

サイプレスはインフィニオン テクノロジーズになりました

この表紙に続く文書には「サイプレス」と表記されていますが、これは同社が最初にこの製品を開発したからです。新規および既存のお客様いずれに対しても、引き続きインフィニオンがラインアップの一部として当該製品をご提供いたします。

文書の内容の継続性

下記製品がインフィニオンの製品ラインアップの一部として提供されたとしても、それを理由としてこの文書に変更が加わることはありません。今後も適宜改訂は行いますが、変更があった場合は文書の履歴ページでお知らせします。

注文時の部品番号の継続性

インフィニオンは既存の部品番号を引き続きサポートします。ご注文の際は、データシート記載の注文部品番号をこれまで通りご利用下さい。

Intel Cyclone 10 GX FPGA を構成するサイプレス QSPI フラッシュの有効化

AN229767 は、Intel Quartus Prime Pro Generic Flash Programmer ツールを使用して、アクティブシリアル (AS) モードで Intel Cyclone 10 GX FPGA を構成するために Cypress S25HS512T QSPI フラッシュを有効にする方法を説明します。同じ方法は、それぞれのフラッシュデータシートに従ってフラッシュを構成することにより、他のサイプレス QSPI フラッシュファミリを有効にする場合にも適用されます。

目次

1 はじめに	1	3.2.4 消去フローテンプレートの変更	13
2 背景情報	1	3.3 .sof ファイルを変換して .jic ファイルを生成	15
3 手順	2	3.4 .jic 構成データをサイプレス QSPI フラッシュデバイスにプログラム	20
3.1 *.sof FPGA 構成ファイルを生成	2	4 結論	26
3.2 新しい Configuration Device を定義	2	5 参考資料	26
3.2.1 新しい構成デバイスを追加	2	改訂履歴	27
3.2.2 初期化フローテンプレートの変更	5	セールス, ソリューションおよび法律情報	28
3.2.3 プログラムフローテンプレートの変更	10		

1 はじめに

このドキュメントは、読者が Intel FPGA の開発に精通していることを前提に書かれています。

Quartus Prime 19.1 以降のバージョンの Intel の Generic Flash Programmer ツールは、サードパーティの QSPI フラッシュを FPGA 構成デバイスとして構成する方法を提供します。

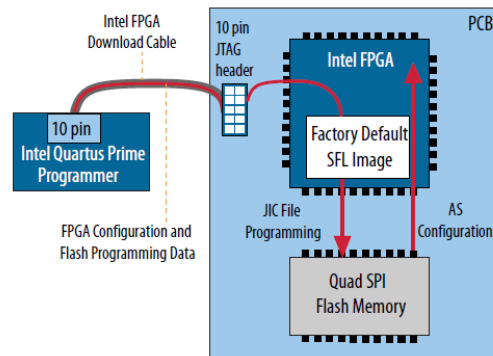
このアプリケーションノートでは、Generic Flash Programmer ツールを使用して FPGA 構成ビットストリームファイルをサイプレス S25HS512T QSPI フラッシュにロードする方法について説明します。その後、サイプレス QSPI フラッシュは、アクティブシリアル (AS) 構成を介して構成データをターゲット FPGA にロードします。

このアプリケーションノートで紹介されている手順は、Quartus Prime Pro 19.2 を使用して、クアド SPI インターフェースを備えた Cypress Semper™ Flash を搭載した Intel Cyclone 10 GX FPGA 開発キット (特に S25HS512T) で検証されました。この手順は、他のすべての Quartus Prime バージョン 19.1 以降に適用される予定です。この手順は、それぞれのデータシートに従ってフラッシュと FPGA デバイスを構成することにより、他のサイプレス QSPI フラッシュファミリと Intel Arria 10 FPGA デバイスにも適用する必要があります。

2 背景情報

Intel Generic Flash Programmer は、Intel Stratix 10、Intel Arria 10、および Intel Cyclone 10 GX FPGA デバイスをサポートします。ただし、Intel Stratix 10 デバイスの場合は、Secure Device Manager (SDM) ファームウェアがフラッシュプログラミングフローを制御します。これらのフローは変更できません。したがって、このアプリケーションノートで紹介されている手順は、Intel Stratix 10 FPGA デバイスには適用されません。

次の図に、Generic Flash Programmer 構成のデータフローを赤で示します。



3 手順

Quartus Prime **Programmer** の **Convert Programming File** ダイアログボックスから、Generic Flash Programmer の設定とコントロールにアクセスできます。

Convert Programming File ダイアログボックスを使用した Generic Flash Programming には、次の高レベルの手順が含まれます。

1. *.sof FPGA 構成ファイルを生成します。
2. 新しい Configuration Device を定義します。
3. *.sof ファイルを*.jic ファイルに変換します。このファイルは、構成データを格納するためにサイプレス QSPI フラッシュをプログラムするために使用されます。
4. Intel Quartus Prime Programmer を使用し、Intel FPGA ダウンロードケーブルを接続して、*.jic 構成データをサイプレス QSPI フラッシュデバイスにプログラムします。

3.1 *.sof FPGA 構成ファイルを生成

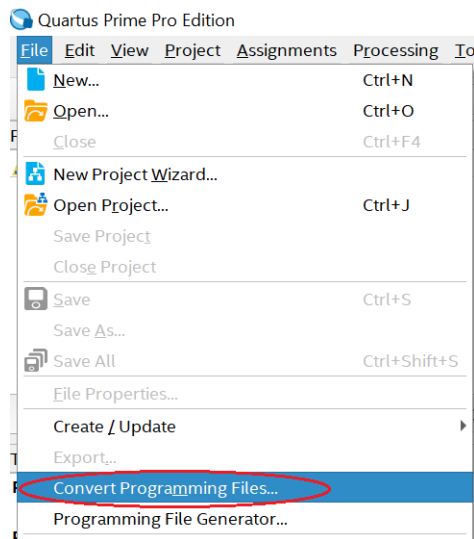
この手順は、このドキュメントではカバーされていません。*.sof ファイルがすでに生成されていることを前提としています。*.sof ファイルの生成については、関連する Intel ドキュメントを参照してください。

3.2 新しい Configuration Device を定義

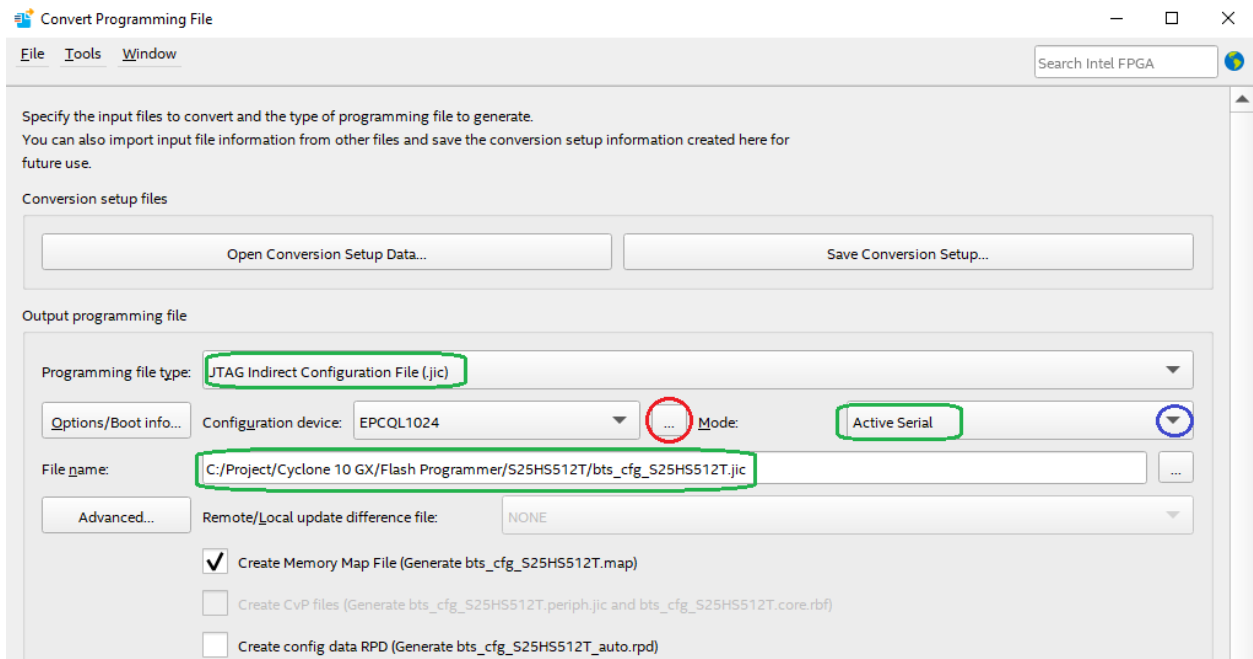
新しい Configuration Device を定義する際、Quartus Prime ソフトウェアは、設定のコレクションを、選択した **Device database directory** の場所にある*.xml ファイルに保存します。この構成は 1 回だけ実行する必要があります。新しい Configuration Device テンプレートは、Quartus Prime にリストされているデフォルトの構成デバイステンプレートと同じように再利用できます。

3.2.1 新しい構成デバイスを追加

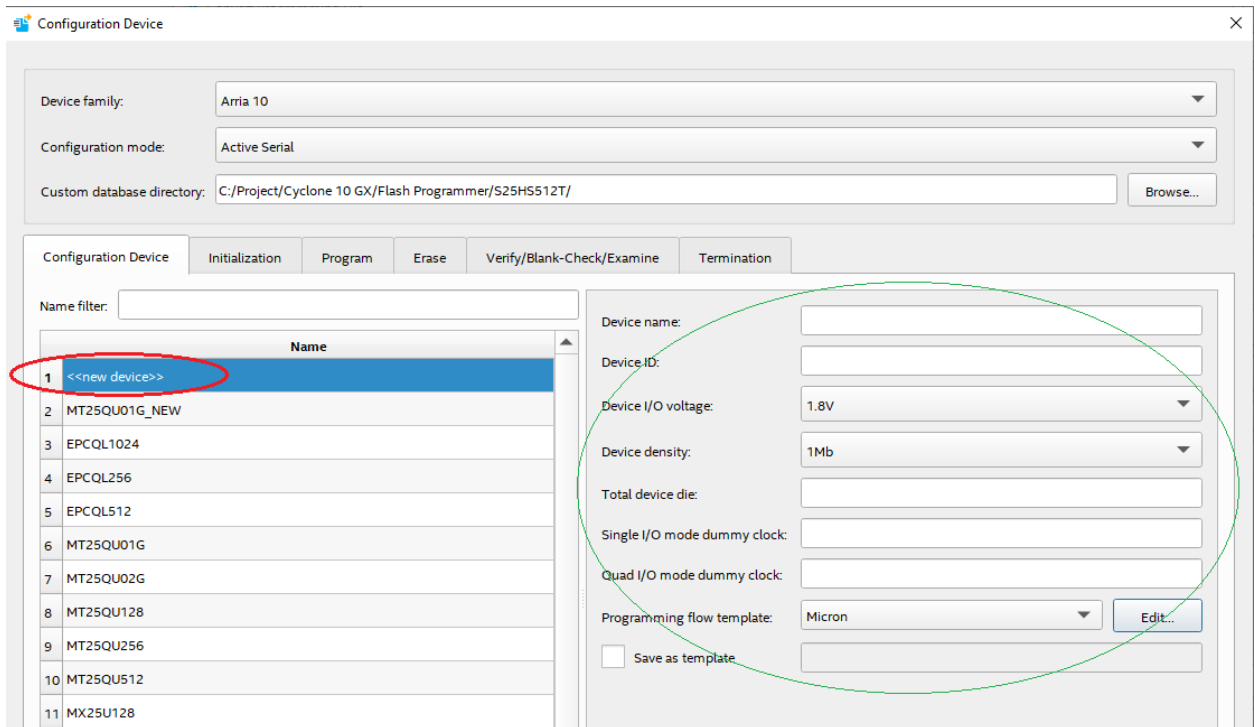
1. Quartus Prime Pro ソフトウェアを開き、**File > Convert Programming Files...** を選択してください。



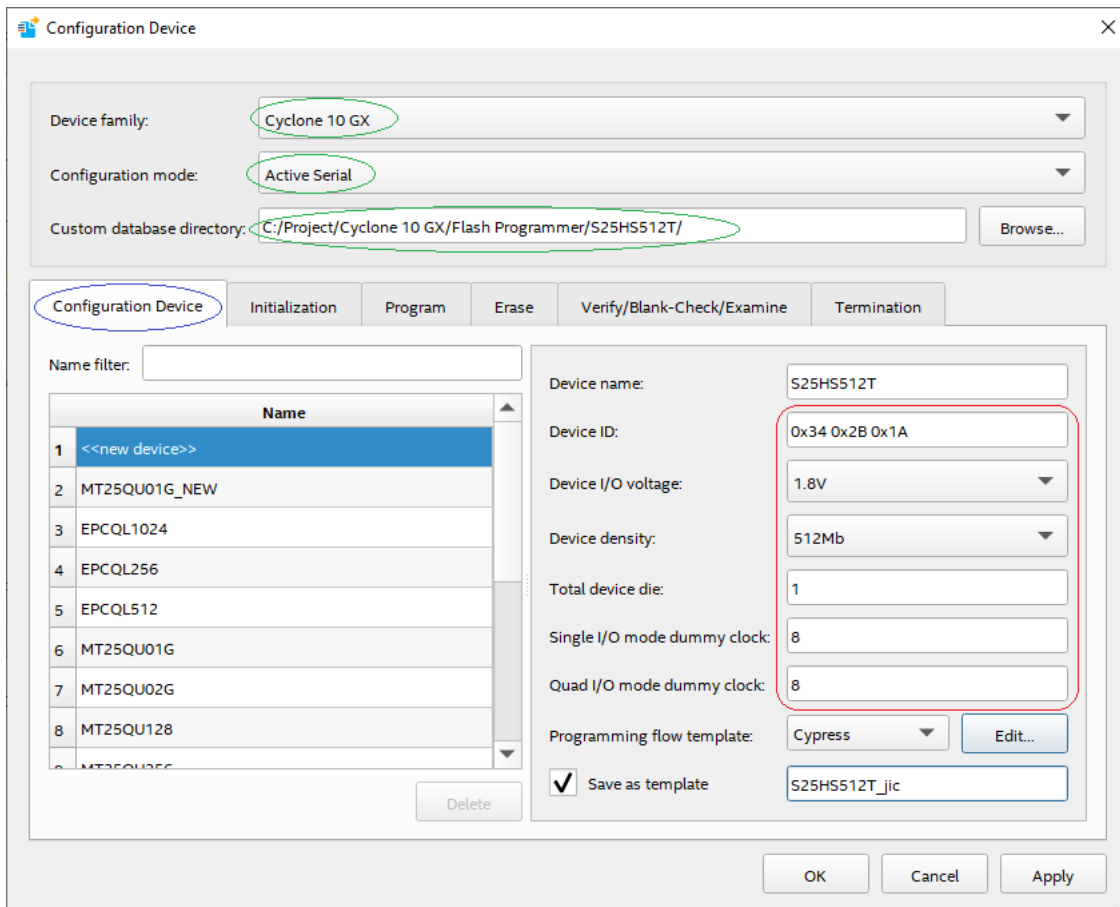
2. **Convert Programming File** 画面で以下を実行してください。



