

Gowin PicoRV32 ハードウェア設計 リファレンスマニュアル

IPUG914-1.9J, 2025-12-30

著作権について(2025)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

GOWIN高云及び **Gowin** は、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

免責事項

当社は、GOWINSEMI Terms and Conditions of Sale (GOWINSEMI取引条件)に規定されている内容を除き、(明示的か又は黙示的に拘わらず)いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ(不具合情報)については、当社に問い合わせる必要があります。

バージョン履歴

日付	バージョン	説明
2020/01/06	1.0J	初版。
2020/03/12	1.1J	● MCU が Wishbone インターフェースの周辺機器 GPIO をサポート。 ● MCU が拡張 AHB バスインターフェースをサポート。 ● MCU がオフチップ SPI-Flash によるダウンロードをサポート。 ● MCU がオフチップ SPI-Flash の読み出し、書き込み、および消去をサポート。 ● MCU が Hardware Stack Protection および Trap Stack Overflow 機能をサポート。
2020/06/01	1.2J	● MCU のオンラインデバッグをサポート。 ● MCU コアの割り込み処理機能を強化。 ● MCU コアの命令を最適化。
2021/07/16	1.3J	● 合成ツールの Synplify Pro を削除。 ● FPGA ソフトウェアのバージョン情報を更新。
2022/02/11	1.4J	GW2AN-9X/GW2AN-18X の、ITCM および DTCM Size の範囲を変更。
2023/08/18	1.5J	Arora V FPGA 製品のサポートを追加。
2024/03/29	1.6J	● A バージョンの GW5AT-60 のサポートを追加。 ● ハードウェア・リファレンス・デザインを更新。
2024/06/14	1.7J	● A バージョンの GW5ART-15 のサポートを追加。 ● ハードウェア・リファレンス・デザインを更新。
2025/09/26	1.8J	● 関連ソフトウェアのバージョン情報を更新。 ● ハードウェア・リファレンス・デザインを追加。
2025/12/30	1.9J	GW3A-20/GW3AR-20 デバイスのサポートを追加。

目次

目次	i
図一覧	iii
表一覧	iv
1 ハードウェア・アーキテクチャ	1
1.1 アーキテクチャ	1
1.2 システムの特徴	2
1.2.1 PicoRV32 Core サブシステム	2
1.2.2 Wishbone Bus サブシステム	3
1.3 システムポートの定義	3
1.4 リソース使用量	5
2 ハードウェア設計フロー	6
2.1 ハードウェア環境	6
2.2 ソフトウェアのバージョン	6
2.3 IP コア生成ツール	6
2.4 ダウンロードツール	6
2.5 設計フロー	7
3 プロジェクトテンプレート	8
3.1 プロジェクトの作成	8
3.1.1 プロジェクトの新規作成	8
3.1.2 プロジェクトの名称とパスの設定	9
3.1.3 デバイスの選択	9
3.1.4 プロジェクト作成の完了	10
3.2 ハードウェア設計	10
3.2.1 PicoRV32 CORE サブシステムの構成	12
3.2.2 Wishbone Bus サブシステムの構成	18
3.3 ユーザーデザイン	25
3.4 制約	25
3.5 プロジェクトの構成	26
3.5.1 合成オプションの構成	26
3.5.2 Post-Place File の構成	26
3.5.3 Dual-Purpose Pin の構成	27

3.6 合成.....	27
3.7 配置配線.....	28
3.8 ダウンロード	29
4 リファレンス・デザイン.....	32

図一覧

図 1-1 アーキテクチャ	1
図 3-1 New Projects	8
図 3-2 Project Name.....	9
図 3-3 Select Device	10
図 3-4 Summary	10
図 3-5 Gowin_PicoRV32 の選択	11
図 3-6 システムの構成	11
図 3-7 Gowin PicoRV32 CORE の構成	12
図 3-8 命令メモリ ITCM の構成	13
図 3-9 データメモリ DTCM の構成	15
図 3-10 Simple UART の構成	17
図 3-11 OPEN AHB INTERFACE の構成	18
図 3-12 WB UART の構成	19
図 3-13 WB I2C Master の構成	19
図 3-14 WB SPI Master の構成	20
図 3-15 WB SPI Slave の構成	21
図 3-16 WB GPIO の構成	23
図 3-17 ADV SPI-Flash の構成	23
図 3-18 Open Wishbone Interfac の構成	24
図 3-19 Top Module の構成	25
図 3-20 合成オプションの構成	26
図 3-21 Post-Place File の構成	27
図 3-22 Dual-Purpose Pin の構成	27
図 3-23 合成	28
図 3-24 Place & Route	28
図 3-25 LittleBee ファミリーの Embedded Flash Mode の構成	29
図 3-26 Arora ファミリーの External Flash Mode の構成	30
図 3-27 Arora V の External Flash Mode の構成	31
図 3-28 ダウンロード	31

表一覧

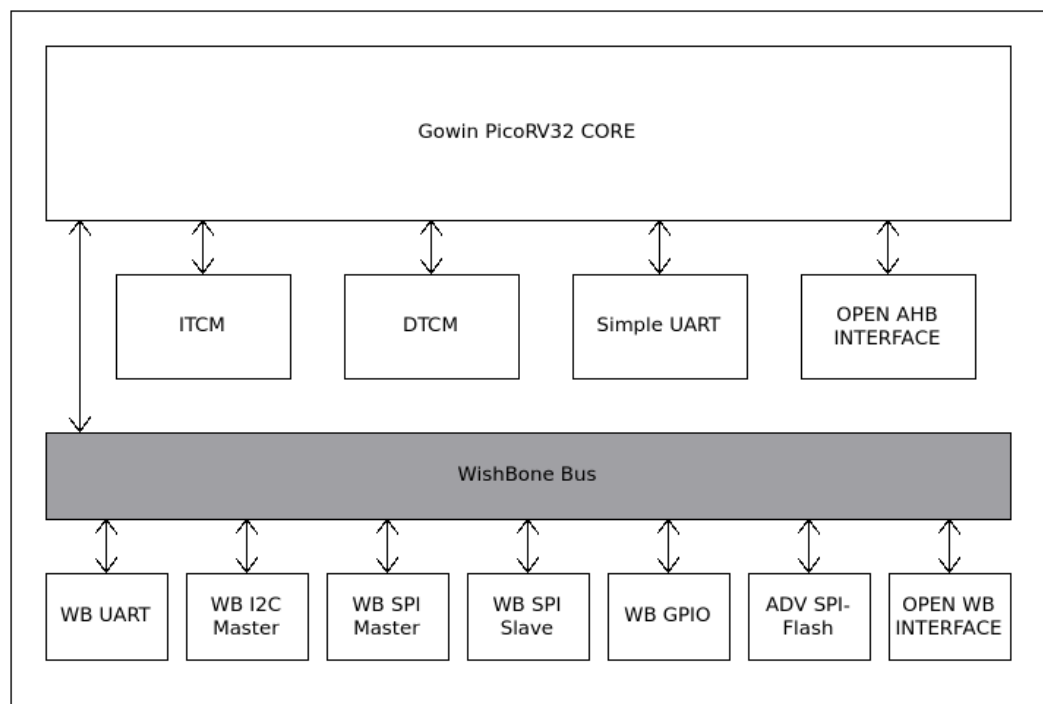
表 1-1 ポート一覧.....	3
表 1-2 リソース使用量	5
表 3-1 Gowin PicoRV32 CORE の構成オプション	12
表 3-2 Wishbone Bus サブシステムの構成オプション	18
表 3-3 WB SPI Master のパラメータの構成オプション	20
表 3-4 WB SPI Slave のパラメータの構成オプション	21
表 3-5 ADV SPI-Flash R/W モードの構成オプション	24

1 ハードウェア・アーキテクチャ

1.1 アーキテクチャ

Gowin_PicoRV32 アーキテクチャには、図 1-1 に示すように、PicoRV32 コア、命令メモリ ITCM、データメモリ DTCM、軽量 UART、AHB バス拡張インターフェース、Wishbone バス、および周辺機器が含まれています。

図 1-1 アーキテクチャ



- Gowin PicoRV32 CORE は 32 ビットの RISC-V アーキテクチャを備えたマイクロコントローラ・コアです。
- ITCM は命令メモリです。
- DTCM はデータメモリです。
- Simple UART は構成可能な軽量 UART です。
- OPEN AHB INTERFACE は、ユーザーが AHB バスインターフェース

の周辺機器を拡張するための AHB バス拡張インターフェースです。

- Wishbone バスは、PicoRV32 コアと、UART、I2C Master、SPI Master、SPI Slave、GPIO、ADV SPI-Flash Memory、および Wishbone バス拡張インターフェースを含む Wishbone インターフェースの周辺機器とを接続します。

1.2 システムの特徴

Gowin_PicoRV32 は 2 つのサブシステムで構成されています：

- マイクロコントローラ・コア、命令メモリ、データメモリ、構成可能な軽量 UART、および AHB バス拡張インターフェースを含む PicoRV32 Core サブシステム。
- Wishbone バスおよび Wishbone インターフェースを介してアクセスされる周辺機器を含む Wishbone Bus サブシステム。

1.2.1 PicoRV32 Core サブシステム

プロセッサコア

- RISC-V 32 ビット整数命令セットアーキテクチャ。
- 構成可能な RISC-V32M 乗算/除算命令セット拡張および構成可能な RISC-V32C 圧縮命令セット拡張。
- オフチップ SPI FLASH Memory によるダウンロードと起動をサポートする構成可能な SPI FLASH Memory インターフェース。
- カスタム割り込み管理命令を採用し、32 の割り込みソース管理をサポートし、割り込み優先度をソフトウェアで制御できる、オンチップ割り込みハンドラー。
- カスタムタイマー操作命令を使用した、オンチップ 32 ビットタイマー。
- オンラインデバッグをサポートする、デバッグモジュール
- ビッグエンディアン/リトルエンディアン：RISC-V はリトルエンディアン形式のみをサポート。
- Trap Stack Overflow 機能をサポート。

Memory

- ITCM：リトルエンディアン形式をサポートする、サイズ (8/16/32/64/128/256/512KB) が構成可能な命令メモリ。
- DTCM：リトルエンディアン形式をサポートし、Hardware Stack Protection 機能をサポートする、サイズ (8/16/32/64/128/256/512KB) が構成可能なデータメモリ。

Simple UART

- 軽量の UART シリアル通信インターフェース
- 必要なロジックリソースはごくわずか

OPEN AHB INTERFACE

- AHB バス拡張インターフェース
- ユーザーは AHB バス拡張インターフェースを使用して AHB バスインターフェースの周辺機器の接続を拡張できます

1.2.2 Wishbone Bus サブシステム

Wishbone Bus サブシステムは、UART、I2C Master、SPI Master、SPI Slave、GPIO、ADV SPI-Flash Memory、および Wishbone バス拡張インターフェースで構成されています。

1.3 システムポートの定義

Gowin_PicoRV32 システムのポートの定義を表 1-1 に示します。

表 1-1 ポート一覧

名称	I/O	幅(ビット)	説明	モジュール
clk_in	入力	1	システムクロック信号	-
resetn_in	入力	1	システムリセット信号	-
irq_in	入力	12	外部割り込み入力信号	OPEN WB INTERFACE と OPEN AHB INTERFACE
jtag_TDI	入力	1	JTAGデータ入力信号	Debug
jtag_TCK	入力	1	JTAGクロック入力信号	
jtag_TMS	入力	1	JTAGモード選択信号	
jtag_TDO	出力	1	JTAGデータ出力	
ser_tx	出力	1	Simple UARTの出力信号	Simple UART
ser_rx	入力	1	Simple UARTの入力信号	
gpio_io	入出力	32	Wishbone GPIOの入出力信号	WB GPIO
wbuart_tx	出力	1	Wishbone UARTの出力信号	WB UART
wbuart_rx	入力	1	Wishbone UARTの入力信号	
wbi2c_sda	入出力	1	Wishbone I2C Masterのデータ	WB I2C Master
wbi2c_scl	入出力	1	Wishbone I2C Masterのクロック信号	
wbspi_master_miso	入力	1	Wishbone SPI MasterのMISO信号	WB SPI Master
wbspi_master_mosi	出力	1	Wishbone SPI MasterのMOSI信号	
wbspi_master_ssn	出力		Wishbone SPI MasterのSLAVE選択信号 各ビットは1つのSLAVEに対応	

