MS104-FPGA/CⅢ
NiosⅡプロセッサを使用したサンプルプログラム解説

2.1 版　2023 年 10 月 02 日

1.　概要

1.1　概要

本アプリケーションノートは MS104-FPGA/CⅢに付属する NiosⅡプロセッサを使用したサンプルプログラムと、NiosⅡ IDE を使用したデバッグ手順について解説します。MS104-FPGA/CⅢには下記の NiosⅡプロセッサを使用したサンプルプログラムが付属しています。

サンプルプログラム	動作内容
NiosⅡプロセッササンプルプログラム	・SDRAM アクセス ・シリアル通信(CN2) ・ロータリスイッチ(RSW1) ・LED 制御(LD1～LD4)

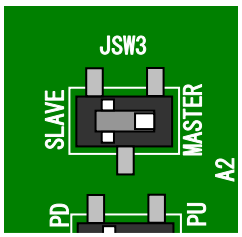
詳細な動作内容に関しては、後述の「4.動作説明」を参照してください。

1.2　準備物

・MS104-FPGA/CⅢ
・USB-Blaster
・AC アダプタ等の電源(DC5V)
・MS104-FPGA/CⅢ付属の CD-ROM(CD-ROM 内の[FPGA]フォルダを PC の QuartusⅡのワークフォルダにコピーしてください。)

1.3　設定

本サンプルプログラムは、MS104-FPGA/CⅢ単体で動作します。PC/104 バスに対応したサンプルプログラムは付属しませんので、他の PC/104 準拠ボードを接続しないでください。
MS104-FPGA/CⅢの設定は、PC/104 バスの設定をマスタとし、それ以外は出荷時設定としてください。



JSW3 PC/104 バスの設定　：　MASTER

1.4　開発環境について

本サンプルプログラムは FPGA/CPLD 開発ソフトウェア QuartusⅡと、NiosⅡ統合開発環境 NiosⅡ IDE を用いて開発されています。サンプルプログラムに対応する開発環境は以下のようになります。

開発環境	バージョン
QuartusⅡ	Version8.1
NiosⅡ IDE	Version8.1

※ QuartusⅡ Web Edition 及び NiosⅡ IDE は Intel 社のウェブサイトからダウンロードできます。

＊ QuartusⅡおよび NiosⅡ IDE のバージョンについて

QuartusⅡと NiosⅡ IDE のバージョンを必ず同じにして開発を行ってください。バージョンが違う場合、正しく動作しない可能性があります。

＊ NiosⅡライセンスについて

MS104-FPGA/CⅢには NiosⅡのライセンスは付属しません。NiosⅡのライセンスがない場合、以下の制約があります。

-コンパイル後に「*_timelimited.sof」ファイルが生成され、FPGA にダウンロードしてから一定期間(30 分程度)でハードウェアの動作が停止する。

-pof ファイルの生成ができない。

NiosⅡの IP を購入後、Intel 社へユーザ登録するとライセンスファイルの取得が可能になり、上記の制約はなくなります。

1.5　開発の流れ

<STEP1>

NiosⅡ IDE からプロジェクトをビルドする。

<STEP2>

NiosⅡ CPU 及び周辺機能システムを FPGA にダウンロードする。

<STEP3>

SDRAM にプログラムをダウンロードし、NiosⅡ IDE からデバッグする。

2.　サンプルプログラムの構成

MS104-FPGA/CⅢに付属の CD-ROM に収録されているサンプル HDL プログラムは下記のようなフォルダ構成になっています。サンプル HDL プログラムはあらかじめ QuartusⅡのワークフォルダにコピーし、開発を行ってください。

```
fpga
 ├── ms104fpga_sample    MS104-FPGA/CⅢサンプル HDL プログラム プロジェクト一式*1
 ├── ms104fpga_write     MS104-FPGA/CⅢPFL プログラム プロジェクト一式*1
 └── ms104fpga_nios2     MS104-FPGA/CⅢNiosⅡプロセッサ用サンプルプログラムプロジェクト一式
```

*1　ms104fpga_sample、ms104fpga_write についての詳細はアプリケーションノート「AN1201 FLASHROM の書き込み方法」を参照してください。

Fig 2-1　サンプル HDL プログラムの構成

3.　動作手順

MS104-FPGA/CⅢに搭載されている FPGA に sof ファイル(NiosⅡ及び周辺機能)をダウンロードし、プログラムを SDRAM にダウンロードしてデバッグする手順を以下に示します。