- a. プログラミングファイルタイプで*.jic ファイルタイプを選択してください。
注：Mode オプションは、*.jic が選択されている場合にのみ選択できます。
 - b. File name フィールドで、変換された*.jic ファイルを保存する場所を指定してください。
 - c. 青い円のプルダウンメニューをクリックし、**Active Serial** を選択してください。
 - d. 赤い丸をクリックしてください。
2. << new device >>をクリックしてください。緑の円で構成可能なフィールドが表示されます。



3. 以下の変更を行ってください。



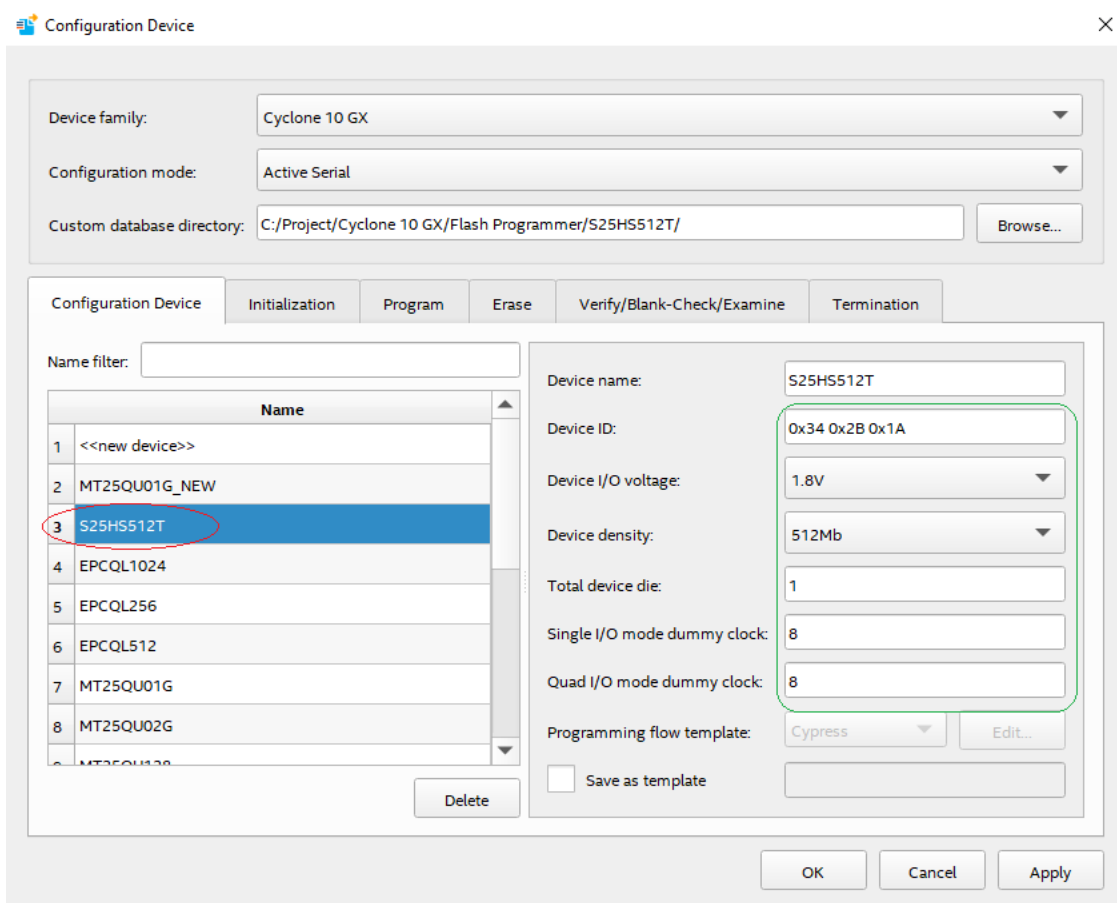
- FPGA デバイスに応じて FPGA の **Device family** と **Configuration mode** を設定してください。
- *.xml ファイルが保存される **Custom database directory** を指定してください。
- QSPI フラッシュデバイスのデータシートに従って、フラッシュデバイスのパラメータを設定してください。
 - デバイス ID
 - デバイスの I/O 電圧
 - デバイス密度
 - 総デバイスダイ
 - シングル I/O モードダミークロック
 - クワッド I/O モードダミークロック
 例えば、S25HS512T デバイスの場合、S25HS512T データシートからこれらのパラメータを入力してください。

4. **Apply** をクリックしてください。新しいデバイス "S25HS512T" が左側のリストに自動的に追加されます。

注：

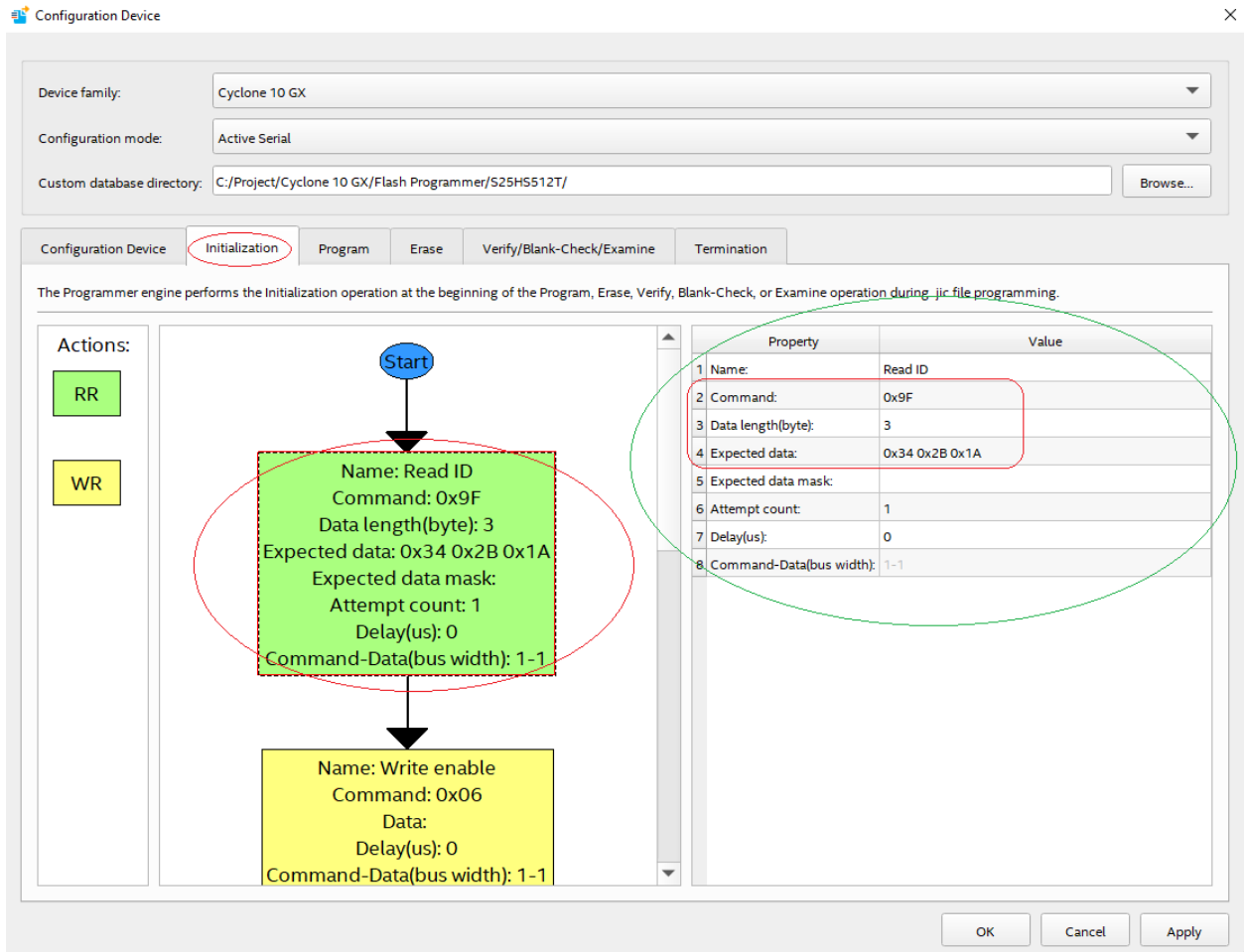
- フラッシュパラメータの設定では、**Device name** とテンプレート名 ("Save as template") を同じにすることはできません。それ以外の場合は、**Apply** ボタンをクリックすると、"テンプレート名 xxxxxx がデバイス名 xxxxxx と競合します" というエラーメッセージが表示されます。
- **Device ID** は、**0xAB 0xCD 0xEF** 形式の 3 バイトの 16 進データです。

3.2.2 初期化フローテンプレートの変更



1. 作成した Configuration Device (例として S25HS512T) を選択し、**Initialization** タブをクリックしてください。

Initialization タブには、フラッシュ初期化フローテンプレートが表示されます。テンプレートのアクションを追加、削除、または変更できます。これは、デフォルトのフローテンプレートに基づく S25HS512T デバイスの変更を示します。必要に応じて、独自の初期化フローテンプレートを定義できます。



The Programmer engine performs the Initialization operation at the beginning of the Program, Erase, Verify, Blank-Check, or Examine operation during jic file programming.

Actions:

- RR
- WR

Flowchart:

```

    graph TD
      Start((Start)) --> ReadID[Name: Read ID  
Command: 0x9F  
Data length(byte): 3  
Expected data: 0x34 0x2B 0x1A  
Expected data mask:  
Attempt count: 1  
Delay(us): 0  
Command-Data(bus width): 1-1]
      ReadID --> WriteEnable[Name: Write enable  
Command: 0x06  
Data:  
Delay(us): 0  
Command-Data(bus width): 1-1]
    
```

Property	Value
1 Name:	Read ID
2 Command:	0x9F
3 Data length(byte):	3
4 Expected data:	0x34 0x2B 0x1A
5 Expected data mask:	
6 Attempt count:	1
7 Delay(us):	0
8 Command-Data(bus width):	1-1

Buttons: OK, Cancel, Apply

2. ウィンドウの左側で **Read ID 動作** をクリックしてください。フラッシュデバイスのデータシートに従って、右側 (緑色の円) のプロパティを構成します。
3. 動作プロパティを設定したら、**Apply** をクリックしてください。

デフォルトの "Write enable" 動作は、S25HS512T の動作と同じです。この例では、このフィールドは変更されていません。

Configuration Device

Device family: Cyclone 10 GX

Configuration mode: Active Serial

Custom database directory: C:/Project/Cyclone 10 GX/Flash Programmer/S25HS512T/

Configuration Device | Initialization | Program | Erase | Verify/Blank-Check/Examine | Termination

The Programmer engine performs the Initialization operation at the beginning of the Program, Erase, Verify, Blank-Check, or Examine operation during .jic file programming.

