:

- セミコロンを記入** (Enter semicolon): Points to the end of the `opb_uartlite_0_TX` signal declaration in `top.vhd`.
- セミコロンを消す** (Delete semicolon): Points to the semicolon at the end of the `opb_uartlite_0_RX` signal declaration in `top.vhd`.
- top.vhd の COMPONENT xps_proj に opb_uartlite_0_RX : in std_logic; opb_uartlite_0_TX : out std_logic を追記** (Add opb_uartlite_0_RX and opb_uartlite_0_TX to the COMPONENT xps_proj in top.vhd): Points to the component instantiation section in `top.vhd`.

Code Snippets:

```

-- top.vhd
...
opb_uartlite_0_RX : in std_logic;
opb_uartlite_0_TX : out std_logic;
...
COMPONENT xps_proj
PART
  SYS_Bus : IN STD_LOGIC;
  SYS_CLK : IN STD_LOGIC;
  ...
  opb_uartlite_0_RX : in std_logic;
  opb_uartlite_0_TX : out std_logic;
END COMPONENT;
...

```

```

-- xps_proj_stub.vhd
...
entity xps_proj_stub is
  port (
    SYS_Bus : in std_logic;
    ...
    opb_uartlite_0_RX : in std_logic;
    opb_uartlite_0_TX : out std_logic;
  );
end xps_proj_stub;
...

```


The screenshot shows the Xilinx Project Navigator interface. The main editor displays two VHDL files: `top.vhd` and `xps_proj_stub.vhd`.

Annotations:

- カンマを記入 (Enter comma):** Points to the line `opb_uartlite_0_RX => opb_uartlite_0_RX,` in `top.vhd`, where a comma is being added after the component name.
- カンマを消す (Delete comma):** Points to the line `opb_uartlite_0_TX => opb_uartlite_0_TX` in `top.vhd`, where a comma is being removed.
- top.vhd の xps_proj_i : xps_proj のインスタンス文に opb_uartlite_0_RX => opb_uartlite_0_RX, opb_uartlite_0_TX => opb_uartlite_0_TX を追記** (Add `opb_uartlite_0_RX => opb_uartlite_0_RX,` and `opb_uartlite_0_TX => opb_uartlite_0_TX` to the instance statement in `top.vhd`): A text box explaining the modification to the component instantiation.

Code Snippets:

```

-- top.vhd
xps_proj_i : xps_proj
PART MAP (
  MEMCON_CSn => MEMCON_CSn,
  SYS_Clk => SYS_Clk,
  SOPAR_core_clk => SYS_Clk,
  SOPAR_CS => PAR_CS,
  MEMCON_Bn => MEMCON_Bn,
  SOPAR_Clk => SOPAR_Clk,
  CONSOLE_RX => CONSOLE_RX,
  SYS_Reset => SYS_Reset,
  SOPAR_Bn => PAR_Bn,
  SOPAR_Cn => PAR_Cn,
  MEMCON_Bn => MEMCON_Bn,
  SOPAR_CS => PAR_CS,
  SOPAR_DataAddr => PAR_Bn,
  CONSOLE_TX => CONSOLE_TX,
  MEMCON_An => MEMCON_An,
  MEMCON_Bn => MEMCON_Bn,
  SOPAR_An => SOPAR_An,
  SOPAR_Dn => SOPAR_Dn,
  SOPAR_Dn => SOPAR_Dn,
  MAC_INTERRUPT_IN => MAC_INTERRUPT_IN,
  MEMCON_Bn_0 => MEMCON_Bn_0,
  MEMCON_Bn_1 => MEMCON_Bn_1,
  MEMCON_Bn_2 => MEMCON_Bn_2,
  SPIE_0_0 => SPIE_0_0,
  SPIE_0_1 => SPIE_0_1,
  SPIE_0_2 => SPIE_0_2,
  opb_uartlite_0_RX => opb_uartlite_0_RX,
  opb_uartlite_0_TX => opb_uartlite_0_TX
);

-- Free B01220
CONSOLE_RX <= CONSOLE_RX;
-- Free B01220
LEDn <= '0';

-- Address bus tri state buffer
SUBST_0 : OBUF
PART MAP (
  0 => L0(0),
  1 => L0(1),
  2 => L0(2)
);
SUBST_1 : OBUF