名称	I/O	幅(ビット)	説明	モジュール
			し、最大8ビットをサポートします	
wbspi_master_sclk	出力	1	Wishbone SPI Masterのクロック信号	
wbspi_slave_miso	出力	1	Wishbone SPI SlaveのMISO信号	WB SPI Slave
wbspi_slave_mosi	入力	1	Wishbone SPI SlaveのMOSI信号	
wbspi_slave_ss_n	入力	1	Wishbone SPI SlaveのSLAVE選択信号	
wbspi_slave_sclk	入力	1	Wishbone SPI Slaveのクロック信号	
io_spi_clk	入出力	1	ADV SPI-Flashのクロック信号	ADV SPI-Flash Memory
io_spi_csn	入出力	1	ADV SPI-Flashのチップセレクト信号	
io_spi_mosi	入出力	1	ADV SPI-FlashのMOSI信号	
io_spi_miso	入出力	1	ADV SPI-FlashのMISO信号	
slv_ext_stb_o	出力	1	Wishboneバス拡張インターフェースのstrb信号	OPEN WB INTERFACE
slv_ext_we_o	出力	1	Wishboneバス拡張インターフェースの書き込み信号	
slv_ext_cyc_o	出力	1	Wishboneバス拡張インターフェースのcyc信号	
slv_ext_ack_i	入力	1	Wishboneバス拡張インターフェースのack信号	
slv_ext_adr_o	出力	32	Wishboneバス拡張インターフェースのアドレス	
slv_ext_wdata_o	出力	32	Wishboneバス拡張インターフェースの書き込みデータ	
slv_ext_rdata_i	入力	32	Wishboneバス拡張インターフェースの読み出しデータ	
slv_ext_sel_o	出力	4	Wishboneバス拡張インターフェースのバイト選択信号	
hrdata	入力	32	AHBバス拡張インターフェースの読み出しデータ信号	OPEN AHB INTERFACE
hresp	入力	2	AHBバス拡張インターフェースのバス転送状態	
hready	入力	1	AHBバス拡張インターフェース準備完了	
haddr	出力	32	Wishboneバス拡張インターフェ	

名称	I/O	幅(ビット)	説明	モジュール
			ースのアドレス	
hwrite	出力	1	AHBバス拡張インターフェースの読み出し/書き込み方向	
hsize	出力	3	AHBバス拡張インターフェースの転送データのサイズ	
hburst	出力	3	AHBバス拡張インターフェースのバースト信号	
hwdata	出力	32	AHBバス拡張インターフェースの書き込みデータ	
hsel	出力	1	AHBバス拡張インターフェースのチップセレクト信号	
htrans	出力	2	AHBバス拡張インターフェースの転送タイプ	

1.4 リソース使用量

例えば、AバージョンのGW5A-25の場合、Gowin_PicoRV32 システムのリソース使用量は表 1-2 に示すとおりです。

表 1-2 リソース使用量

Configuration \ Resources	LUT	Register	BSRAM	DSP
MCUコア	5937	3583	32	2
MCUコア+バス・周辺機器	8542	5405	32	2

2 ハードウェア設計フロー

2.1 ハードウェア環境

- DK_START_GW2A18 V2.0
GW2A-LV18PG256C8/I7
GW2A-18(C バージョン)
- DK_START_GW2A-LV55PG484C8I7 V1.3
GW2A-LV55PG484C8/I7
GW2A-55(C バージョン)
- DK_START_GW5AST-LV138FPG676A V1.0
GW5AST-LV138FPG676AC1/I0
GW5AST-138 (B バージョン)
- DK_DP_GW5AT-LV60UG225 V1.0
GW5AT-LV60UG225C2/I1
GW5AT-60 (B バージョン)
- DK_START_GW5A-LV25UG324 V2.0
GW5A-LV25UG324C2/I1
GW5A-25 (A バージョン)

2.2 ソフトウェアのバージョン

テスト済みバージョン : Gowin_V1.9.11.02 (64-bit)

2.3 IP コア生成ツール

Gowin ソフトウェアは、Gowin_PicoRV32 IP 設計を構成および生成するための IP Core Generator を提供します。

2.4 ダウンロードツール

Gowin ソフトウェアは、ビットストリーム・ファイルをダウンロード

するための **Programmer** を提供します。

Programmer の使用方法については、『**Gowin Programmer ユーザーガイド**([SUG502](#))』を参照してください。

2.5 設計フロー

Gowin_PicoRV32 ハードウェア設計フロー：

1. **IP Core Generator** で **PicoRV32 Core** サブシステムと **Wishbone Bus** サブシステムを構成して **Gowin_PicoRV32 IP** 設計を生成してプロジェクトにインポートします。

注記：

Wishbone バス拡張インターフェースまたは **AHB** バス拡張インターフェースを介してカスタム周辺機器にアクセスする必要がある場合は、**OPEN WB INTERFACE** または **OPEN AHB INTERFACE** を有効にし、「**Use Gowin PicoRV32 as top module**」オプションを無効にします。

2. **Gowin_PicoRV32 IP Top Module** をインスタンス化してユーザーデザインをインポートし、**Gowin_PicoRV32 IP Top Module** およびユーザーデザインを接続させます。
3. 物理制約とタイミング制約を与えます。
4. 合成ツールである **GowinSynthesis** で合成してネットリスト・ファイルを生成します。
5. 合成が完了したら、配置配線ツールの **Place & Route** を実行してビットストリーム・ファイルを生成します。
6. ダウンロードツールである **Programmer** を使用してビットストリーム・ファイルを **FPGA** にダウンロードします。

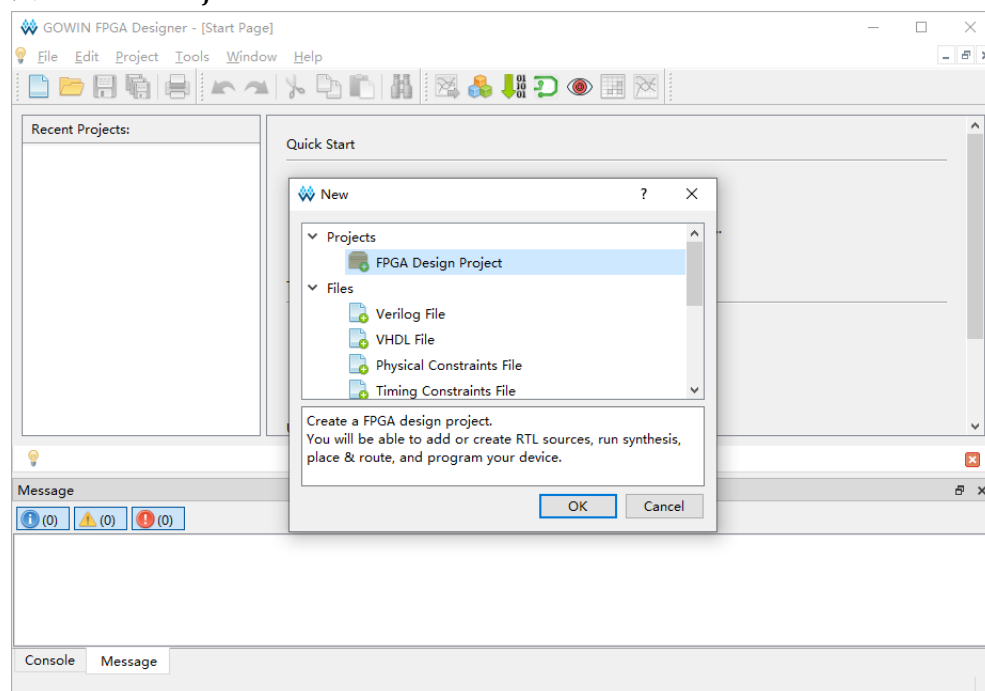
3 プロジェクトテンプレート

3.1 プロジェクトの作成

3.1.1 プロジェクトの新規作成

Gowin ソフトウェアを起動した後、メニューバーで「File > New… > FPGA Design Project…」を選択するか、ツールバーの「」または「Quick Start」ページの「New Project…」をクリックしてプロジェクトを新規作成します(図 3-1)。

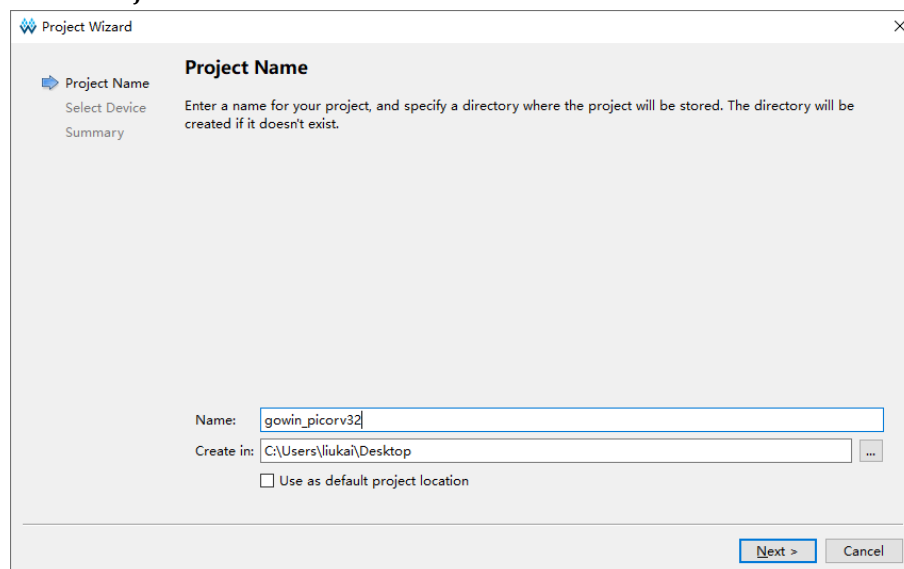
図 3-1 New Projects



3.1.2 プロジェクトの名称とパスの設定

プロジェクト名を入力し、プロジェクトのパスを選択します(図 3-2)。

図 3-2 Project Name



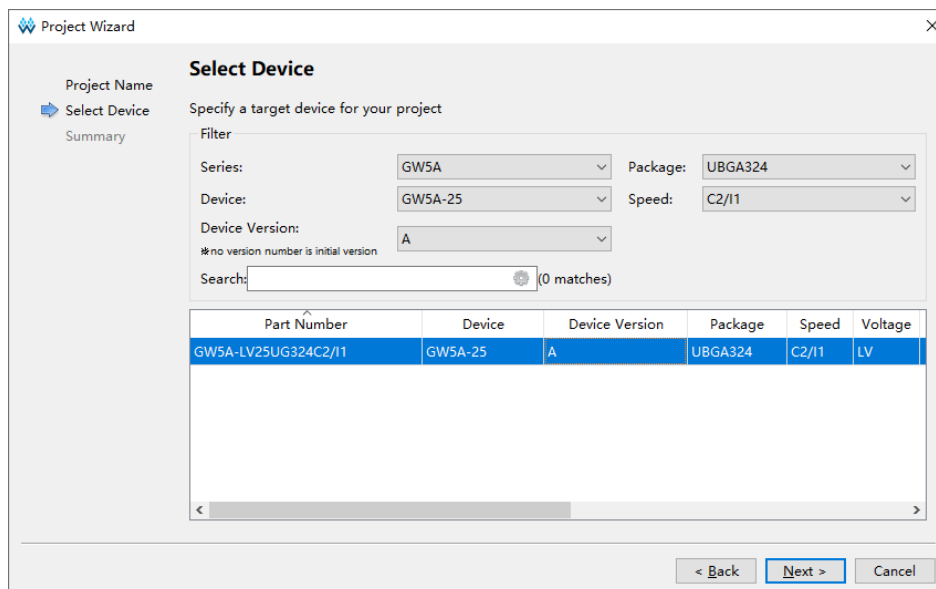
3.1.3 デバイスの選択

「Series」、「Device」、「Device Version」、「Package」、「Speed」、および「Part Number」を選択します(図 3-3)。