<STEP1>　NiosⅡ IDE からプロジェクトをビルドする

①NiosⅡ IDE の起動

NiosⅡ IDE を起動します。

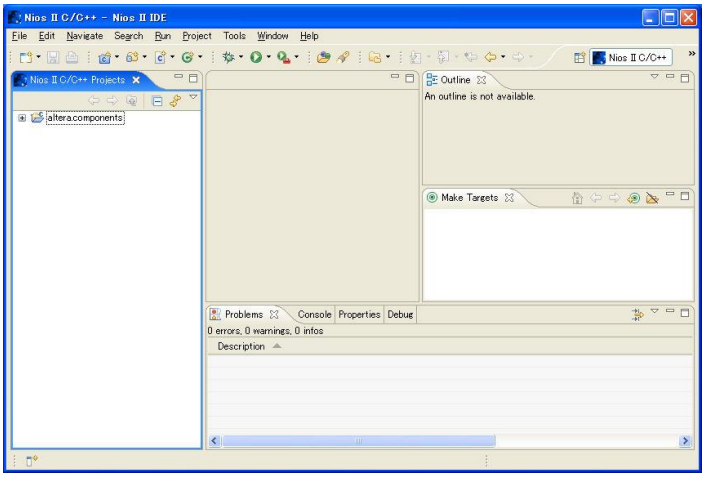


Fig 3-1　NiosⅡ IDE の起動

②プロジェクトのインポート 1

NiosⅡ C/C++ Project ウィンドウ内で右クリックし、[Import...]を選択します。

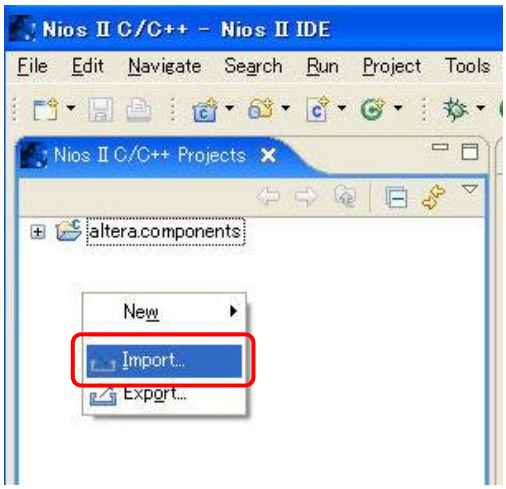


Fig 3-2　プロジェクトのインポート 1

③プロジェクトのインポート 2

[Exsiting NiosⅡ IDE project into work space]を選択し、[Next]を押します。

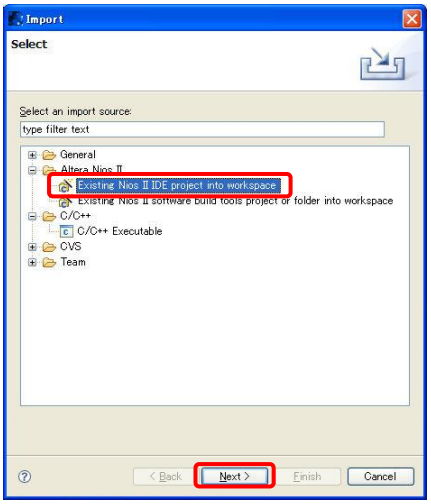


Fig 3-3　プロジェクトのインポート 2

④プロジェクトのインポート 4

Project content に、インポートするフォルダを指定します。添付の CD-ROM に収録されている[fpga]フォルダ内の以下のフォルダを指定し、[Finish]を押します。