```

```

-- xps_proj_stub.vhd
xps_proj_stub.vhd
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

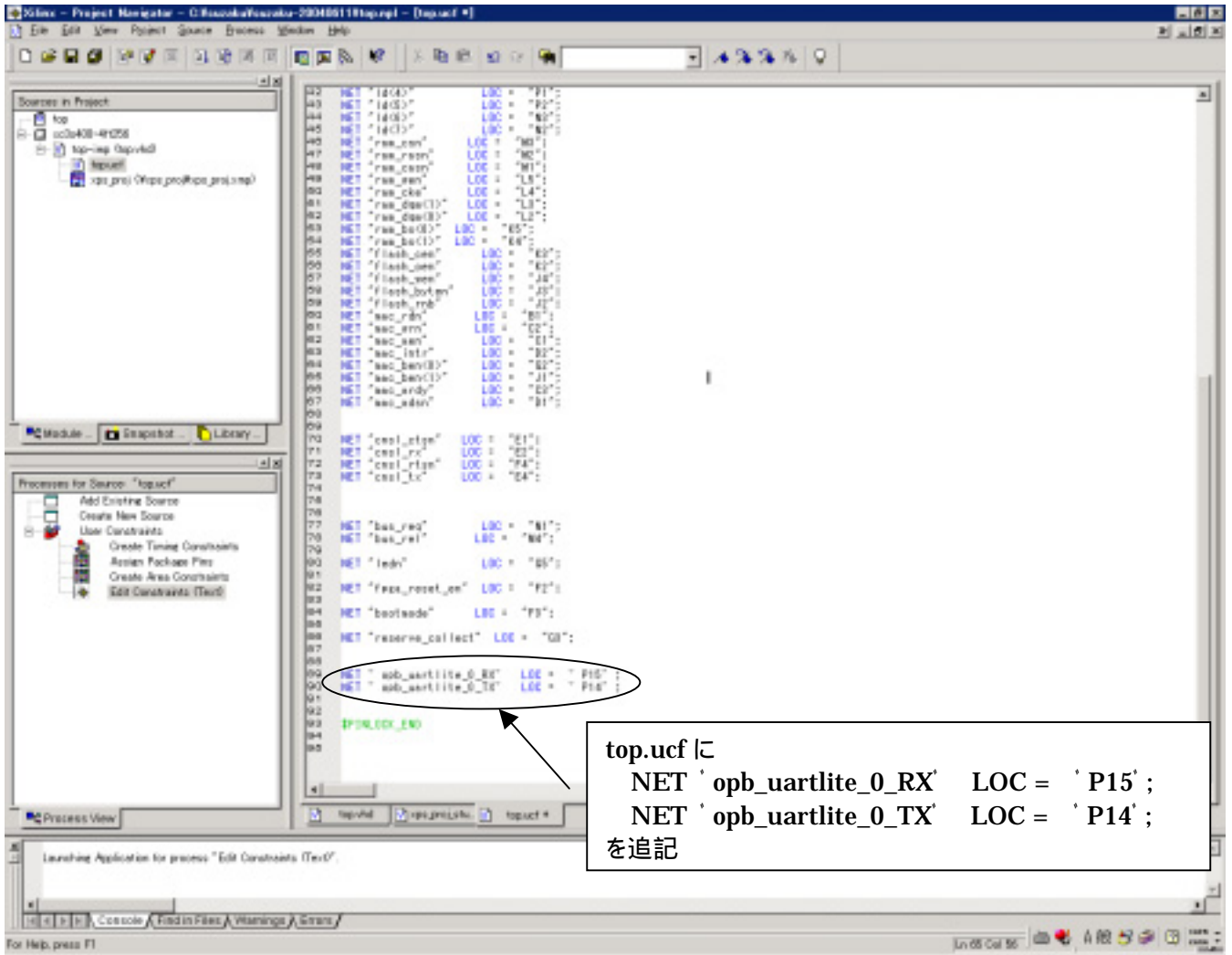
-- synthesis translate_off
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;

-- synthesis translate_on
entity xps_proj_stub is
port (
  SYS_Clk : in std_logic;
  SOPAR_CSn : out std_logic;
  SOPAR_Bn : out std_logic;
  SOPAR_core_clk : in std_logic;
  SOPAR_Cn : out std_logic;
  SOPAR_CS : out std_logic;
  SOPAR_DataAddr : out std_logic_vector(8 to 1);
  SOPAR_Dn : in out std_logic_vector(8 to 1);
  SOPAR_An : out std_logic_vector(8 to 1);
  CONSOLE_TX : out std_logic;
  CONSOLE_RX : in;
  MEMCON_An : out;
  MEMCON_Bn : out;
  MEMCON_Bn_0 : out std_logic_vector(8 to 1);
  MEMCON_Bn_1 : out std_logic_vector(8 to 1);
  MEMCON_Bn_2 : out std_logic_vector(8 to 1);
  SPIE_0_0 : in out std_logic;
  SPIE_0_1 : in out std_logic;
  SPIE_0_2 : in out std_logic;
  opb_uartlite_0_RX : in;
  opb_uartlite_0_TX : out std_logic;
);
end entity;

architecture STRUCTURE of xps_proj_stub
component xps_proj
port (
  SYS_Clk : in std_logic;
  MEMCON_CSn : out std_logic;
  MEMCON_Bn : out std_logic;
  MEMCON_Bn_0 : out std_logic_vector(8 to 1);
  MEMCON_Bn_1 : out std_logic_vector(8 to 1);
  MEMCON_Bn_2 : out std_logic_vector(8 to 1);
  SPIE_0_0 : in out std_logic;
  SPIE_0_1 : in out std_logic;
  SPIE_0_2 : in out std_logic;
  opb_uartlite_0_RX : in;
  opb_uartlite_0_TX : out std_logic;
);
end component

```

¥suzaku_v0¥xps_proj¥top.ucf ファイルを開き、Spartan-3 のフリー I/O ピンに信号を割当てます。
 今回は' CN5 の 3 番ピン(P15)、4 番ピン(P14)' に割当てることとします。
 以上でプロジェクトトップファイルの編集は完了です。