Actions:

- RR
- WR

Flowchart:

```

    graph TD
      A[Command-Data(bus width): 1-1] --> B[Name: Write enable  
Command: 0x06  
Data:  
Delay(us): 0  
Command-Data(bus width): 1-1]
      B --> C[Name: Write registers  
Command: 0x01  
Data: 0x00 0x82  
Delay(us): 0  
Command-Data(bus width): 1-1]
      C --> D[Name: Read status register]
    
```

Property Value

Property	Value
1 Name:	Write registers
2 Command:	0x01
3 Data:	0x00 0x00 0x88 0x08
4 Delay(us):	0
5 Command-Data(bus width):	1-1

OK Cancel Apply

4. 左側のウィンドウを下にスクロールして、**Write registers** 動作をクリックしてください。プロパティを編集して、フラッシュレジスタを設定してください。

この例では、S25HS512T を 4 バイトのアドレスモードに構成し、セクターアーキテクチャを均一な 256KB に設定してください。

S25HS512T 書き込みレジスタコマンドシーケンスは次のとおりです。書き込みレジスタコマンド (0x01) + STR1 (ステータスレジスタ-1) 値+ CFR1 (コンフィギュレーションレジスタ-1) 値+ CFR2 (コンフィギュレーションレジスタ-2) 値+ CFR3 (コンフィギュレーションレジスタ-3) 値。

STR1 のデフォルト値は 0x00 です。CFR1 のデフォルトは 0x00 です。

CFR2 のデフォルト値は 0x08 です。4 バイトアドレスモードの場合は、CFR2 [7]を'1'に設定してください。CFR2 の新しい値は 0x88 です。

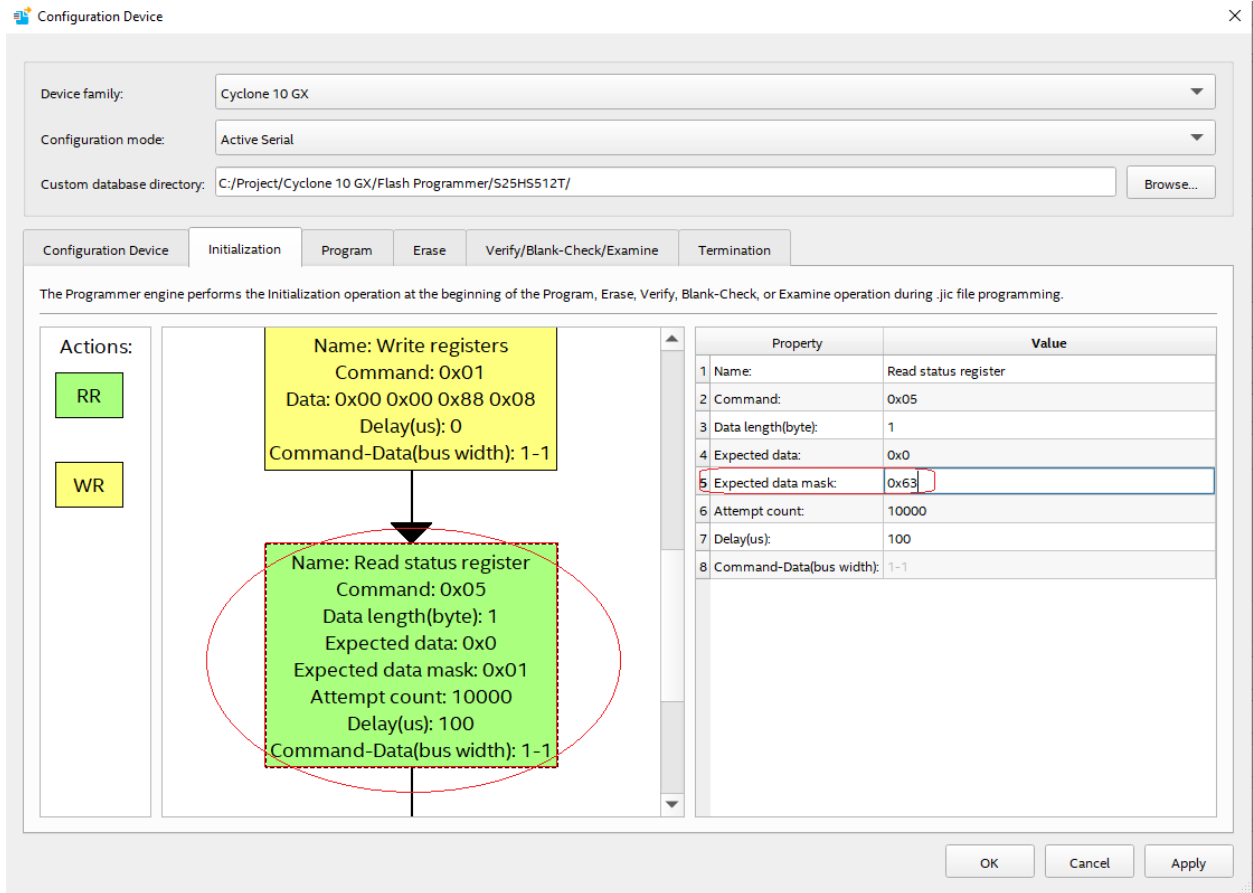
CFR3 のデフォルト値は 0x00 (ハイブリッドセクターアーキテクチャ 4KB + 256KB セクター) です。均一な 256KB セクターの場合は、CFR3 [3]を'1'に設定してください。CFR3 の新しい値は 0x08 です。

したがって、0x01 コマンドに続くデータは (右側の赤い円の上のスクリーンショットに示されているように) 0x00 0x00 0x88 0x08 です。レジスタの詳細については、S25HS512T のデータシートを参照してください。

注 :

- Intel Cyclone 10 GX および Arria10 デバイスには、4 バイトのアドレス指定フラッシュメモリが必要です。フラッシュは 4 バイトアドレス指定をサポートする必要があり、4 バイトアドレスモードに設定する必要があります。

- デフォルトの消去フローテンプレートは、均一な 256KB セクターアーキテクチャ用です。消去フローを単純化するためには、初期化フローでフラッシュを均一な 256KB セクターアーキテクチャとして設定します。そうでない場合、消去フローはフラッシュアレイを完全に消去することはできません。これはエラーメッセージなしで発生する可能性があることに注意してください。



The screenshot shows the 'Configuration Device' window with the 'Initialization' tab selected. The 'Actions' list on the left includes 'RR' and 'WR'. The main workspace shows a flowchart with two actions:

- Name: Write registers**
Command: 0x01
Data: 0x00 0x00 0x88 0x08
Delay(us): 0
Command-Data(bus width): 1-1
- Name: Read status register**
Command: 0x05
Data length(byte): 1
Expected data: 0x0
Expected data mask: 0x01
Attempt count: 10000
Delay(us): 100
Command-Data(bus width): 1-1

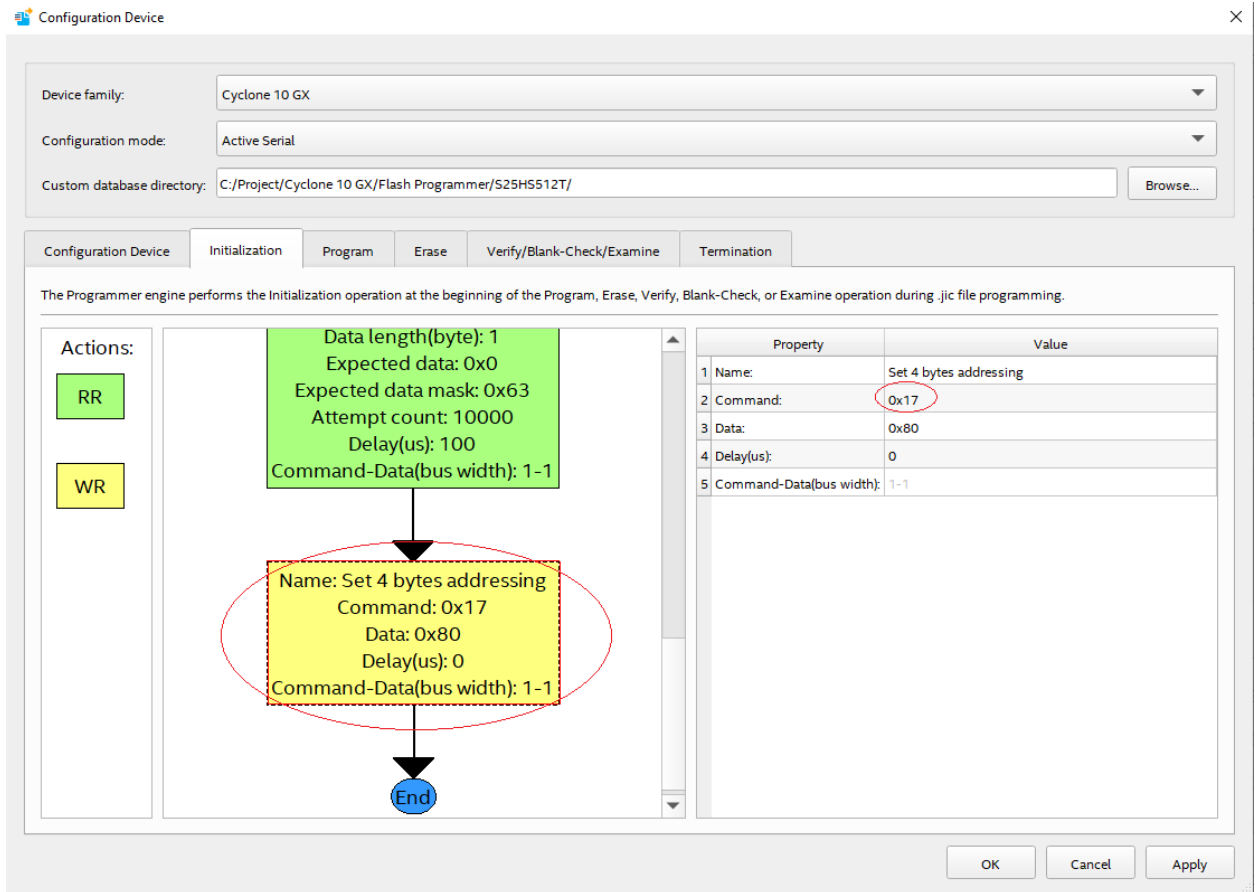
The 'Read status register' action is circled in red. To the right, a table shows the properties of the selected action:

Property	Value
1 Name:	Read status register
2 Command:	0x05
3 Data length(byte):	1
4 Expected data:	0x0
5 Expected data mask:	0x63
6 Attempt count:	10000
7 Delay(us):	100
8 Command-Data(bus width):	1-1

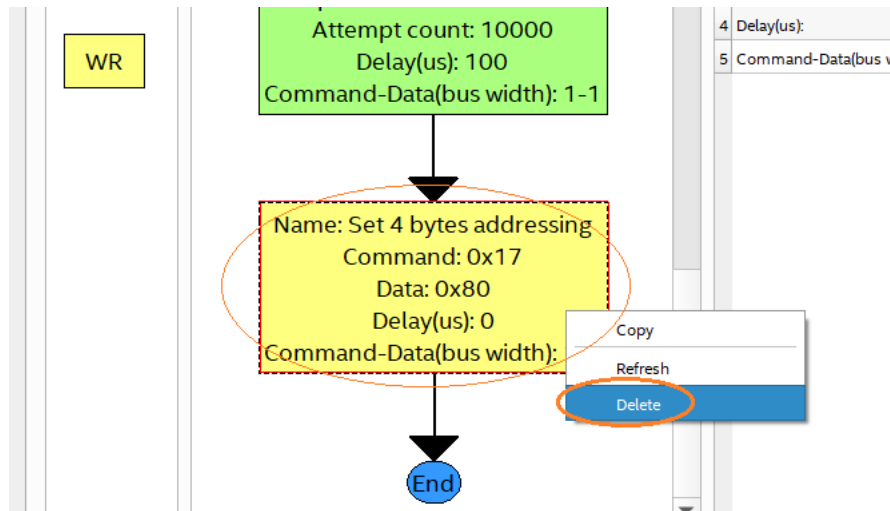
5. Write register 動作の後、書き込み操作のステータスを確認するために、ステータスレジスタを読み出してください。

STR1 [6]は P_ERR、STR1 [5]は E_ERR、STR1 [1]は WEL、STR1 [0]は RDYBSY です。プログラム操作が正常に完了すると、これらの 4 ビットはすべて 0 になります。他のビットはドントケアです。したがって、期待されるデータマスクは 0110 0011b (0x63) です。マスクされた期待データは 0x0 です。ステータスレジスタビットの定義については、S25HS512T データシートを参照してください。

この動作 (Read Status Register) は、フラッシュアレイプログラム/消去操作のステータスを確認する場合と同じです。プログラムおよび消去フローテンプレートでも使用できます。

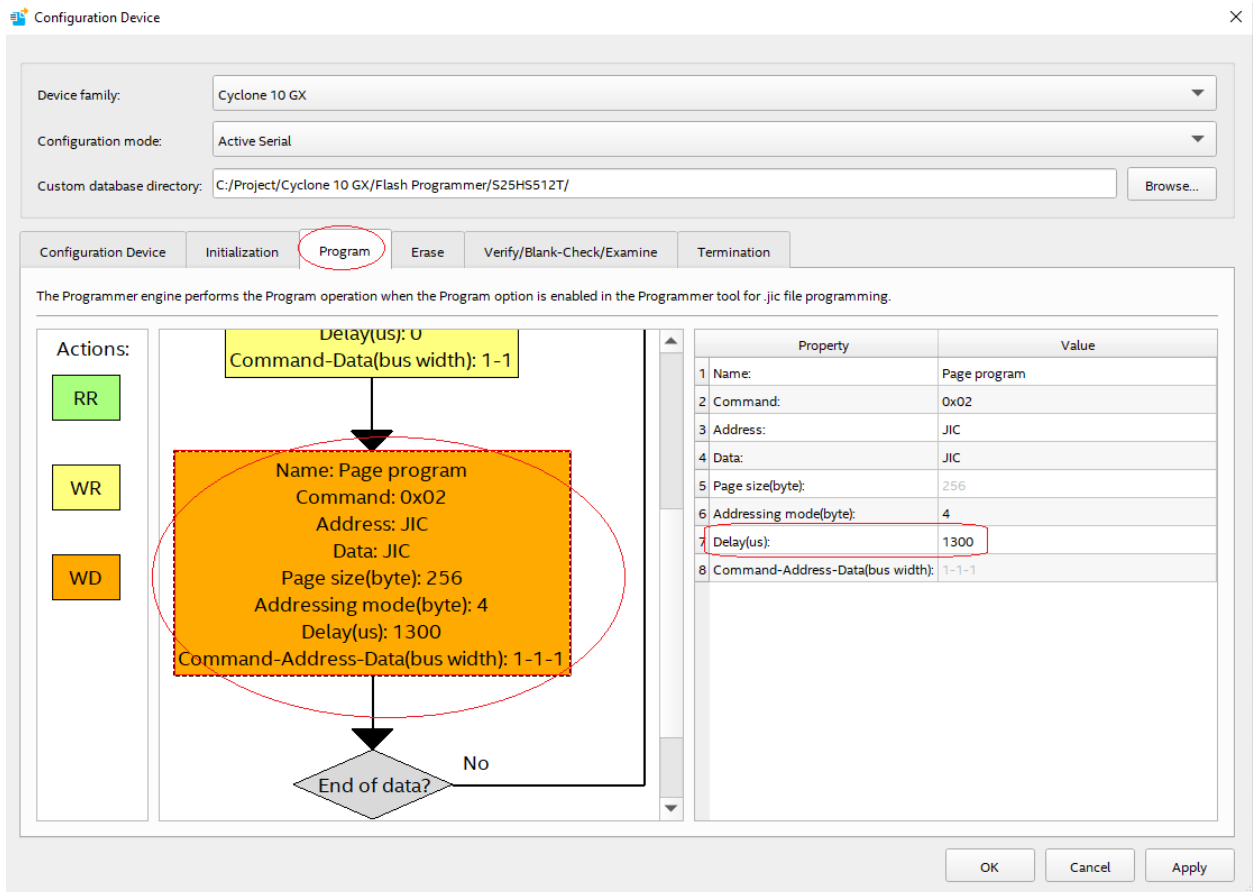


6. **Set 4 byte addressing** 動作を右クリックし、**Delete** をクリックしてから、**Apply** をクリックして変更を保存してください。これは、このコマンドがサイプレスフラッシュデバイスに適用できないためです。サイプレスフラッシュ (S25HS512T) の場合、前の書き込みレジスタ動作ですでに行われた 4 バイトアドレス指定を設定します。



