## Sundance Multiprocessor Technology Limited Application Note

# Application Note for SMT6058

日本語参考資料

注:

本参考資料内の説明文には、Xilinx 社 EDK のヘルプファイルの 文章を引用しています。



本資料は、XPS のヘルプの内容を抜粋したものに、Sundance のソフトのコンパイルに関して、 追記したものです。 詳細は、XPS のヘルプを参照してください。

1. XPS デザインフロー

次の図は、ISE と XPS を使用した別のデザイン フローを示しています。



### 2. XPS プロジェクト ファイル

Xilinx® EDK のプロジェクト ファイルの種類は、複数あります。 特に記述がない場合、これらのファイ ルはデザイン要件に合わせて編集できます。

2.1 プロジェクト ファイル

### XMP

XMP (Xilinx Microprocessor Project) は、EDK デザインの最上位のプロジェクト ファイルで、プロ ジェクト管理に使用されます。 このファイルは変更しないでください。

### BSB

BSB (Base System Builder) ファイルは Base System Builder の生成するプレイバック ファイル です。このファイルは、前に定義したオプションで新しいプロジェクトを作成する際に使用します。このファイルは編集できません。

### 2.2 入力ファイル

#### BBD

BBD (Black Box Definition) ファイルは、デザインのブラック ボックス部分を表すネットリストの場所 を指定します。

### MDD

MDD (Microprocessor Driver Description) ファイルには、ソフトウェア ドライバをカスタマイズする ための指示子が含まれます。

### MHS

MHS (Microprocessor Hardware Specification) ファイルは、ハードウェア コンポーネントを定義 します。MHS ファイルは、<u>Platform Generator (Platgen) への入力ファイル</u>として使用されます。 MHS ファイルはエンベデッド プロセッサ システムのコンフィギュレーションを定義するファイルで、次の 内容が含まれます。 バス アーキテクチャ

ペリフェラル プロセッサ システム接続 アドレス空間

### MLD

MLD (Microprocessor Library Definition) ファイルには、ソフトウェア ライブラリをカスタマイズし、 OS の BSP (Board Support Packages) を生成するための設定が含まれています。

### MPD

MPD (Microprocessor Peripheral Definition) ファイルは、ペリフェラルのインターフェイスを定義します。 MPD ファイルには、次のような特徴があります。

バス インターフェイスのポートおよびデフォルトの接続を記述します。 パラメータおよびデフォルト値を記述します。 MPD のパラメータは、MHS の同等の設定に書き換えられます。

#### MSS

MSS (Microprocessor Software Specification) ファイルは、<u>Library Generator (Libgen) の</u> 入力ファイルとなります。 MSS ファイルには、**OS、ライブラリ、ドライバの定義**が含まれます。

### PAO

PAO (Peripheral Analyze Order) ファイルは、合成が必要な HDL ファイルと、コンパイルの解析順 を定義します。

### UCF

UCF (User Constraints File) は、FPGA デザインのタイミングおよび配置制約を指定します。

### 2.3 インプリメンテーション ファイル

Bitgen オプション ファイル (etc/bitgen.ut) このファイルには、ビットストリーム生成ツールのオプションが含まれます。

iMPACT コマンド ファイル (etc/download.cmd) このファイルは、iMPACT でビットストリームをダウンロードする際に使用されるスクリプト ファイルです。

### インプリメンテーション オプション ファイル (etc/fast\_runtime.opt)

このファイルには、MAP や PAR のような ISE® インプリメンテーション ツールに対するデザイン イン プリメンテーション用コマンド ライン オプションが含まれます。

#### 2.4 出力ファイル

#### BMM

各ブロック RAM がどのように連続した論理データ空間を構成するかが記述されています。メモリ初期 化データ (通常は実行プログラム)を含めた FPGA のビットストリームをアップデートする場合、 Data2MEM はこのファイルを使用してデータを最適な初期化形式に変換します。 BMM ファイルはテキスト ファイルですが、直接編集はしないでください。

このファイルは、Platform Generator (Platgen) で生成され、Bitstream Generator (Bitgen) で 物理的なロケーション情報を含めてアップデートされます。

### ELF

ELF (Executable and Linkable Format) は、コンピュータでプログラムとして認識可能な標準ファ イル形式です。ほとんどの場合、特定プロセッサのマシン命令をバイナリ表記していますが、インタープ リタを実行する必要がある中間形式も含んでいます。

### MHS

MHS (Microprocessor Hardware Specification) ファイルは、ハードウェア コンポーネントを定義 します。MHS ファイルは、Platform Generator (Platgen) への入力ファイルとして使用されます。 MHS ファイルはエンベデッド プロセッサ システムのコンフィギュレーションを定義するファイルで、次の 内容が含まれます。

バス アーキテクチャ ペリフェラル プロセッサ システム接続 アドレス空間

### MSS

MSS (Microprocessor Software Specification) ファイルは、Library Generator (Libgen)の入力ファイルとなります。 MSS ファイルには、OS、ライブラリ、ドライバの定義が含まれます。

### NGC

論理デザイン データと制約の情報を含むネットリストで、EDIF および NCF (Netlist Constraints File) からの情報がこのファイルに含まれています。

### xplorer\_fast\_runtime.opt

Xplorer を選択して XPS でデザインをインプリメントすると作成されます。 このファイルには、 最適な結果になるツール オプションがコピーされます。 Xplorer を使用しない場合は、 fast\_runtime.opt でこれらのオプションを使用できます。 詳細は、『Platform Format Specification Reference Manual』を参照してください。

### 3. XPS ディレクトリ構造の概要



- 3.1 XPS プロジェクト ディレクトリ詳細
- (1) **\_\_\_xps**

XPS で生成される中間ファイルとさまざまな管理ツールが含まれます。このディレクトリはユーザーは使用しませんが、削除しないでください。

(2) **blkdiagram** このディレクトリには、SVG、PNG、または JPG フォーマット ファイル (UNIX のみ) など、デザインを

ブロック図で表したファイルが含まれています。 (3) **bootloops** PowerPC® のブートループ実行ファイルを含むディレクトリです。このファイルにより、PowerPC プロセ

ッサはリセットベクタでループするようになり、既知の良好な状態に保持されます。 XPS では、各新規プロジェクトに対してこのブートループが自動的に作成されます。 ブートループが使用されるようにするには、XPS の BRAM を初期化する際にブートループ アプリケー ションをオンにします。 <u>この際、複数のアプリケーションが BRAM</u> に同時に初期化されないようにしてく <u>ださい。</u>

(4) **bsp** 

オプションのディレクトリです。 このディレクトリを使用すると、カスタマイズされた BSP (Board Support Package) を含むことができたり、既存の BSP を変更したりできます。

- (5) data 必須のディレクトリです。このディレクトリには、UCF (User Constraints File) が含まれます。
- (6) **drivers** オプションのディレクトリです。このディレクトリには、カスタム ドライバを含めることができます。
- (7) etc
  - このディレクトリには、次のファイルが含まれます。
- bitgen.ut : ISE ビットストリーム (BIT) ファイルを作成するのに使用されるオプションが含まれます。
- download.cmd : JTAG チェーンの仕様です。
- fast\_runtime.opt : NGDBuild、マップ、配置配線を実行するためのコマンドおよびオプションが含まれます。
- xplorer\_fast\_runtime.opt: Xplorer を選択して XPS でデザインをインプリメントすると作成されます。このファ イルには、最適な結果になるツール オプションがコピーされます。 Xplorer を使用しない場合は、fast\_run time.opt でこれらのオプションを使用できます。
- BSDL ファイル: JTAG チェーンに含まれるデバイスの BSDL (Boundary Scan Description Language) フ ァイルすべてが含まれます。

