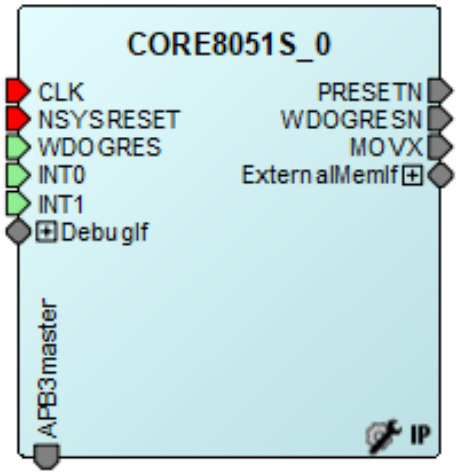
Microsemi Core8051s

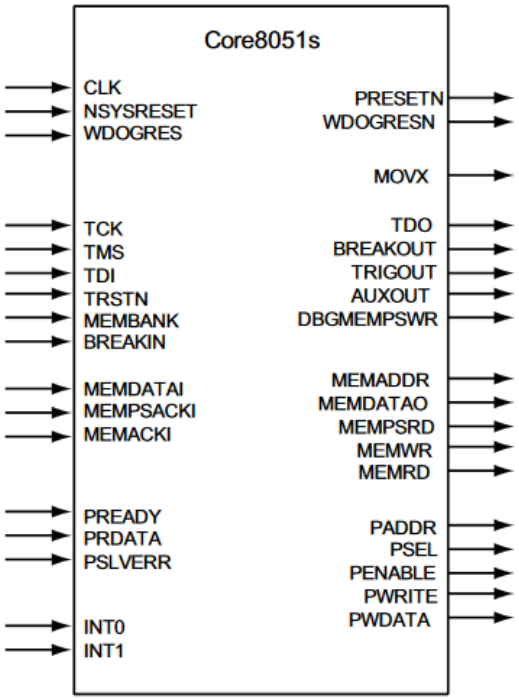
Core8051sの使い方

1. Core8051sの概要

Microsemi社が提供するソフトCPU IP



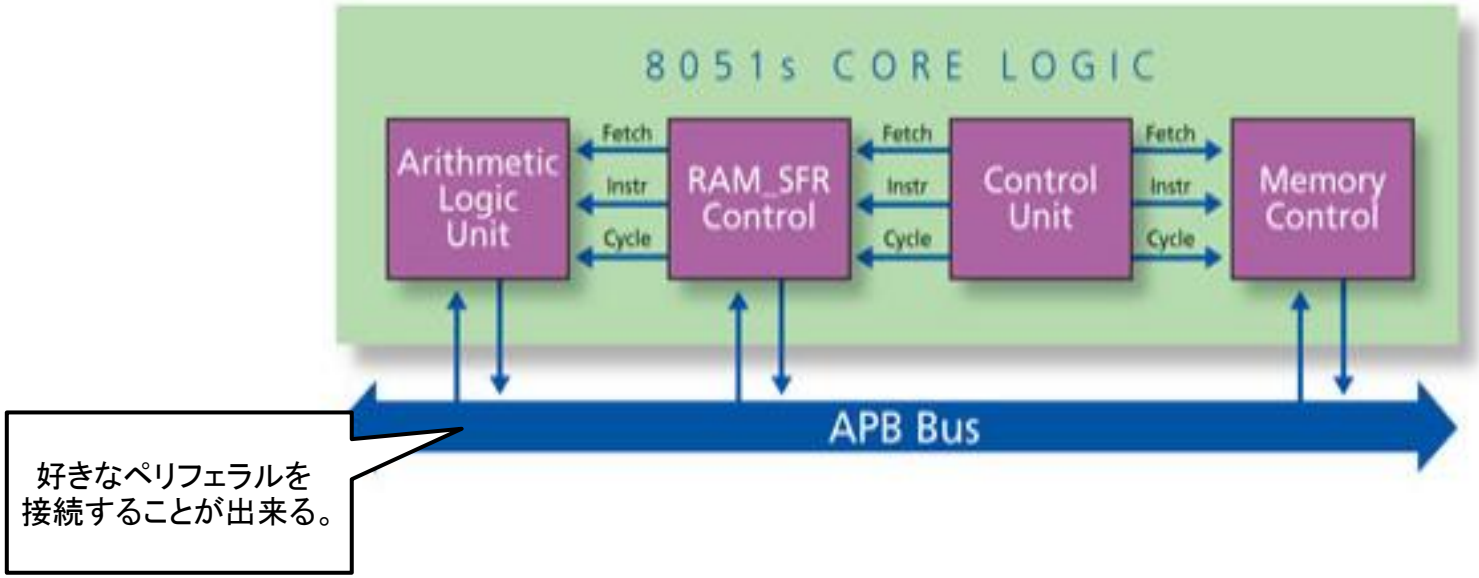
Core8051s



- 8051をベースに作られたCPU IP
- デフォルトのインタフェースはほとんどなくAPB3バス経由でインタフェースを拡張する形になっている。
- 開発言語は、SoftConsole環境にてC言語での開発が可能。コンパイラにはSDCCが使用されている。

2. Core8051sシステム

Core8051sの構成



- ARM社が提唱するAPBバスが実装されており、対応するペリフェラルを接続・拡張することができる。もともとの8051には、APBバスはなく、独自のバスが実装されていた。
- 8051マイクロコントローラの命令セットと互換性がある。
- いくつかの機能及び性能がオプション、もしく縮小されている。

3. 対応デバイス

Core8051sが使用可能なデバイス

Supported Actel FPGA Families for the Core8051s are as follows:

- IGLOO®/e/PLUS
- ProASIC3®/E/L
- Fusion
- ProASIC PLUS®
- Axcelerator®
- RTAX-S

多くのMicrosemi社製FPGAで使用することが出来る。

4. ピンアサイン

System Signals

ピン名称	方向	極性/BUS	説明
CLK	入力	↑	内部ロジック用のクロック入力。 この信号は、すべてのAPBペリフェラルのクロックにも使用する必要があります。
NSYSRESET	入力	負論理	ハードウェアリセット入力 発振器が動作している間に2クロックサイクルの間この信号が論理0になると、デバイスがリセットされます。