例えば、DK_START_GW5A-LV25UG324 V2.0 開発ボードのリファレンス・デザインの場合の構成は次のとおりです：

- Series : GW5A
- Device : GW5A-25
- Device Version : A
- Package : UBGA324
- Speed : C2/I1
- Part Number : GW5A-LV25UG324C2/I1

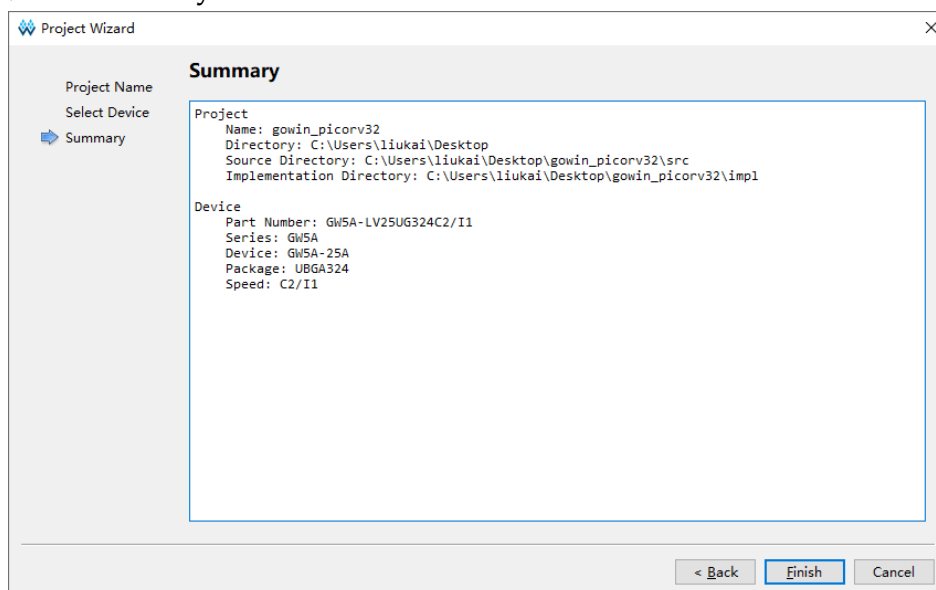
図 3-3 Select Device



3.1.4 プロジェクト作成の完了

図 3-4 に示すように、プロジェクトの新規作成を完了します。

図 3-4 Summary



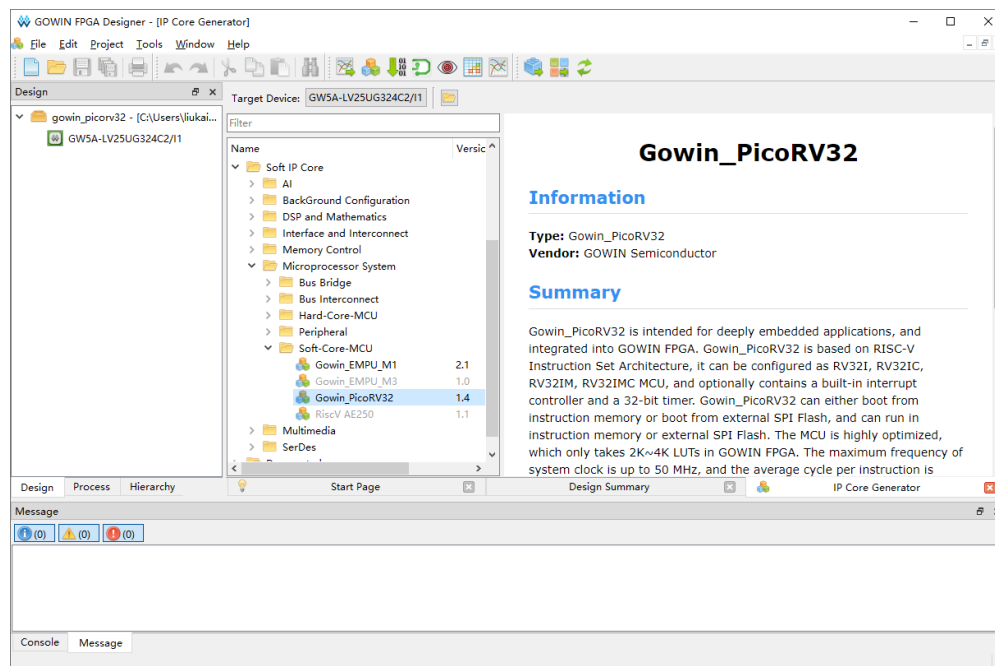
3.2 ハードウェア設計

IP Core Generator を使用して Gowin_PicoRV32 IP 設計を生成します。

メニューから「Tools > IP Core Generator」を選択するか、ツールバーで「」を選択して IP Core Generator を開きます。

「Soft IP Core > Microprocessor System > Soft-Core-MCU > Gowin_PicoRV32 1.4」を選択します(図 3-5)。

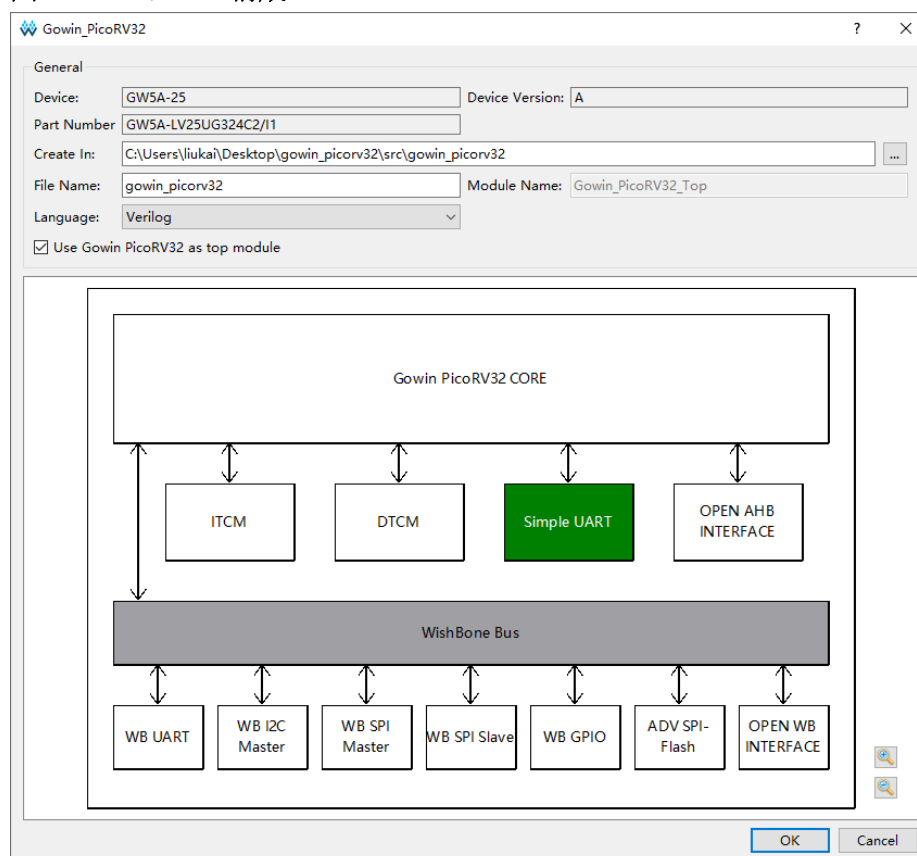
図 3-5 Gowin_PicoRV32 の選択



Gowin_PicoRV32 をダブルクリックして開きます。図 3-6 に示すように、GowinPicoRV32 アーキテクチャには、Gowin PicoRV32 CORE サブシステムと Wishbone Bus サブシステムが含まれています。

構成されたモジュールは緑色で表示されます。

図 3-6 システムの構成

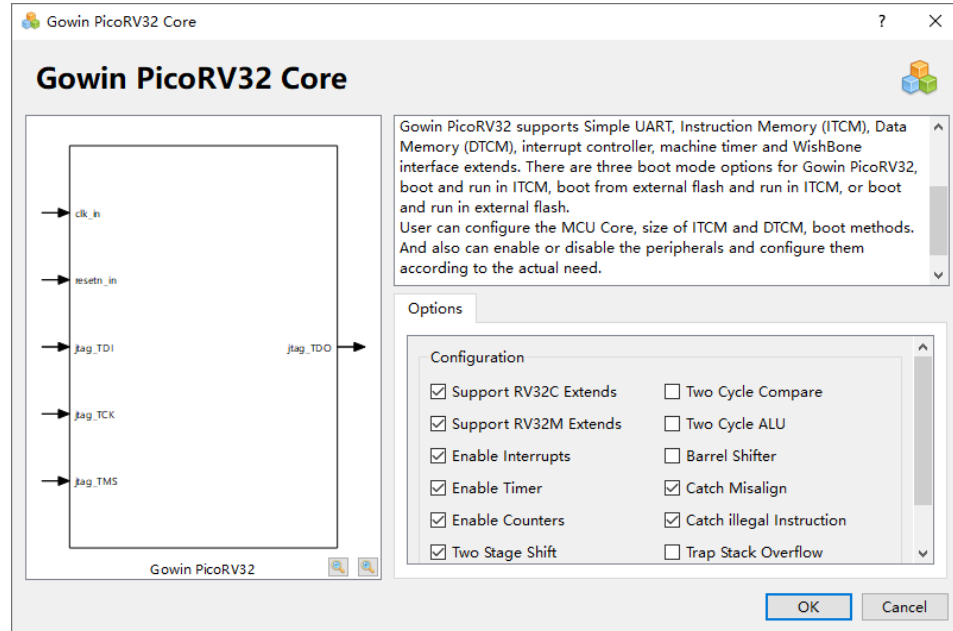


3.2.1 PicoRV32 CORE サブシステムの構成

Gowin PicoRV32 CORE の構成

Gowin PicoRV32 CORE をダブルクリックして構成します(図 3-7)。

図 3-7 Gowin PicoRV32 CORE の構成



Gowin PicoRV32 CORE の構成オプションを表 3-1 に示します。

表 3-1 Gowin PicoRV32 CORE の構成オプション

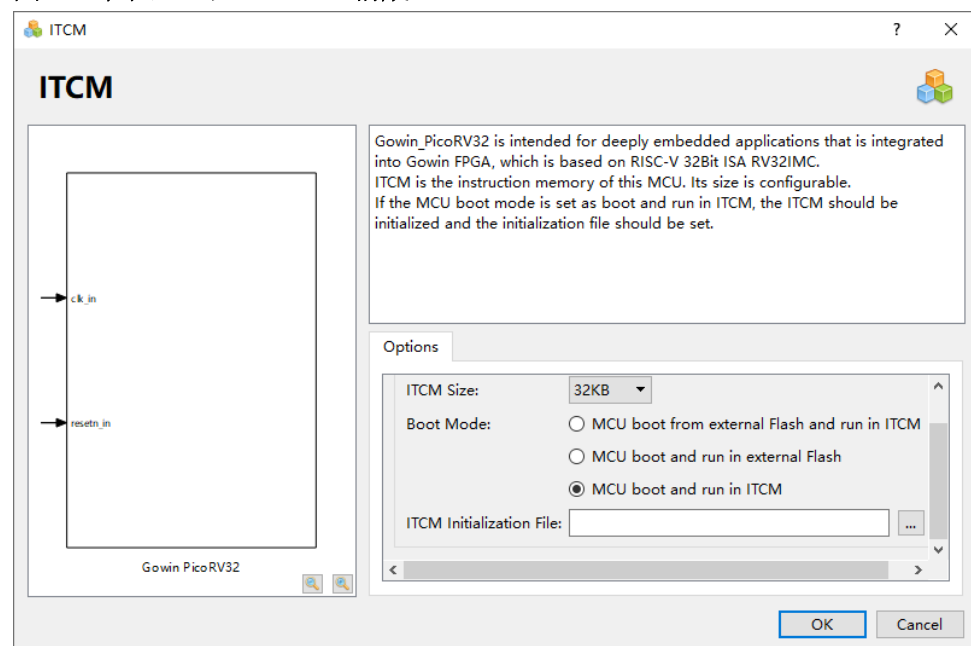
構成オプション	説明
Support RV32C Extends	RISC-V 圧縮命令セット拡張をサポートするかどうかを選択します。デフォルトではチェックされています。
Support RV32M Extends	RISC-V 乗算/除算命令セット拡張をサポートするかどうかを選択します。デフォルトではチェックされています。
Enable Interrupts	割り込み処理をサポートするかどうかを選択します。デフォルトではチェックされています。
Enable Timer	タイマーをサポートするかどうかを選択します。デフォルトではチェックされています。
Enable Counters	カウント命令 RDCYCLE[H]/RDTIME[H]/RDINSTRET[H] をサポートするかどうかを選択します。デフォルトではチェックされています。
Two Stage Shift	2 段階シフト機能をサポートするかどうかを選択します (チェックする場合、シフト操作を高速化できますが、ロジックリソースの使用量も増加します)。デフォルトではチェックされています。
Two Cycle Compare	2 サイクルの比較機能をサポートするかどうかを選択します (チェックする場合、データのパスの長さを短くしてタイミングの品質を向上させることができますが、比較命令を実行するために 1 クロックサイクルが増加しま