3.2.3 プログラムフローテンプレートの変更

1. Program タブをクリックし、Page program 動作をクリックしてください。



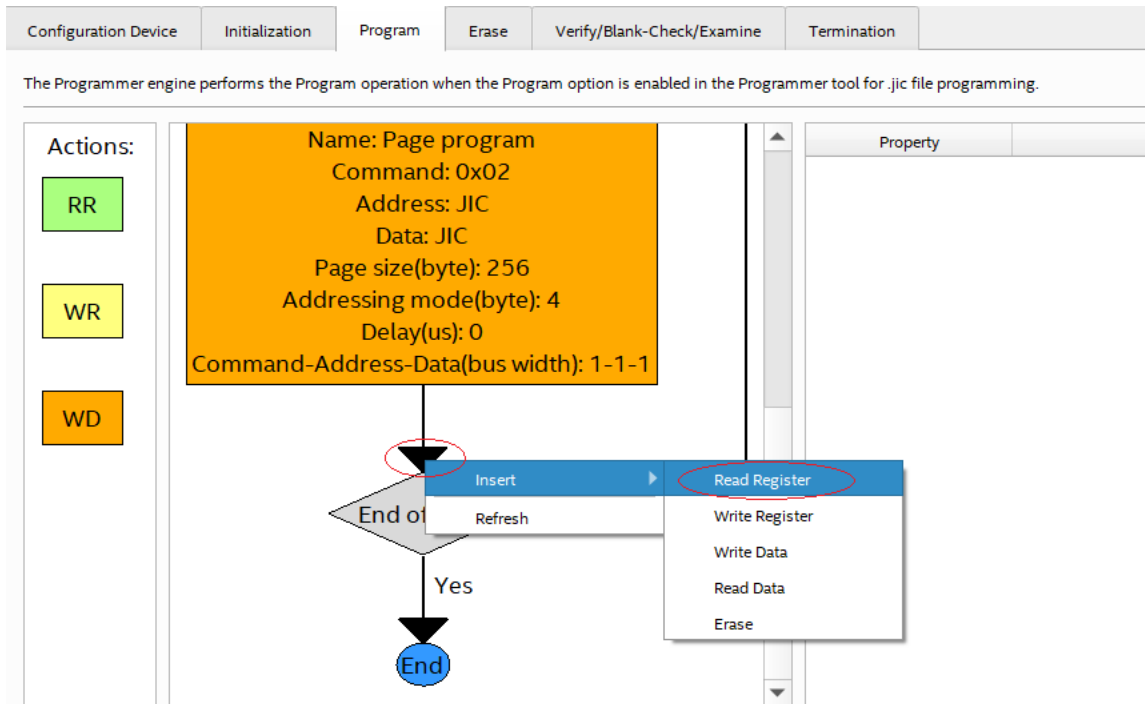
The Programmer engine performs the Program operation when the Program option is enabled in the Programmer tool for .jic file programming.

Property	Value
1 Name:	Page program
2 Command:	0x02
3 Address:	JIC
4 Data:	JIC
5 Page size(byte):	256
6 Addressing mode(byte):	4
7 Delay(us):	1300
8 Command-Address-Data(bus width):	1-1-1

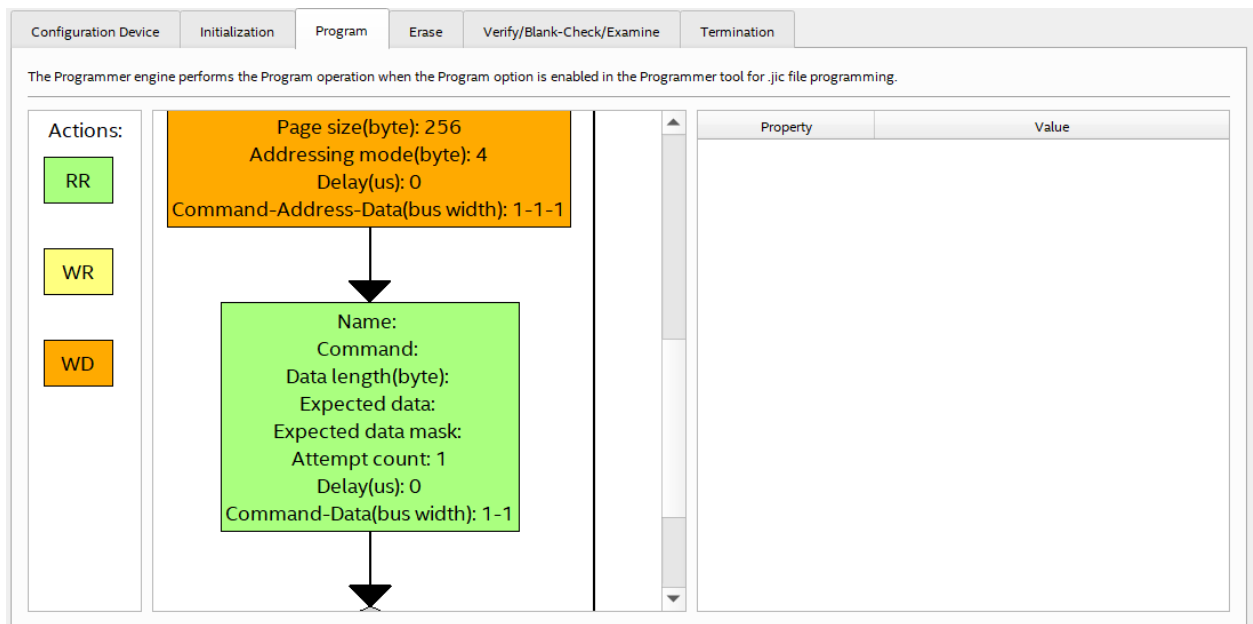
2. フラッシュデータシートで指定されている最大ページプログラム時間に従って、Delay の値を変更してください。例えば、S25HS512T の場合、最大ページプログラム時間は 2175 μ s です。

もう 1 つのオプションは、遅延値を '0' に設定し、ステータスレジスタの読み出しアクションを追加して、次のようにページプログラム操作のステータスを確認することです。

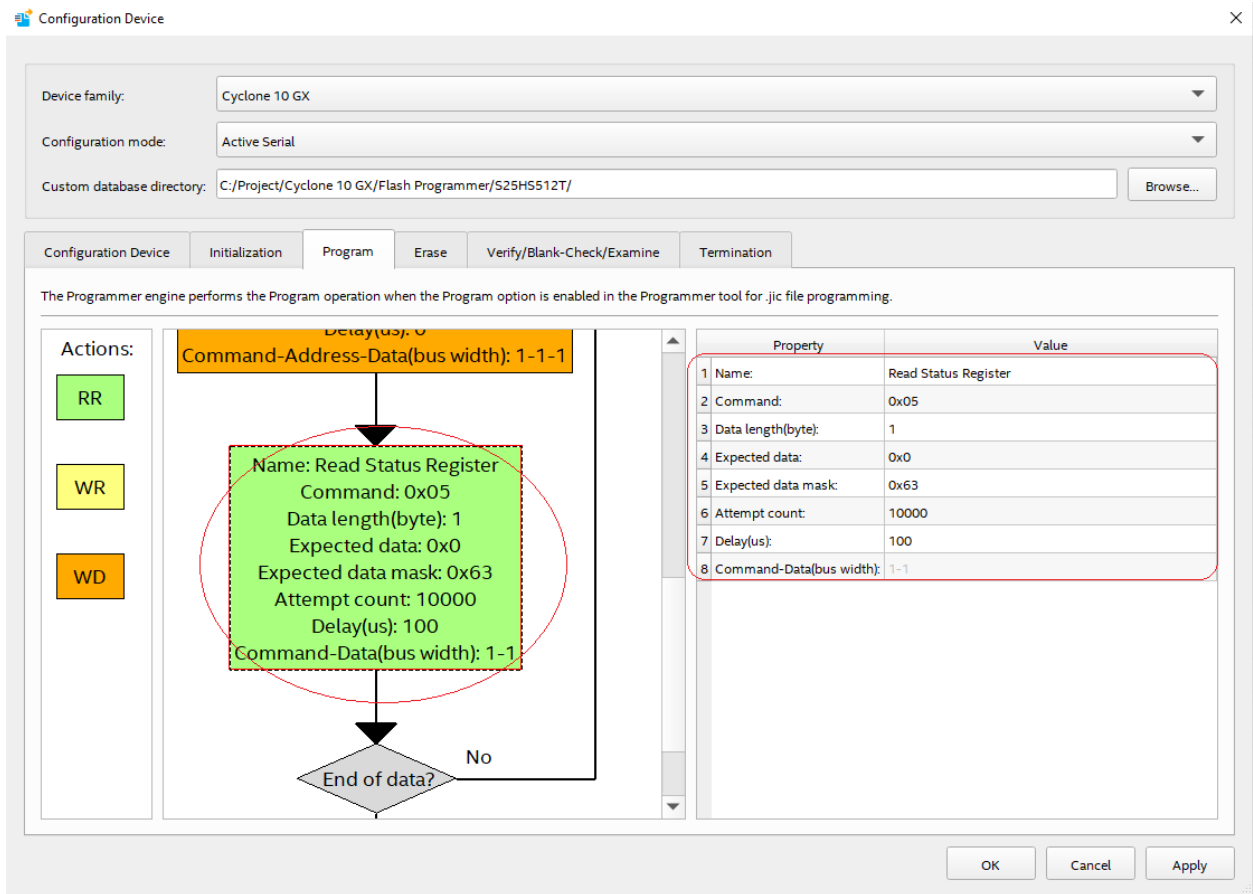
ページプログラムのサイズは、デフォルトのプログラムフローテンプレートと S25HS512T : 256 バイトで同じです。S25HS512T は、512 バイトのより大きなページプログラムサイズもサポートします。CFR3x [4] = 1 を設定することにより、この値を 512 バイトに構成できます。S25HS512T に 512 バイトのページプログラムサイズを使用するためには、初期化フローテンプレートの Write register 動作で値を設定してください。



- a. 矢印を右クリックし、**Insert > Read Register** を選択して、Read Register 動作をフローに追加してください。



- b. 追加した動作をクリックして、そのプロパティを構成してください。



The screenshot shows the Configuration Device software interface. At the top, the device family is set to "Cyclone 10 GX" and the configuration mode is "Active Serial". The custom database directory is "C:/Project/Cyclone 10 GX/Flash Programmer/S25HS512T/". Below this, there are tabs for "Configuration Device", "Initialization", "Program", "Erase", "Verify/Blank-Check/Examine", and "Termination". The "Program" tab is selected, and a message states: "The Programmer engine performs the Program operation when the Program option is enabled in the Programmer tool for .jic file programming."

The main area displays a flowchart for the "Read Status Register" action. The flowchart starts with a box labeled "Command-Address-Data(bus width): 1-1-1" with a "Delay(us): 0" above it. An arrow points down to a green box containing the following properties: "Name: Read Status Register", "Command: 0x05", "Data length(byte): 1", "Expected data: 0x0", "Expected data mask: 0x63", "Attempt count: 10000", "Delay(us): 100", and "Command-Data(bus width): 1-1". An arrow points down from this box to a diamond-shaped decision box labeled "End of data?". A "No" path loops back from the diamond to the top of the green box.

On the left side, there are three action buttons: "RR" (highlighted in green), "WR", and "WD". On the right side, there is a table with the following properties and values:

Property	Value
1 Name:	Read Status Register
2 Command:	0x05
3 Data length(byte):	1
4 Expected data:	0x0
5 Expected data mask:	0x63
6 Attempt count:	10000
7 Delay(us):	100
8 Command-Data(bus width):	1-1

At the bottom right, there are "OK", "Cancel", and "Apply" buttons.

- c. フラッシュデータシートに従ってプロパティを構成してください。これは、初期化テンプレートに示される構成と同じです。

時間 (Delay x Attempt count) は、フラッシュデータシートで指定されている最大ページプログラム時間よりも大きくする必要があります。例えば、S25HS512T の場合、最大ページプログラム時間は 2175 us です。したがって、Delay x Attempt count 値は 2175 us より大きくなければなりません。Read Register アクションが期待されるマスクされたデータを返した後、フローは最大試行回数に達することなく次のステップに進むため、より大きな値を入力できます。

- d. プロパティを構成したら、**Apply** をクリックしてください。

3.2.4 消去フローテンプレートの変更

Configuration Device

Device family: Cyclone 10 GX
 Configuration mode: Active Serial
 Custom database directory: C:/Project/Cyclone 10 GX/Flash Programmer/S25HS512T/

Configuration Device | Initialization | Program | Erase | Verify/Blank-Check/Examine | Termination

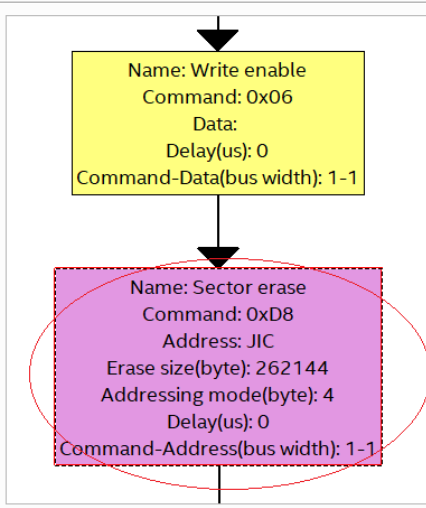
The Programmer engine performs the Erase operation when the Erase option is enabled in the Programmer tool for .jic file programming. The Programmer engine uses all erase actions specified between "Erase data" and "End of data?" to decide the optimal way of erasing the data.