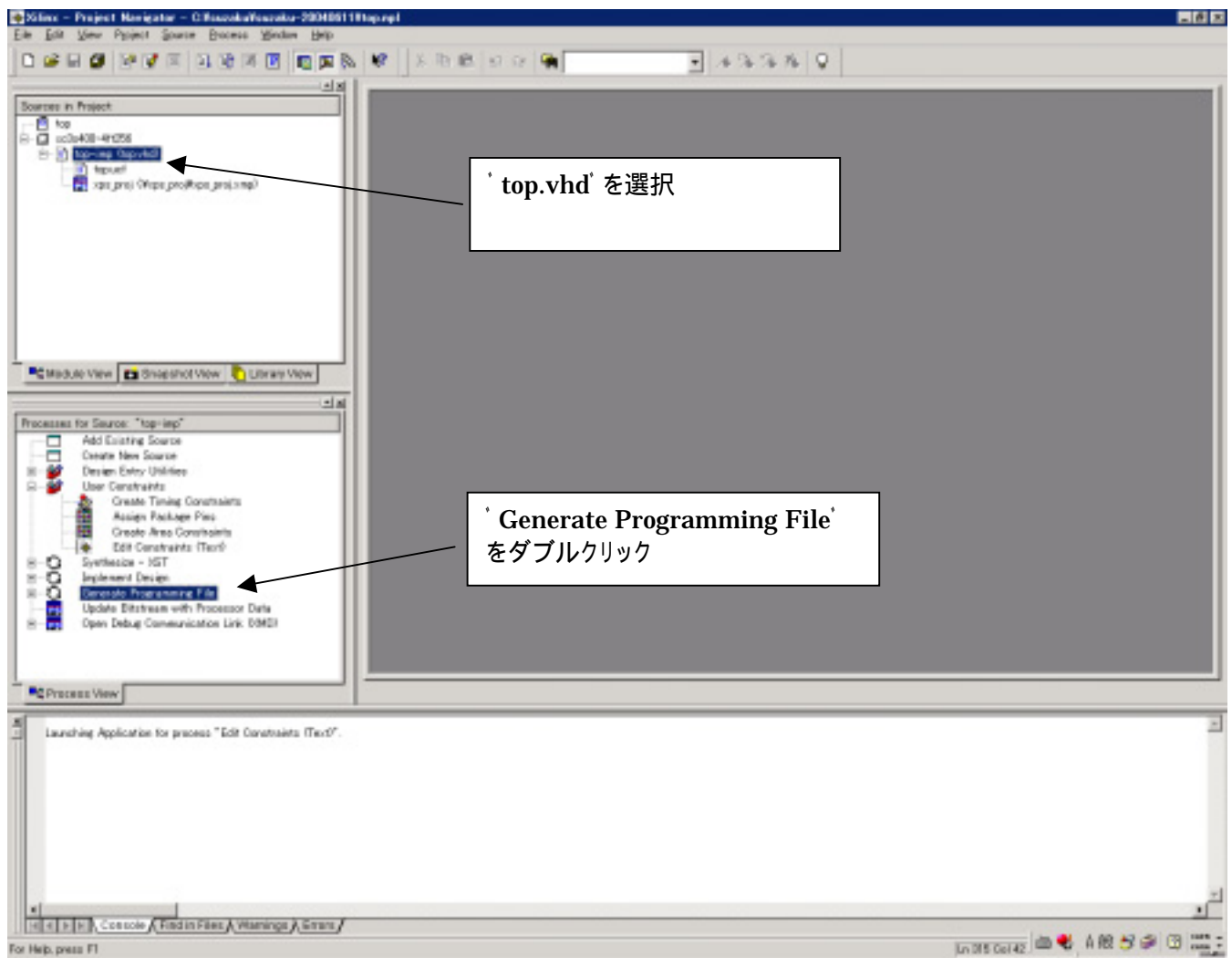


9.6. Platform Studio コンパイルの実行

'Sources in Project' で 'top.vhd' を選択し、'Generate Programming File' をダブルクリックし、コンパイルを実行します。

以上で設定、コンパイルが全て終了し、'top.bit' ファイルが生成されます。

(top.bit は FPGA コンフィグレーション用の元ファイルになります。実際にコンフィグレーションプログラムするためには、top.bit を MCS ファイルに変換する必要があります。次項の 'FPGA コンフィグレーション' を参照ください)