メモ: XPS では、プロジェクトを作成するとこのディレクトリが自動的に作成されます。このディレクトリのファイルの中には、選択したボードに対応するように修正する必要があるものもあります。

(8) hdl

このディレクトリは、[Hardware] → [Generate Netlist] をクリックすると Platgen で自動的に作成 されます。このディレクトリには、プロセッサ システムを記述した HDL ファイルすべてが含まれます。

(9) implementation

このディレクトリは、[Hardware]  $\rightarrow$  [Generate Bitstream] をクリックすると自動的に作成されます。 含まれるのは、UCF ファイル、BMM ファイル、および BIT ファイルを含むインプリメンテーション結果 です。 Xflow か Xplorer のいずれかが内部で実行され、このディレクトリにファイルが生成されます。

(10) **ppc405\_0** 

これらのディレクトリは、[Software]  $\rightarrow$  [Generate Libraries and BSPs] をクリックすると Libgen で自動的に作成されます。このディレクトリには、次のサブディレクトリが含まれます。

- include: ドライバに必要な Cヘッダ ファイルが含まれます。また、Libgen により、xparameters.h という include ファイルもこのディレクトリに作成されます。このファイルは、システムのペリフェラルのベ ース アドレス、ドライバ、OS、ライブラリ、およびユーザー プログラムで必要な #defines、関数のプ ロトタイプを定義します。
- lib: libc.a、libm.a、libxil.a などのライブラリが含まれます。libxil ライブラリには、プロセッサからアクセス可能なドライバ関数が含まれます。
- libsrc: OS、ライブラリ、ドライバのコンパイルに必要な中間ファイルおよび makefile が保存されます。このディレクトリには、ペリフェラルに固有のドライバ ファイル、OS 用の BSP ファイル、ライブラリ ファイルが含まれます。
- code : MicroBlaze  $\mathcal{OH}_{\circ}$

上記のディレクトリ(ppc405\_0 以下)はすべて、<u>LibGen を実行するたびに削除されます</u>。 Libgen は プロセッサ インスタンスごとに、そのプロセッサ インスタンスと同じ名前でディレクトリを作成します。 ソースおよび実行ファイルを保存するには、TestApp などのディレクトリを別に作成する必要がありま す。別に作成するディレクトリには、プロセッサインスタンスと同じ名前を付けないでください。

このディレクトリは、プロセッサ インスタンスと同じ名前にする必要があります。 ディレクトリに付けた 名前がプロセッサのインスタンス名になります。 (11) pcores

オプションのディレクトリです。 プロセッサ システム用にカスタマイズされたハードウェア ペリフェラルを 含めるために使用します。

(12) SDK\_Projects

Platform Studio ソフトウェア開発キット (SDK) を起動すると作成されるディレクトリです。 SDK のソ フトウェア アプリケーションに関する情報、設定、その他の SDK 特有の情報が含まれます。

(13) simulation

XPS では、[Simulation] → [Generate Simulation HDL Files] をクリックするとこのディレクトリが 自動的に作成されます。Simgen により、該当するサブディレクトリに各シミュレーションのシミュレーシ ョン ファイルが作成されます。このファイルには、ペリフェラル、最上位のエンベデッド システム ファイ ル、オプションのテストベンチ ファイルの HDL ラッパが含まれます。シミュレータのコンパイルおよびへ ルパ スクリプトも生成されます。

- (14) sw\_services オプションのディレクトリです。このディレクトリには、カスタム ライブラリを含めることができます。。
- (15) **synthesis** このディレクトリは、[Hardware] → [Generate Netlist] をクリックすると Platgen で自動的に作成

このティレクトリは、[Hardware] → [Generate Netlist] をクリックすると Platgen で自動的に作成 されます。含まれるのは、インプリメンテーションで使用されるネットリストを作成する XST 合成スクリプ トとログ ファイルすべてです。

(16) TestApp\_Memory および TestApp\_Peripheral

BSB (Base System Builder) ウィザードで作成されます。 含まれるのは、基本的なソフトウェア テス ト アプリケーションとリンカ スクリプトです。 これらのディレクトリは、 BSB でユーザーが指定した場合 にのみ作成されます。指定しない場合は作成されません。

### 4. プロジェクト作成

### 4.1 準備

プロジェクトを作成する前に、下記の準備をしておく。

- EDK のパッチを適用する。
   Sundance より提供されている、EDK\_Source\_Patch のパッチを行う。

  - (b)  $lwip_v_3_0_a \notin : C: Xilinx = 10.1 EDK = wathing Party = wathing Services$
- (2) xps\_ll\_temac コア用の評価ライセンス xps\_ll\_soft\_core\_parmanent\_eval.lic を ダウンロードし、C:¥Xilinx¥10.1¥ISE¥coregen¥core\_licences 内にコピーしておく。

<u>http://www.xilinx.com/support/answers/32054.htm</u> を参照 メンバーページへの login が必要

- (3) SMT6058 で提供されている SMT6058¥Hardware¥SMT6058¥FPGA¥FX60¥IPCORE¥ フォルダの下の、 edk\_user\_repository フォルダを、特定なフォルダ(例えばc:ドライブ)の下にコピーする。 (c:¥edk\_user\_repository)
- (4) PPC のプロジェクトを作成するフォルダを新しく作成しておく。
   例えば、c: ¥ new\_PPC



### 4.2 白紙からの XPS プロジェクトの作成

 XPSを起動し、起動時に表示される [Xilinx Platform Studio] ダイアログ ボックスで [Blank XPS project] をオンにします。



(5) ターゲット デバイスの種類、既存のmhsファイルを指定します。

| Create New XPS Project   |   |
|--|---|
| C:/test/New_PPC/system.xmp Browse  |   |
| Target device         Architecture       Device Size       Package       Speed Grade         virtex4       ▼       xc4vfx60       ▼       ff1152       −10       ▼ | SMT6058で提供された既存の MHS ファ<br>イルを指定する                              |
| Advanced options (optional: F1 for help)<br>MHS file to import   | 指定した MHS ファイルがプロジェクト デ<br>プロジェクト ディレクトリにコピーされます。                |
| system.mhs     Browse       Set Project Peripheral Repositories       Browse   | これは、プロジェクト ディレクトリ以外のディ<br>レクトリの MHS ファイルを直接修正するこ<br>とはできないからです。 |
| OK Cancel  |   |
| 既存の MHS ファイルを含めない新規のプロジェ<br>XPS プロセッサ アーキテクチャおよびコアに精通し   |   |

ターゲットの FPGA ファミリおよびデバイスを指定します。XPS でツールを実行する前に、ターゲット ア ーキテクチャを指定する必要があります。ただし、デバイス サイズ、パッケージ、スピード グレードなど は、[Generate Bitstream] でデザインをインプリメントするまで使用されません。 これは、後で [Project Options] ダイアログ ボックスで変更できます。