Project contents　　ms104fpga_nios2¥software¥ms104fpga_nios2*1

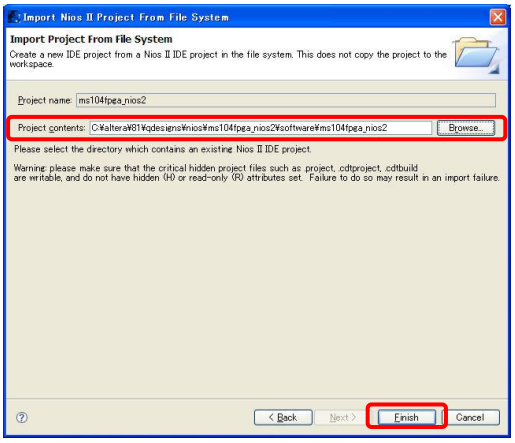


Fig 3-4　プロジェクトのインポート 3

*1　あらかじめ CD-ROM 内の[FPGA]フォルダを QuartusⅡのワークフォルダにコピーし、コピー先のフォルダを指定してください。

⑤プロジェクトのインポート完了

インポートが完了すると、インポートされたフォルダが NiosⅡ IDE の NiosⅡ C/C++ Project ウィンドウ内に表示されます。

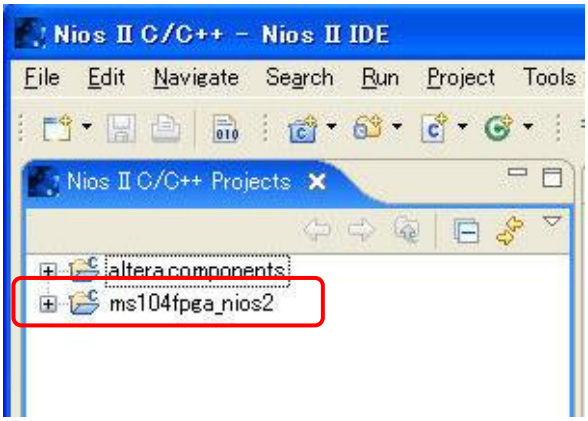


Fig 3-5　プロジェクトのインポート完了

⑥システムライブラリのインポート 1

次にシステムライブラリフォルダのインポートをします。NiosⅡ C/C++ Project ウィンドウ内で右クリックし、[Import...]を選択します

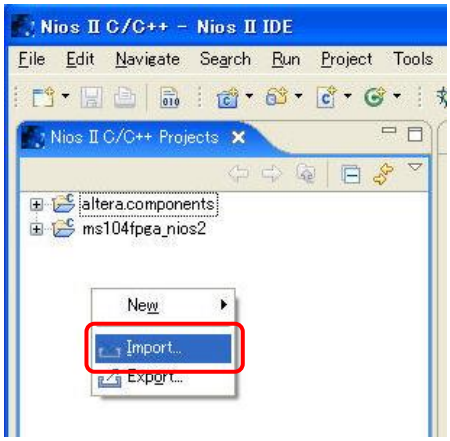


Fig 3-6　システムライブラリのインポート 1

⑦システムライブラリのインポート 2

[Exsiting NiosⅡ IDE project into work space]を選択し、[Next]を押します。

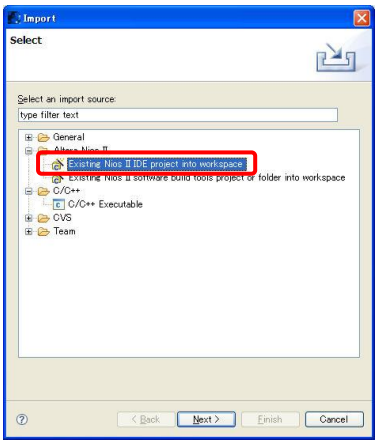


Fig 3-7　システムライブラリのインポート 2

⑧システムライブラリのインポート 3

Project content に、インポートするフォルダを、SOPC Builder System PTF File に PTF ファイルを指定します。以下のフォルダ、ファイル*1 を指定し[Finish]を押してください。

Project contents	ms104fpga_nios2¥software¥ms104fpga_nios2_syslib
SOPC Builder System PTF File	ms104fpga_nios2¥ms104fpga_sopc.ptf

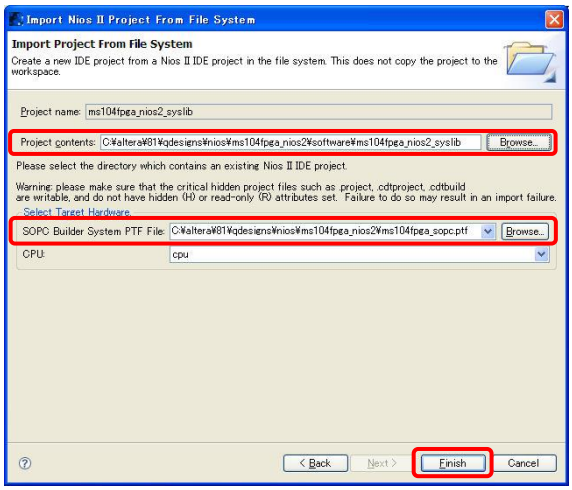


Fig 3-8　システムライブラリのインポート 3

*1　あらかじめ CD-ROM 内の[FPGA]フォルダを QuartusⅡのワークフォルダにコピーし、コピー先のフォルダを指定してください。

⑨システムライブラリのインポート完了

インポートが完了すると、インポートされたフォルダが NiosⅡ IDE の NiosⅡ C/C++ Project ウィンドウ内に表示されます。

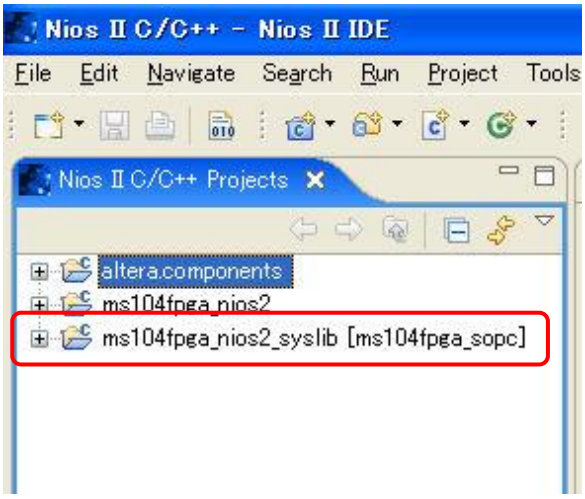


Fig 3-9　システムライブラリのインポート完了

⑩システムライブラリの設定 1

[ms104fpga_nios2_syslib]を右クリックし、[properties]を選択します。

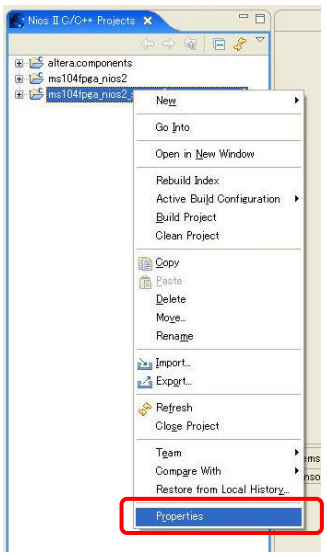


Fig 3-10　システムライブラリの設定 1

⑪システムライブラリの設定 2

[System Library]を選択し、[Small C library]のチェックボックスを有効にし、以下の項目を設定します。

Program memory	:	SDRAM
Read-only data memory	:	SDRAM
Read/write data memory	:	SDRAM
Heap memory	:	SDRAM
Stack memory	:	SDRAM

