

# Nios II - オンチップ・メモリ からのブート手順

ver.14

## Nios II - オンチップ・メモリからのブート手順

### 目次

1. はじめに .....	3
1-1. ブート用オンチップ・メモリの設定 .....	3
1-2. Nios II プロセッサのリセット・ベクタと例外ベクタの設定 .....	4
2. Nios II SBT を使用したときのオンチップ・メモリからのブート手順 .....	5
2-1. BSP Editor の設定 (ROM 設定のオンチップメモリからブートする場合) .....	5
2-2. HEX ファイルの生成方法 .....	6
2-3. Quartus II での設定とコンパイル .....	9
2-4. HEX ファイル更新時の Quartus II プロジェクトへの反映方法 .....	11
改版履歴 .....	13

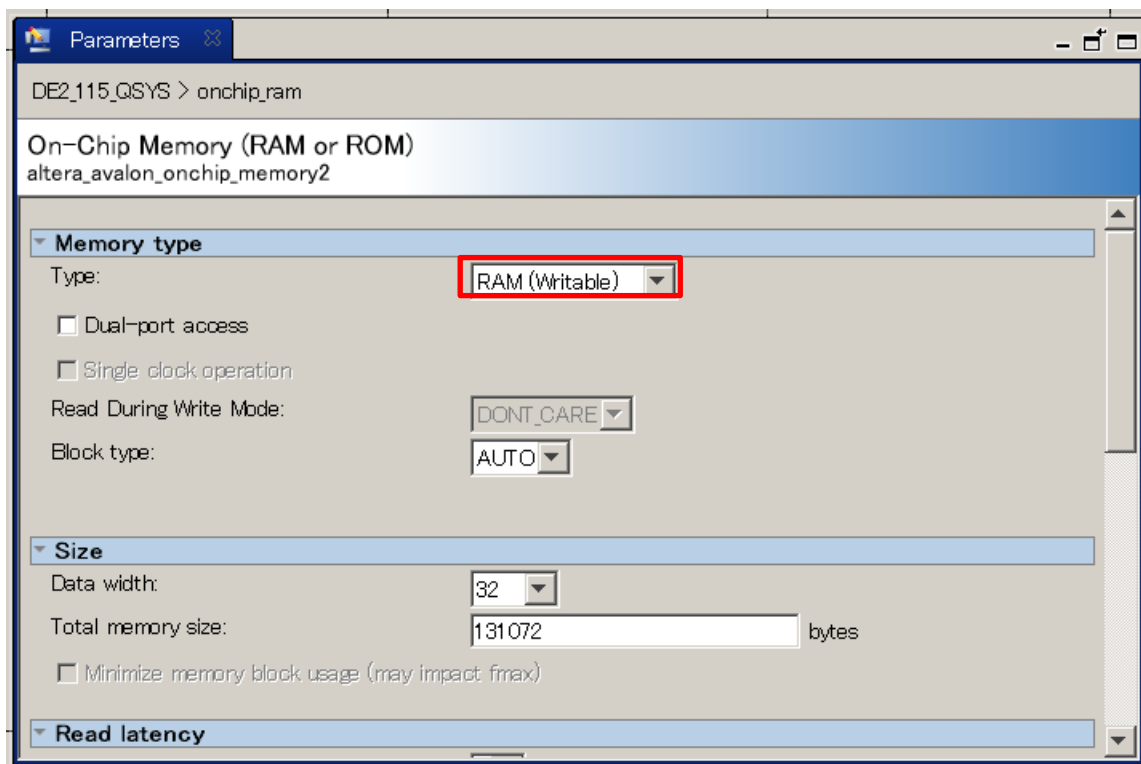
## 1. はじめに

この資料は、Nios® II のソフトウェア・プログラムを FPGA デバイス内のオンチップ・メモリからブートする際の設定方法について説明します。前提として、Quartus® II（本例では、ver 14.0 を使用）にてハードウェア・デザインが完成していて、Nios II Software Build Tool for Eclipse（以下、Nios II SBT）から JTAG 経由でアプリケーション・ソフトウェアが正常に動作することが必要となります。

### 1-1. ブート用オンチップ・メモリの設定

ソフトウェア・プログラムを配置するオンチップ・メモリをシステムに追加しますが、ROM/RAM のどちらかを選択する必要があります。以下の特徴を参考に、用途によって選択してください。