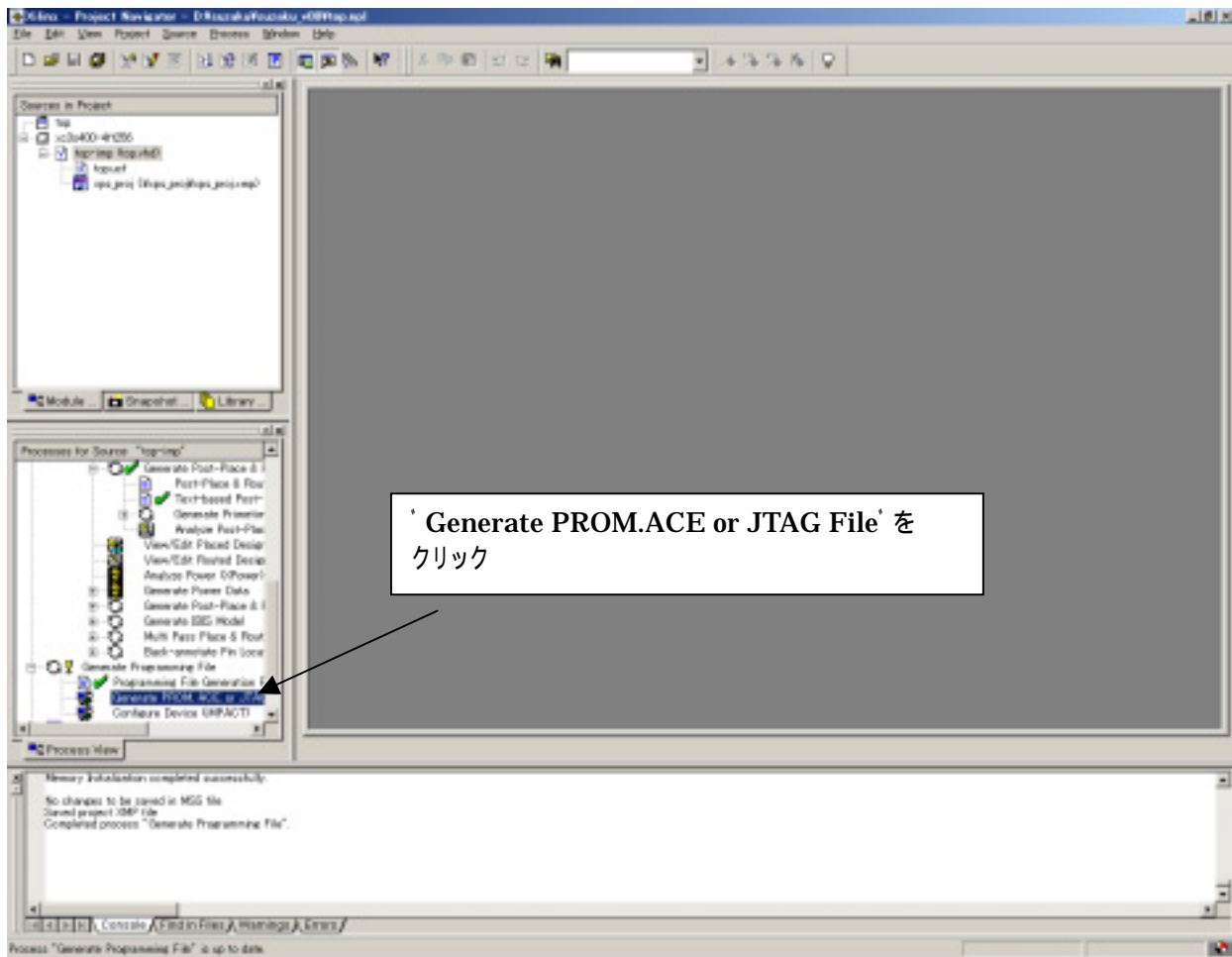


10. FPGA コンフィグレーション

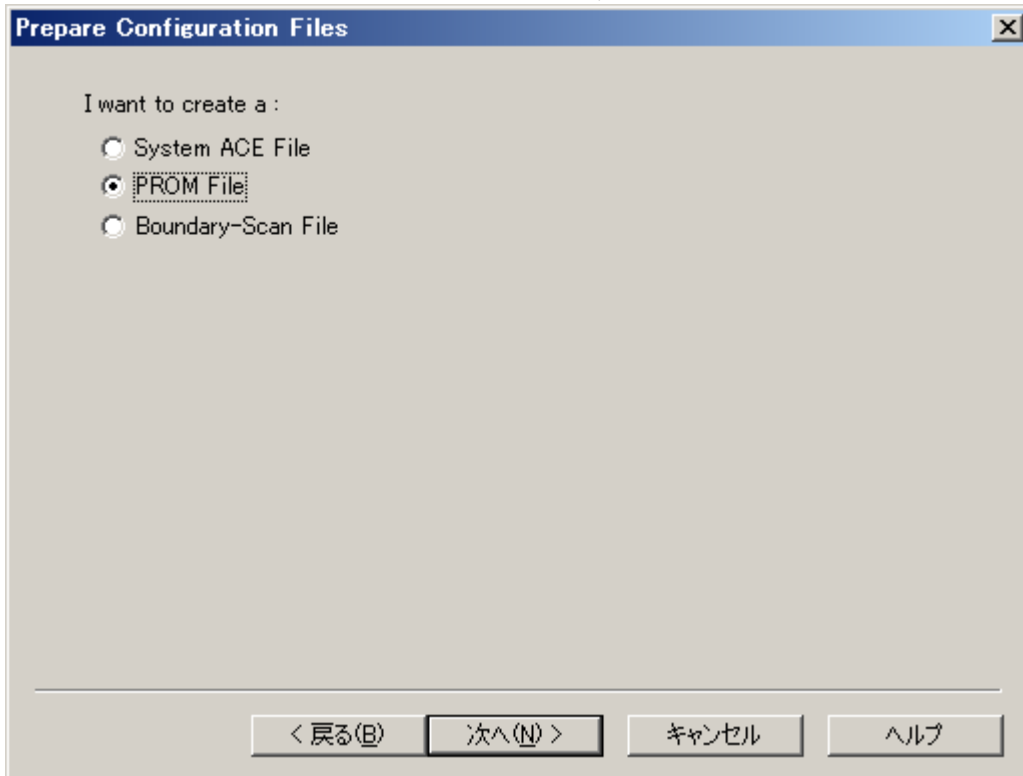
SUZAKU のコンフィグレーション方法について説明します。

10.1. BIT ファイルから MCS ファイルへの変換

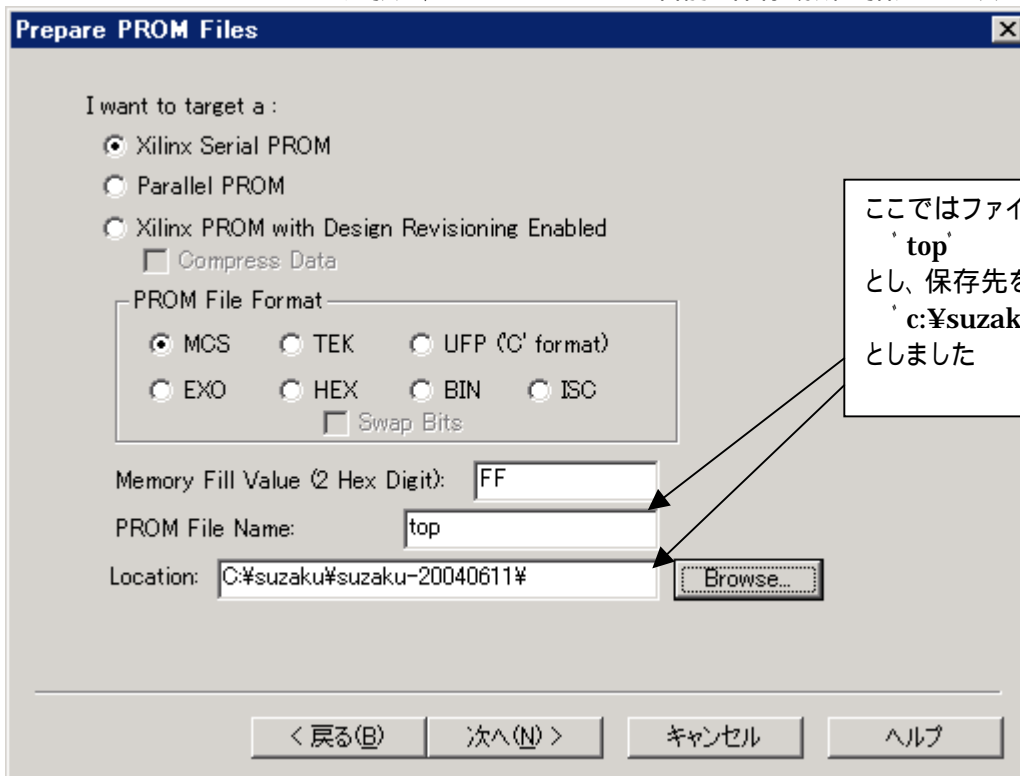
'Generate PROM.ACE or JTAG File' をクリックします。