構成オプション	説明
	す)。デフォルトではチェックされていません(つまり、オフです)。
Two Cycle ALU	2 サイクルの演算機能をサポートするかどうかを選択します(チェックする場合、データのパスの長さを短くしてタイミングの品質を向上させることができますが、演算命令を実行するために 1 クロックサイクルが増加します)。デフォルトではチェックされていません。
Barrel Shifter	バレルシフト機能をサポートするかどうかを選択します。デフォルトではチェックされていません。
Catch Misalign	メモリアクセス中にアドレスミスアライメント・エラーが発生したときに TRAP に入って動作を停止するかどうかを選択します。デフォルトではチェックされています。
Catch illegal Instruction	不当命令を実行するときに TRAP に入って動作を停止するかどうかを選択します。デフォルトではチェックされています。
Trap Stack Overflow	スタックオーバーフロー防止機能を有効にするかどうかを選択します。デフォルトではチェックされていません。
Enable Debug	オンラインデバッグをサポートするかどうかを選択します。デフォルトではチェックされています。

命令メモリ ITCM の構成

命令メモリ ITCM をダブルクリックして構成します。図 3-8 に示すように、ITCM Size、Gowin_PicoRV32 の起動方法、および ITCM の初期値ファイルを構成できます。

図 3-8 命令メモリ ITCM の構成



- ITCM Size

- サイズは、8KB、16KB、32KB、64KB、128KB、256KB、または 512KB です。
- GW1N/R-9 の場合、最大値は 32KB で、デフォルトは 16KB です。
- GW2AN-9X/18X の場合、最大値は 32KB で、デフォルトは 16KB です。
- GW2A/R/NR-18 の場合、最大値は 64KB で、デフォルトは 32KB です。
- GW2A/N-55 の場合、最大値は 256KB で、デフォルトは 64KB です。
- GW5A/T/ST/S-138 の場合、最大値は 512KB で、デフォルトは 128KB です。
- GW5AT-75 の場合、最大値は 256KB で、デフォルトは 128KB です。
- GW5A/R/S-25 の場合、最大値は 64KB で、デフォルトは 32KB です。
- GW5AT/RT-15 の場合、最大値は 64KB で、デフォルトは 32KB です。
- GW5A/T-60 の場合、最大値は 128KB で、デフォルトは 64KB です。
- GW3A/R-20 の場合、最大値は 64KB で、デフォルトは 32KB です。

- **Boot Mode**

- オフチップ SPI FLASH Memory によるダウンロードおよび起動、命令メモリ ITCM による実行(MCU boot from external Flash and run in ITCM)。
- オフチップ SPI FLASH Memory によるダウンロードおよび起動、オフチップ SPI FLASH Memory による実行(MCU boot and run in external Flash)。
- 命令メモリ ITCM による起動、命令メモリ ITCM による実行(MCU boot and run in ITCM)

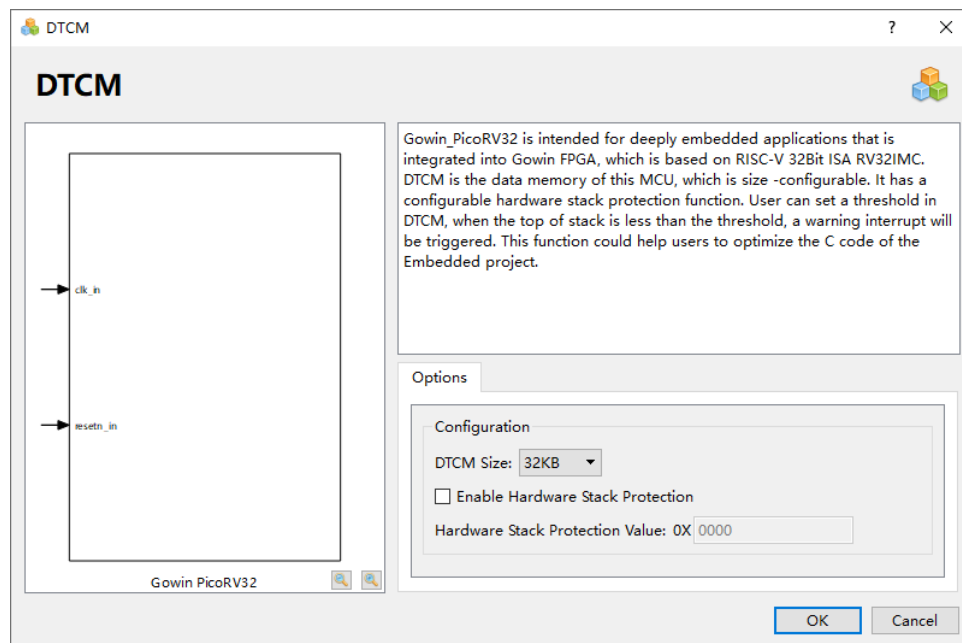
注記：

- MCU boot and run in ITCM がチェックされた場合、ITCM Initialization File に ITCM 初期化ファイル(つまり、Gowin_PicoRV32 ソフトウェア設計 ram32.hex)をインポートする必要があります。
- ITCM Initialization File のパスには、数字または¥n、¥t などのエスケープ文字を使用してはなりません。

データメモリ DTCM の構成

データメモリ DTCM をダブルクリックして構成します。図 3-9 に示すように、DTCM Size、ハードウェアスタック保護およびその範囲を構成できます。

図 3-9 データメモリ DTCM の構成



- DTCM Size

- サイズは、8KB、16KB、32KB、64KB、128KB、256KB、または 512KB です。
- GW1N/R-9 の場合、最大値は 32KB で、デフォルトは 16KB です。
- GW2AN-9X/18X の場合、最大値は 32KB で、デフォルトは 16KB です。
- GW2A/R/NR-18 の場合、最大値は 64KB で、デフォルトは 32KB です。
- GW2A/N-55 の場合、最大値は 256KB で、デフォルトは 64KB です。
- GW5A/T/ST/S-138 の場合、最大値は 512KB で、デフォルトは 128KB です。
- GW5AT-75 の場合、最大値は 256KB で、デフォルトは 128KB です。
- GW5A/R/S-25 の場合、最大値は 64KB で、デフォルトは 32KB です。
- GW5AT/RT-15 の場合、最大値は 64KB で、デフォルトは 32KB です。
- GW5A/T-60 の場合、最大値は 128KB で、デフォルトは 64KB です。
- GW3A/R-20 の場合、最大値は 64KB で、デフォルトは 32KB です。

- Hardware Stack Protection

- Enable Hardware Stack Protection をチェックすると、

Gowin_PicoRV32 はハードウェアスタック保護をサポートするようになります。

- **Hardware Stack Protection Value** : ハードウェアスタック保護の範囲であり、**DTCM Size** 未満である必要があります。

ITCM と DTCM Size の構成の制限

- GW1N/R-9 の場合、ITCM または DTCM のサイズは最大 32KB に構成できますが、ITCM または DTCM が 32KB に構成されている場合、もう 1 つのメモリは最大 16KB になります。
- GW2AN-9X/18X の場合、ITCM または DTCM のサイズは最大 32KB に構成できますが、ITCM または DTCM が 32KB に構成されている場合、もう 1 つのメモリは最大 16KB になります。
- GW2A/R/NR-18 の場合、ITCM または DTCM のサイズは最大 64KB に構成できますが、ITCM または DTCM が 64KB に構成されている場合、もう 1 つのメモリは最大 16KB になります。
- GW2A/N-55 の場合、ITCM または DTCM のサイズは最大 256KB に構成できますが、ITCM または DTCM が 256KB に構成されている場合、もう 1 つのメモリは最大 16KB になります。
- GW5A/T/ST/S-138 の場合、ITCM または DTCM のサイズは最大 512KB に構成できますが、ITCM または DTCM が 512KB に構成されている場合、もう 1 つのメモリは最大 128KB になります。
- GW5AT-75 の場合、ITCM または DTCM のサイズは最大 256KB に構成できますが、ITCM または DTCM が 256KB に構成されている場合、もう 1 つのメモリは最大 256KB になります。
- GW5A/R/S-25 の場合、ITCM または DTCM のサイズは最大 64KB に構成できますが、ITCM または DTCM が 64KB に構成されている場合、もう 1 つのメモリは最大 32KB になります。
- GW5AT/RT-15 の場合、ITCM または DTCM のサイズは最大 64KB に構成できますが、ITCM または DTCM が 64KB に構成されている場合、もう 1 つのメモリは最大 8KB になります。
- GW5A/T-60 の場合、ITCM または DTCM のサイズは最大 128KB に構成できますが、ITCM または DTCM が 128KB に構成されている場合、もう 1 つのメモリは最大 64KB になります。
- GW3A/R-20 の場合、ITCM または DTCM のサイズは最大 64KB に構成できますが、ITCM または DTCM が 64KB に構成されている場合、もう 1 つのメモリは最大 32KB になります。