PRESETN	出力	負論理	同期リセット出力 この信号を使用してすべてのAPBペリフェラルをリセットする必要があります。
WDOGRES	入力	正論理	ウォッチドッグタイムアウト表示
WDOGRESN	出力	負論理	ウォッチドッグ用リセット信号
MOVX	出力	正論理	MOVX命令の実行

4. ピンアサイン

On-Chip Debug Interface (Optional)

ピン名称	方向	極性/BUS	説明
TCK	入力	↑	JTAGテストクロック OCIを使用しない場合は、ロジック1に接続してください。
TMS	入力	正論理	JTAGテストモード選択 OCIを使用しない場合は、ロジック0に接続してください。
TDI	入力	正論理	JTAGテストデータを入力します。OCIを使用しない場合は、ロジック0に接続します。
TDO	出力	正論理	JTAGテストデータ出力
TRSTN	入力	負論理	JTAGテストリセット OCIを使用しない場合は、ロジック1に接続してください。
MEMBANK	入力	4	オプションのコードメモリバンクの選択 使用しない場合は、ロジック0に接続してください。

4. ピンアサイン

On-Chip Debug Interface (Optional)

ピン名称	方向	極性/BUS	説明
BREAKIN	入力	正論理	バス入力を切断します。ハイにサンプリングされると、ブレークポイントが生成されます。使用しない場合は、論理0に接続してください。
BREAKOUT	出力	正論理	バス出力を中断します。これは、Core8051sがエミュレーションを停止したときにハイに駆動されます。これは、複数のプロセッサを接続するオープンドレインブレークバスに接続できるため、CPUが停止すると、バス上の他のすべてが数クロックサイクル以内に停止します。
TRIGOUT	出力	正論理	トリガ出力 この信号は、オプションで外部のテスタ機器に接続して、Core8051sの内部アクティビティとクロストリガすることができます。
AUXOUT	出力	正論理	補助出力 この信号は、OCIデバッガソフトウェアを介して制御できるオプションの汎用出力です。
DBGMEMPSWR	出力	正論理	デバッグプログラムストアの書き込み