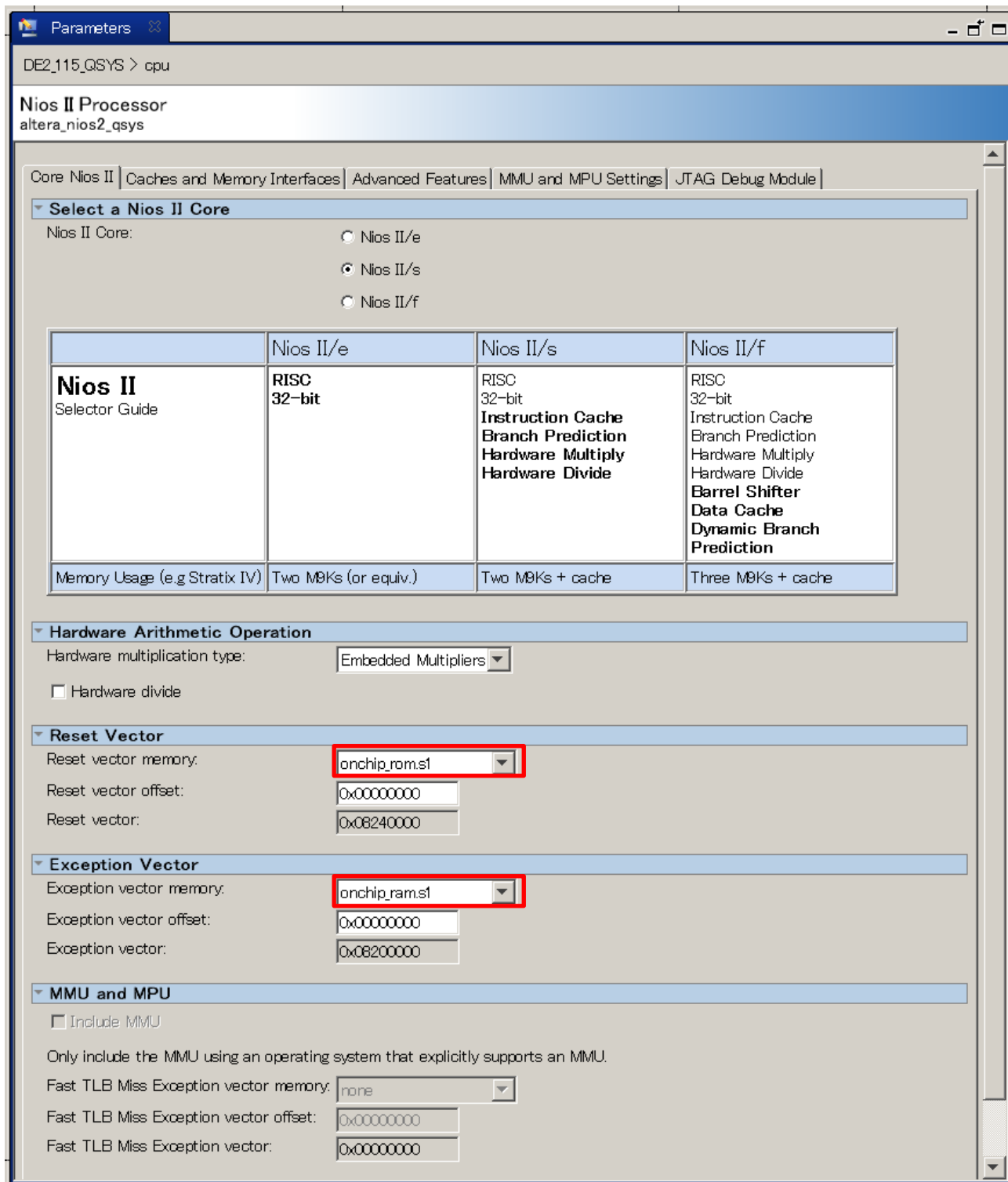
- ・ RAM の場合：RAM 領域としても使用可能のため、別途 RAM を用意する必要がありません。  
ただし、ソフトウェアの作りによってはソフトウェア・プログラム本体を破壊してしまい起動できなくなる恐れがあります。
- ・ ROM の場合：RAM 領域が別途必要です。  
(この資料では、RAM 領域としてオンチップ・メモリを使用した例を紹介します。)



## 1-2. Nios II プロセッサのリセット・ベクタと例外ベクタの設定

Qsys 上にある Nios II のリセット・ベクタの設定をオンチップ・メモリに設定する必要があります(下図参照)。この設定で、FPGA 起動後に Nios II がリセット・ベクタで指定されたアドレスからソフトウェア・プログラムをロードしブートが開始します。

ブート用オンチップ・メモリが ROM の場合、例外・ベクタの設定を RAM 領域に設定する必要があります。今回の例では、オンチップ・メモリを追加します。



DE2\_115\_QSYS > cpu

**Nios II Processor**  
altera\_nios2\_qsys

Core Nios II | Caches and Memory Interfaces | Advanced Features | MMU and MPU Settings | JTAG Debug Module

▼ **Select a Nios II Core**

Nios II Core:  Nios II/e  Nios II/s  Nios II/f

	Nios II/e	Nios II/s	Nios II/f
<b>Nios II</b> Selector Guide	RISC 32-bit	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide Barrel Shifter Data Cache Dynamic Branch Prediction
Memory Usage (e.g Stratix IV)	Two M9Ks (or equiv.)	Two M9Ks + cache	Three M9Ks + cache

▼ **Hardware Arithmetic Operation**

Hardware multiplication type: Embedded Multipliers ▼

Hardware divide

▼ **Reset Vector**

Reset vector memory: onchip\_rom.s1 ▼

Reset vector offset: 0x00000000

Reset vector: 0x08240000

▼ **Exception Vector**

Exception vector memory: onchip\_ram.s1 ▼

Exception vector offset: 0x00000000

Exception vector: 0x08200000

▼ **MMU and MPU**

Include MMU

Only include the MMU using an operating system that explicitly supports an MMU.