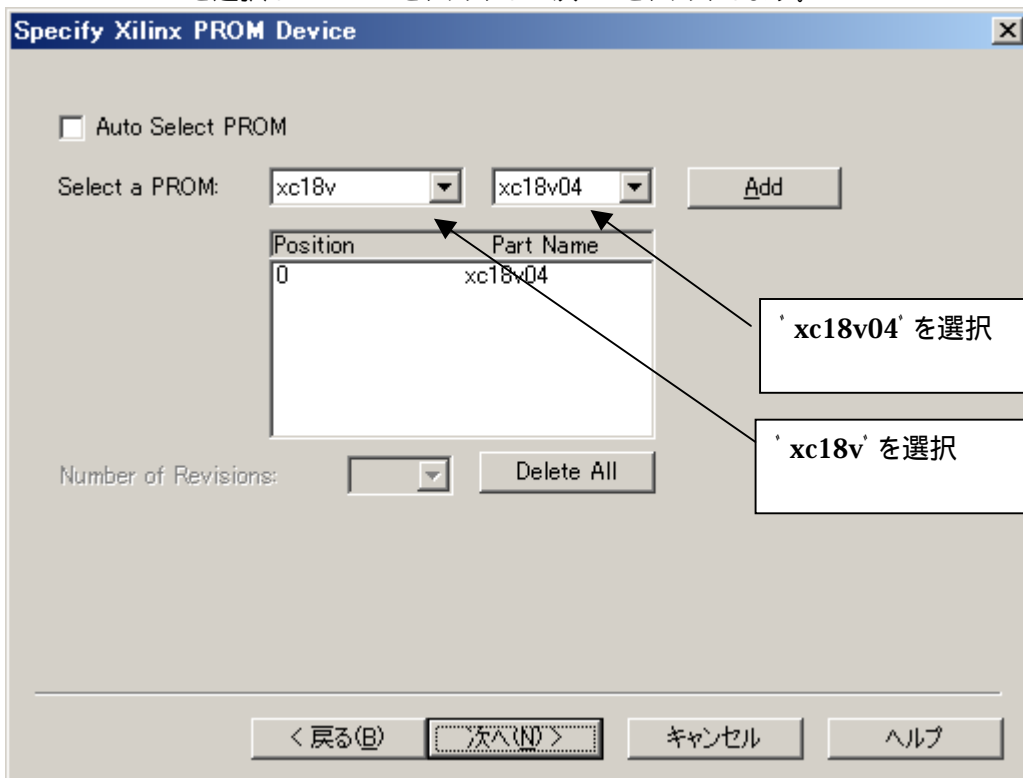
'PROM File' を選択し '次へ' をクリックします。



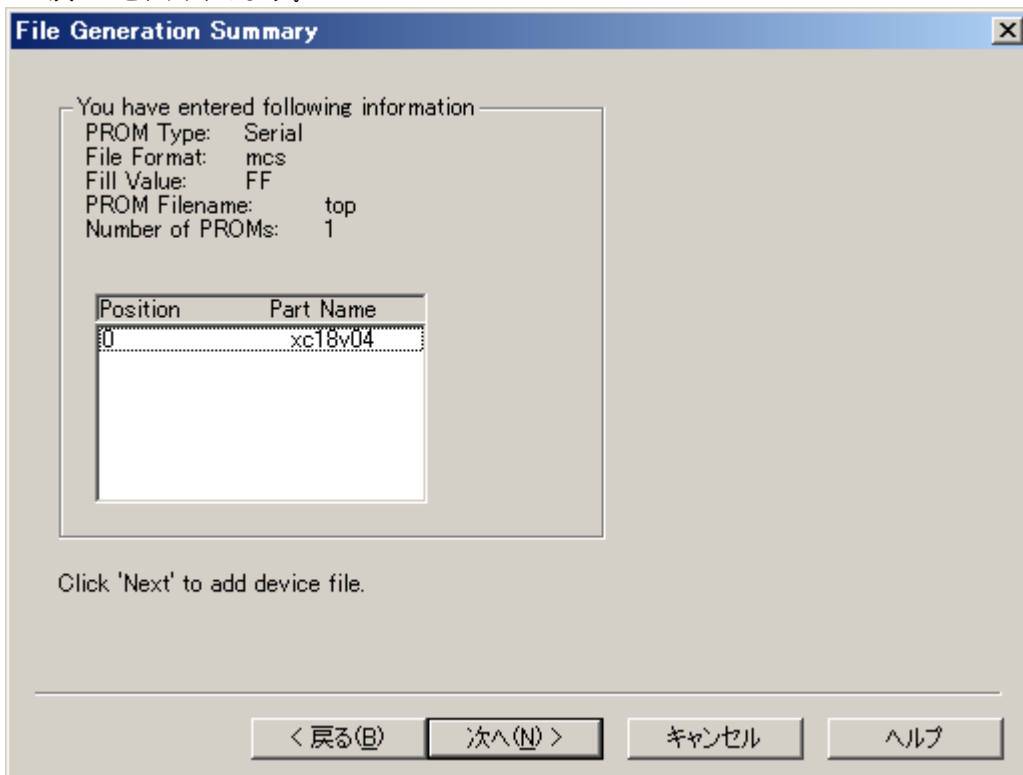
'Xilinx Serial PROM' を選択し、PROM ファイルの名前と保存場所を指定し '次へ' をクリックします。