Actions:

RR

WR

E



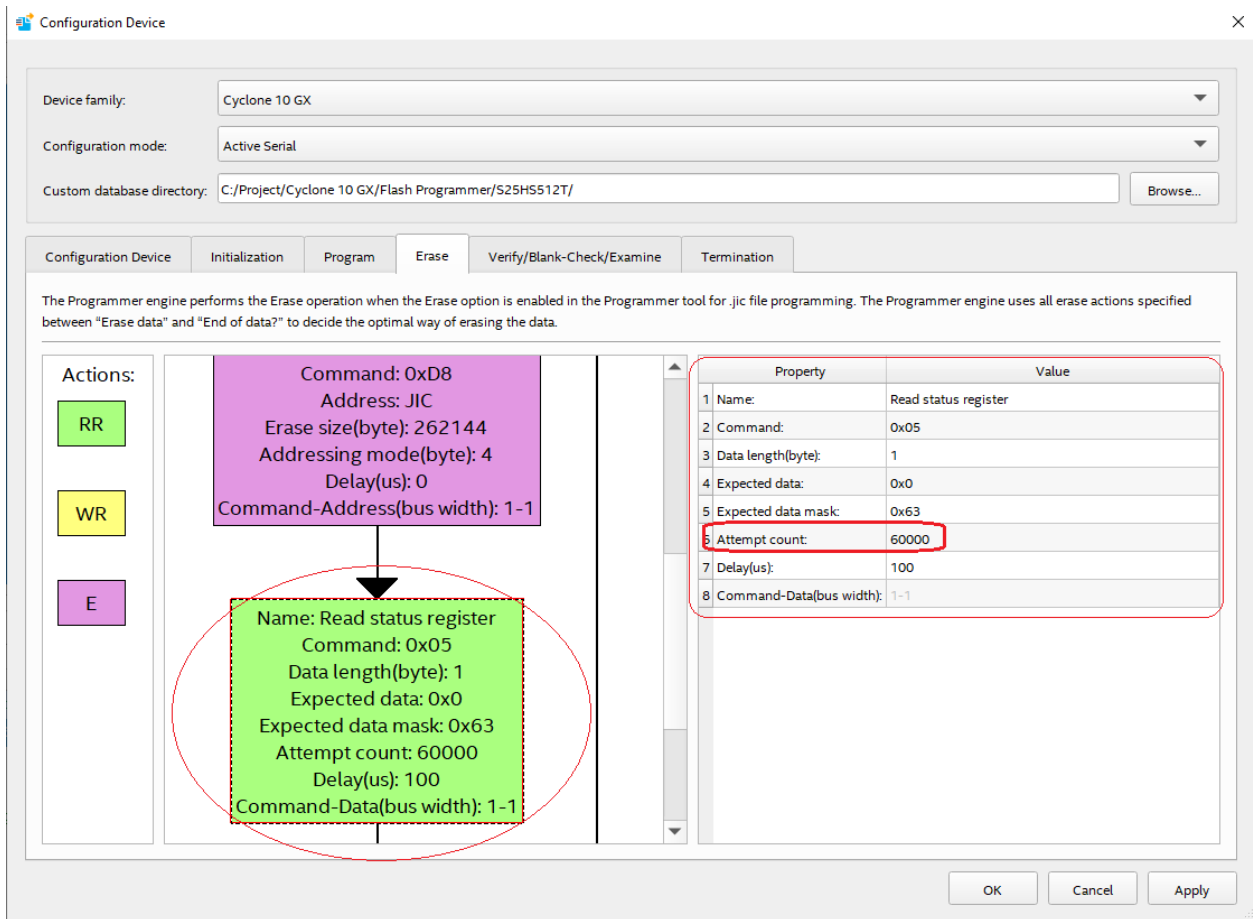
```

graph TD
    A["Name: Write enable  
Command: 0x06  
Data:  
Delay(us): 0  
Command-Data(bus width): 1-1"] --> B["Name: Sector erase  
Command: 0xD8  
Address: JIC  
Erase size(byte): 262144  
Addressing mode(byte): 4  
Delay(us): 0  
Command-Address(bus width): 1-1"]
          
```

Property	Value
1 Name:	Sector erase
2 Command:	0xD8
3 Address:	JIC
4 Erase size(byte):	262144
5 Addressing mode(byte):	4
6 Delay(us):	0
7 Command-Address(bus width):	1-1

OK Cancel Apply

デフォルトの“Write enable”、および“Sector erase”動作は、S25HS512T の均一な 256KB セクター消去操作と互換性があります。S25HS512T は、初期化フローテンプレートのレジスタ Write register 動作で 256KB セクターアーキテクチャを統一するようにすでに構成されているため、これらの値は変更しないでください。



The Programmer engine performs the Erase operation when the Erase option is enabled in the Programmer tool for .jic file programming. The Programmer engine uses all erase actions specified between "Erase data" and "End of data?" to decide the optimal way of erasing the data.

Property	Value
1 Name:	Read status register
2 Command:	0x05
3 Data length(byte):	1
4 Expected data:	0x0
5 Expected data mask:	0x63
6 Attempt count:	60000
7 Delay(us):	100
8 Command-Data(bus width):	1-1

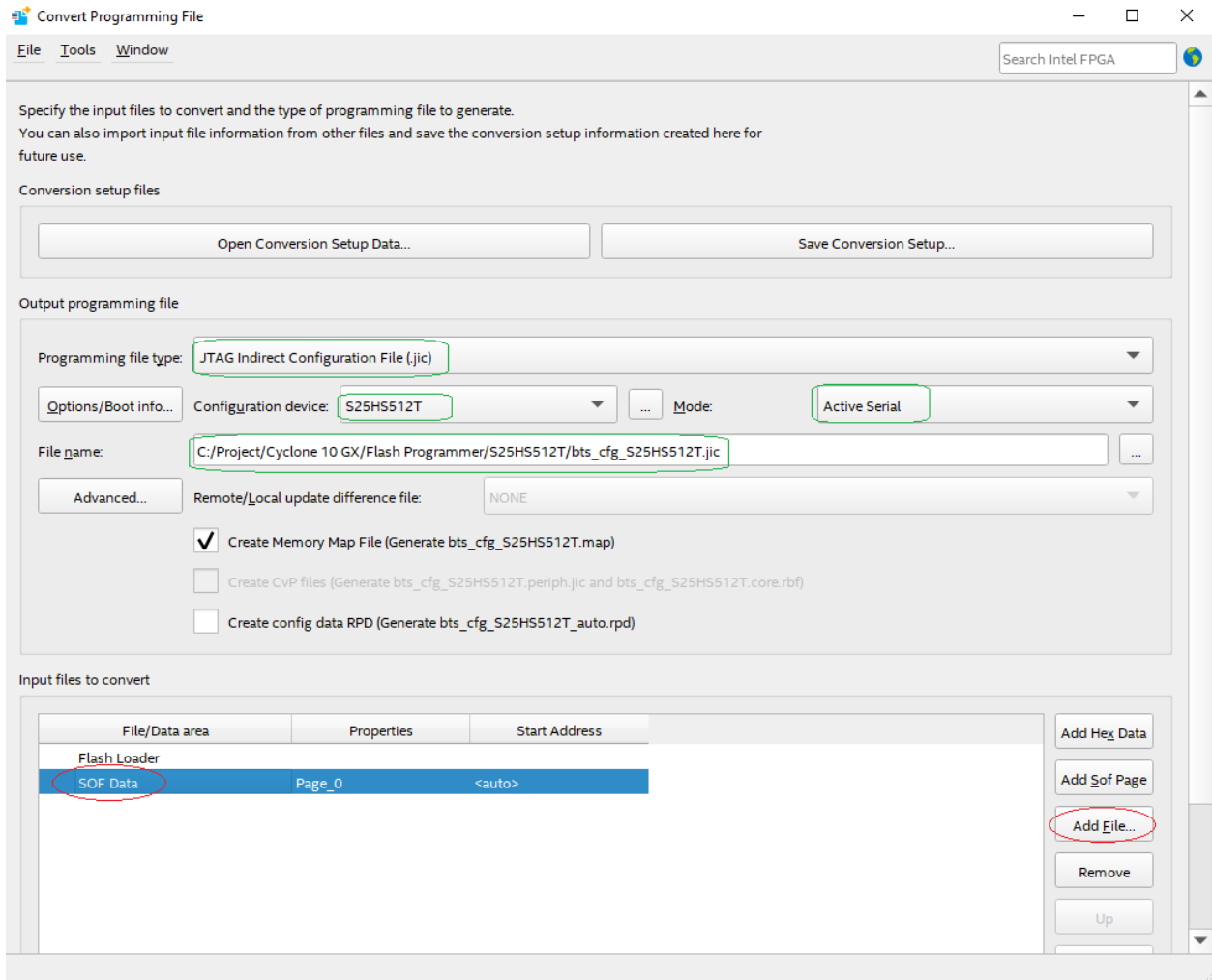
1. **Read status register** 動作をクリックし、Command、Expected data、data mask などの S25HS12T のプロパティを更新してください。この動作は、初期化フローおよびプログラムフローの **Read status register** 動作に似ています。

注：(Delay x Attempt count) は、フラッシュデータシートで指定されている最大 256KB セクター消去時間よりも大きくする必要があります。S25HS12T の場合、256KB EnduraFlex™ アーキテクチャが有効になっていると 5869 ミリ秒になります。詳細については、S25HS12T のデータシートを参照してください。