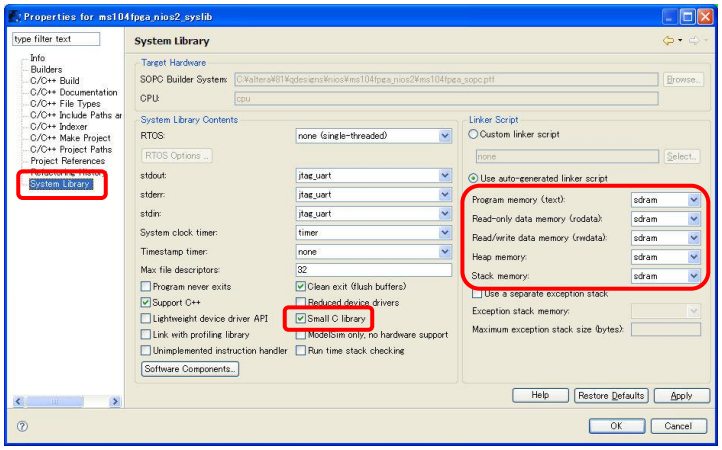


Fig 3-11　システムライブラリの設定 2

⑫プロジェクトのビルド 1

[ms104fpga_nios2]を右クリックし、[Build Project]を選択します。

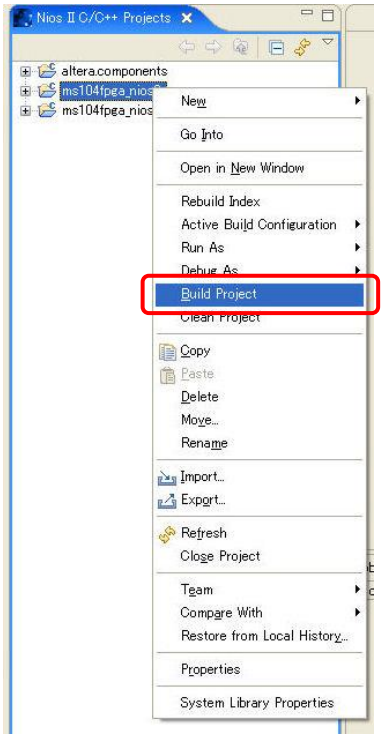


Fig 3-12　プロジェクトのビルド 1

⑬プロジェクトのビルド 2

ビルドが開始され、経過が表示されます。ビルドには数分かかる場合があります。

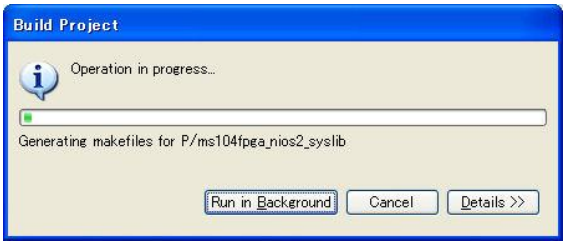


Fig 3-13　プロジェクトのビルド 2

⑭ビルド完了

ビルドが完了すると、Console ウィンドウ内に「Build complete in xxx seconds」と表示されます。

以上でプロジェクトのビルドが完了しました。続けてデバッグを行う場合は、NiosⅡ IDE を終了せずに続けて〈STEP2〉に進んでください。

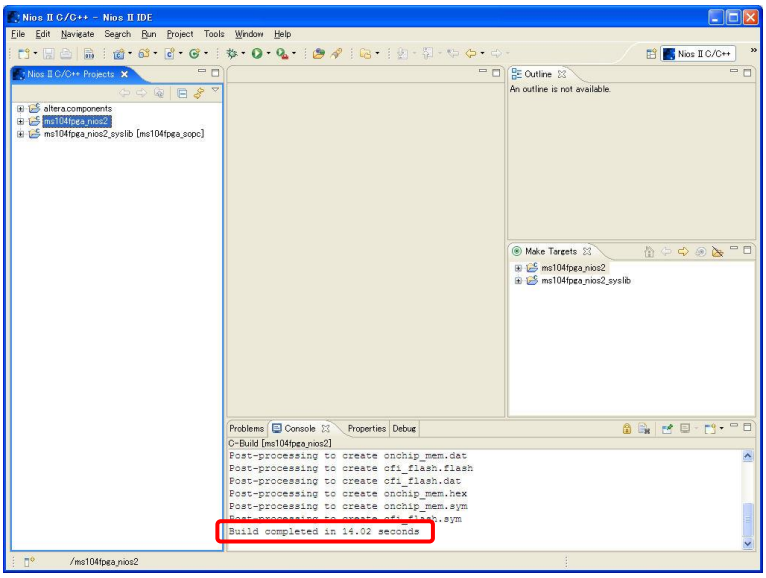


Fig 3-14　ビルド完了

<STEP2>　NiosⅡ CPU 及び周辺機能システムを FPGA にダウンロードする

①接続

MS104-FPGA/CⅢ、USB-Blaster、PC を以下のように接続してください。