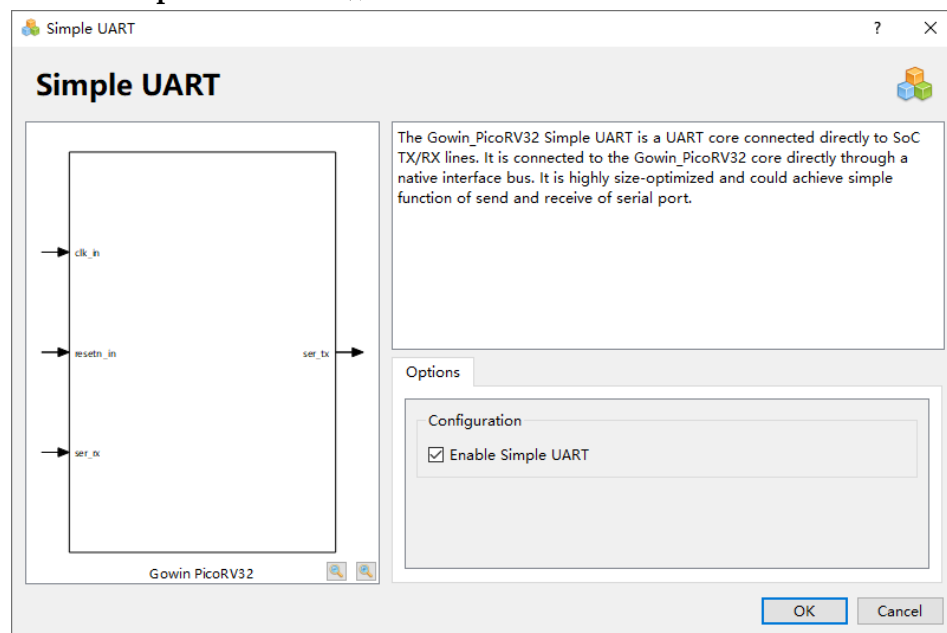
Simple UART の構成

Simple UART をダブルクリックして構成します。図 3-10 に示すように、Simple UART を有効にするかどうかを構成できます。

Enable Simple UART をチェックすると、Gowin_PicoRV32 は Simple

UART をサポートするようになります。デフォルトではチェックされています。

図 3-10 Simple UART の構成

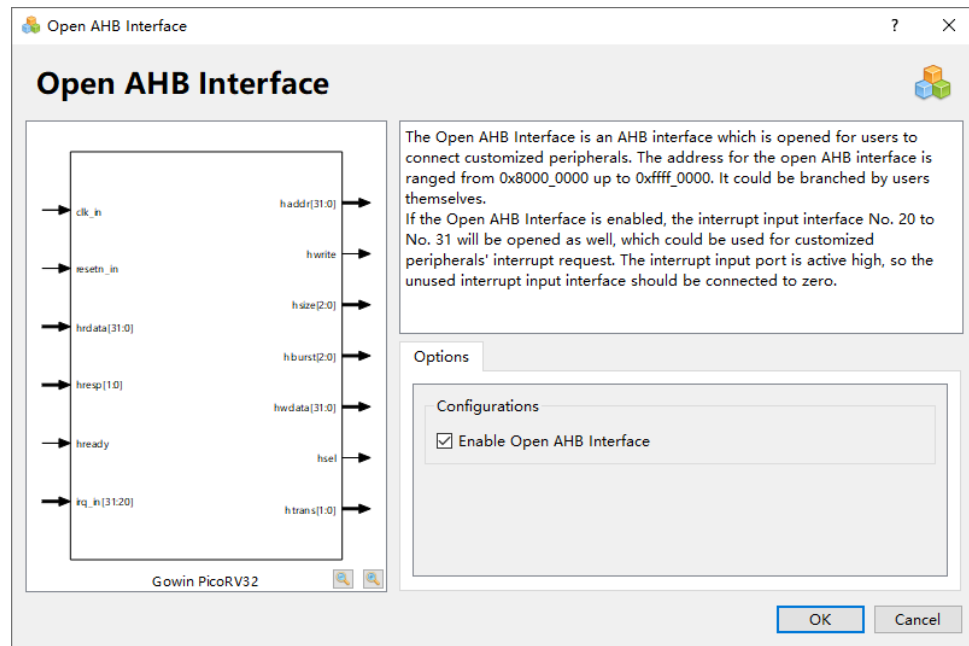


OPEN AHB INTERFACE の構成

OPEN AHB INTERFACE をダブルクリックして構成します(図 3-11)。

- デフォルトではチェックされていません。
- Enable Open AHB Interface をチェックすると、Gowin_PicoRV32 は OPEN AHB INTERFACE をサポートすることになります。ユーザーはこのインターフェースで AHB バス拡張インターフェース経由の周辺機器を接続できます。
- ユーザーが拡張した AHB 周辺機器用の 12 個の外部割り込み信号 irq_in [31:20] は予約されています。

図 3-11 OPEN AHB INTERFACE の構成



3.2.2 Wishbone Bus サブシステムの構成

Wishbone Bus サブシステムは、WB UART、WB I2C Master、WB SPI Master、WB SPI Slave、WB GPIO、ADV SPI-Flash Memory、および OPEN WB INTERFACE を含む周辺機器をサポートできます。

Wishbone Bus サブシステムの構成オプションを表 3-2 に示します。

表 3-2 Wishbone Bus サブシステムの構成オプション

構成オプション	説明
Enable Wishbone UART	WB UART を有効にします。デフォルトではオフです。
Enable Wishbone I2C Master	WB I2C Master を有効にします。デフォルトではオフです。
Enable Wishbone SPI Master	WB SPI Master を有効にします。デフォルトではオフです。
Enable Wishbone SPI Slave	WB SPI Slave を有効にします。デフォルトではオフです。
Enable Wishbone GPIO	WB GPIO を有効にします。デフォルトではオフです。
Enable ADV SPI-Flash	ADV SPI-Flash Memory を有効にします。デフォルトではオフです。
Enable Open Wishbone Interface	OPEN WB INTERFACE を有効にします。デフォルトではオフです。

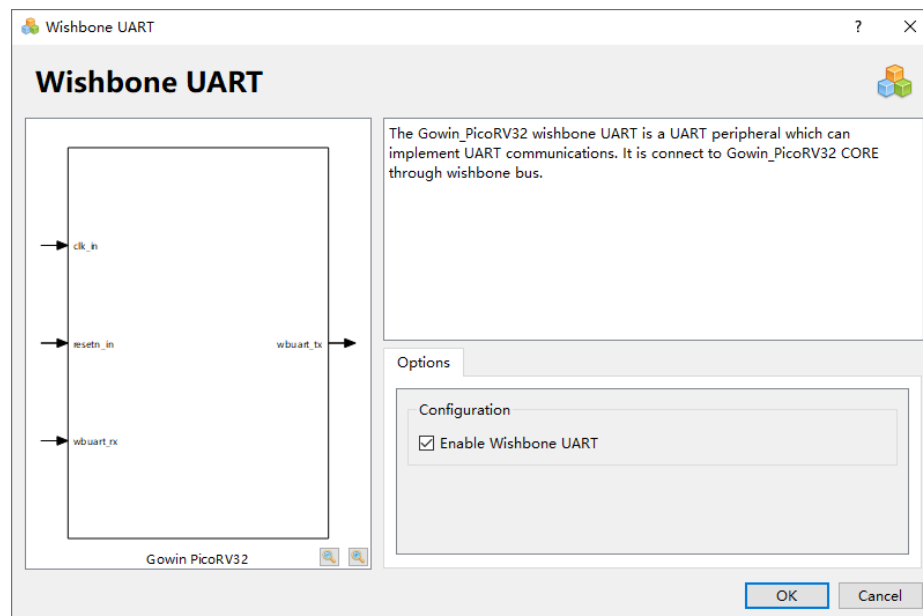
WB UART の構成

WB UART をダブルクリックして構成します(図 3-12)。

Enable Wishbone UART をチェックすると、Gowin_PicoRV32 は

Wishbone UART をサポートするようになります。デフォルトではチェックされていません。

図 3-12 WB UART の構成

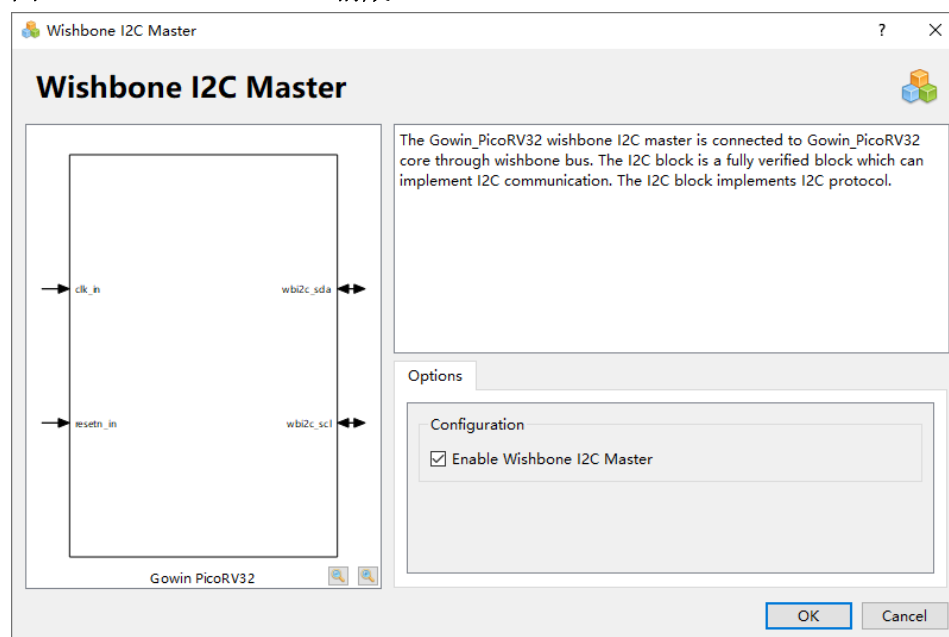


WB I2C Master の構成

WB I2C Master をダブルクリックして構成します(図 3-13)。

Enable Wishbone I2C Master をチェックすると、Gowin_PicoRV32 は Wishbone I2C Master をサポートするようになります。デフォルトではチェックされていません。

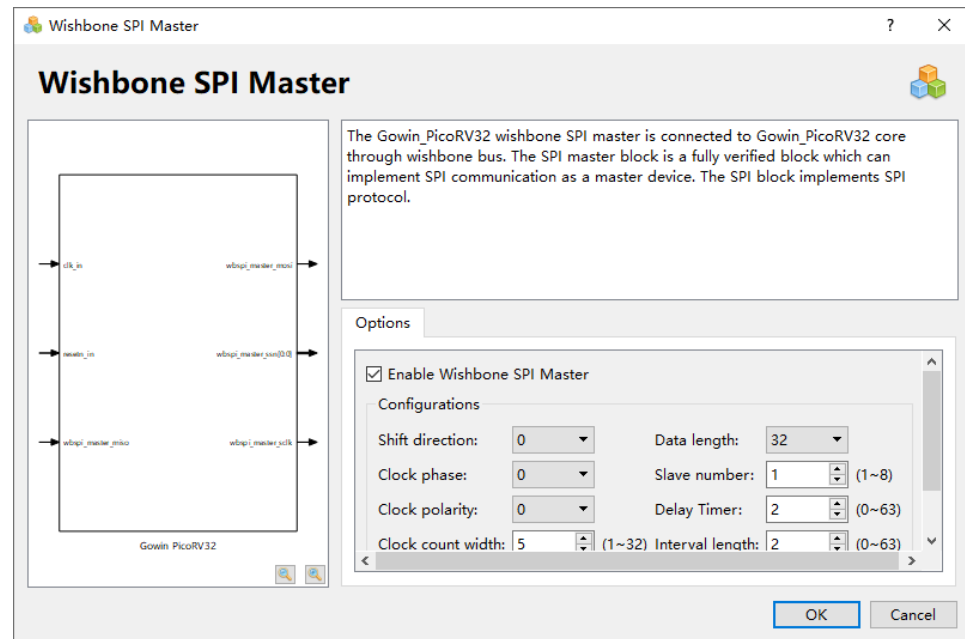
図 3-13 WB I2C Master の構成



WB SPI Master の構成

WB SPI Master をダブルクリックして構成します(図 3-14)。

図 3-14 WB SPI Master の構成



Enable Wishbone SPI Master をチェックすると、Gowin_PicoRV32 は Wishbone SPI Master をサポートするようになります。デフォルトではチェックされていません。

Wishbone SPI Master を有効にした場合、Wishbone SPI Master のパラメータを構成できます(表 3-3)。

表 3-3 WB SPI Master のパラメータの構成オプション

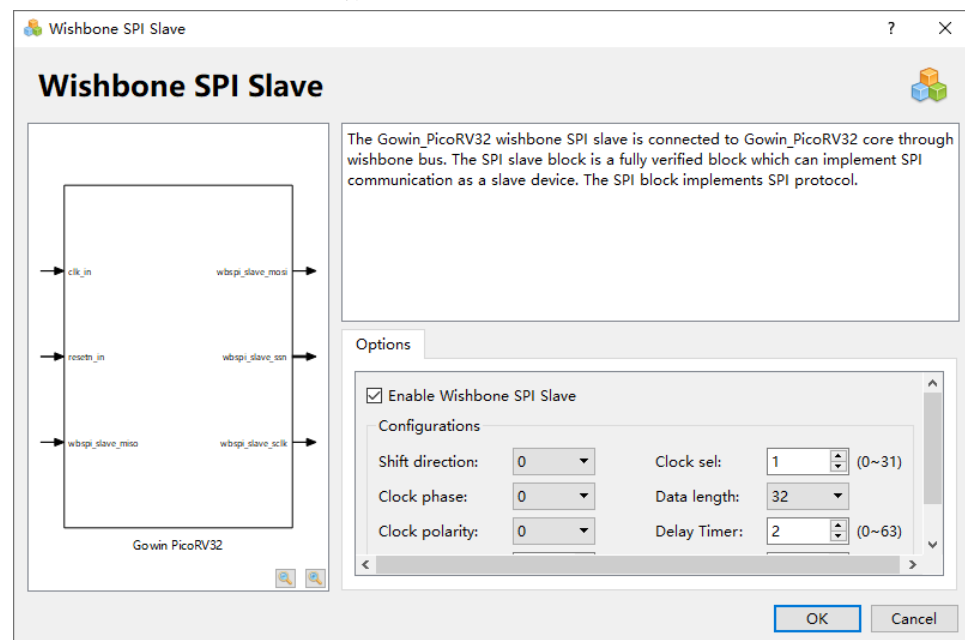
パラメータ	説明
Shift direction	データの転送形式を指定します： 0の場合、データの最上位ビット(MSB)が最初に送信されます 1の場合、データの最下位ビット(LSB)が最初に送信されます
Clock phase	WB SPI Masterのクロック位相を指定します： 0の場合、データはSCLKの最初のエッジで有効です。 1の場合、データはSCLKの2番目のエッジで有効です。
Clock polarity	WB SPI Masterのクロック極性を指定します： 0の場合、アイドル状態の間、SCLKはLowです 1の場合、アイドル状態の間、SCLKはHighです
Clock count width	クロックカウンターの範囲を指定します SCLKのデータ幅を実現するには、十分なビット幅が必要です。
Clock sel	CLK_I分周によりSCLKを生成するために必要な分周係数を指定します SCLKは次の式で計算可能： $SCLK = CLK_I / (2 * (CLOCK_SEL) + 1)$ 値の範囲：0 ~ $2^{(clock\ count\ width)} - 1$
Data length	データの幅を指定します