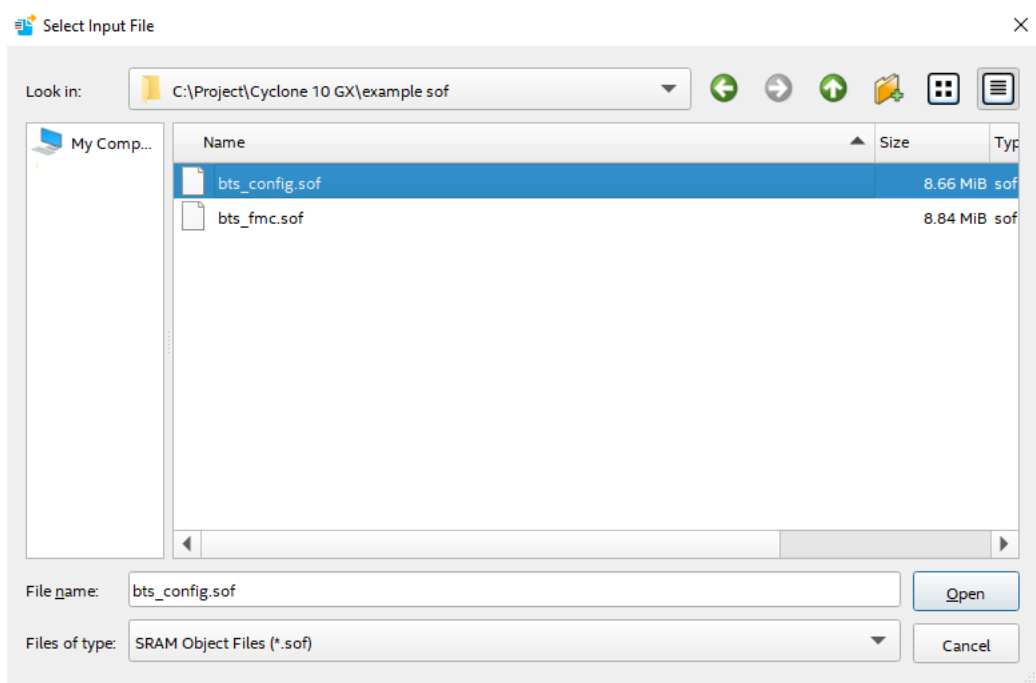
2. **OK** をクリックして変更を保存してください。

これらの動作を構成した後、.sof ファイルをフラッシュデバイスにロードできる .jic ファイルに変換するように構成デバイスを設定してください。

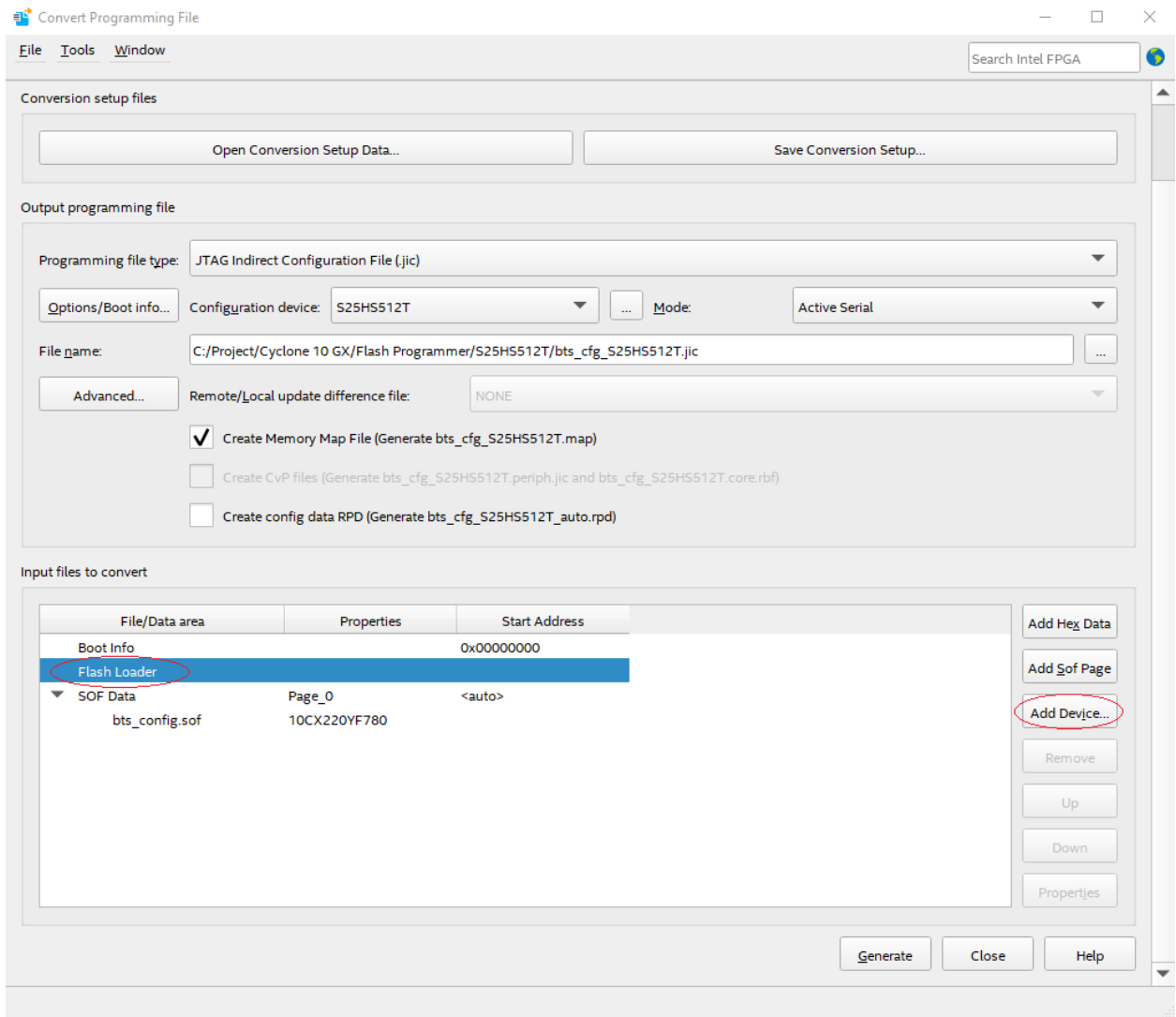
3.3 .sof ファイルを変換して.jic ファイルを生成



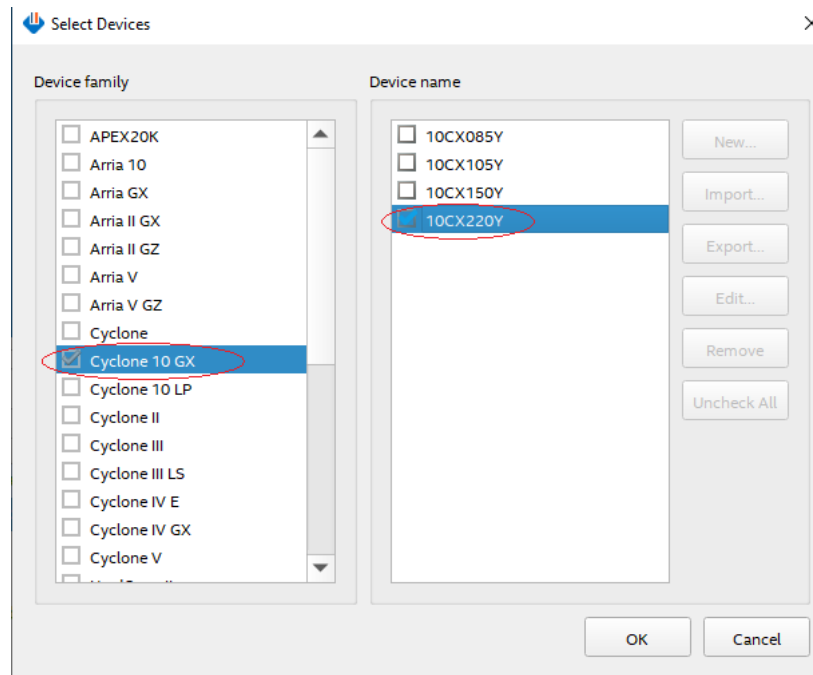
1. **Programming file type** を JTAG Indirect Configuration File (.jic) に設定してください。
2. Configuration device として S25HS512T (前の手順で作成) を選択してください。
3. Mode を **Active Serial** として選択してください。
4. 生成する .jic ファイル名と生成した .jic ファイルを保存する場所を指定してください。
5. **SOF Data** を選択し、**Add File...** をクリックしてください。
6. 変換する .sof ファイルを選択してください。



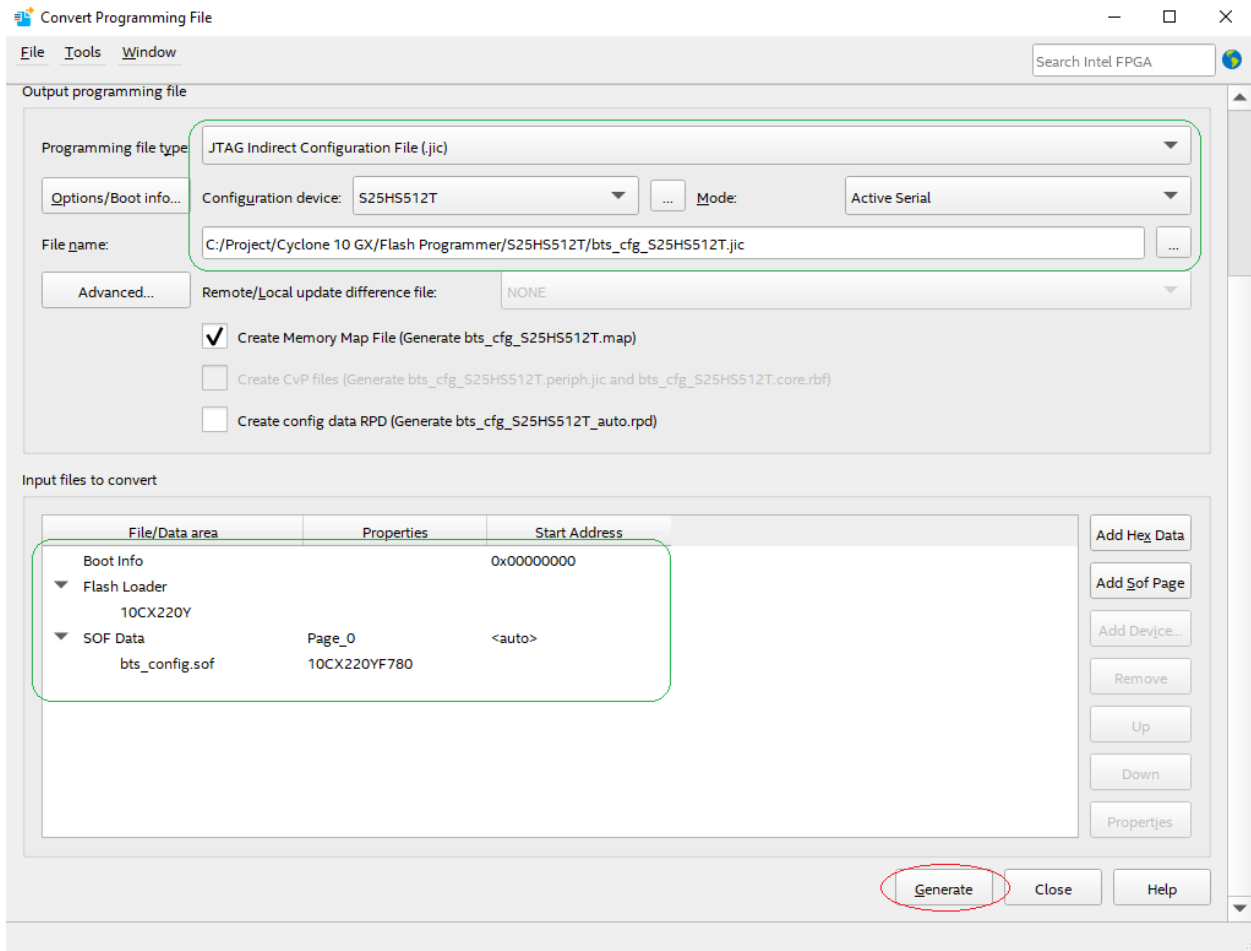
7. .sof ファイルを追加した後、**Flash Loader** を選択し、**Add Device...** をクリックしてください。



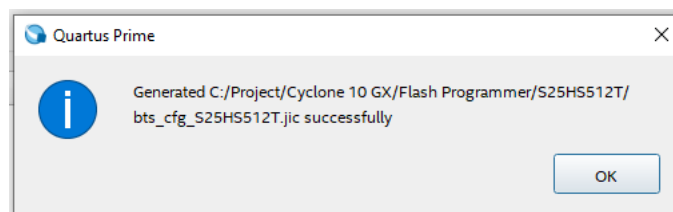
8. 使用する FPGA デバイスに応じてデバイスを選択し、**OK** をクリックしてください。



このアプリケーションノートでは、Cyclone10 GX 10CX220Y FPGA デバイスを搭載した Cyclone10 GX Dev キットで手順を確認します。

9. **Generate** をクリックしてください。

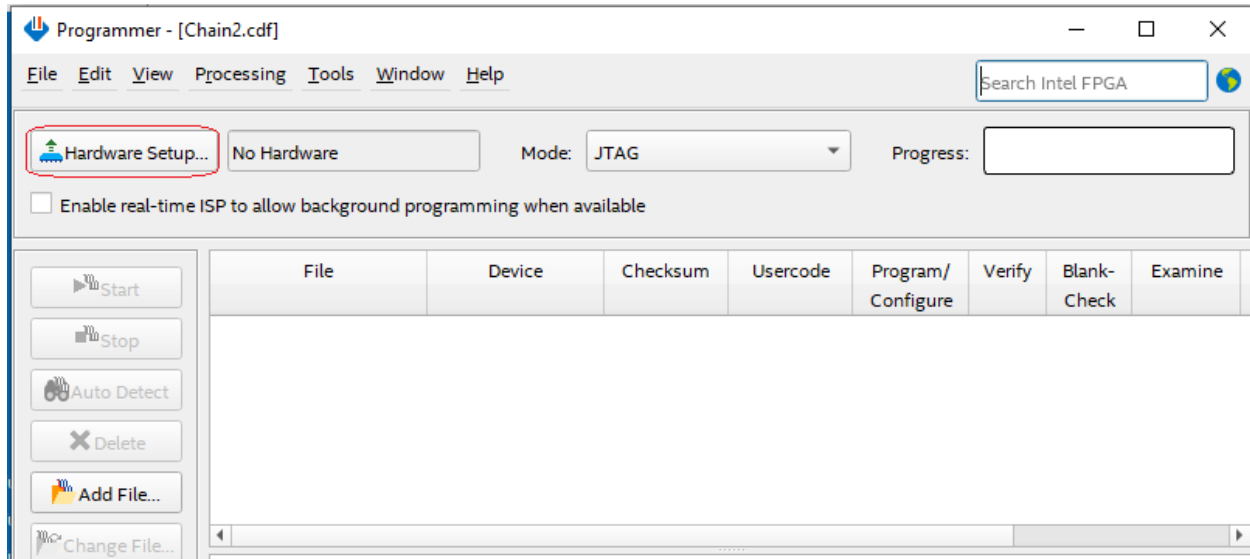
しばらくすると、.jic ファイルが正常に生成されたことを示すポップアップウィンドウが表示されます。



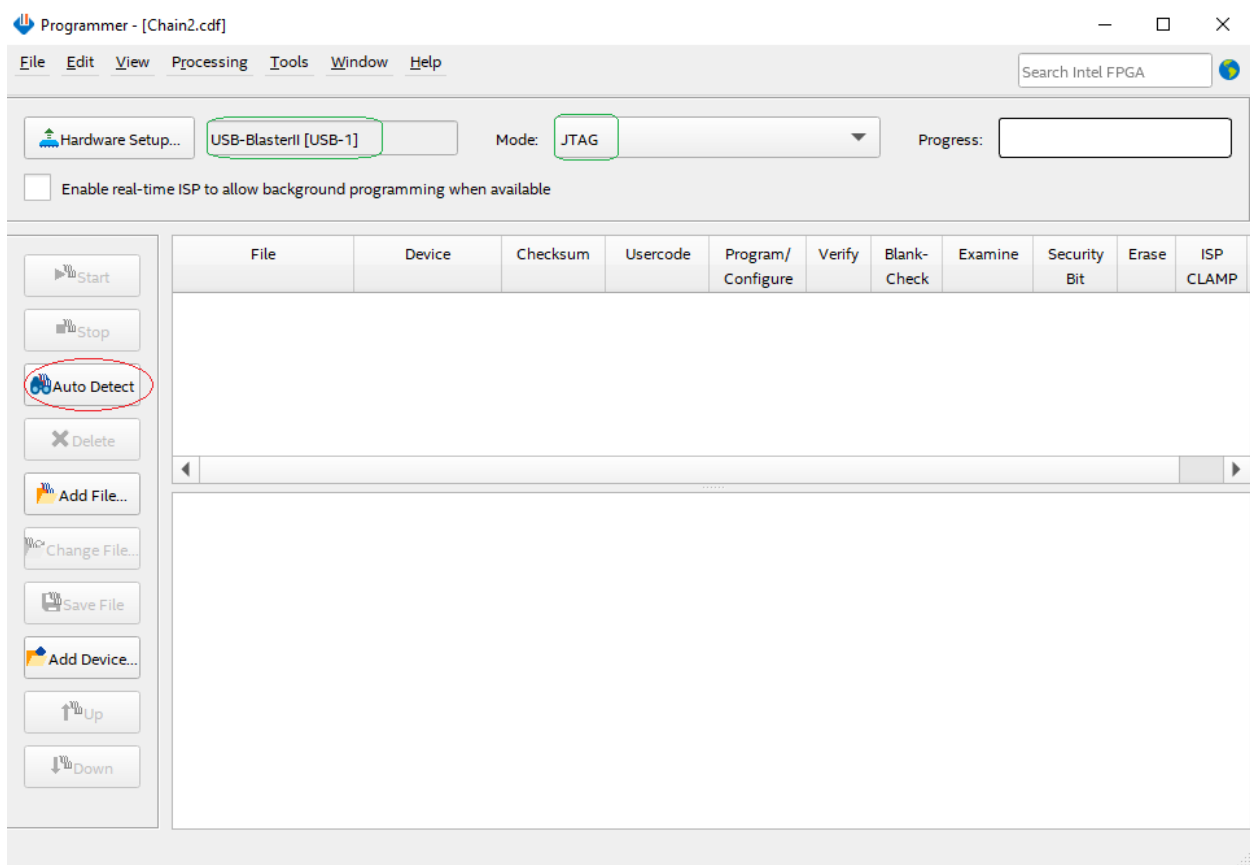
これで、生成された.jic ファイルを構成フラッシュ (この例では S25HS512T) にプログラムできます。