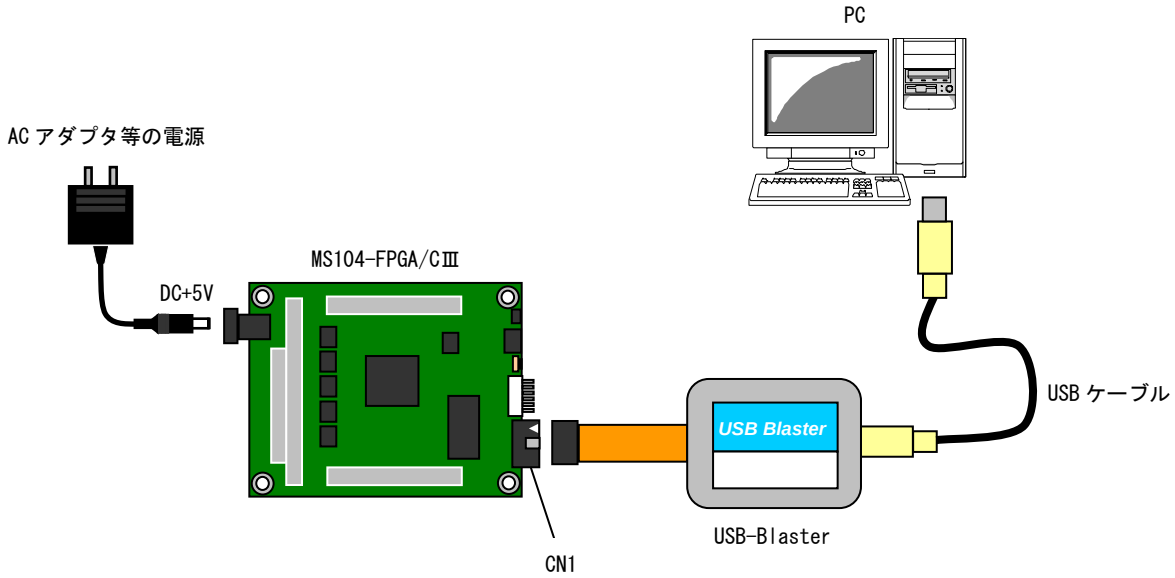


Fig 3-15　MS104-FPGA/CⅢ、USB-Blaster、PC の接続

②Programmer の起動

NiosⅡ IDE の[Tool]メニューから[QuartusⅡ Programmer]を選択します。

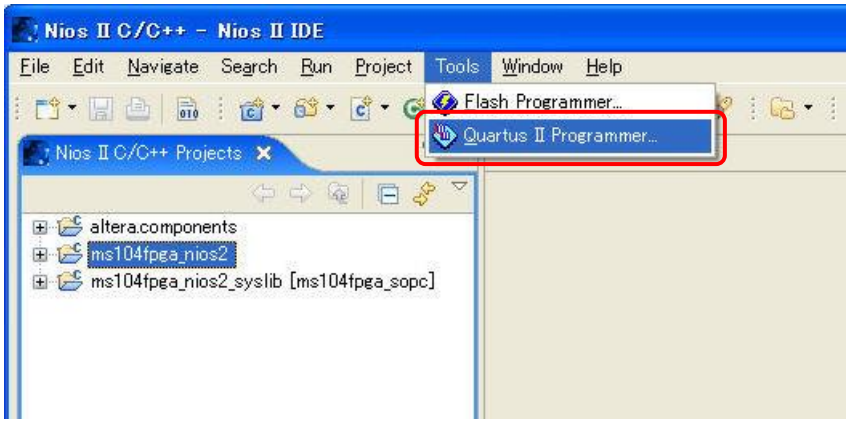


Fig 3-16　programmer の起動

③ハードウェアセットアップ

[Hardware Setup...]を押し、Hardware Setup 画面の[Currently selected hardware]から[USB Blaster[USB-0]]を選択します。

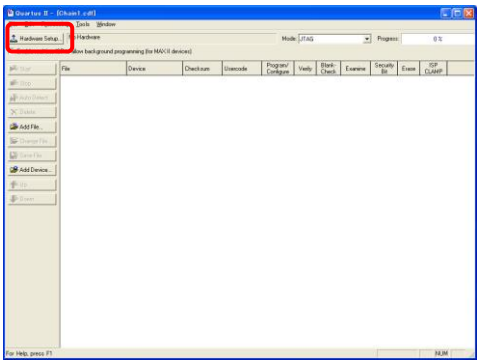


Fig 3-17　ハードウェアセットアップ 1

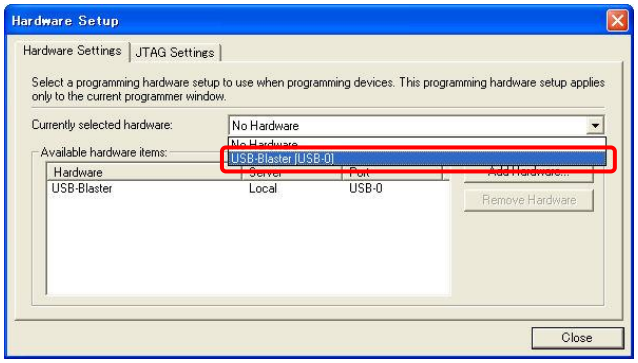


Fig 3-18　ハードウェアセットアップ 2

④Auto Detect の開始

MS104-FPGA/CⅢに電源を投入し、[Auto Detect]を押すと、デバイスの検出を開始します。検出が完了すると FPGA EP3C16 が表示されます。