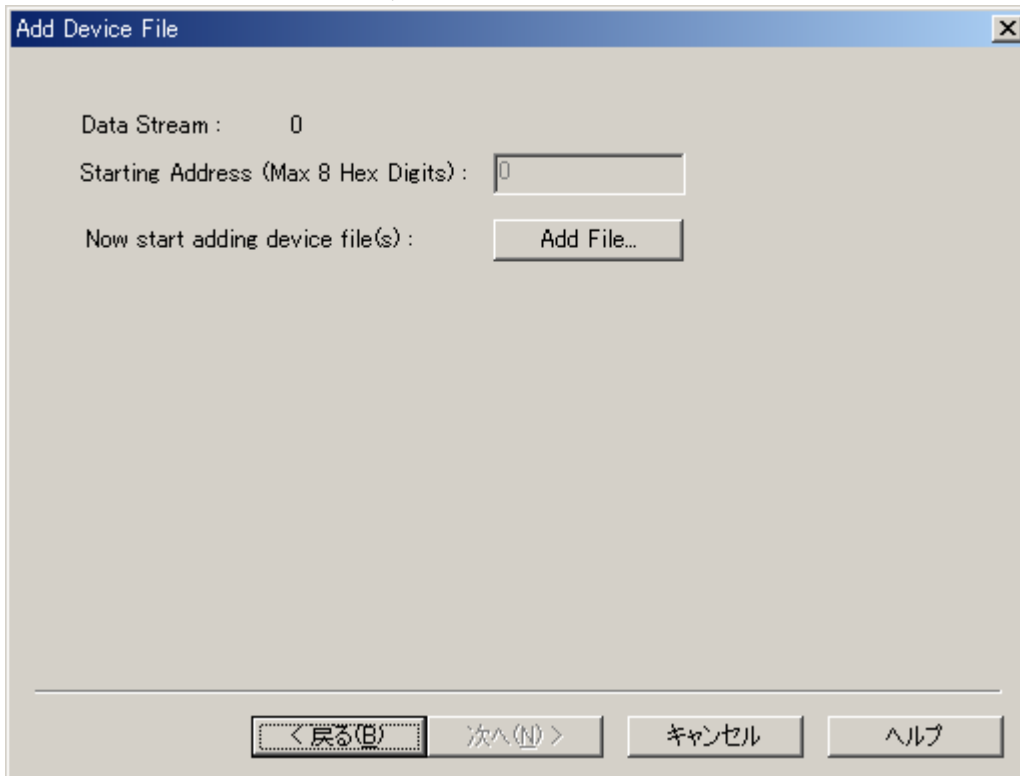
'xc18v 04' を選択し 'Add' をクリックし '次へ' をクリックします。