SMT148 では、 Virtex4, xcvf60 ff1152 -10 を指定する。

[Project Peripheral Repository] は指定しない。

(6) [OK] をクリックする

|   | A FFP                                   | Bus Interfaces Ports Addresses   |                         | Stell Filter |
|---|---|--|-------------------------|--------------|
| roject Applications IP Catalog                    |   | Name Bus Connection  | IP Type IP Version      |              |
| latform   | P + + + + + + + + + + + + + + + + + + + | ⊕  | ppo406_virtex4 = 2.01.a |              |
| Project Files                                     |   | fch_v10_0  | fch_v10 1.00.a          |              |
| MHS File; system.mhs                              |   | → optrx_fst_0_ta_ppo405_0  | fsl_v20 2.11.a          |              |
| -MSS File: system.mss                             |   |  | fel_v20 211 a           |              |
| UOF File: data/system.ucf                         |   |  | fst_v20 Z.11.a          |              |
| -IMPACT Command File: etc/download.cmd            |   | → ppc405_0_ta_cptrx_ts[_1  | fel_v20 211 a           |              |
| Implementation Options File: etc/fast_runtime.opt |   |  | pt0_040 1.08.a          |              |
| Parts at Oatland                                  |   | He sps_bram_if_cnfir_1   | xps_bram_it_cntir1.00.a | -            |
| - Device: verteful@Off11E2=10                     |   | Generic_External_Memory  | xps_mch_emc 2.00.a      |              |
| -Device: xc4vixcom152-10                          |   | +- > ph bram # cnts 1 bram   | bram_block 100a         |              |
| -Implementation: VDC (Vflaur)                     |   | epux_is_u  | opunctisi 1.00.a        |              |
|   |   | in child him a   | California 1000         |              |
| Sim Madal: REHAVIODAL                             |   | the state of the sector of the | house only 2.01 a       |              |
| Pafarance Files                                   | 4 3                                     | in- pagpar char o  | pagpin chine 2 01 c     |              |
| i-log Files                                       | Legend                                  | h  |                         |              |
| Synthesis Report Files                            | Haster O Slave                          | Master/Slave Target Connected  | Unconnected             |              |
|   | [Pintform Studio]                       | System Assembly View Dirick Discom   |                         |              |
|   |   | Direct Diagram   |                         |              |
|   |   |  |                         |              |
|   |   |  |                         |              |
|   |   |  |                         |              |
|   |   |  |                         |              |
|   |   |  |                         |              |
|   |   |  |                         |              |
|   |   |  |                         |              |
|   |   |  |                         |              |
|   |   |  |                         |              |

(7) 起動の途中で、下記 Error が表示される。



フォルダの参照より、4-1(3)で設定した、edk\_user\_repository フォルダを選択する。

(8) 正常に、プロジェクトの作成が終了したら、EDK を、クローズする。

(9) sundance の SMT6058 フォルダ内の **data¥system.ucf**、etc¥download.cmd を、 新しく作成したプロジェクト内の同じフォルダ位置にコピーする。

参考:download.cmd setMode setCable -p auto identify assignfile -p l -file implementation/download.bit program -p l quit

(10) system.xmp を、テキストエディタで開き、ModuleSearchPath の値の、
 (フォルダ)/edk\_user\_repository/MyProcessor/・・・・・
 の、MyProcessor 以下を削除し、(フォルダ)/edk\_user\_repository/ として system.xmp を保存する。

### 5. ハードウェア プラットフォームのインプリメンテーション

### 5.1 デザイン制約(UCF)を指定する。

デザインをダウンロードおよびインプリメントする BIT ファイルを生成するには、まずユーザー制約ファイル (UCF)を設定する必要があります。

UCF のベース ファイル名は、MHS (Microprocessor Hardware Specification) と同じで、プロジェクト ディレクトリの data フォルダ内にあります。

Sundance の SMT6058 で提供されたプロジェクト内の ucf ファイルを 新規作成したプロジェクト内の data¥system.ucf へ上書きコピーして使用します。

### 5.2 Bbitgen.ut の修正



- 5.3 ハードウェア プラットフォームすべてを記述したビットストリーム (BIT) ファイルを生成する。 ハードウェア プラットフォームを生成するには、次の手順に従います。
  - (1) ネットリストの生成

[Hardware] → [Generate Netlist] をクリックします。 これでプラットフォーム構築ツールである Platgen が起動され、次が実行されます。

- デザインのプラットフォーム コンフィギュレーション、MHS (Microprocessor Hardware Specification) ファイルを読み込みます。
- VHDL を生成します。
- XST を実行します。

ネットリスト ファイルをザイリンクスの NGC 形式で生成します。

最初から、次の[Generate Bitstream]を行ってもいい。

(2) ビットストリームの生成

XPS でハードウェア デザインを含む BIT ファイルを作成 (ビットストリームを生成) します。

[Hardware] → [Generate Bitstream] をクリックします。必要であれば、Platgen が起動され、

- (1) ネットリストが生成されます。
- 次に ISE® インプリメンテーション ツールが実行され、UCF ファイルが読み込まれて、 ハードウェア デザインを含む BIT ファイルが生成されます。

アプリケーションソフトウェア(elf)のパターンは含まれません

■エンベデッド ソフトウェア プラットフォームでは、各プロセッサに対し、ハードウェア プラットフォーム (ボード サポート パッケージ) に含まれるペリフェラルのドライバ、ライブラリ、標準入力/出力デバイス、割り込みハンドラ ルーチン、その他のソフトウェア機能が定義されます。

さらに XPS プロジェクトでは、各プロセッサ上で実行されるソフトウェア アプリケーションも定義されます。 これらのソフトウェア アプリケーションは、ソフトウェア プラットフォームに基づいています。

ハードウェアにプロセッサとペリフェラルを含めたら、アドレス マップが生成されます。 これで、ソフトウェア プラットフォームを設定したり、XPS を使用して BSP (ボード サポート パッケージ) を生成することができます。この後、アプリケーションースから参照できるプラットフォーム別ヘッダ ファイ ルが生成できます。

■ソフトウェア プラットフォームは、MSS (Microprocessor Software Specification) ファイルで定義さ <u>れます。</u>このテキスト形式の MSS ファイルと、ソフトウェア アプリケーションが、エンベデッド システムの ソフトウェア コンポーネントを表す主要なソース ファイルです。

EDK と共にインストールされるライブラリとドライバ、およびカスタム ペリフェラル用のライブラリとドライバ を使用して、XPS により MSS で定義されるソフトウェア コンポーネントを含むソフトウェア アプリケーショ ンが、プロセッサ ハードウェア プラットフォームで実行できる ELF (Executable and Linked Format) ファイルとしてコンパイルされます。

ELF 実行ファイルは、オンチップかオフチップ メモリ、またはその両方に含めることができます。

■オンチップの ELF ファイルは、FPGA のコンフィギュレーション用の ビットストリームにマージして、オンチ ップの BRAM ブロックを初期化できます。

オフチップの ELF ファイルは、ユーティリティ (SMT148FXの場合は、Sundance のSMT6002)を 使用してフラッシュのような不揮発性メモリ デバイスに格納することもできます。

■プロトタイプの作成中(デバッグ時)には、FPGA に接続されている JTAG ケーブルを介して、ELFファイル をダイナミックにダウンロードできます。

また、XPS のソフトウェア デバッグ機能を使用すると、同じ JTAG ケーブルに接続された FPGA のアプリケーションを制御および監視できます。

■ XPS には、アプリケーションを開発しコンパイルする単純なソース コードの開発環境が含まれます。

■Xilinx SDK

EDK にはソフトウェア アプリケーションの開発用に、XPS の補助プログラムである Platform Studio ソ フトウェア開発キット (SDK) が含まれています。このプログラムは Eclipse™ オープン ソース標準に基 づいており、高度な C/C++ コード エディタおよびコンパイル環境を提供します。

SDK では、ソフトウェア プラットフォームのコンフィギュレーション、アプリケーション プロジェクトの作成と 管理、フラッシュ プログラミング、オンチップ BRAM 初期化、フラッシュ メモリのプログラム、ブートローダ 一の作成、アプリケーションのデバッグとプロファイルなど、XPS で提供されるソフトウェア機能がすべてサ ポートされます。 7. ソフトウェア プラットフォームの生成