Fast TLB Miss Exception vector memory: none ▼

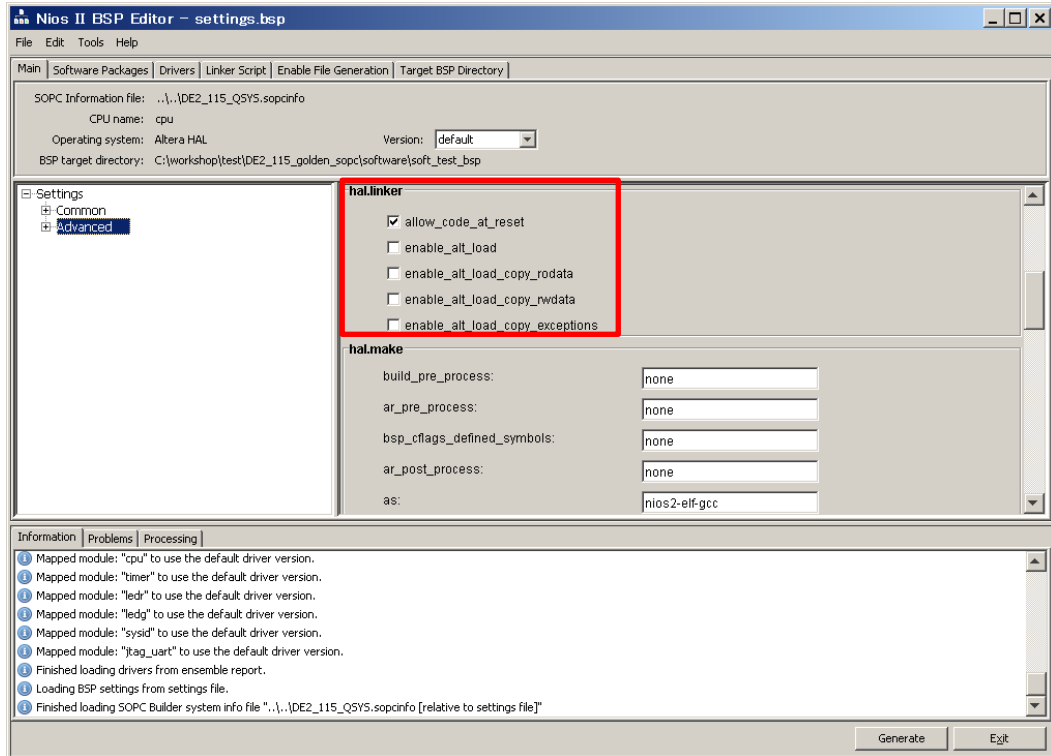
Fast TLB Miss Exception vector offset: 0x00000000

Fast TLB Miss Exception vector: 0x00000000

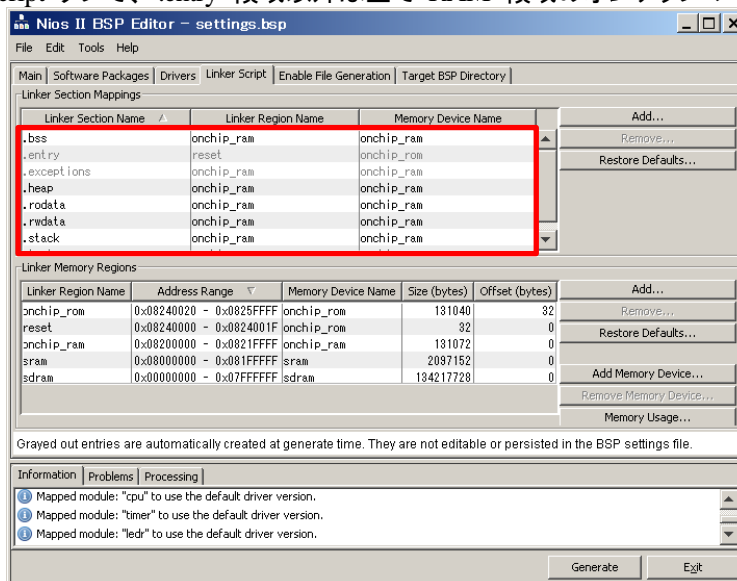
## 2. Nios II SBT を使用したときのオンチップ・メモリからのブート手順

### 2-1. BSP Editor の設定 (ROM 設定のオンチップ・メモリからブートする場合)

Nios II SBT にて BSP ・ソフトウェア・プロジェクト(本例では、soft\_test\_bsp)を右クリックして、プルダウン・メニューから BSP Editor を起動します。そこで、Main タブで下図のように hal.linker の設定をします。