パラメータ	説明
	値の範囲 : 8/16/32/64
Slave number	サポートできるスレーブの数を指定します 値の範囲 : 1 ~ 32
Delay time	SS_N信号がアサートされた後、データの最初のビットが送信される前に待機する遅延時間を指定します : 遅延時間の計算式 : $\text{Delay} = \text{Delay Time} * (\text{SCLK period} / 2)$ 値の範囲 : 0 ~ 63
Interval length	SPI送信要求後にSS_N信号が待機するSCLKサイクルの数を指定。 値の範囲 : 0 ~ 63

WB SPI Slave の構成

WB SPI Slave をダブルクリックして構成します(図 3-15)。

図 3-15 WB SPI Slave の構成



Enable Wishbone SPI Slave をチェックすると、Gowin_PicoRV32 は Wishbone SPI Slave をサポートするようになります。デフォルトではチェックされていません。

Wishbone SPI Slave を有効にした場合、Wishbone SPI Slave のパラメータを構成できます(表 3-4)。

表 3-4 WB SPI Slave のパラメータの構成オプション

パラメータ	説明
Shift direction	データの転送形式を指定します : 0の場合、データの最上位ビット(MSB)が最初に送信されます。 1の場合、データの最下位ビット(LSB)が最初に送信されます。

パラメータ	説明
Clock phase	WB SPI Slaveのクロック位相を指定します： 0の場合、データはSCLKの最初のエッジで有効です。 1の場合、データはSCLKの2番目のエッジで有効です。
Clock polarity	WB SPI Slaveのクロック極性を指定します： 0の場合、アイドル状態の間、SCLKはLowです。 1の場合、アイドル状態の間、SCLKはHighです。
Clock count width	クロックカウンターの範囲を指定します： SCLKのデータ幅を実現するには、十分なビット幅が必要です。
Clock sel	CLK_I分周によりSCLKを生成するために必要な分周係数を指定します： SCLKは次の式で計算可能： $SCLK = CLK_I / (2 * (CLOCK_SEL) + 1)$ 値の範囲：0 ~ $2^{(clock\ count\ width)} - 1$
Data length	データの幅を指定します： 値の範囲：8/16/32/64
Delay time	SS_N信号がアサートされた後、データの最初のビットが送信される前に待機する遅延時間を指定します： 遅延時間の計算式： $Delay = Delay\ Time * (SCLK\ period / 2)$ 値の範囲：0 ~ 63
Interval length	SPI送信要求後にSS_N信号が待機するSCLKサイクルの数を指定します： 値の範囲：0 ~ 63

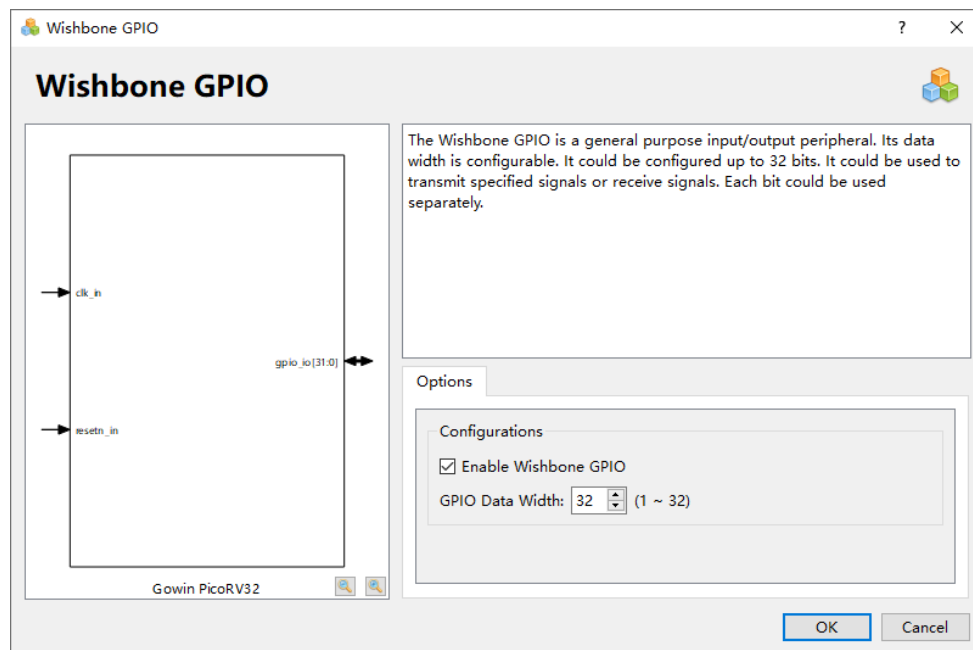
WB GPIO の構成

WB GPIO をダブルクリックして構成します(図 3-16)。

Enable Wishbone GPIO をチェックすると、Gowin_PicoRV32 は Wishbone GPIO をサポートするようになります。デフォルトではチェックされていません。

Wishbone GPIO を有効にした場合、そのデータ幅(GPIO Data Width)を構成できます。その範囲は1~32 です。

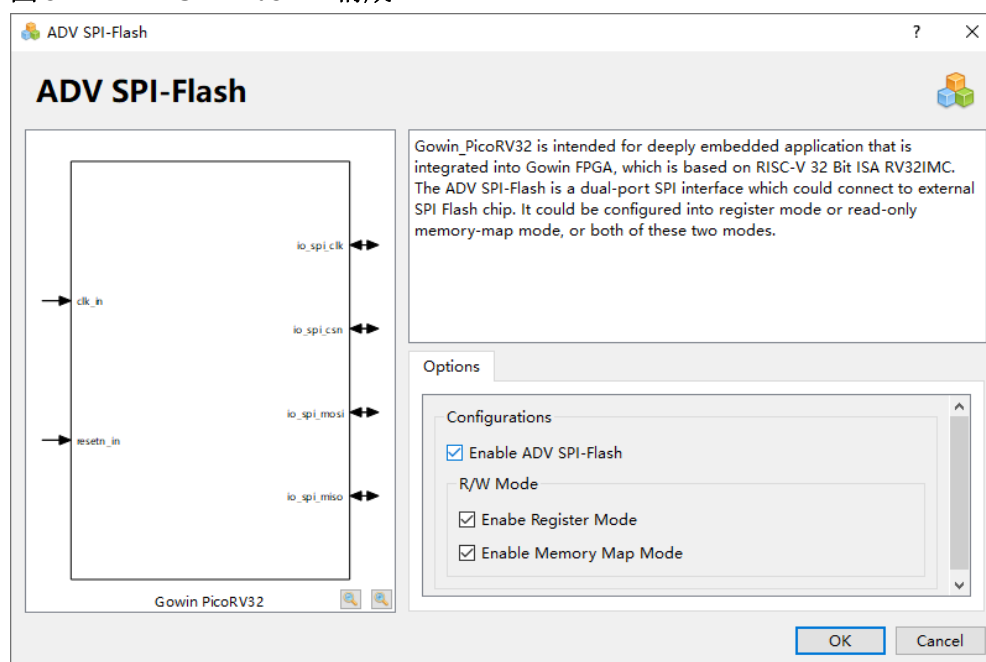
図 3-16 WB GPIO の構成



ADV SPI-Flash の構成

ADV SPI-Flash をダブルクリックして構成します(図 3-17)。

図 3-17 ADV SPI-Flash の構成



Enable ADV SPI-Flash をチェックすると、Gowin_PicoRV32 は ADV SPI-Flash Memory をサポートするようになります。デフォルトではチェックされていません。

ADV SPI-Flash は、MCU ソフトウェア設計における BIN ファイルのダウンロードおよび命令実行をサポートします。

ADV SPI-Flash は、読み出し、書き込み、および消去機能をサポートします。その読み出し/書き込みモードの構成を表 3-5 に示します。

表 3-5 ADV SPI-Flash R/W モードの構成オプション

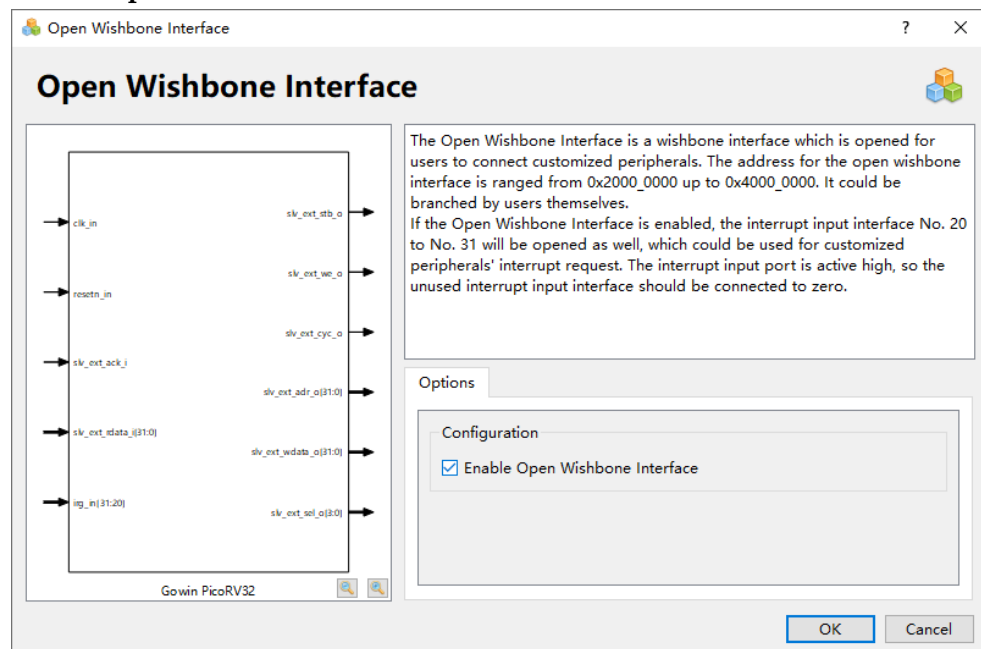
パラメータ	説明
Enable Register Mode	レジスタ・アクセス・モードを有効にします。デフォルトではチェックされています。
Enable Memory Map Mode	メモリ・マップ・アクセス・モードを有効にします。デフォルトではチェックされています。

OPEN WB INTERFACE の構成

OPEN WB INTERFACE をダブルクリックして Open Wishbone Interface を構成します(図 3-18)。

- デフォルトではチェックされていません。
- Enable Open Wishbone Interface をチェックすると、Gowin_PicoRV32 は OPEN WB INTERFACE をサポートすることになります。ユーザーはこのインターフェースで Wishbone バス拡張インターフェース経由の周辺機器を接続できます。
- ユーザーが拡張した Wishbone 周辺機器用の 12 個の外部割り込み信号 irq_in [31:20]は予約されています。