3.4 .jic 構成データをサイプレス QSPI フラッシュデバイスにプログラム

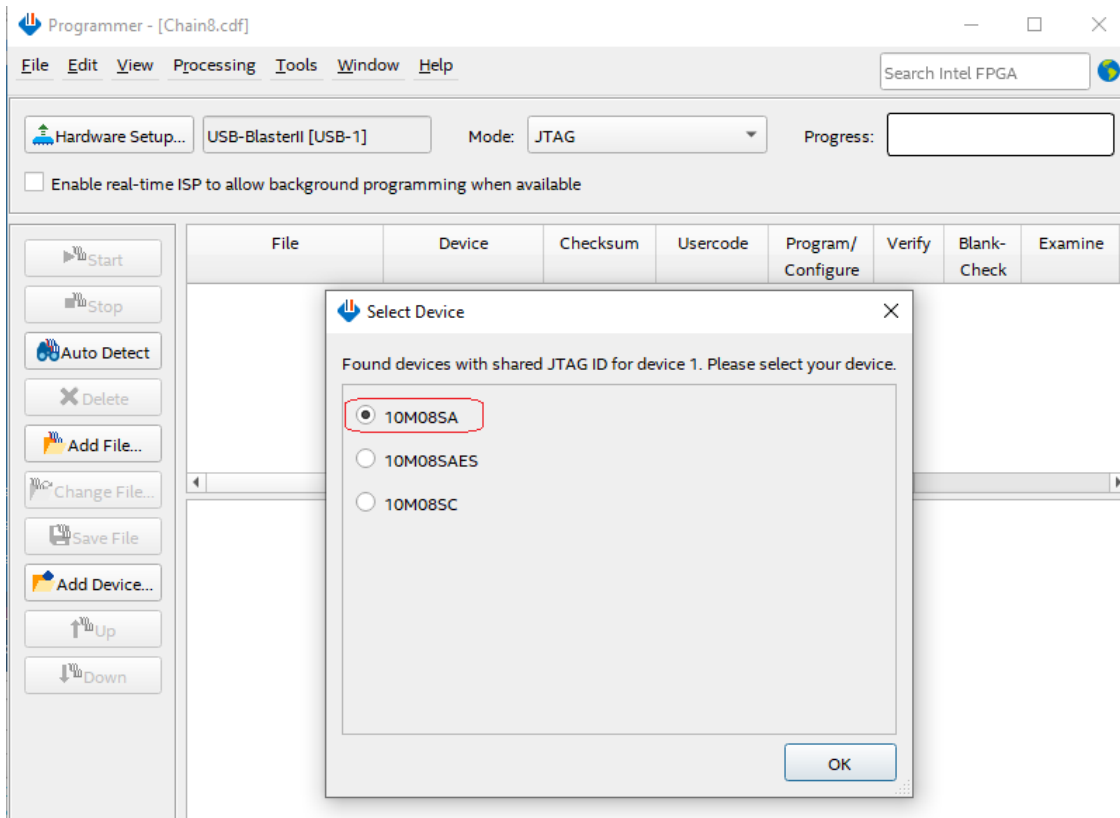
1. Intel FPGA JTAG USB ダウンロードケーブルを介して Intel FPGA ボードを PC に接続し、FPGA ボードの電源を入れてください。
2. Quartus Prime Pro から、**Tools > Programmer** を選択してください。
3. **Hardware Setup...** をクリックしてください。



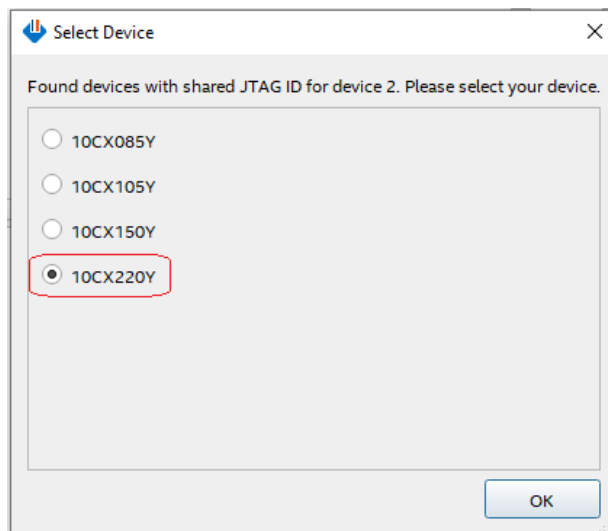
4. **USB-Blaster...** をクリックしてください。



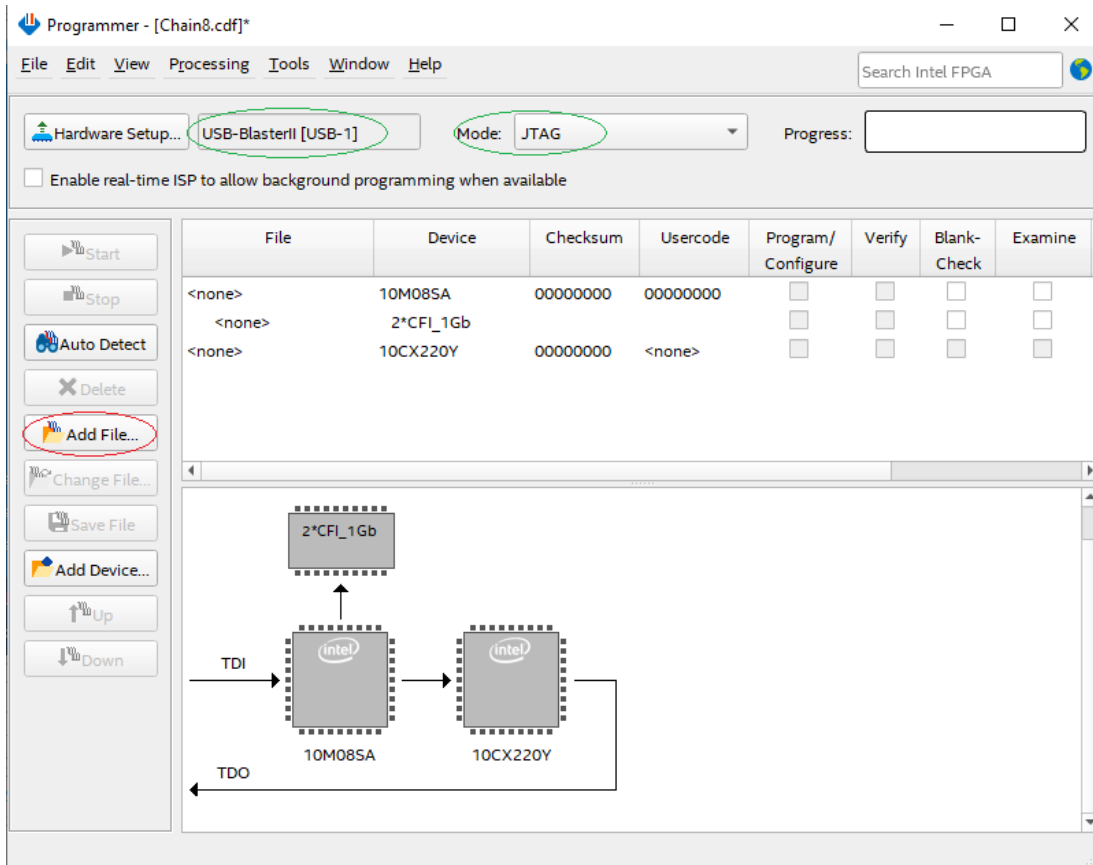
5. **Auto Detect** をクリックしてください。
6. ポップアップウィンドウで **10M08SA** を選択してください。これは、手順が検証された Cyclone10 GX 開発ボード上の MAX10 FPGA デバイスです。



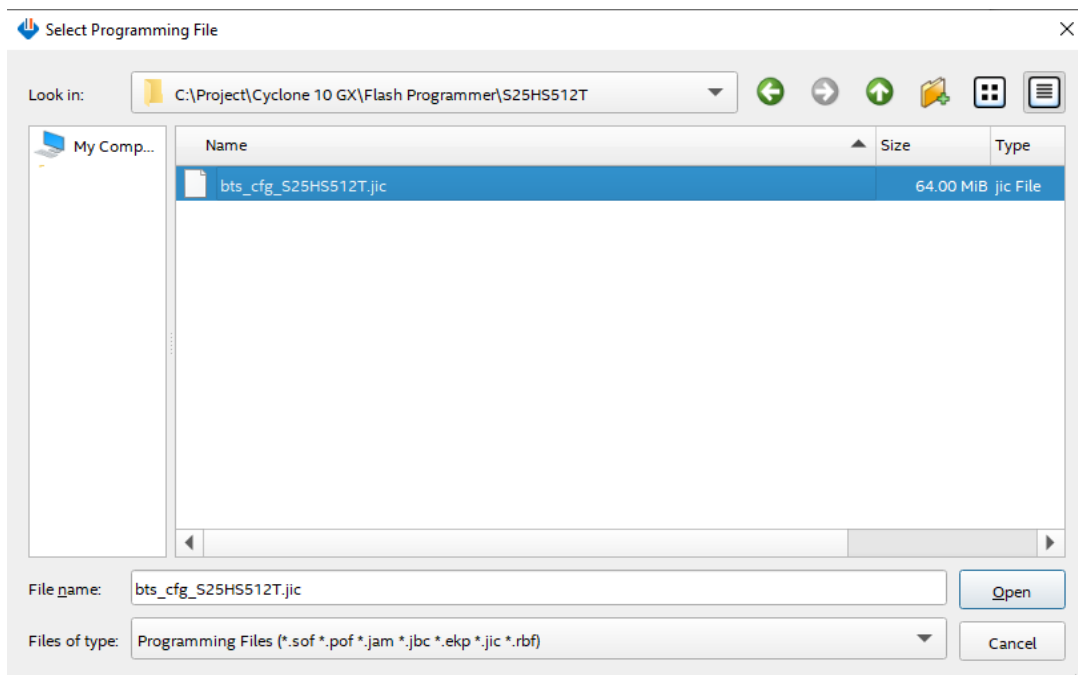
7. ポップアップウィンドウで **10CX220Y** を選択してください。これは、開発ボード上の Cyclone 10 GX FPGA デバイスです。