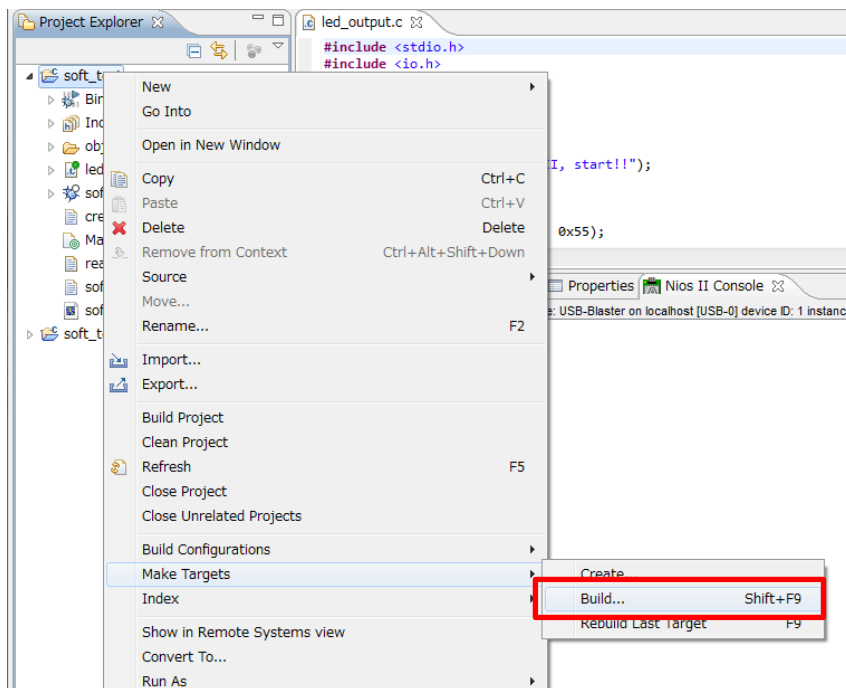


Linker Script タブで、.entry 領域以外は全て RAM 領域のオンチップ・メモリに設定する。

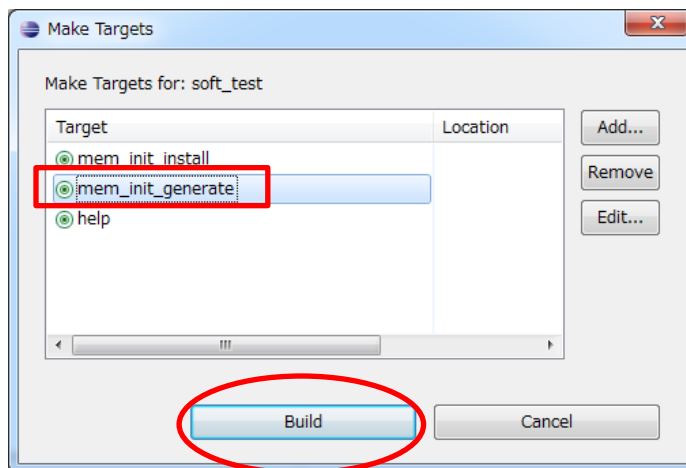


## 2-2. HEX ファイルの生成方法

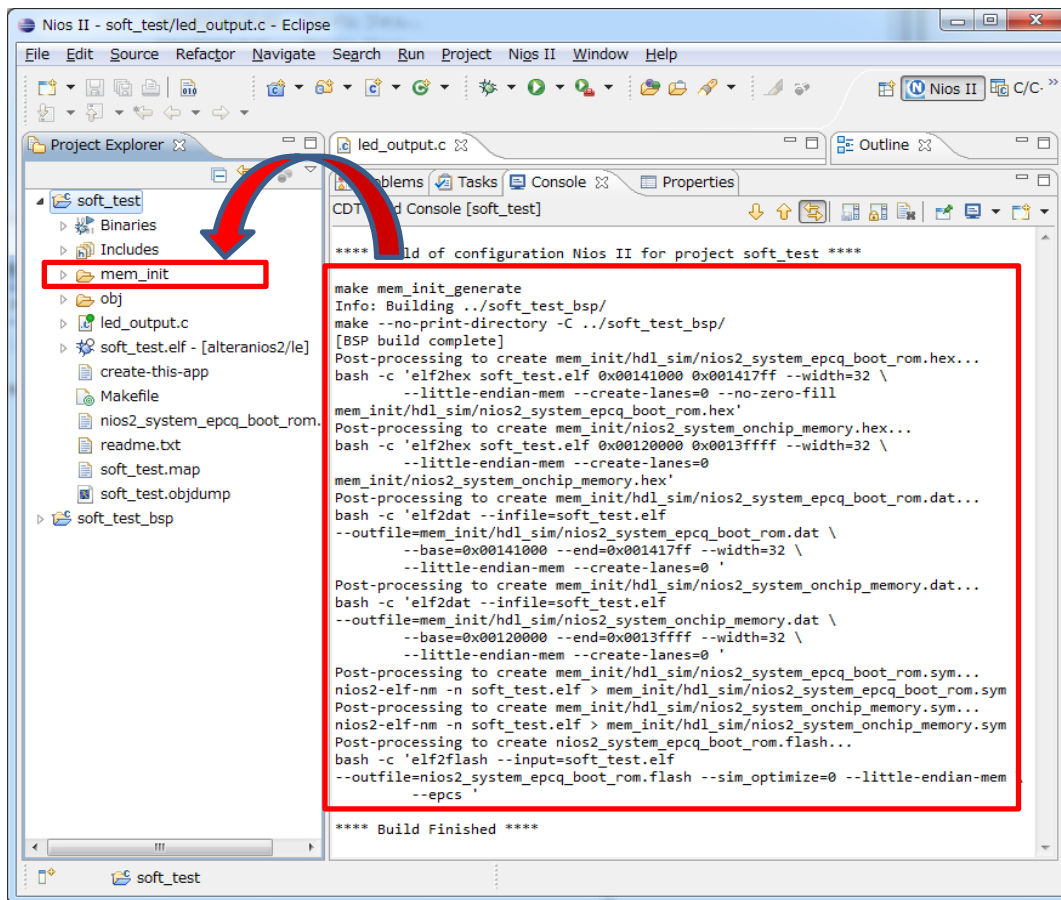
Nios II SBT にてビルド済のアプリケーション・ソフトウェア・プロジェクト(本例では、soft\_test) を右クリックして、プルダウン・メニューから Make Targets ⇒ Build を選択します。