図 3-18 Open Wishbone Interface の構成



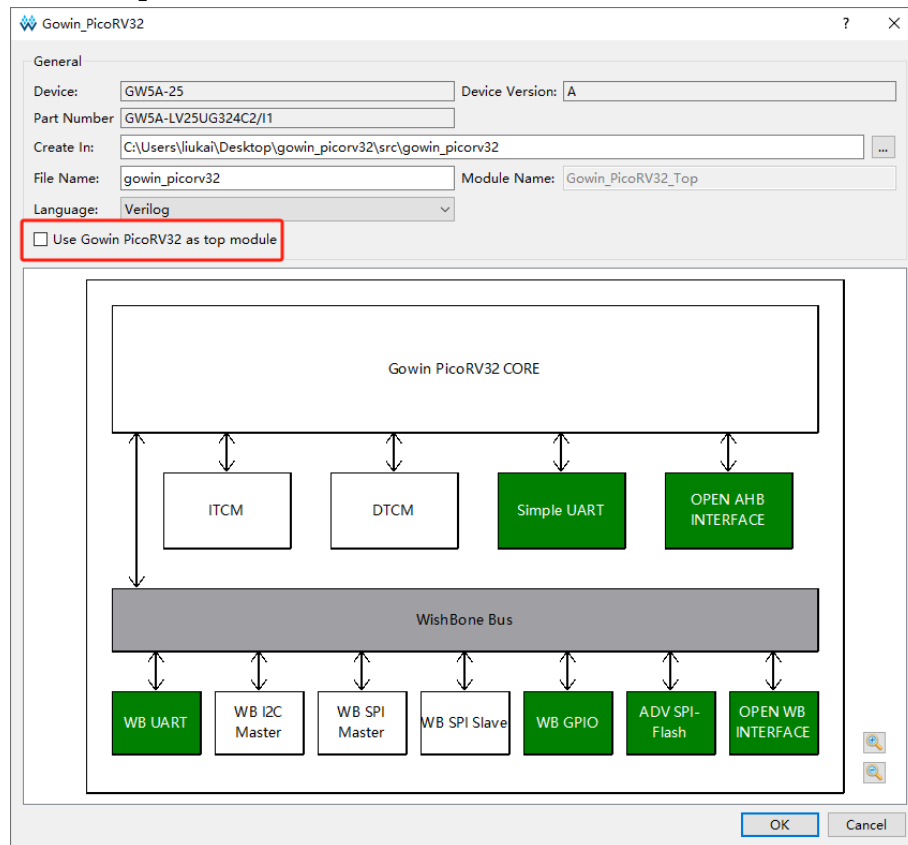
Top Module の構成

- Gowin_PicoRV32 を設計の Top Module として使用する場合は、「Use Gowin PicoRV32 as top module」をチェックして Gowin_PicoRV32 を

Top Module として設定してください。

- OPEN WB INTERFACE または OPEN AHB INTERFACE を構成して Wishbone バスインターフェース経由の周辺機器を拡張する場合、図 3-19 に示すように、「Use Gowin PicoRV32 as top module」オプションをチェックしないでください。

図 3-19 Top Module の構成



3.3 ユーザーデザイン

Gowin_PicoRV32 の構成が完了すると、Gowin_PicoRV32 IP 設計が生成されます。

Gowin_PicoRV32 Top Module をインスタンス化し、Top Module として設定するか、ユーザーデザインに接続します。

ユーザーデザインをインポートして Gowin_PicoRV32 Top Module とユーザーデザインを接続させて完全な RTL 設計を作成します。

3.4 制約

ユーザーRTL 設計が完了したら、使用する開発ボードと出力が必要な IO に基づいて物理制約ファイルを生成します。

タイミング要件およびタイミング解析レポートに基づいてタイミング制約ファイルを生成します。

物理制約の生成方法については、『Gowin 物理制約ユーザーガイド

([SUG935](#))』および『Arora V 物理制約ユーザーガイド([SUG1018](#))』を参照してください。

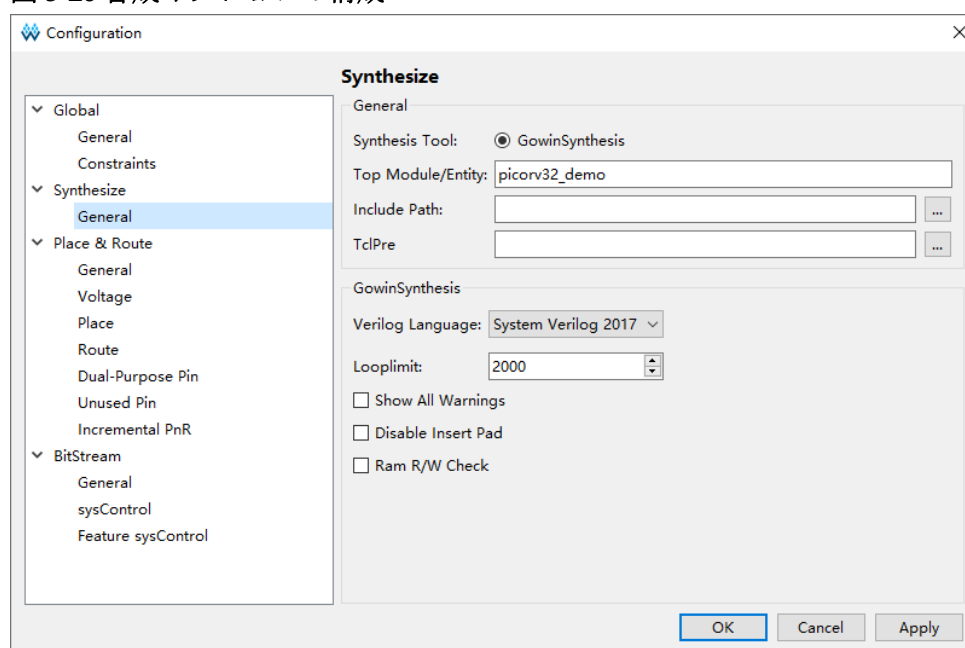
3.5 プロジェクトの構成

3.5.1 合成オプションの構成

「Synthesize > General」オプションを構成します(図 3-20)。

- Top Module/Entity で実際のトップ・モジュール名を入力します。
- Include Path で実際のインクルードパスを入力します。
- Verilog Language を選択します。例えば、System Verilog 2017。

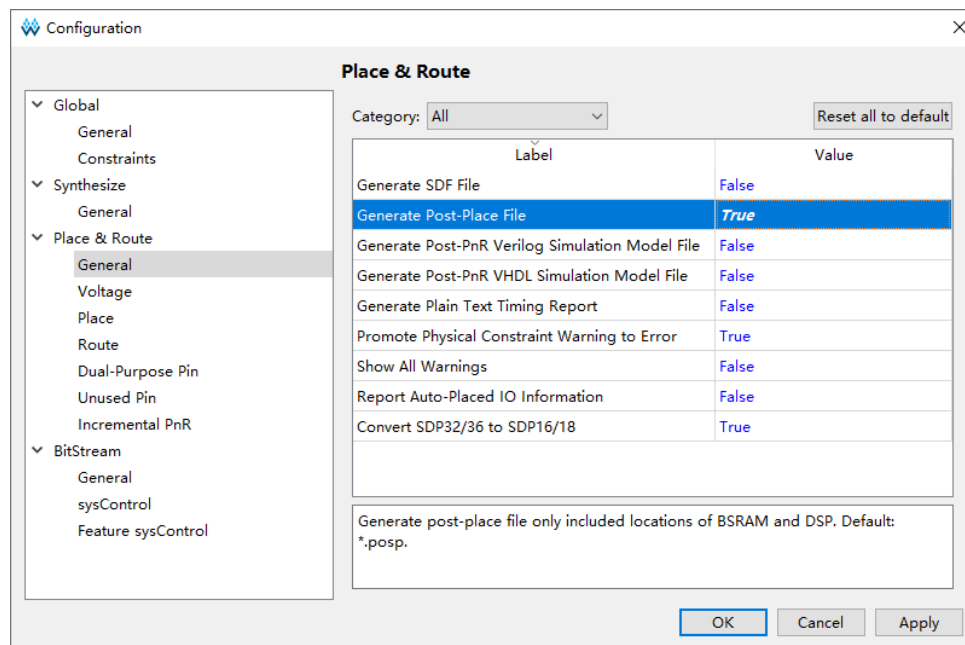
図 3-20 合成オプションの構成



3.5.2 Post-Place File の構成

Gowin_PicoRV32 ソフトウェア設計とハードウェア設計のマージされたファイルをダウンロードするという方法を採用する場合、「Place & Route > General > Generate Post-Place File」オプションを「True」として構成して Post-Place File を生成する必要があります(図 3-21)。

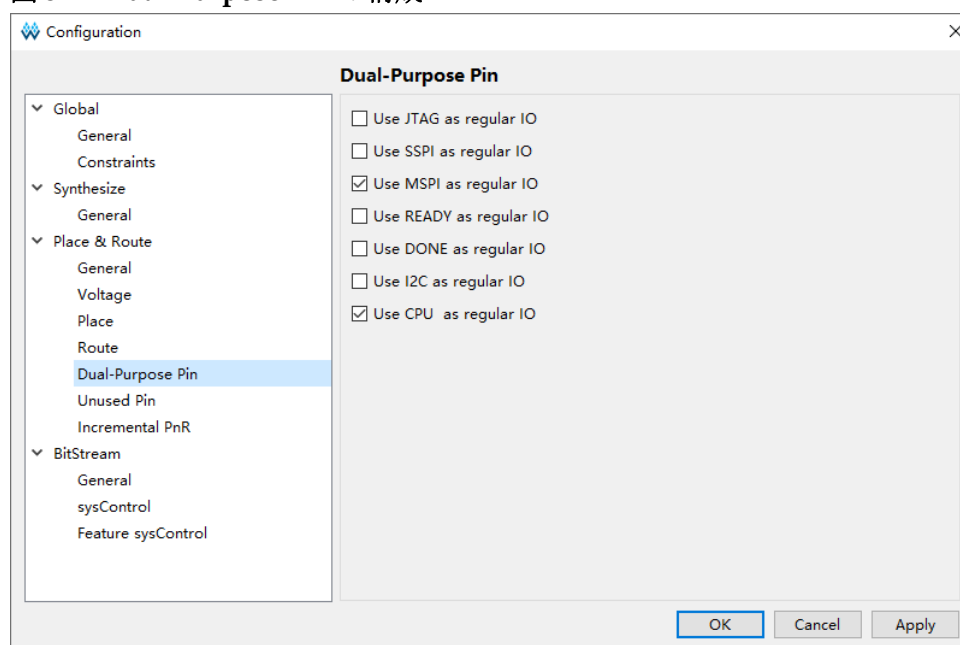
図 3-21 Post-Place File の構成



3.5.3 Dual-Purpose Pin の構成

MCU boot from external Flash and run in ITCM または MCU boot and run in external Flash を選択した場合、「Place & Route > Dual-Purpose Pin」オプションで専用のポート(MSPI、CPU など)を通常の IO として多重化します(図 3-22)。それ以外の場合、ピンの多重化は不要です。