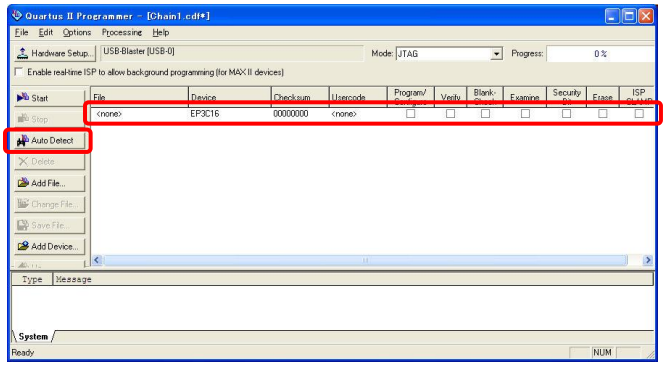


Fig 3-19　Auto Detect の開始

⑤ファイルの選択

検出された EP3C16 の File<none>をダブルクリックするか、または EP3C16 を選択し、[Change File]をクリックするとプログラムファイルの選択画面が表示されます。ここで FPGA にダウンロードするファイルを選択します。ファイルは CD-ROM 内に収録されている[fpga]-[ms104fpga_nios2]内にある「MS104FPGA_time_limited.sof」を選択してください。

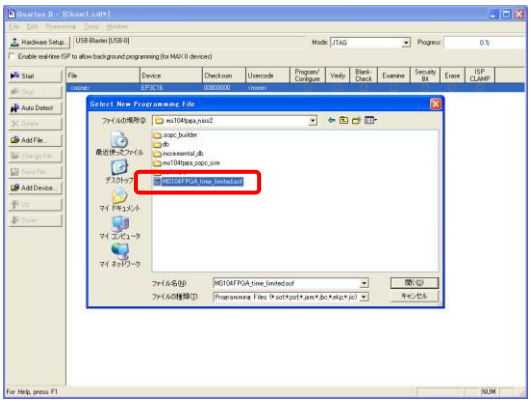


Fig 3-20　ファイルの選択

⑥ファイルのダウンロード 1

⑤で選択した SOF ファイルを FPGA にダウンロードします。Program/Configure ボックスをチェックし、[Start]をクリックします。

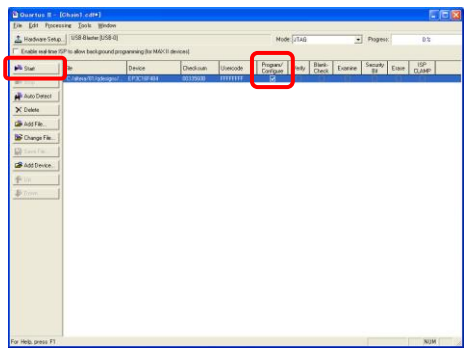


Fig 3-21　ファイルのダウンロード 1

⑦ファイルのダウンロード 2

ダウンロードが完了すると Open Core Plus Status 画面が表示されます。以上で NiosⅡ CPU 及び周辺機能システムのダウンロードが完了しました。ダウンロード完了後は MS104-FPGA/CⅢの電源を落とさず、また Programmer を終了させずに<STEP3>に進んでください。

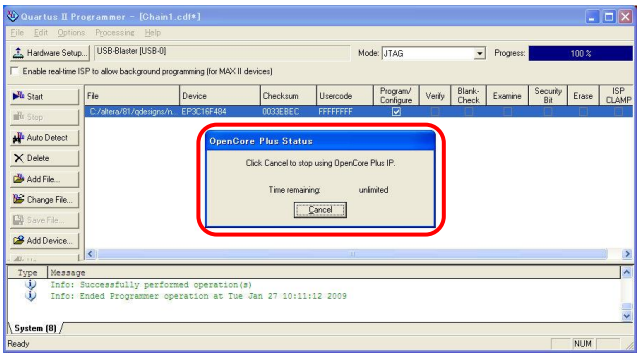


Fig 3-22　ファイルのダウンロード 2

＊ OpenCore Plus Status ダイアログについて

「*_time_limited.sof」ファイルを FPGA にダウンロードした場合、OpenCore Plus Status ダイアログが表示されます。[Cancel]ボタンを押すと FPGA の動作が停止してしまいますので、ご注意ください。

<STEP3>　SDRAM にプログラムをダウンロードし、NiosⅡ IDE からデバッグする

①デバッグの開始

FPGA に NiosⅡ CPU 及び周辺機能システムがダウンロードされている状態で、NiosⅡ IDE の NiosⅡ C/C++ Project ウィンドウ内の[ms104fpga_nios2]を右クリックし、[Debug As]-[Nios Ⅱ Hardware]を選択します。