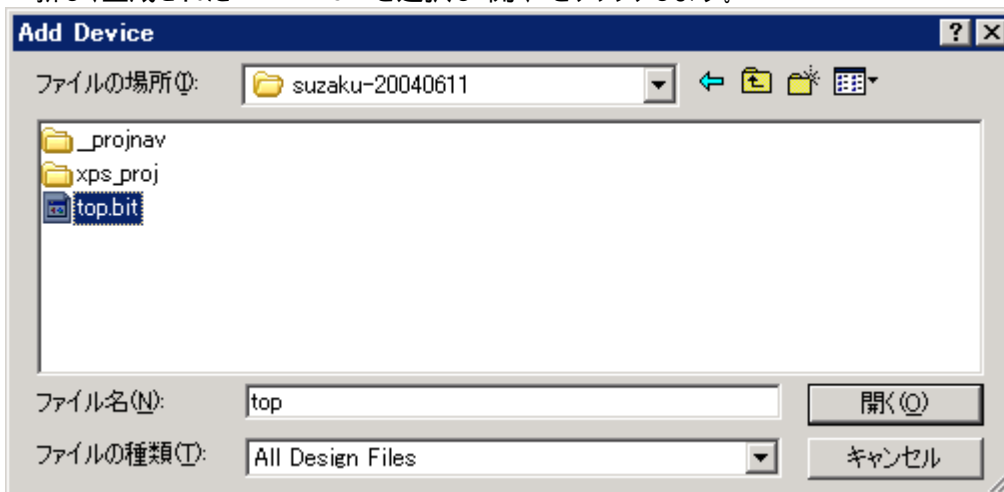
'次へ' をクリックします。



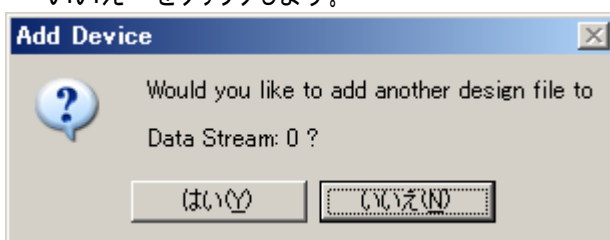
'Add File...' をクリックします。



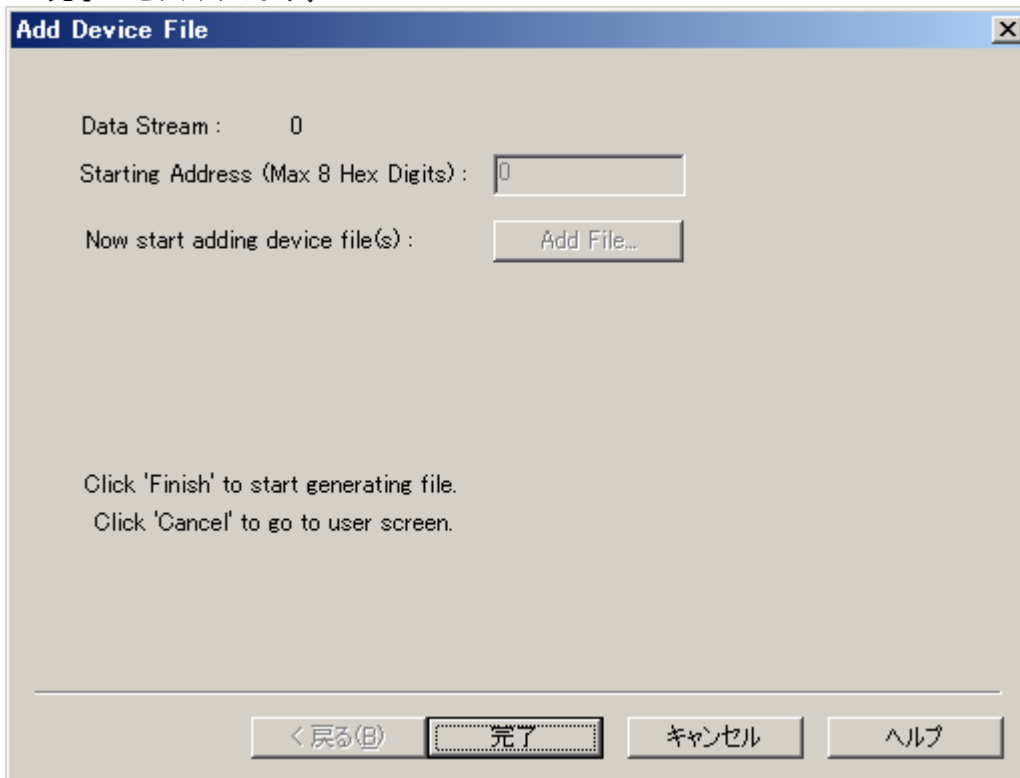
新しく生成された 'BIT file' を選択し '開く' をクリックします。



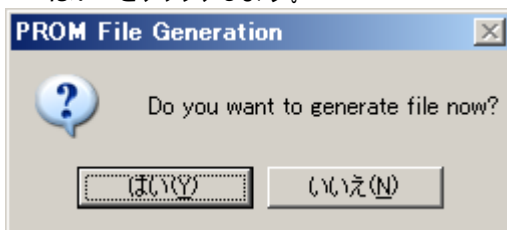
'はいえ' をクリックします。



‘完了’ をクリックします。



‘はい’ をクリックします。



以上で BIT ファイルから MCS ファイルへの変換は終了です。

10.2. SUZAKU へコンフィグレーションデータのプログラム

前項で生成した MCS ファイルを実際に SUZAKU にプログラムします。

SUZAKU では FPGA のコンフィグレーションに東京エレクトロデバイスの TE7720 を使用しています。

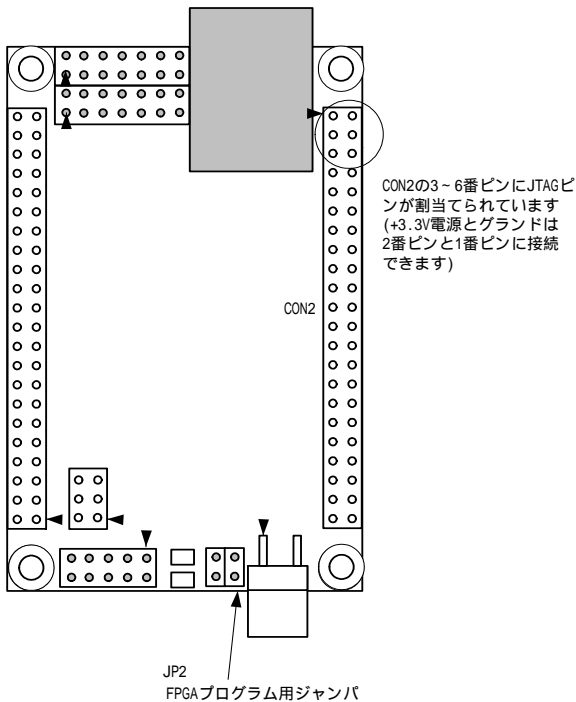
TE7720 の動作については、4.4.9 項 FPGA コンフィグレーションを参照ください。