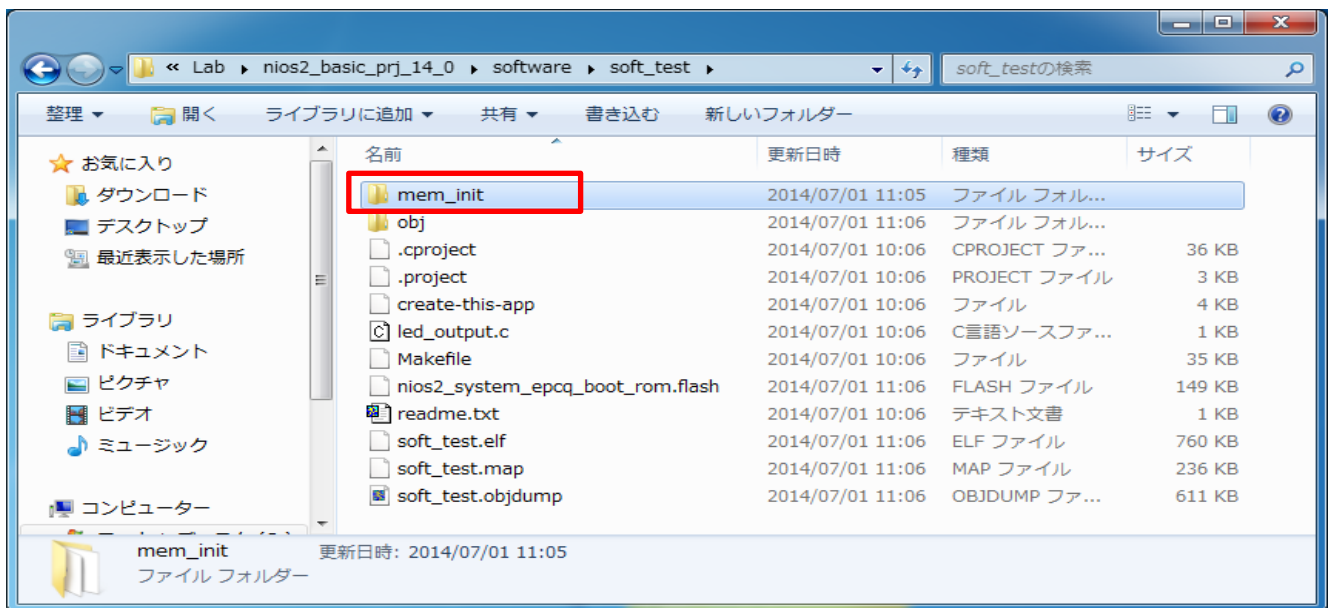
下図のように、Make Targets 画面が起動したら、mem\_init\_generate をハイライトして、[Build] ボタンをクリックします。