SMT6058 で提供されている、system.mss をプロジェクトのフォルダへコピーすることで、下記(7.1 ~ 7.3)の操作は不要。

- 7.1 ソフトウェア プラットフォームの概要 アプリケーションを構築するソフトウェア ドライバおよび OS をまとめたもの。 ハードウェア プラットフォームの流動性、およびザイリンクスおよびサードパーティ パートナ ーのサポートにより、ハードウェア プラットフォームに対してさまざまなソフトウェア プラッ トフォームを作成できます。
- 7.2 ソフトウェア プラットフォームの生成

[Software] → [Software Platform Settings] をクリックして表示されるダイアログ ボ ックスから、ペリフェラルのドライバおよびライブラリのパラメータ、プロセッサのパラメータ、 標準 I/O デバイス、割り込みハンドラ ルーチン、その他のソフトウェア機能が指定できます。

XPS は、ユーザーの指定に基づいてソフトウェア プラットフォームを生成します。 このソフトウェア プラットフォームを使用して、エンベデッド アプリケーション ソフトウェア コードを開発します。

| 😔 Software Pl                                    | atform Settings  |   |  |   | ×  | ] |            |              |
|--|--|---|--|---|--|---|------------|--------------|
| Processor Informati<br>Processor Instance        | ion<br>: ppc405_0 💌  |   |  |   |  |   | ppc405_0   | を選択          |
| Software Platform<br>OS and Libraries<br>Drivers | Processor Settings<br>CPU Driver: cpu_ppc405   | CPU Driver Version:   | 1.10.b 💌   |   |  |   | ppc400_0   |              |
|  | Name<br>ExtRa_COMPILER_FLAG<br>ARCHIVER<br>-COMPILER<br>-COMPILER<br>-CORE_CLOCK_FREQ_HZ             | Current Value<br>S -g<br>powerpc-eabi-ar<br>powerpc-eabi-gcc<br>30000000                                | Default Value<br>-g<br>powerpc-eabi-ar<br>powerpc-eabi-gcc<br>400000000  | Type<br>string<br>string<br>string<br>int           | Description<br>Extra compiler flags used in BSP and libra<br>Archiver used to archive libraries for bott<br>Compiler used to compile both BSP librar<br>Core Clock Frequency in Hz |   |            |              |
|  | OS & Library Settings  |   |  | _   |  |   | xulkernel  | を選択          |
|  | OS: xilkernel ▼ Version<br>Use Library Version<br>▼ xilmfs 1.00a<br>T xilfash 1.01a<br>xilfatfs 2000 | Xilinx Memory File<br>Xilinx Memory File<br>Xilinx In-system a<br>Xilinx Flash III<br>Provides read/wri | kernel software platfor<br>reads, scheduling, syno<br>e System<br>and Serial Flash Libran<br>For Intel/AMD CFI co<br>te routines to access | rm. Provides<br>shronization<br>mpliant<br>files st | POSIX like kernel services such as , time etc. Requires an XPS timer on a v  | - | xilmfs、lwi | <b>p</b> を選択 |
|  | Milp 3.00.a  | WIP TCP/IP Stac   | sk libnary V3.00.a<br>sk libnary: IwIP v1.3.0, X   | (Ilinx ac   | OK Cancel Help   |   |            |              |

ソフトウェア プラットフォームを指定すると、MSS (Microprocessor Software Specification) ファイルが自動的に生成されます。 このファイルには、定義したソフトウェア プラットフォームに関する情報が含まれます。

また、ソフトウェア アプリケーションは XPS の代わりに SDK でも開発できます。 SDK を使 用したソフトウェア開発フローおよびソフトウェア プラットフォームの生成の詳 細は、「Platform Studio ソフトウェア開発キット (SDK)の使用」を参照してください。

MSS ファイルの詳細は、『Platform Specification Format Reference Manual』の「Microprocessor Software Specification (MSS)」を参照してください。 <u>MSS ファイルは、Library Generator (Libgen)の入力として、ドライバ、ライブラリ、および</u> 割り込みハンドラをカスタマイズするために使用されます。 詳細は、『エンベデッド システム ツール リファレンス マニュアル』の「Library Generator (Libgen)」を参照してください。

### 7.3 ソフトウェア ライブラリの生成

EDK ソフトウェア ライブラリには、生成されたライブラリとボード サポート パッケージ (BSP) が含まれます。

● EDK ソフトウェア ライブラリおよび BSP を生成するには、 [Software] → [Generate Libraries and BSPs] をクリックします。

ソフトウェア ライブラリを生成するツール、Libgen が実行されます。

●プロジェクトのソフトウェア ライブラリは、次のプロジェクト ディレクトリに作成されます。 <Project Directory>¥<Processor Instance>¥lib¥libxil.a

●システムのアドレス マップは、次のヘッダ ファイルに作成されます。 <Project Directory>¥<Processor Instance>¥include¥xparameters.h

この段階までに作成された、ハードウェア、ライブラリ等のデータは、プロジェクトフォルダ内 のデータを新しい別のフォルダにコピーすることで、再利用できる。 コピーすることで、各ハードウェアプロジェクトごとに、ハードウェアの生成、ライブラリを再度 生成する必要は無い。

### 8. ソフトウェア アプリケーションのサポート

XPS では、デザインに含まれるシステムのソフトウェア アプリケーションを管理できます。 ソフトウェア アプリケーションの情報は、プロジェクト設定の一部として XPS に取り込まれます。

通常のソフトウェア アプリケーション フローには、次が含まれます。

- (1) システムのプロセッサ インスタンスに対するソフトウェア アプリケーションの新規作成
- (2) アプリケーションのコンパイルおよび ELF (Executable and Linkable Format) ファイルの作成
- (3) ELF 情報を使用したシステム ブロック RAM の初期化 (ELF が完全にブロック RAM に含まれ る場合)
- (4) ビットストリームのダウンロード
- (5) アプリケーションのデバッグ
- 8.1 アプリケーションの作成

XPS では、[Applications] タブで複数のアプリケーション プロジェクトを指定できます。 各アプリケーションは、そのアプリケーションを実行するプロセッサ インスタンスに関連付けます。 各アプリケーション プロジェクトには、固有の名前を付ける必要があります。

XPS では、ソフトウェア フローの設定に Library Generator (Libgen) を使用します。 Libgen は、エンベデッド プロセッサ デザイン用にライブラリ、デバイス ドライバ、ファイル システム、お よび割り込みハンドラを設定します。

**ソフトウェア アプリケーション**のソース コードは、C や C++ のような高級言語、またはアセンブリ言語 で記述します。

8.2 アプリケーションのコンパイル

作成したソース コード ファイルをコンパイルし、これらのファイルをリンクして ELF フォーマットの実行ファイルを生成します。

デフォルトでは PowerPC® 用の GNU コンパイラが使用されますが、ハードウェア プラットフォームで 使用しているプロセッサをサポートするコンパイラであればどれでも使用できます。

8.3 アプリケーションのデバッグ

ソフトウェア アプリケーションのデバッグには、XMD (Xilinx Microprocessor Debugger) と GNU デバッガ (GDB) を使用します。

XMD には、命令セット シミュレータが含まれています。

また、動作中のハードウェア プラットフォームに接続し、GDB でユーザー アプリケーションを実行する ことも可能です。

### 9. XPS を使用したソフトウェア アプリケーション プロジェクトの新規作成

ソフトウェア アプリケーション プロジェクトを新規作成する方法は 2 つあります。 <u>・XPS を使用したソフトウェア アプリケーション プロジェクトの新規作成</u> ソフトウェア アプリケーション プロジェクトを XPS で直接作成します。