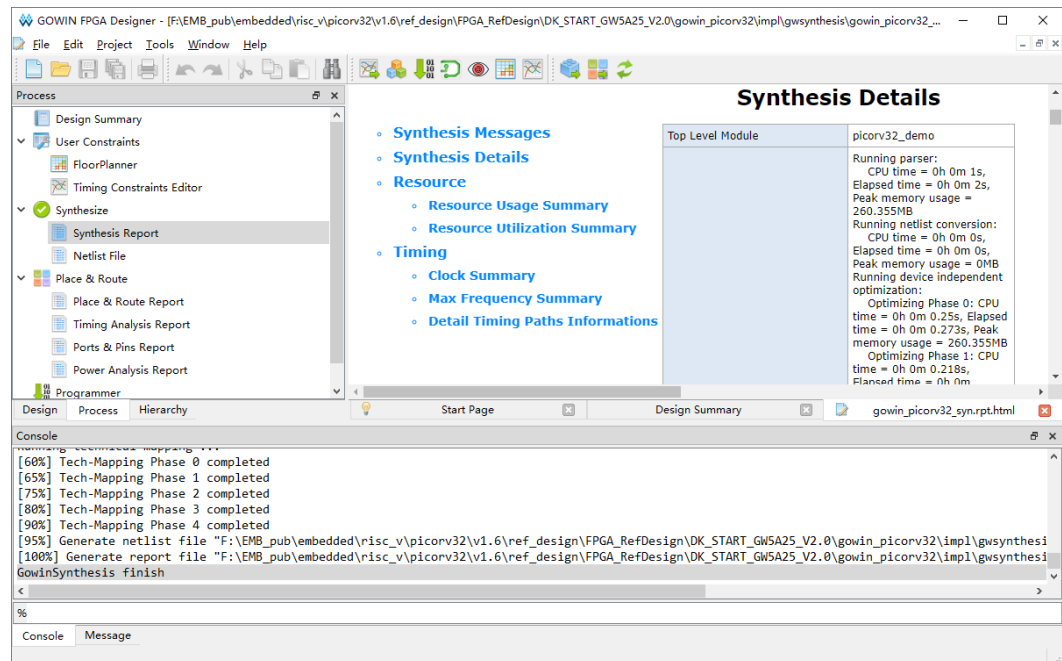
図 3-22 Dual-Purpose Pin の構成



3.6 合成

Gowin ソフトウェアの合成ツール GowinSynthesis で RTL 設計の合成を実行してネットリスト・ファイルを生成します(図 3-23)。

図 3-23 合成

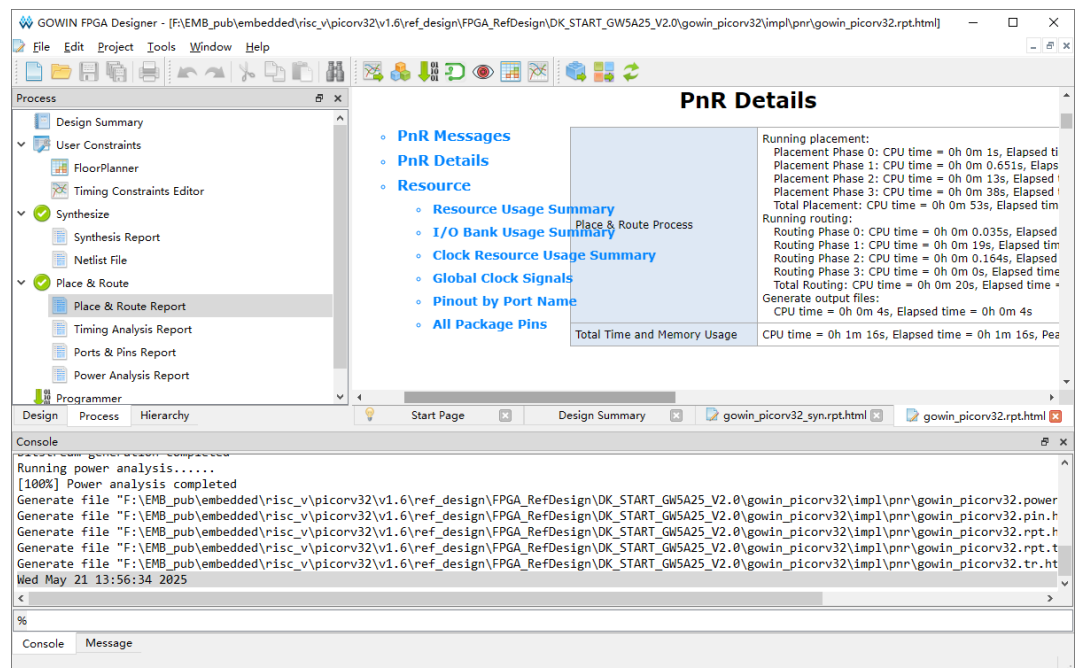


合成ツールの使用方法については、『Gowin ソフトウェア ユーザーガイド([SUG100](#))』を参照してください。

3.7 配置配線

Gowin ソフトウェアの配置配線ツール Place & Route を実行して配置配線を行い、ビットストリーム・ファイルを生成します(図 3-24)。

図 3-24 Place & Route



配置配線ツールの使用方法については、『Gowin ソフトウェア ユーザーガイド([SUG100](#))』を参照してください。

3.8 ダウンロード

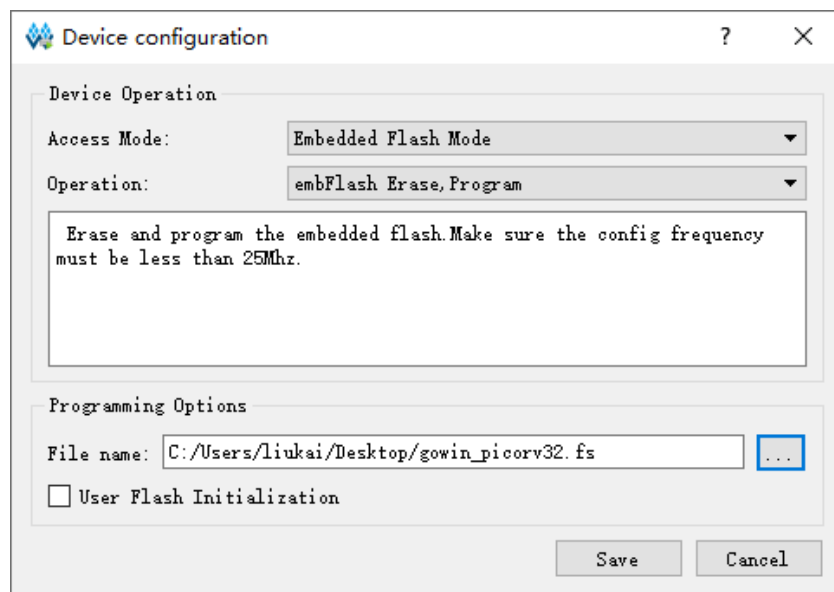
ダウンロードツールである **Programmer** を使用してビットストリーム・ファイルをダウンロードします。

Gowin ソフトウェアの画面またはインストールパスで **Programmer** を開いた後、メニューバーで「**Edit > Configure Device**」またはツールバーで「**Configure Device**」をクリックして、「**Device configuration**」ダイアログボックスを開きます。

開発ボード上のデバイスが **LittleBee** ファミリーFPGA 製品の場合、ダウンロード・オプションの構成は図 3-25 に示すとおりです。

- 「**Access Mode**」のドロップダウン・リストから「**Embedded Flash Mode**」を選択します。
- 「**Operation**」のドロップダウン・リストから「**embFlash Erase, Program**」または「**embFlash Erase, Program, Verify**」を選択します。
- 「**Programming Options > File name**」で、ダウンロードする必要のあるビットストリーム・ファイルをインポートします。
- 最後に「**Save**」をクリックします。

図 3-25 LittleBee ファミリーの **Embedded Flash Mode** の構成



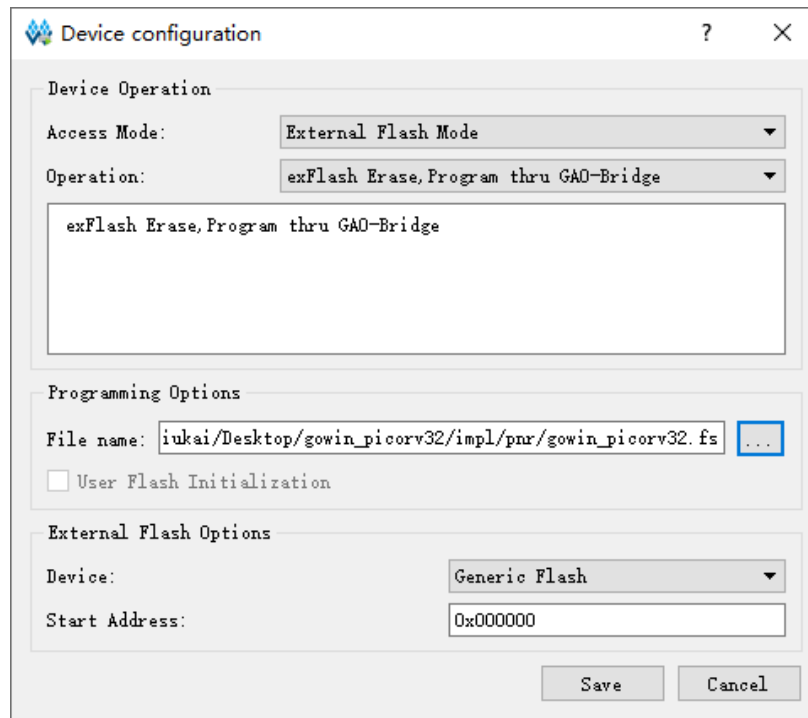
開発ボード上のデバイスが **Arora** ファミリーFPGA 製品の場合、ダウンロード・オプションの構成は図 3-26 に示すとおりです。

- 「**Access Mode**」のドロップダウン・リストから「**External Flash Mode**」を選択します。
- **Operation** のドロップダウン・リストから「**exFlash Erase, Program thru GAO-Bridge**」または「**exFlash Erase, Program, Verify thru GAO-Bridge**」を選択します。
- 「**Programming Options > File name**」で、ダウンロードする必要のある

るビットストリーム・ファイルをインポートします。

- 「External Flash Options > Device」 オプションで「Generic Flash」を選択します。
- 「External Flash Options > Start Address」は「0x000000」に構成します。
- 最後に「Save」をクリックします。

図 3-26 Arora ファミリーの External Flash Mode の構成



開発ボード上のデバイスが Arora V FPGA 製品の場合、ダウンロード・オプションの構成は図 3-27 に示すとおりです。

- 「Access Mode」のドロップダウン・リストから「External Flash Mode 5A」を選択します。
- 「Operation」のドロップダウン・リストから「exFlash Erase, Program 5A」または「exFlash Erase, Program, Verify 5A」を選択します。
- 「Programming Options > File name」で、ダウンロードする必要のあるビットストリーム・ファイルをインポートします。
- 「External Flash Options > Device」オプションで「Generic Flash」を選択します。
- 「External Flash Options > Start Address」は「0x000000」に構成します。
- 最後に「Save」をクリックします。

図 3-27 Arora V の External Flash Mode の構成

Device configuration

Device Operation

Access Mode: External Flash Mode 5A

Operation: exFlash Erase, Program 5A

exFlash Erase, Program 5A

Programming Options

File name: T_GW5A25_V2.0/gowin_picorv32/impl/pnr/gowin_picorv32.fs

☐ Retained User Flash Data

☐ User Flash Initialization

External Flash Options

Device: Generic Flash

Start Address: 0x000000

Save Cancel

Device configuration が完了したら、Programmer のツールバーの Program/Configure 「」 をクリックしてビットストリーム・ファイルをダウンロードします(図 3-28)。

図 3-28 ダウンロード

Gowin Programmer Version V1.9.11.02 (64-bit)

File Edit Tools About

USB Cable Setting

Enable	Series	Device	Operation	FS File	User Code	IDCODE
1 ☒	GW5A	GW5A-25A	exFlash Erase, Program 5A	F:/EMB_pub/embedded/risc_v/picorv32/v1.6/...	0x00007E44	0001281B

Output

```

Info Target Cable: Gowin USB Cable(WINUSB)/0/404/null@10MHz
Info Target Device: GW5A-25A(0x0001281B)
Info Operation "exFlash Erase, Program 5A" for device#1...
Info Status Code is: 0x36020238
Info Try to program spi flash: 0xC84017I
Info Programmng Flash starts from 0x000000.
Info Programmng Flash ends from 0x0D2700.
Info Program Flash finished.
Info Finished.
Info Cost 9.66 second(s)

```

Ready

Programmer の使用方法については、『Gowin Programmer ユーザーガイド([SUG502](#))』を参照してください。

4 リファレンス・デザイン

Gowin_PicoRV32 の[ハードウェア・リファレンス・デザイン](#) :

...

¥ref_design¥FPGA_RefDesign¥DK_START_GW2A18_V2.0¥gowin_picorv32 ;

...

¥ref_design¥FPGA_RefDesign¥DK_START_GW2A55_V1.3¥gowin_picorv32 ;

...

¥ref_design¥FPGA_RefDesign¥DK_START_GW5A25_V2.0¥gowin_picorv32 ;

...

¥ref_design¥FPGA_RefDesign¥DK_DP_GW5AT60_V1.0¥gowin_picorv32 ;

...

¥ref_design¥FPGA_RefDesign¥DK_START_GW5AST138_V1.0¥gowin_picorv32。