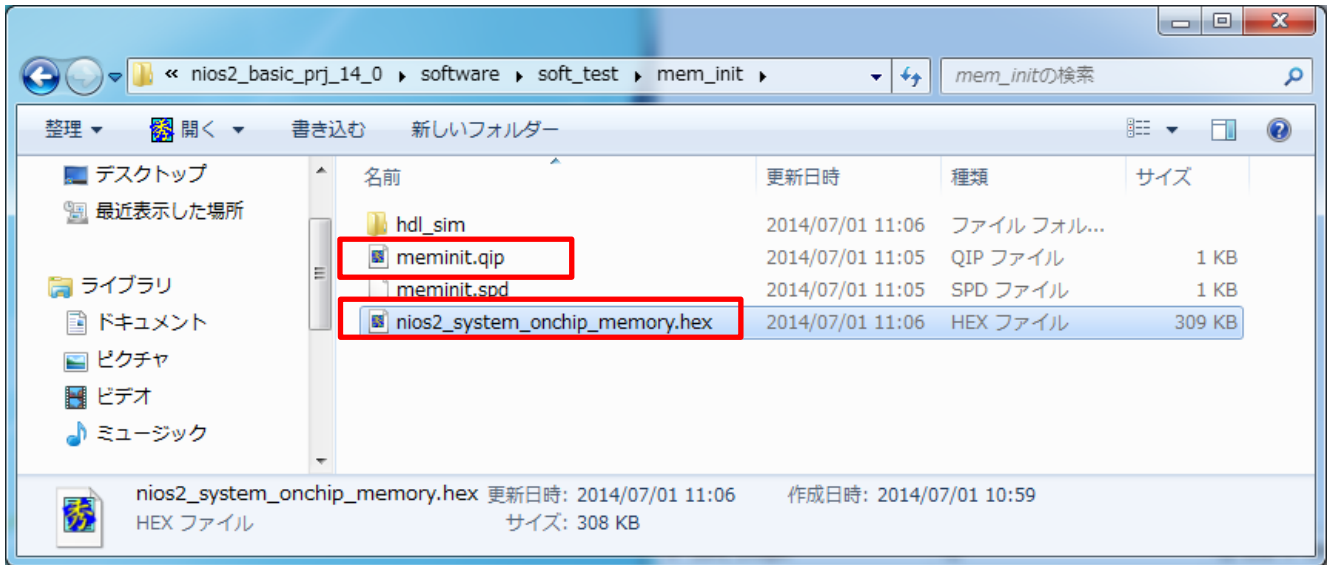
この操作によって、mem\_init フォルダがアプリケーション・プロジェクトの中に生成されます。



下図のように、Windows エクスプローラーから見ても mem\_init フォルダが生成されていることが確認できます。



mem\_init フォルダ内には、下図のように、meminit.qip ファイルと nios2\_system\_onchip\_memory.hex ファイル (※このファイル名は、プロジェクトごとに異なります。)を確認することができます。



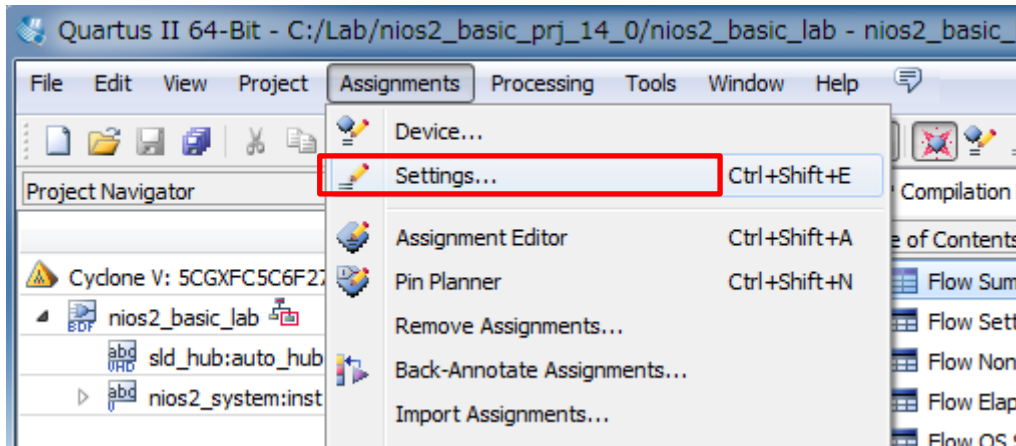
nios2\_system\_onchip\_memory.hex ファイルをテキスト・エディタで開くと、Hex フォーマットでの記述となりますが、ソフトウェア・コードがデータとして反映されていることが確認できます。