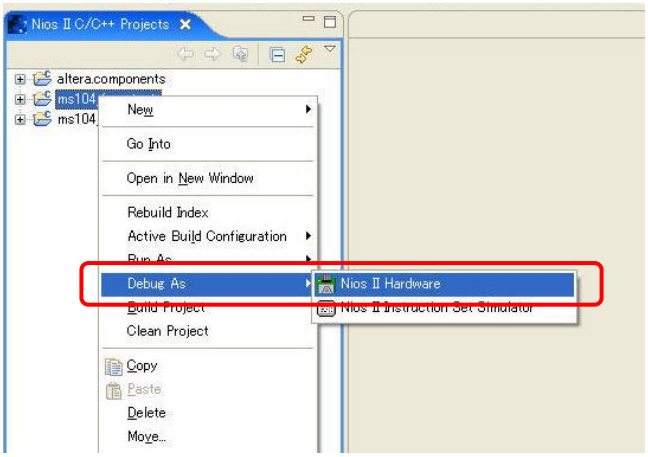


Fig 3-23　デバッグの開始

②デバッグ画面の選択

Confirm Perspective Switch 画面が表示されます。[Yes]を選択するとデバッグ画面が表示されます。

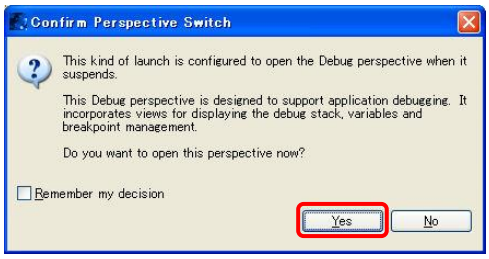


Fig 3-24　デバッグ画面の選択

③デバッグ作業

デバッグウィンドウでデバッグ操作ができます。詳細なデバッグ方法は NiosⅡ IDE のマニュアル等を参照してください。

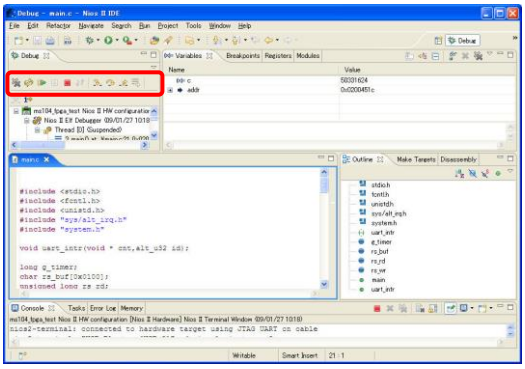


Fig 3-25　デバッグ作業

4. 機能

4.1 アドレスマップ

本サンプルプログラムでは、NiosⅡ CPU の周辺機能のアドレスは以下のように割り当てられています。

アドレス	機能	割り込み
H’0x00000000 H’0x00007FFF	On Chip Memory 32KByte	
H’0x00008000 H’0x00008007	JTAG UART	IRQ0
H’0x00008008 H’0x00008009	未使用	
H’0x00008010 H’0x0000801F	LED	
H’0x00008020 H’0x0000803F	タイマ	IRQ1
H’0x00008040 H’0x0000805F	SDRAM PLL	
H’0x00008060 H’0x0000807F	シリアル通信	IRQ2
H’0x00008080 H’0x0000808F	ロータリスイッチ RSW1	
H’0x00008090 H’0x000087FF	未使用	
H’0x00008800 H’0x00008FFF	CPU JTAG モジュール	
H’0x00009000 H’0x00FFFFFF	未使用	
H’0x01000000 H’0x017FFFFF	FLASH ROM 8MByte	
H’0x01800000 H’0x01FFFFFF	未使用	
H’0x02000000 H’0x02FFFFFF	SDRAM 16MByte	

Table 4-1　アドレスマップ

4.2　レジスタ

4.2.1　LED 制御レジスタ

LED(LD1～LD4)は LED 制御レジスタに値を書き込むことで制御します。

LED 制御レジスタ(H’00008010)

ビット	15	14	13	12	11	10	9	8
名称	–	–	–	–	–	–	–	–
初期値	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

ビット	7	6	5	4	3	2	1	0
名称	–	–	–	–	LD4_CTR	LD3_CTR	LD2_CTR	LD1_CTR
初期値	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 4-2　LED 制御レジスタ

ビット	ビット名称	初期値	R/W	機能
15～8	–	0	R	未使用
3～0	LDx_CTR	0	R/W	0:LDx 点灯 1:LDx 消灯

Table 4-3　LED 制御レジスタの説明

4.2.2　RSW1 ステータスレジスタ

ロータリスイッチ RSW1 の状態は RSW1 ステータスレジスタ反映されます。

LED 制御レジスタ (H’ 00008080)

ビット	15	14	13	12	11	10	9	8
名称	–	–	–	–	–	–	–	–
初期値	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

ビット	7	6	5	4	3	2	1	0
名称	–	–	–	–	RSW1_8_STS	RSW1_4_STS	RSW1_2_STS	RSW1_1_STS
初期値	0	0	0	0	X*1	X*1	X*1	X*1
R/W	R	R	R	R	R	R	R	R

*1　ロータリスイッチ RSW1 の状態による

Table 4-4　RSW1 ステータスレジスタ

ビット	ビット名称	初期値	R/W	機能
15～8	–	0	R	未使用
3～0	RSW1_x_STS	X*1	R	RSW1 のポジションを表示 Table4-6 参照

*1　ロータリスイッチ RSW1 の状態による

Table 4-5　RSW1 ステータスレジスタの説明

ロータリスイッチ RSW1 のポジションと RSW1_x_STS ビットの関係は以下のようになります。

ポジション	RSW1_8_STS	RSW1_4_STS	RSW1_2_STS	RSW1_1_STS
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	1	0	0
4	1	0	1	1
5	1	0	1	0
6	1	0	0	1
7	1	0	0	0
8	0	1	1	1
9	0	1	1	0
A	0	1	0	1
B	0	1	0	0
C	0	0	1	1
D	0	0	1	0
E	0	0	0	1
F	0	0	0	0