•Platform Studio ソフトウェア開発キット (SDK)の使用

ソフトウェア アプリケーション プロジェクトを Platform Studio ソフトウェア開発キット(SDK)で作成 します。

9.1 XPS を使用したソフトウェア アプリケーション プロジェクトの新規作成

SMT6058 で提供された、プロジェクト内の、app フォルダを、新規プロジェクトのフォルダに コピーしておく。同じプロジェクト内に作成した方が、後日フォルダを他の PC にコピーしたりす る場合フォルダのパスが保存されているので、扱いやすい。

ソフトウェア アプリケーション プロジェクトを新規作成するには

(1) [Project Information Area] ウィンドウの [Applications] タブをクリックします。
 このタブには、ソフトウェア プロジェクトとその関連プロセッサ、実行ファイル、コンパイラ オプション、
 ソース ファイル、およびヘッダ ファイルがツリー ビュー形式で表示されます。



(2) [Add Software Application Project] をダブルクリックし、システム内のプロセッサに対して新しいソフトウェア プロジェクトを作成します。

[Add Software Application Project] ダイアログ ボックスが表示されます。



- (3) プロジェクトの名前を入力する。Application Project名で、フォルダが作成される。
- (4) プロセッサを選択します。

(5) [OK] をクリックします。 ツリー ビューには、新規プロジェクトが表示されます。

| Xilinx Platform Studio - C:¥test¥SMT6058¥system.xmp - [<br>Eile Edit View Project Hardware Software Device Configuration Debug Simulation<br>Help          Image: Software Project       Image: Software Application       Image: Software Appl | Constant of the second se |
|---|---|
| Output Warning Error  | Pk Syst Blo   |

### 9.2 既存のソース ファイルの追加

既存のソース ファイルは、プロジェクト ディレクトリまたはリモート ディレクトリに含めることができます。

- (1) 既存のソース ファイルをプロジェクトに追加するには、次の手順に従います。
  - 1. メイン ウィンドウで [Applications] タブをクリックします。
  - 2. ソース ファイルを追加するプロジェクトの [Sources] を右クリックします。
  - 3. [Add Existing File] をクリックします。ファイルを指定するダイアログ ボックスが表示されます。
  - 4. 追加するソース ファイルを検索し、選択します.



5. ソース ファイルがプロジェクトに追加されます。

注意 : ソースをプロジェクトに追加してもファイルがコピーされる訳ではなく、プロジェクト データにパスと ファイル名が追加されるだけです。

C ソース ファイルに加えて、ヘッダ ファイルもプロジェクトに追加できます。 この場合は、XPS メイン ウィンドウの[Applications] タブで [Headers] を右クリックし、ソース ファイルと同様の操作を行います。

各 Example のスレッドで使用するファイル一覧 (複数のスレッドの選択が可能)

|   | ファイル名             | APP_WEBSERVER | APP_ETH2CP | APP_TFTPSERVER | APP_ECHOSERVER |
|---|-------------------|---------------|------------|----------------|----------------|
|   | main.c            |               |            |                |                |
| С | webserver.c       |               |            |                |                |
|   | et2cp.c           |               | •          |                |                |
|   | tftpserver.c      |               |            |                |                |
|   | echo.c            |               |            |                |                |
|   | webutils.c        |               |            |                |                |
|   | tftputils.c       |               |            |                |                |
|   | AccessFlashFpga.c |               |            |                |                |
|   | FlashFpgaLib.c    |               |            |                |                |
|   | http_response.c   |               |            |                |                |
|   | platform_fs.c     |               |            |                |                |
|   | platform_gpio.c   |               |            |                |                |
|   | prot_malloc.c     |               |            |                |                |
|   | AccessFlashFpga.h |               |            |                |                |
| h | flash_baseaddr.h  |               |            |                |                |
|   | FlashFpgaLib.h    |               |            |                |                |
|   | platform_fs.h     |               |            |                |                |
|   | platform_gpio.h   |               |            |                |                |
|   | prot_malloc.h     |               |            |                |                |
|   | tftpserver.h      |               |            |                |                |
|   | tftputils.h       |               |            |                |                |
|   | webserver.h       |               |            |                |                |
|   | Stack size        | 0x2000        | 0x2000     | 0x2000         | 0x2000         |
|   | Heap size         | 0x2000        | 0x2000     | 0x2000         | 0x2000         |

### 9.3 コンパイラの環境変数の設定

コンパイラの環境変数を設定するには、次の手順に従います。 (1) XPS メイン ウィンドウで [Applications] タブをクリックします。

| 😵 Xilinx Platform Studio - C:¥test¥SMT6058¥system.xmp - [Sys                     |              |
|--|--------------|
| Eile Edit View Project Hardware Software Device Configuration Debug Simulation W | (indow       |
| Help   | _ 8 ×        |
| ] 🗅 🏓 🗒 🕹 ] 🗗 🗑 🗗 ] 🕫 🖉 着 🛱 🖓 🔒 🛱 🖄 🔒 🖄 👘 📥 🔌 🤶 🗌 🎄                              |              |
|  |              |
| ×  | F            |
| Project Applications IP Catalog  | S<br>L       |
| Software Projects  |              |
| Cadd Software Application Project  |              |
| Project: Webserver   |              |
| -Processor: ppc405_0   |              |
| Executable: C:¥test¥SMT6058¥Webserver¥executable.elf                             |              |
| E-Compiler Options<br>   |              |
| Mode: executable   |              |
| -Sources   | ▶++++        |
| <sup>t</sup> Headers   | <u> </u>     |
|  |              |
|  | -legend      |
|  | 🗆 Master 🔾   |
|  | [PlaSyst Blo |
| x  |              |
| Output Warning Error   |              |
|  |              |

(2) ツリー ビューでソフトウェア プロジェクトの階層を展開表示して
 [Compiler Options] をダブルクリックし、[Compiler Options] ダイアログボックスを開きます。

(3) [Environment] タブをクリックします。



(7) [OK] をクリックすると、コンパイラが設定され、system.xmp に設定が保存されます。

### 9.4 コンパイラの検索パスの設定

コンパイラの最適化レベルを"Medium(-O2)に設定した後、そのコンパイラの検索パスを設定します。 コンパイラの検索パスを設定するには

(1) [Compiler Options] ダイアログ ボックスで [Paths and Options] タブをクリックします。
 [Search Paths] フィールドで、次のディレクトリを設定できます。

| Compiler Options     Compiler Tools: powerpc-eabi-gcc     Environment Debug and Optimization Paths and Options     All paths should be relative to project directory. Separate multiple | (2) <b>[Library (-L)]</b> にはコンパイラ<br>で 使用されるライブラリ検索パス<br>のリストに追加するディレクトリを<br>入力します。                       |
|---|---|
| options with a space.  Search Paths  Librarv (-L)  Browse Browse Browse   | (3) <b>[Include (-l)]</b> には、コンパイラ<br>で 使用されるインクルード検索<br>パスのリストに追加するディレクト<br>リを入力します。                     |
| Libraries to Link against (-I)<br> wip4 xilkernel<br>Other Compiler Options to Append<br>These options will be appended to the compiler command line                                    | (4) [Libraries to Link against (-I)]<br>には、未定義のシンボルに対し、<br>リンカによって検索されるライブラ<br>リのリストに追加するライブラリを<br>入力します。 |
|   | lwip4、xilkernel を設定する。  |
| OK Cancel Help  | (6) [OK] をクリックすると、コンパ<br>イラが設定されます。<br>結果は、system.xmp に保存され<br>ます。  |