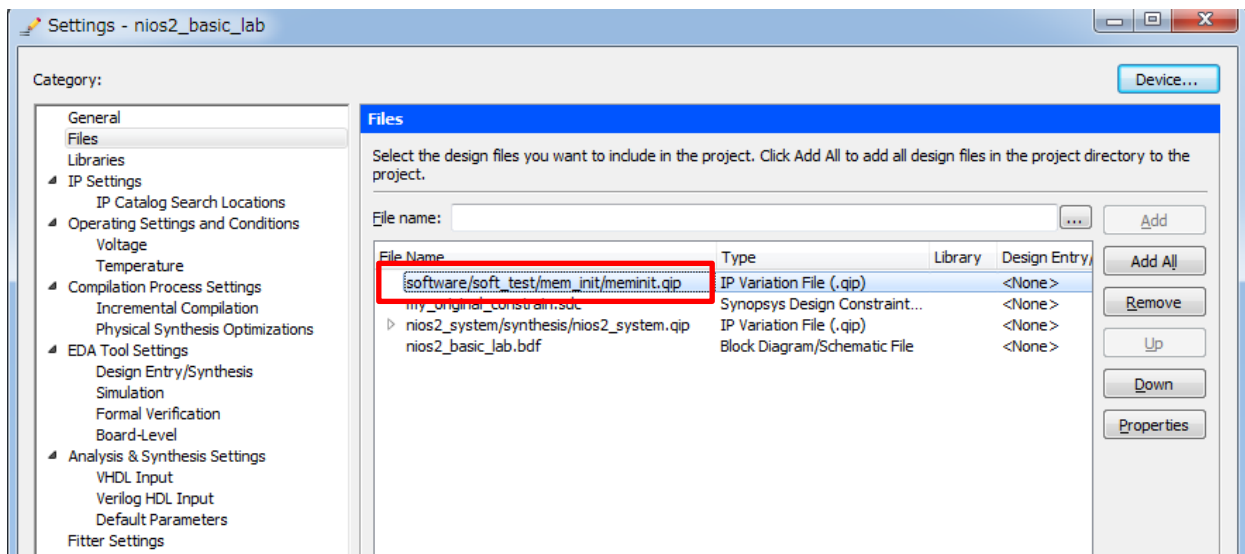
### 2-3. Quartus II での設定とコンパイル

Assignments メニューから Settings を選択します。

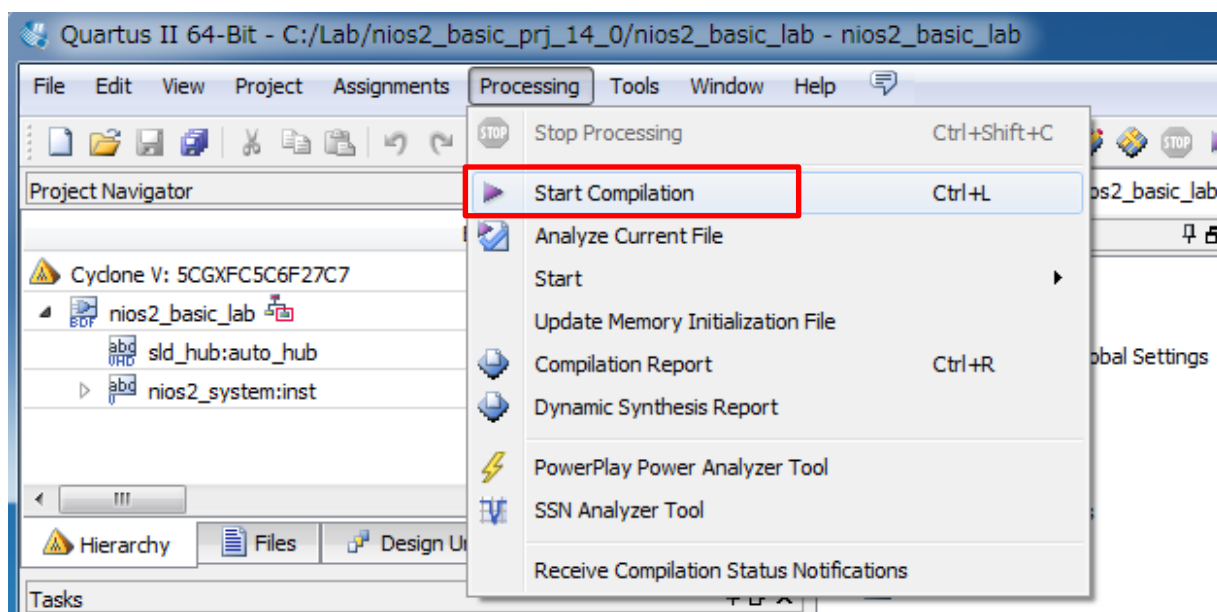


下図のように、Settings 画面が起動します。画面左側の Category 欄から Files を選択して、右側の File name 欄に、前述の meminit.qip ファイルを指定した後、[Add] ボタンをクリックして追加登録を行います。

下図のように、File name 枠内に追加されていることが確認できたら、[OK] ボタンをクリックします。



Processing メニューから Start Compilation を選択して、フル・コンパイルを行います。

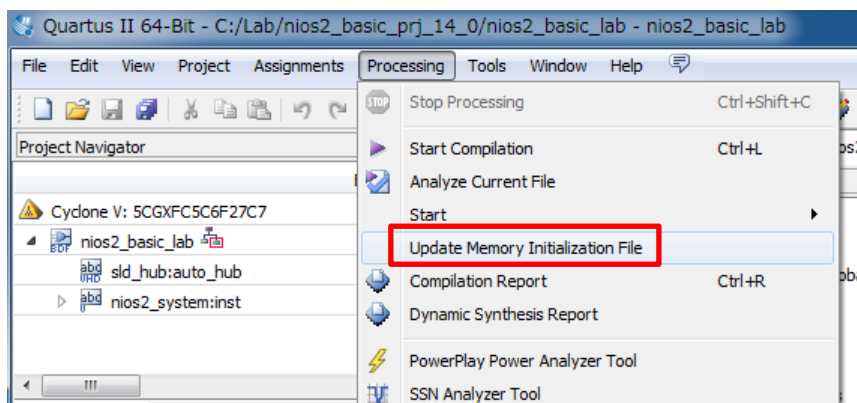


コンパイルが完了すると作成した HEX ファイルの内容を含んだ SOF/POF ファイルが生成されるので Quartus II Programmer で SOF ファイルを書き込み、実機上で正常に動作するか確認してください。本プログラムでは、SOF を書き込んだ時点で Nios II が制御する LED が規定の動作を行っていることを確認できました。正常に動作が確認できれば、FPGA 内のオンチップメモリに Nios II のプログラムが正常に初期化されていることとなります。