また、TE7720 について東京エレクトロデバイスのホームページから詳細資料をダウンロードできます

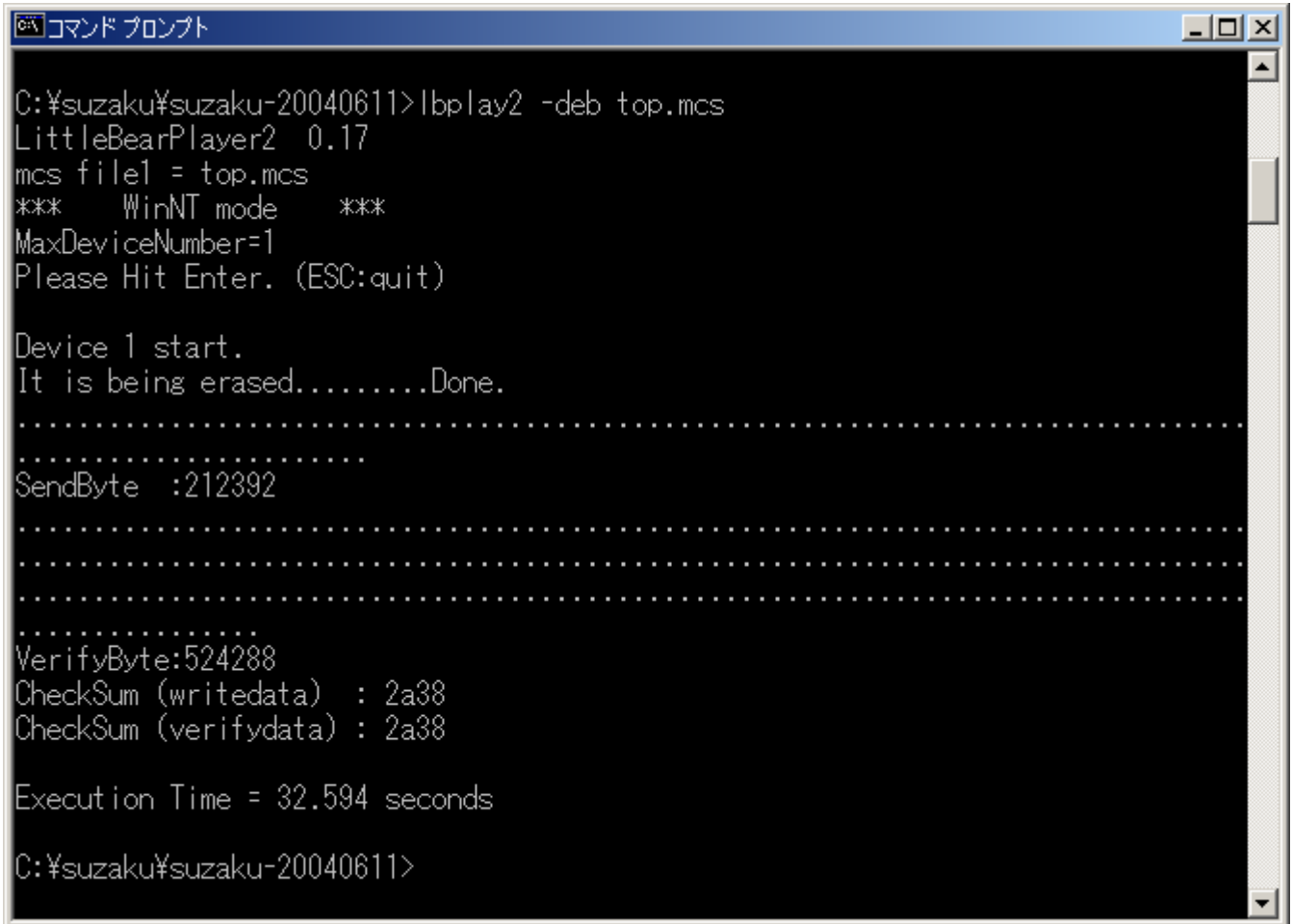
まず、新しく生成された 'MCS file' を 'device.def' と 'lbplay2.exe' と同じフォルダにコピーします。

('device.def' と 'lbplay2.exe' は、プロジェクトトップファイルと同じフォルダ 'szaku-*****' にも入っています。)

- ・SUZAKU の電源がオフとなっていることを確認してください。
- ・CON2 の 'FPGA プログラム用 JTAG' に Xilinx 製 Parallel Cable 等の JTAG ケーブルを接続します。
- ・JP2 ジャンパをショートします。



- ・SUZAKU の電源をオンにします。
- ・コマンドプロンプトを開き、先ほどの MCS file や 'device.def' 、 'lbplay2.exe' があるフォルダに移動します。
- ・ 'lbplay2 -deb top.mcs' とタイプします。
- ・以下のエラーが発生した場合は、CD-ROM の¥fpga_proj ¥lbplay2 driver install.txt に記載してある手順で、ドライバをインストールしてください。
"ERROR: Please check WINNT¥system32¥drivers¥windrvr.sys."



```
コマンド プロンプト
C:\¥suzaku¥suzaku-20040611>lbplay2 -deb top.mcs
LittleBearPlayer2 0.17
mcs file1 = top.mcs
*** WinNT mode ***
MaxDeviceNumber=1
Please Hit Enter. (ESC:quit)

Device 1 start.
It is being erased.....Done.
.....
SendByte :212392
.....
VerifyByte:524288
Checksum (writedata) : 2a38
Checksum (verifydata) : 2a38

Execution Time = 32.594 seconds

C:\¥suzaku¥suzaku-20040611>
```

・プログラム完了後、チェックサム値が合っていることを確認し、SUZAKU の電源をオフにします。
何かの原因でエラーを起こし、チェックサムが違っていた場合は、SUZAKU を動作させないでください。
間違ったデータをプログラムしたり、またプログラム中に何かの原因でエラーを起こした場合は、電源を切断し、必ず JP2 をショートし、再プログラミングを行ってください。

・JP2 ジャンパをオープンにします。

・SUZAKU の電源をオンにします。

・新しいコンフィグレーションデータでプログラムされた SUZAKU が起動します。

尚、この例では、UART を外部 I/O にピンアサインしているため、+3.3V の入出力レベルとなります。このため、パソコン等の RS232C とは直接接続できません。

実際にパソコン等の RS232C と接続する方法についての例を、

SUZAKU Official Site <http://suzaku.atmark-techno.com/> の HowTo ページに記載してありますのでご参照ください。

改訂履歴

Ver.	年月日	改訂内容
1.0	2004/04/29	初版作成
1.0.1	2004/06/04	・5.1 項 SUZAKU メモリマップ 誤記訂 誤 0x00000000 – 0x00000FFF BRAM 正 0x00000000 – 0x00001FFF BRAM ・9.5 項 プロジェクトのトップファイルの編集 誤記訂正 誤 ¥suzaku_v00¥xps_proj¥top.vhd 正 ¥suzaku_v00¥top.vhd
1.0.2	2004/06/11	・CD-ROM の FPGA プロジェクトフォルダ名変更 ・LBPLAY2 エラー発生時のドライバインストールについて追記 ・UART をパソコン等 RS232C への接続について追記
1.0.3	2004/06/16	・ソフトウェアリセットの方法について追記