Application Note for SMT6058

### 9.5 アプリケーション実行ファイルの生成

プログラム サイズを取得した後は、アプリケーションの実行ファイルを生成します。

9.5.1 アプリケーション実行ファイルの生成

[Applications] タブでプロジェクト名を右クリックして、[Build Project] をクリックするとBuilが実行される。

- 注: [Build project] では、make.exe が呼び出される。 XPS では、make.exe の version3.79 を使用しており、他のソフト等で、それより新しい Version が使用されていて、そちらの make が呼び出されてしまうと、フォルダ構造が正しく認 識されず、make で"No rule to make target" のエラーが発生する。 これを避けるために
  - (1) c:\Xilinx\10.1\EDK\cygwin\bin の path を PATH のトップに設定する。
  - (フォルダ名はデフォルト時の例、 実際は EDK をインストールしたフォルダを使用)
  - (2) Windows¥system32内に、make.exe がある場合、名前を変更するか、EDK で提供されて
  - いる make.exe をコピーしておく。

**executable.elf** が、Application Project フォルダ内に生成される。

### 9.6 リンカ スクリプトの生成

生成された ELF ファイルに基づいたリンカスクリプトファイルが生成できる。 プログラムセクションの修正等に使用できる。

 XPS のメイン ウィンドウの [Applications] タブで、プロジェクト名を右クリックして、 [Generate Linker Script] をクリックします。

アプリケーション用の ELF ファイルがある場合は、[Generate Linker Script] ダイアログ ボックスの [Sections View] と [Heap and Stack View] に ELF ファイルの設定が表示されます。

[ELF file used to populate section information] は、ELF ファイルの場所を示します。

- 例 : Webserver 用の場合設定、9.3 項で設定した、デフォルトの Heap : 0x2000、 Stack : 0x2000 が表示されている。
- (2) [Sections View] では、セクションをメモリにマップできます。 各セクションの、Memory 配置位置を設定できます。
- (3) [Heap and Stack View] では、ヒープとスタックのサイズ、およびメモリを指定できます。

[Memories View] では、システムに含まれるメモリのリストと共に、その開始アドレスおよび サイズが示されます。 このセクションは、変更できません。

[Boot and Vector Sections View] には、アプリケーションを実行するプロセッサのブート セクションおよびベクタ セクションが表示されます。これらも変更できません。

(4) [Output Linker Script] には、リンカ スクリプトのファイル名を入力できます。 このスクリプトは、アプリ ケーションのコンパイラ設定に自動的に追加されます。

| 📀 Genera     | ate Linker S   | cript           |               |                  |                   |                  |           | X      |
|--------------|----------------|-----------------|---------------|------------------|-------------------|------------------|-----------|--------|
| Sections Vie | w:             |                 |               | Heap and Stack   | View:             |                  |           |        |
| Section      | Size (bytes)   | Memory          |               | Section          | Size (bytes)      | Memory           |           |        |
| vectors      | 0x00000000     | Generic_Exter   |               | Heap             | 0x20000           | Generic_Exte 💌   | 1         |        |
| .text        | 0x00000000     | Generic_Exter   |               | Stack            | 0x20000           | Generic_Exter    | ]         |        |
| .rodata      | 0x00000000     | Generic_Exter   |               |                  |                   |                  |           |        |
| .rodata1     | 0x00000000     | Generic_Exter   |               |                  |                   |                  |           |        |
| .sdata2      | 0x00000000     | Generic_Exter   |               | 1                |                   |                  |           |        |
| .sbss2       | 0x00000000     | Generic_Exter   |               |                  |                   |                  |           |        |
| .data        | 0x00000000     | Generic_Exte    |               | Memories View:   | :                 |                  |           |        |
| .data1       | 0x00000000     | Generic_Exter   |               | Memory           | Start Address     | Length           |           |        |
| .fixup       | 0x00000000     | Generic_Exte    |               | Generic_Extern   | 0x00000000        | 2048K            | 0         |        |
| .sdata       | 0x00000000     | Generic_Exter   |               | xps_bram_if_cnt  | 0xFFFF0000        | 64K              |           |        |
| .sbss        | 0x00000000     | Generic_Exte    |               |                  |                   |                  |           |        |
| .bss         | 0x00000000     | Generic_Exter   |               |                  |                   |                  |           |        |
|              |                | Add Section D   | elete Section | ELF file used to | ) populate sectio | on information:  |           |        |
| Boot and Ve  | ctor Sections: |                 |               | C:¥test¥SMT60    | )58¥Webserver¥    | executable.elf   |           |        |
| Section      | Address        | Memory          |               |                  |                   |                  |           |        |
| .boot0       | 0xFFFFFFEC     | xps_bram_if_cnt |               | Output Linker S  | Script: Webserve  | er/Webserver_lir | nker_scri | ipt.ld |
| .boot        | 0xFFFFFFFC     | xps_bram_if_cnt |               |                  |                   |                  |           |        |
|              |                |                 |               |                  |                   | OK Car           | ncel      | Help   |

- (5) [Generate] をクリックします。
- (6) リンカ スクリプト ファイルが既に存在する場合、上書きするかどうかを確認するメッセージが表示され ます。 上書きする場合は [OK] を、上書きしない場合は [Cancel] をクリックします。
- (7) エラーが発生しなかった場合は、ダイアログ ボックスで [OK] をクリックします。
- (8) ソフトウェアプロジェクト用フォルダ内に、ファイル名が 〈プロジェクト名〉\_linker\_sclipt.ld の リンカスクリプトファイルが生成されます。

### 10. ソフトウェア実行のための、ビットストリームの初期化

XPS の [Generate Bitstream] コマンドでは、ハードウェアのビットストリームのみが生成されています。

このビットストリームは、アプリケーション ソフトウェア コンポーネント(Elfファイル)をビットストリームに含めるまでFPGA には適用できません。

アプリケーション ソフトウェアの開発が完了したら、次のいずれかの方法で、ハードウェアでの実行を 行います。

- (1) ELFファイルを、ハードウェア ビットストリームにマージして実行 サイズの小さな、アプリケーションプログラム(実行ファイルがオンチップ メモリ内に実装可能)は ELFファイルを、ハードウェア ビットストリームにマージできます。 この場合、ELF ファイルはオンチップ メモリに読み込まれ、FPGA がコンフィギュレーションされるたびに実行されます。
- (2) JTAGで実行 (ブートループを使用) デバッグ時に、FPGA に接続されている JTAG ケーブルを介して、実行ファイルをボードにダイナミ ックにダウンロードできます。この場合、オンチップ メモリを初期化するためにブートループをビット ストリームにマージして、ソフトウェアのダウンロードが完了するまでプロセッサをスタティックの状態 にします。
- (3) フラッシュより実行 (ブートローダーを使用) 製品システムの場合は、オフチップメモリ内にある実行ファイルをフラッシュ PROM などの不揮発性 のメモリ デバイスに格納します。この場合、ブートローダーをハードウェア ビットストリームにマー ジして、オンチップ メモリを初期化します。これで、FPGA がコンフィギュレーションまたはリセットされるたびに、ブートローダーによりアプリケーション実行ファイル(Elf)が適切な不揮発性メモリ デバ イスにコピーされ、実行されます。

FPGAのビットストリームに対し、使用する初期化実行ファイル(オンチップ アプリケーション、ブートループ、 またはブートローダー)を指定し、ビットストリームをアップデートします。

ビットストリームのアップデートを行うことで、指定の初期化実行ファイルがハードウェア ビットストリーム にマージされ、FPGA に適用するための、download.bitが生成されます。

### 10.1 ブートループを使用したビットストリームの初期化 (JTAG使用時)

FPGA がビットストリームを使用してコンフィギュレーションされると、プロセッサがリセット状態から解除され、実行を開始します。

システムがソフトウェア アプリケーションを使用して初期化されない場合、初期化されていないメモリから 無効なコードが実行されて、ソフト リセットでは解除できない状態になることがあります。このため、プロセ ッサはシステムが完全に初期化できるようになるまで良好な状態にしておく必要があります。