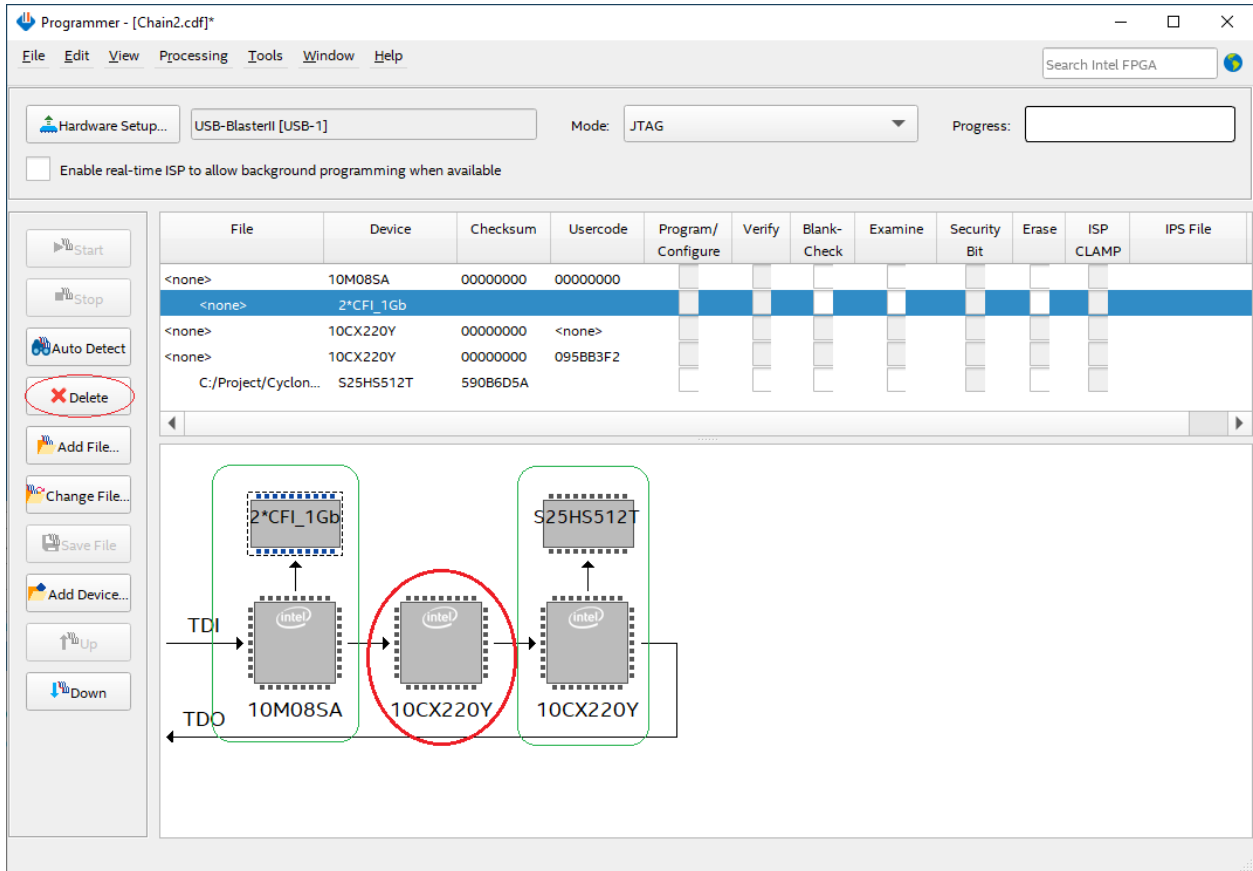
8. **Add File...**をクリックしてください。



9. 前の手順で生成された.jic ファイルを開いてください。



10. .jic ファイルを追加した後に追加のデバイスが表示された場合は、それを選択して **Delete** をクリックしてください。



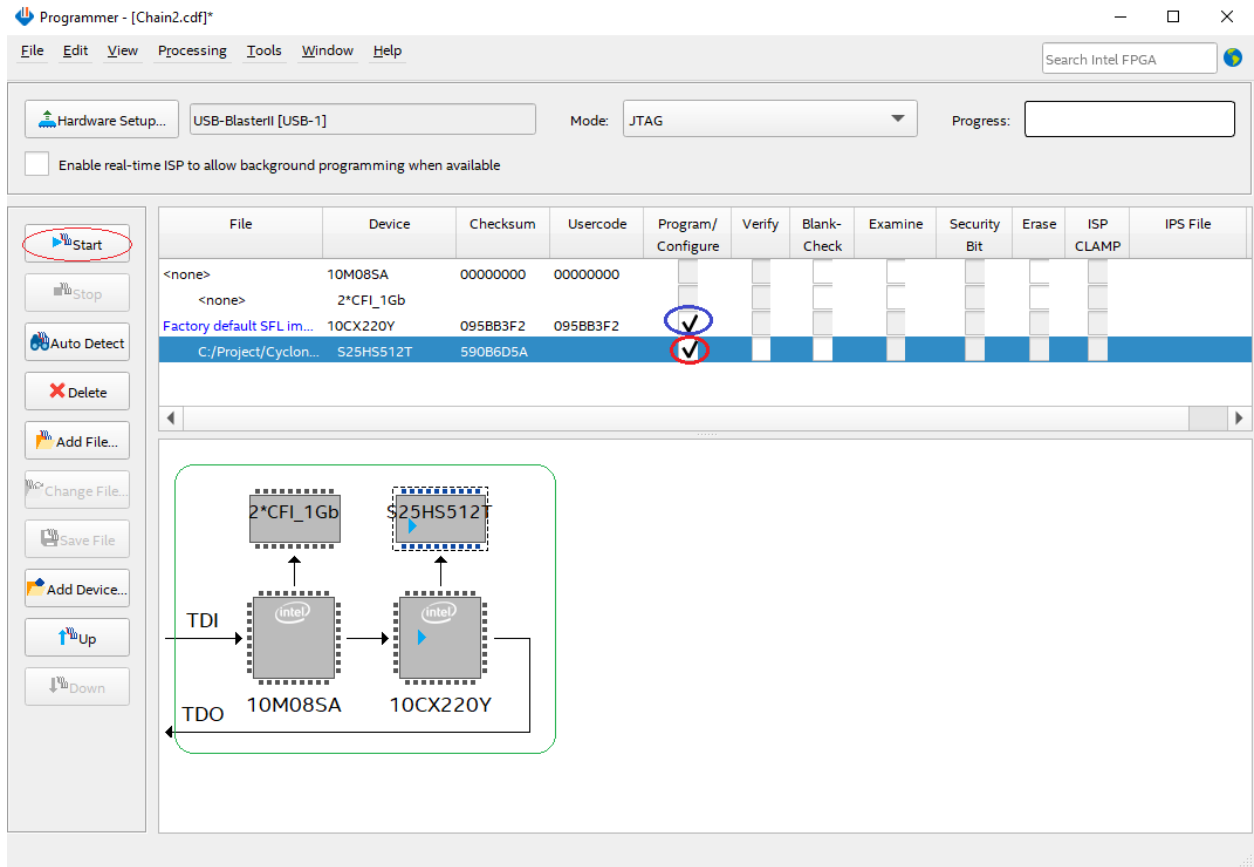
The screenshot shows the Cypress Programmer software interface. The top menu includes File, Edit, View, Processing, Tools, Window, and Help. Below the menu, there are options for Hardware Setup (USB-BlasterII [USB-1]), Mode (JTAG), and a Progress bar. A checkbox for "Enable real-time ISP to allow background programming when available" is present.

The main area contains a table with the following columns: File, Device, Checksum, Usercode, Program/Configure, Verify, Blank-Check, Examine, Security Bit, Erase, ISP CLAMP, and IPS File. The table lists several devices, with the second row (2*CFI_1Gb) highlighted in blue.

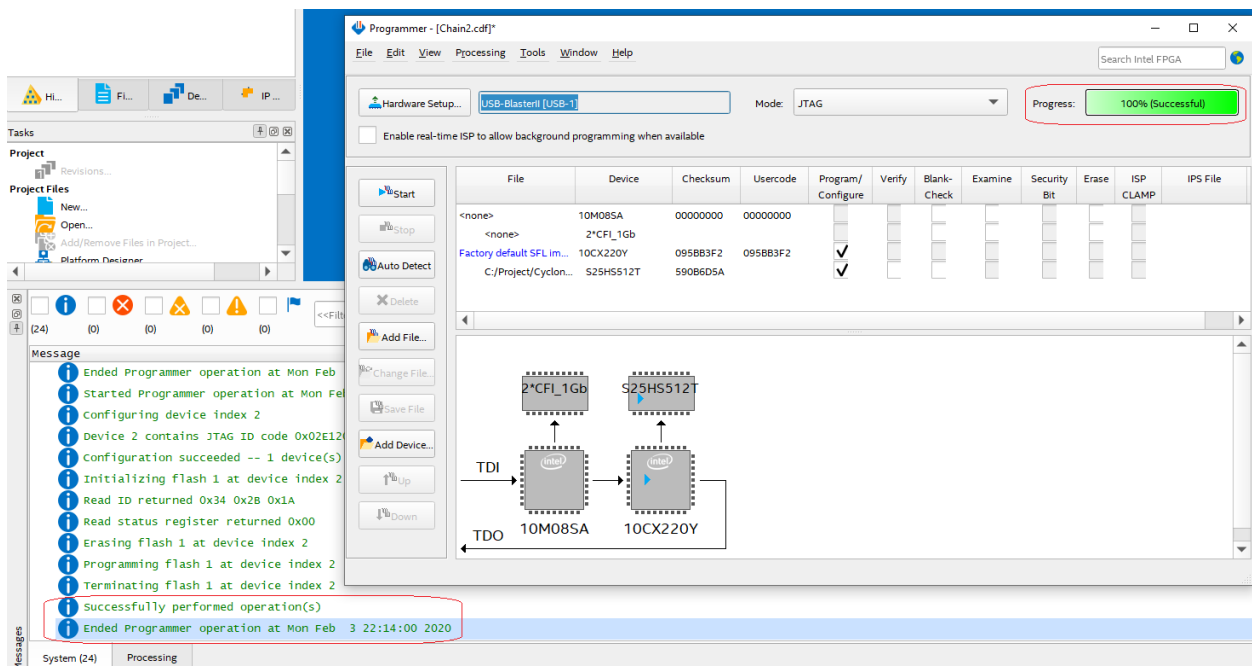
File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP	IPS File
<none>	10M08SA	00000000	00000000								
<none>	2*CFI_1Gb										
<none>	10CX220Y	00000000	<none>								
<none>	10CX220Y	00000000	0958B3F2								
C:/Project/Cyclon...	S25HS512T	590B6D5A									

Below the table is a hardware diagram showing three Intel Cyclone 10GX FPGA chips connected in a chain. The middle chip, labeled 10CX220Y, is circled in red. Above the first chip is a 10M08SA device, and above the third chip is an S25HS512T device. The diagram shows TDI and TDO signals and data flow between the chips.

11. **Program/Configure** オプションを有効にしてください (赤い円のチェックボックス。青い円のチェックボックスも自動的に有効になります)。



12. **Start** をクリックしてください。



13. プログレスバーに 100% (Successful) と表示され、メッセージに”Successfully performs operations”と表示されたら、ターゲットボードの電源を入れ直してください。

4 結論

このアプリケーションノートでは、Cypress S25HS512T QSPI Semper Flash を有効にして Intel Cyclone 10 GX FPGA デバイスを構成する方法について説明しました。

このアプリケーションノートで紹介されている手順は、それぞれのデータシートに従ってフラッシュと FPGA デバイスを構成することにより、他のサイプレス QSPI フラッシュファミリと Intel Arria 10FPGA デバイスにも適用されます。

5 参考資料

- Intel Cyclone 10 GX FPGA 開発キットユーザーガイド
<https://www.intel.com/content/www/us/en/programmable/documentation/hvu1509010715799.html?wapkw=Intel%20Cyclone%2010%20GX%20FPGA%20Development%20Kit%20User%20Guide>
- Generic Flash Programmer ユーザーガイド : Intel Quartus Prime Pro Edition
<https://www.intel.com/content/www/us/en/programmable/documentation/pah1554479151020.html?wapkw=Generic%20Flash%20Programmer>
- サイプレス S25HS-T データシート

改訂履歴

文書名: AN229767 – Intel Cyclone 10 GX FPGA を構成するサイプレス QSPI フラッシュの有効化

文書番号: 002-32734

版数	変更内容
**	本版は英語版 002-29767 Rev. **について、CYPRESS DEVELOPER COMMUNITYの参画者によって日本語に翻訳されたドキュメントです。

セールス、ソリューションおよび法律情報

ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューションセンター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、[サイプレスのロケーションページ](#)をご覧ください。

製品

Arm® Cortex® Microcontrollers	cypress.com/arm
車載用	cypress.com/automotive
クロック&バッファ	cypress.com/clocks
インターフェース	cypress.com/interface
IoT (モノのインターネット)	cypress.com/iot
メモリ	cypress.com/memory
マイクロコントローラ	cypress.com/mcu
PSoC	cypress.com/psoc
電源用 IC	cypress.com/pmuc
タッチセンシング	cypress.com/touch
USB コントローラー	cypress.com/usb
ワイヤレス	cypress.com/wireless

PSoC®ソリューション

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

サイプレス開発者コミュニティ

[コミュニティ](#) | [サンプルコード](#) | [Projects](#) | [ビデオ](#) | [ブログ](#) | [トレーニング](#) | [Components](#)

テクニカルサポート

cypress.com/support

本書で言及するその他すべての商標または登録商標は、それぞれの所有者に帰属します。



Cypress Semiconductor
An Infineon Technologies Company
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2020-2021. 本書面は、Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社 (以下「Cypress」という。) に帰属する財産である。本書面 (本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア (以下「本ソフトウェア」という。) を含む) は、アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し、本段落で特に記載されているものを除き、その特許権、著作権、商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず、かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合、Cypress は、(1) 本ソフトウェアの著作権に基づき、(a) ソースコード形式で提供されている本ソフトウェアについて、Cypress ハードウェア製品と共に用いるためののみ、かつ組織内部でのみ、本ソフトウェアの修正及び複製を行うこと、並びに (b) Cypress のハードウェア製品ユニットに用いるためののみ、(直接又は再販売者及び販売代理店を介して間接的) か (c) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること、並びに (2) 本ソフトウェア (Cypress により提供され、修正がなされていないもの) が抵触する Cypress の特許権のクレームに基づき、Cypress ハードウェア製品と共に用いるためののみ、本ソフトウェアの作成、利用、配布及び輸入を行うことについての非独占的譲渡不能な一身専属的ライセンス (サブライセンスの権利を除く) を付与する。本ソフトウェアのその他の使用、複製、修正、変換又はコンパイルを禁止する。

適用される法律により許される範囲内、Cypress は、本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても、明示又は黙示を問わず、いかなる保証 (商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない) も行わない。いかなるコンピューティングデバイスも絶対に安全ということはない。従って、Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず、Cypress は、Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて、本書面に記載された製品には、エラーと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり、公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内、Cypress は、別途通知することなく、本書面を変更する権利を留保する。Cypress は、本書面に記載のある、いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報 (あらゆるサンプルデザイン情報又はプログラムコードを含む) は、参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計、プログラム、かつテストすることは、本書面のユーザーの責任において行われるものとする。Cypress 製品は、兵器、兵器システム、原子力施設、生命維持装置若しくは生命維持システム、蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム、汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用、又は装置若しくはシステムの不具合が人身傷害、死亡若しくは物的損害を生じさせるようなその他の使用 (以下「本目的外使用」という。) のためには設計、意図又は承認されていない。重要な構成部分とは、その不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ、若しくは本目的外使用に関連するいかなる請求、損害又はその他の責任についても、Cypress はその全部又は一部を問わず一切の責任を負わず、かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の本来目的外使用から生じ又は本目的外使用に関連するあらゆる請求、費用、損害及びその他の責任 (人身傷害又は死亡に基づく請求を含む) から免責補償される。

Cypress, Cypress のロゴ, Spansion, Spansion のロゴ及びこれらの組み合わせ, WICED, PSoC, CapSense, EZ-USB, F-RAM, 及び Traveo は、米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは、cypress.com を参照すること。その他の名称及びブランドは、それぞれの権利者の財産として権利主張がなされている可能性がある。