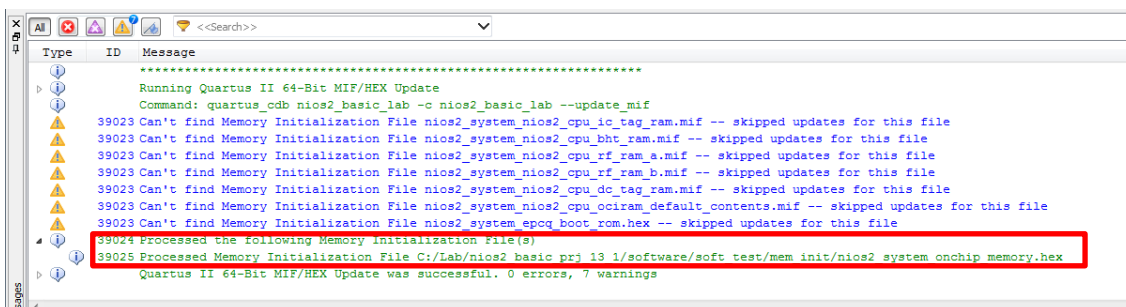
## 2-4. HEX ファイル更新時の Quartus II プロジェクトへの反映方法

Nios II の実行ファイルである ELF ファイルが変更された際には、2-1 章の手順により HEX ファイルを更新しますが、そのたびに Quartus II をフル・コンパイルするのは効率的ではありません。ここでは、HEX ファイルに対応するオンチップ・メモリの内容のみ更新する方法をご紹介します。

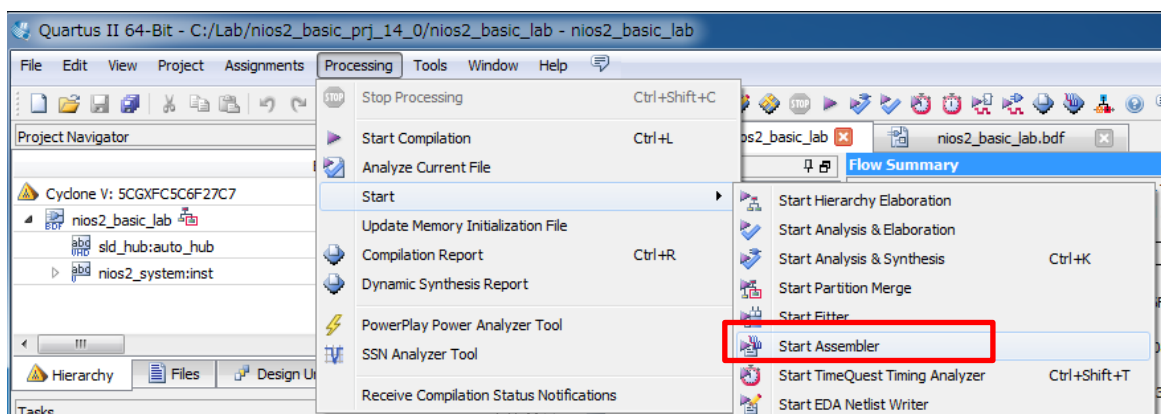
まず、Processing メニューから Update Memory Initialization File を選択します。この処理で更新された HEX ファイルを解析してファイルに問題があるかどうかをチェックします。この時点では、HEX ファイルの内容は SOF/POF ファイルには反映されないことに注意してください。



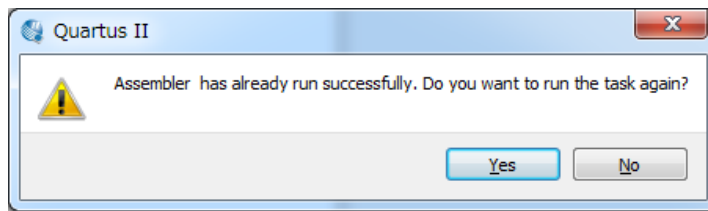
正常に解析されると、Quartus II のメッセージが以下のように表示されます。



次に、Processing メニューから Start ⇒ Start Assembler を選択します。この処理で SOF/POF ファイルに新しい HEX ファイルの内容が反映されます。フル・コンパイルに比べて、FPGA 内部のオンチップ・メモリのみ更新するので処理が短時間で済みます。



以下のメッセージが表示される場合があるので、Yes をクリックし処理を進めます。



この処理が完了したら、SOF/POF ファイルのタイム・スタンプが更新されていることを確認してください。また、改めて FPGA に書き込んで実機での動作を確認してください。

## 改版履歴

Revision	年月	概要
1	2014 年 7 月	新規作成
2	2014 年 9 月	ブート用オンチップ・メモリが ROM の場合を追加

### 免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。  
 株式会社アルティマ ホームページ: <http://www.altima.co.jp>      技術情報サイト EDISON: <https://www.altima.jp/members/index.cfm>  
 株式会社エルセナ ホームページ: <http://www.elsena.co.jp>      技術情報サイト ETS : <https://www.elsena.co.jp/elspear/members/index.cfm>
4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカー発行の英語版の資料もあわせてご利用ください。