<u>ブートループは、実際にアプリケーションがダウンロードされて実行されるまで、プロセッサを定義された</u> <u>状態に保つソフトウェア アプリケーションです。</u>この単純な分岐命令を含むアプリケーションは、プロセッ サのブート ロケーションにあります。 XPS には定義済みのブートループ アプリケーションが含まれてい ます。

ブートループを使用する場合、ソフトウェア アプリケーションは通常通りに作成します。 使用するリンカ スクリプトは、ブートループを使用しないときのスクリプトと同じです。 つまり、ソフトウェア アプリケーショ ンにはプロセッサのブート ロケーションに命令を含める必要があります。

ソフトウェア アプリケーションは、システム ブロック RAM (BRAM)の初期化には使用しないでください。

ブートループを使用してビットストリームを初期化するには、次の手順に従います。

- 1. [Applications] タブのツリー ビューからプロジェクト名 (ユーザー ソフトウェア アプリケーション) を右クリックし、[Mark to Initialize BRAMs] が選択されていないことを確認してください。 選択され ている場合は、それをオフにしてください。
- 2. デフォルトのブートループ プロジェクト (<processor name>\_bootloop) を右クリックし、[Mark to Ini tialize BRAMs] をオンにします。
- 3. XPSメイン ウィンドウで [Device Configuration] → [Update Bitstream] をクリックすると、ビット ストリームがブートループをマージしてアップデートされたdownload.bitがimplementationフォルダ内に 生成されます。

### 10.2 製品用ビットストリームの初期化(ブートローダー使用時)

ハードウェアとソフトウェアのデバッグが終了したら、システムを製品(SMT148)に設定する必要があります。

このためには、ブートローダーを使用する必要があります。ブートローダーは、ハードウェア ペリフェラ ルを初期化し、ソフトウェア アプリケーションをメモリにコピーまたは初期化し、実行します。

SMT6058では、サンプルのブートローダーのソフトを提供しています。

プロジェクトの SMT6058\_Bootloader をビットストリームにマージします。

SMT6058\_Bootloader のみ、[Mark to Initialize BRAMs] をオンにして、 [Update Bitstream] を行う。

FPGAコンフィギュレーションのビットデータ及びブートローダーがマージされた download.bit ファイルが、implementation フォルダ内に生成されます。

Bootloader を新しくコンパイルする場合は、リンカースクリプトファイルで、 各メモリーセクションの指定を、xps\_bram\_if\_cntlr\_1 に設定する。

### 11. FPGA コンフィギュレーションの概要

エンベデッド プロセッサ システムを起動するには、<u>システムのハードウェアと、ソフトウェア の両方のコ</u> <u>ンポーネントを、FPGA とプログラム メモリ</u>それぞれにダウンロードする必要があります。

■ デバッグ時

デバッグ時には、JTAG ケーブルを、ホスト コンピュータから開発ボードの JTAG ポートに接続し、ハードウェア ビットストリームおよび ELF ファイル イメージをダウンロードできます。

XPS で [Device Configuration] → [Download Bitstream] をクリックし、ビットストリーム を使用して FPGA をプログラムします。

ソフトウェアをダウンロードする場合は、FPGA の内部ブロック RAM に収まるサイズであれば、ソフトウェアをビットストリームにマージして初期化します。 または XMD 、GNU デバッガ、SDK などのデバッグ ツールを使用してプログラムをボードにダウンロードします。

■ 製品段階(フラッシュメモリ使用) 製品段階では、コンフィギュレーション ROM (PROM)を使用し、ハードウェア ビットストリームで FPGA をコンフィギュレーションします。

### 11.1 FPGA へのビットストリームのダウンロード (JTAG使用)

XPS で iMPACT が自動的に実行され、ビットストリーム(download.bit)がダウンロードされます。 SMT148とJTAG用パラレル ケーブルを接続しておく必要があります。

(1) 選択したデザインをダウンロードし、実行するには [Device Configuration] → [Download Bitstream] をクリックすると、バッチモードの iMPACT を使用してビットストリーム (download.bit ファイル) をターゲット ボードにダウンロードで きます。この処理には、etc ディレクトリにある download.cmd ファイルが使用されます。

このコマンドによって makefile が呼び出され、コマンド ラインから iMPACT が実行され、生成され たビットストリームがターゲット ボードにダウンロードされます。

ビットストリームをダウンロードすると、プロセッサがリセットされ、コードの実行が開始されます。

参考:download.cmd setMode setCable -p auto identify assignfile -p l -file implementation/download.bit program -p l quit

#### 12. 外部メモリからのアプリケーションの実行

データ ファイルまたはソフトウェア アプリケーション (ELF ファイル)を使用して外部メモリを初期化す るには、次の方法があります。

- (1) XMD を使用したデザインのダウンロード
- (2) ブートローダーを使用したデザインのダウンロード
- 12.1 XMD を使用したデザインのダウンロード FPGA は、11.1 の手順でコンフィギュレーションされている必要があります。

XPS から XMD を起動するには、次の手順に従います。

- (1) XPS のXPSメイン ウィンドウで [Debug] → [Launch XMD] をクリックし、XMD を起動し ます。
- (2) プロセッサに接続し、ソフトウェア アプリケーションをダウンロードします。 ソフトウェア実行ファイルをダウンロードするには、dow <path to executable file> を使用します。

例:

(1) 実行ファイルのダウンロード dow Webserver/executable.elf

(2) データファイルのダウンロード dow -data image.mfs 0x130000

なお、一連のコマンドは、xmd.ini として、定義しておくことで、XMDの起動時に 自動的に実行されます。

xmd.ini 例

dow Webserver/executable.elf ----- elf ファイルを指定 run

### 13. 製品用デバイスのコンフィギュレーションの概要 (SMT6058マニュアル参照)

製品段階でエンベデッド プロセッサ システムをコンフィギュレーションする場合、デザインのハードウェア /ソフトウェア コンポーネントの両方が設定されている必要があります。

ハードウェア ビットストリームには、標準の FPGA コンフィギュレーション手法を使用できます。 ソフトウェア アプリケーションの製品設定を外部から実行する必要がある場合、ブートローダーを使用し て RAM にコピーします。

- 13.1 フラッシュ メモリのプログラムの概要
  - フラッシュには、次のものをプログラムできます。
  - アプリケーションの実行可能イメージ
  - FPGA のハードウェア ビットストリーム
  - サンプル データやアルゴリズム テーブルなどのファイル システム イメージおよびデータ ファイル

アプリケーションの実行可能イメージが最も一般的にプログラムされます。デザインのプロセッサのリ セットが完了すると、ブロック RAM のプロセッサのリセット位置に保存されている実行コードが開始し ます。通常ブロック RAM は、ソフトウェア アプリケーションのイメージ全体を保存するには小さすぎ るので(数 KB)、フラッシュ メモリ(単位 MB)を使用します。ブートローダーをブロック RAM に 収まるサイズで作成し、リセット時にフラッシュ メモリからソフトウェア アプリケーションのイメージを読 み込んで容量の大きい外部メモリにコピーし、ソフトウェア アプリケーションに制御を移行して続行する ようにします。

プロジェクトで作成したソフトウェア アプリケーションは、ELF (Executable and Linked Format) フォーマットです。ELF イメージをブートロードするとブートローダーの複雑性が増すので、通常は ELF イメージを保存してブートロードすることはありません。 通常は、ELF イメージを SREC (Motorola S-record) などのブートロード可能なイメージ フォーマットに変換します。このようにす ると、ブートローダーが単純になり、サイズも小さくなります。