Table 4-6　ロータリスイッチ RSW1 と RSW1_x_STS ビットの関係

5.　サンプルプログラム

5.1　サンプルプログラム動作説明

NiosⅡプロセッサを使用したサンプルプログラムは以下の動作を行います。

- SDRAM アクセス
  本サンプルプログラムは SDRAM にプログラムをダウンロードし、動作します。

- シリアル通信エコーバック(送受信割り込み使用)
  MS104-FPGA/CⅢの通信コネクタ(CN2)から受信した値をそのまま送信します。
  ボーレート　115200bps、ビット長 8、パリティなし、ストップビット 1
  動作確認は、以下のような接続で、PC 上のターミナルソフト(ハイパーターミナル等)を使用してください。

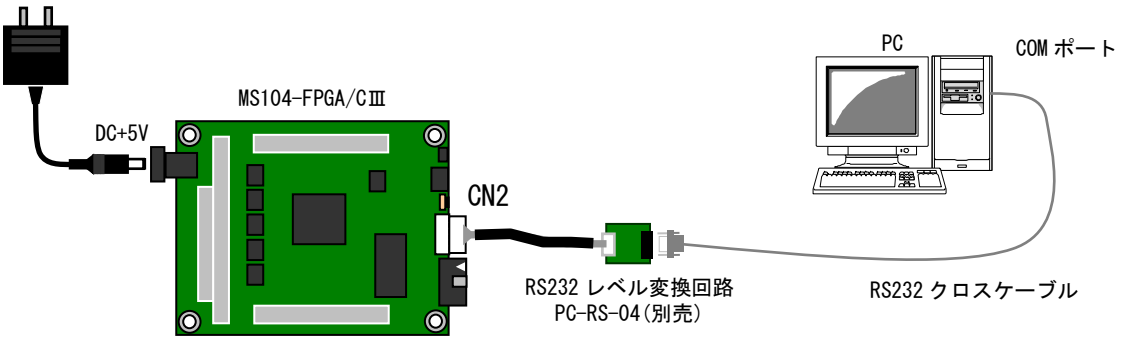


Fig 5-1　シリアル通信接続例

弊社製 RS232 レベル変換アダプタ(PC-RS-04)を使用しない場合には、シリアル入出力を外部回路にて RS232 レベルに変換する必要があります。以下に RS232 レベル変換回路の設計例を示します。

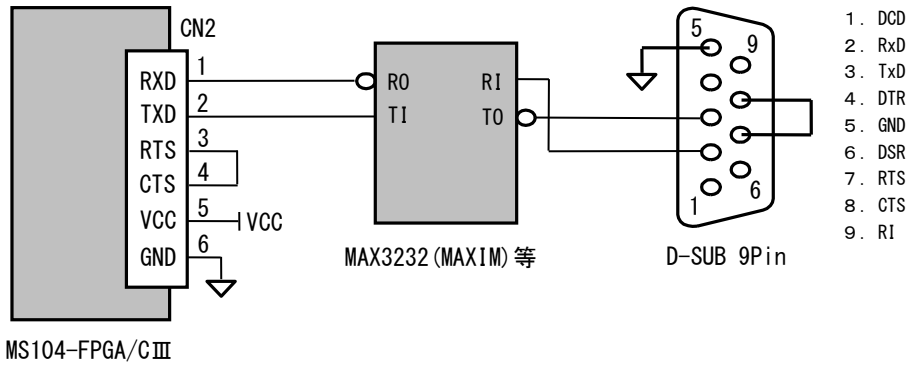


Fig 5-2　RS232 レベル変換回路例

● ロータリスイッチ RSW1 のポジションを読み出し、LD1～LD4 に表示
ロータリスイッチ RSW1 のポジションと、LD1～LD4 の関係を以下に示します。

RSW1 ポジション	LD4	LD3	LD2	LD1
0	消灯	消灯	消灯	消灯
1	消灯	消灯	消灯	点灯
2	消灯	消灯	点灯	消灯
3	消灯	消灯	点灯	点灯
4	消灯	点灯	消灯	消灯
5	消灯	点灯	消灯	点灯
6	消灯	点灯	点灯	消灯
7	消灯	点灯	点灯	点灯
8	点灯	消灯	消灯	消灯
9	点灯	消灯	消灯	点灯
A	点灯	消灯	点灯	消灯
B	点灯	消灯	点灯	点灯
C	点灯	点灯	消灯	消灯
D	点灯	点灯	消灯	点灯
E	点灯	点灯	点灯	消灯
F	点灯	点灯	点灯	点灯

Table 5-1　ロータリスイッチ RSW1 と LED の状態

ご注意

・本文書の著作権は株式会社アルファプロジェクトが保有します。
・本文書の内容を無断で転載することは一切禁止します。
・本文書の内容は、将来予告なしに変更されることがあります。
・本文書に記載されている内容についての質問等のサポートは一切受け付けておりませんのでご了承ください。
・本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡下さい。
・本文書の内容に基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

商標について

・CycloneⅢは Intel Corporation の登録商標、商標または商品名称です。
・Nios®Ⅱは Intel Corporation の登録商標、商標または商品名称です。

・その他の会社名、製品名は、各社の登録商標または商標です。

ALPHA PROJECT
株式会社アルファプロジェクト
〒431-3114
静岡県浜松市中央区積志町 834
https://www.apnet.co.jp
E-Mail: query@apnet.co.jp