SREC ファイルの生成は下記手順によります。

(1) [Device Configuration] - [Program Flash Memory] を選択する。

| Program Flash Memory      K   | SREC ファイルを生成するアプリケ<br>ーションの executable.elf を選択する。                         |
|---|---|
| ✓ Auto-convert file to bootloadable SREC ▼ format when programming flash  |   |
| Processor Instance: ppc405_0<br>Flash Memory Properties<br>Instance Name: Generic_External_Memory_c_mem0_baseaddr                 | — SREC への自動変換を選択する。   |
| Base Address: 0x00000000 Size: 2 Mbytes Bus Width: 16 bits<br>Program at Offset: 0x00000000                                       | xps_bram_if_cntlr_1 を選択   |
| Scratch Memory Properties<br>Instance Name: xps_bram_if_cntlr_1<br>Base Address: 0xFFFFC000 Size: 16 Kbytes                       | Bootloader アプリケーションの生成<br>を選択し、ファイル名を bootloader 0                        |
| Create Flash Bootloader Application<br>SW Application Project: [bootloader_0<br>Bootloader File Format: SREC                      | とする。<br>実際には、Sundance 提供のブートロ<br>ーダーを使用するので、生成された、<br>bootloader 0は使用しない。 |
| Note<br>FPGA must be pre-programmed with a bitstream from an EDK design containing an<br>EMC peripheral connected to Flash Memory |   |
| OK Cancel Help  |   |

- (2) OK をクリックすることで、executable.elf.srec というファイルが、プロジェクトの ルートフォルダの下に生成される。
- (3) SMT148FX のSW3,SW4を次のように設定し、電源を投入する。



- (4) PC で、SMT6002 の、"Flash Utility for FPGA module"を起動する。
- (5) フラッシュメモリー内のメモリーマップは下記のようになっている。

| Position | アドレス       | 内容                        | 備考                |
|----------|------------|---------------------------|-------------------|
| 0        | 0x00000000 | Spartan 用コンフィキ、ュレーションテ、ータ | SMT148 内の CP 接続定義 |
| 1        | 0x00800000 | Virtex4用 コンフィキ゛ュレーションテ゛ータ |                   |
|          | 0x00AA0000 | PPC 用アプリケーション (SREC ファイル) |                   |
|          | 0x01C00000 | PPC 用データファイル              |                   |

Virtex 用コンフィギュレーションファイル、PPC 用アプリケーションファイル、 PPC 用データファイルを必要に応じて設定する。

Spartan 用のデータは、SMT148のファームウェアを変更する場合以外、修正する必要はない。

| 💘 Sundance Flash Programming Uti  | ility for                     | FPG/          | \-Only                  | Module                 |                              |                      |
|---|-------------------------------|---------------|-------------------------|------------------------|------------------------------|----------------------|
| ♥ FPGA modules  |                               |               |                         |                        |                              |                      |
| General Advanced  |                               |               |                         |                        |                              |                      |
| TIM type<br>TIM SMT148FX<br>FPGA bitstream  | etails                        |               |                         |                        |                              |                      |
| Ind   File Name   | Type                          | Posi          | Address                 | Address                | Bytes                        | Byte                 |
| U U:¥Sundance¥I-UNUY_hirmware¥smt148<br>1 C:¥test¥PPC_New¥download_bootloader.bit<br>2 C:¥test¥PPC_New¥et2cp_echo_executabl | Bitstre<br>Bitstre<br>Bitstre | U<br>1<br>N/A | 0x0<br>0x8000<br>0xAA00 | basic<br>basic<br>user | 0x9F24C<br>0x280F<br>0x73E58 | 0x9F<br>0x25<br>0x7F |
|   |                               |               |                         |                        |                              | <b>&gt;</b>          |
| <u>Add</u>  |                               |               |                         |                        |                              |                      |
|   |                               |               |                         |                        | Ci                           | ommit                |
| Help  |                               |               |                         | Cancel                 |                              | ОК                   |

注:SMT6002で、SRECファイルを指定するには、

| File:       C:¥test¥PPC_New¥et2cp_echo_executable.elf.srec       選択時にクリック。         Type:       Bitstream       User を設定         Address:       0x aa0000       Address mode:       User | st¥PPC_New¥et2cp_echo_executable.elf.srec 選択時にクリック。<br>ream ▼ |          |
|---|---|----------|
| Type: Bitstream<br>Position in Userを設定  | ream 💌  |          |
| Position in User を設定  |   |          |
|   | User を設定  |          |
| Address. OX 1 Address mode. 1   | aa0000 Address mode: User                                     |          |
| Version: ツールのバグにより、"Us  | ツールのバグにより、"U  | ser"     |
| Comment: aa0000 選択時にはアドレス入力欄/   | 選択時にはアドレス入力欄  | に、       |
| 直接/トレス値を入力する とができない。  |   | ວ –      |
|   |   | , →      |
| Cancel OK カレ、コピーアンドペース  | Cancel OK カし、コピーアンドペーン  | を入<br>スト |

でアドレス欄に入力する。



(6) 必要なファイルの指定が終わったら、"Commit" をクリックする。 フラッシュの書込が開始する。

| Sundance Flash Programming Utility for FPGA-Only Module   | <                     |
|---|-----------------------|
| General Advanced TIM type TIM SMT148FX Details FPGA bitstream Ind File Name OC:¥Sundance¥T-UNIV_Firmware¥smt148 Bitstre 0 0x0 basic 0x9F24C 0x9F 1 C:¥test¥PPC_New¥download_bootloader.bit Bitstre 1 0x8000 basic 0x280F 0x28 2 C:¥test¥PPC_New¥et2cp_echo_executabl Bitstre N/A 0xAA00 user 0x73E58 0x7F |                       |
| Add     Edit     Delete       Commit     Commit       Help     Cancel   | クリックすること<br>で書込を開始する。 |

(7) 書込が終わったら、電源を OFF して、SMT148 基板上の SW を下記のように設定する。



- (8) SW を ON することで、ブートローダーが動作し、アプリケーションソフトがロードされる。
- 13.2 PPC による、フラッシュの書き換え PPC が、正常に起動している場合、PPC を使用し、フラッシュの書き換えができる。
- (1) SMTY148 に電源を投入し、PPC を起動する。
- (2) SundanceのSMT6058..¥SMT6058¥Host¥SMT6058\_Flash\_Programming\_Tool¥ のフォルダ内に提供されている、"SMT6058\_Flash\_Programming\_Tool"を 起動する。



| 😵 SMT6058 FLASH Programming Too |                                       |
|---------------------------------|---------------------------------------|
| File Selection Spartan          |                                       |
|                                 |                                       |
|                                 | 0s/0s                                 |
| In Flash :                      |                                       |
| Virtex                          |                                       |
|                                 |                                       |
| In Flash                        | 0s/0s                                 |
|                                 |                                       |
| PPC ET                          | · · · · · · · · · · · · · · · · · · · |
|                                 |                                       |
| In Flash :                      |                                       |
| PPC MFS                         |                                       |
|                                 |                                       |
|                                 | 0s/0s                                 |
| In Flash : ]                    |                                       |
| File                            | 05/05                                 |
|                                 | File at address : 0x 01 000000        |
| IP Address                      | SMT148FX Reset                        |
| IP: 192 . 168 . 1 . 10 Port: 20 | Reset after programming Reset         |
| Initialization                  | Program Flash                         |

SMT6058 の、フラッシュ 書込ツールが起動したら、 書き込みたいファイルを選 択し、Program Flash を クリックする。