4. ピンアサイン

External Interrupts

ピン名称	方向	極性/BUS	説明
INT0	入力	正論理	外部割り込み0（低優先）
INT1	入力	正論理	外部割り込み1（高優先）

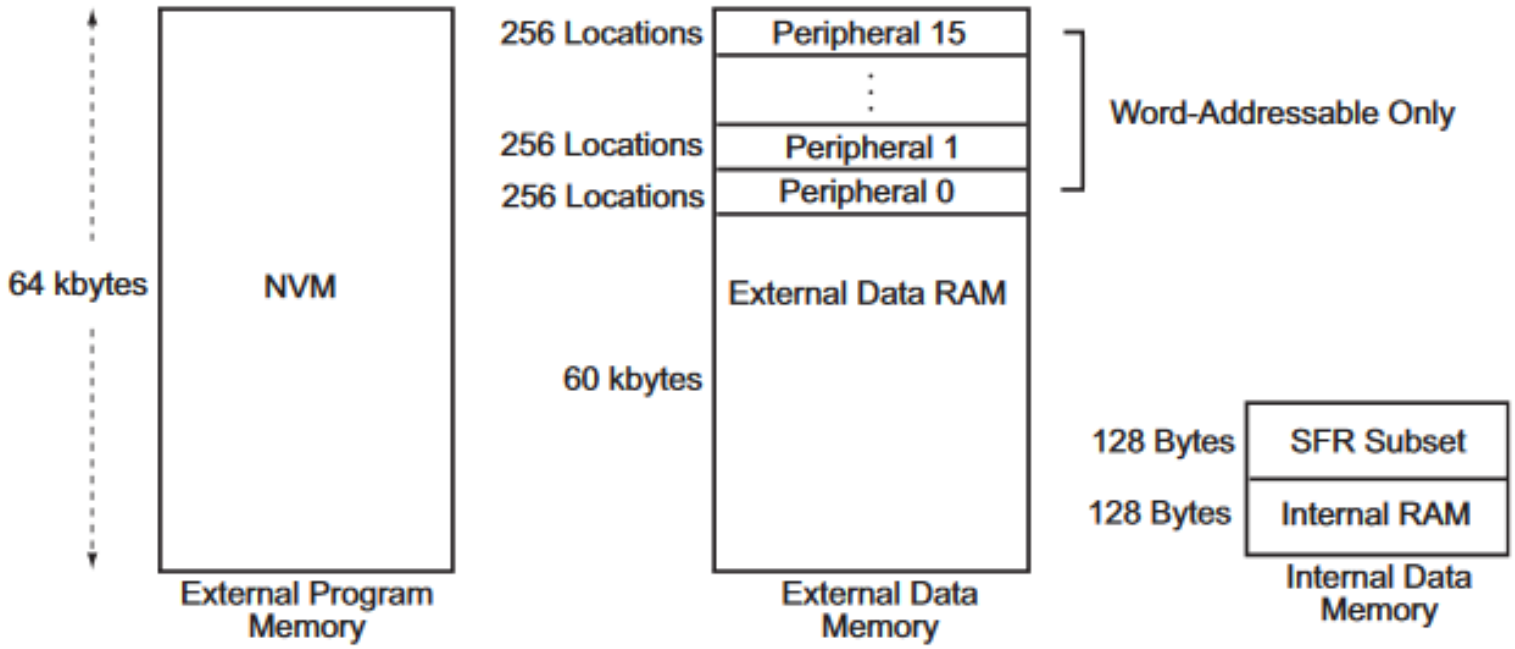
4. ピンアサイン

External Memory Bus Interface

ピン名称	方向	極性/BUS	説明
MEMPSACKI	入力	正論理	プログラムメモリリードアクノリッジ
MEMACKI	入力	正論理	データメモリアクノリッジ
MEMDATAI	入力	8	メモリデータ入力
MEMDATAO	出力	8	メモリデータ出力
MEMADDR	出力	16	メモリアドレス
MEMPSRD	出力	正論理	プログラムストアリードイネーブル
MEMWR	出力	正論理	データメモリライトイネーブル
MEMRD	出力	正論理	データメモリリードイネーブル

5. メモリ空間

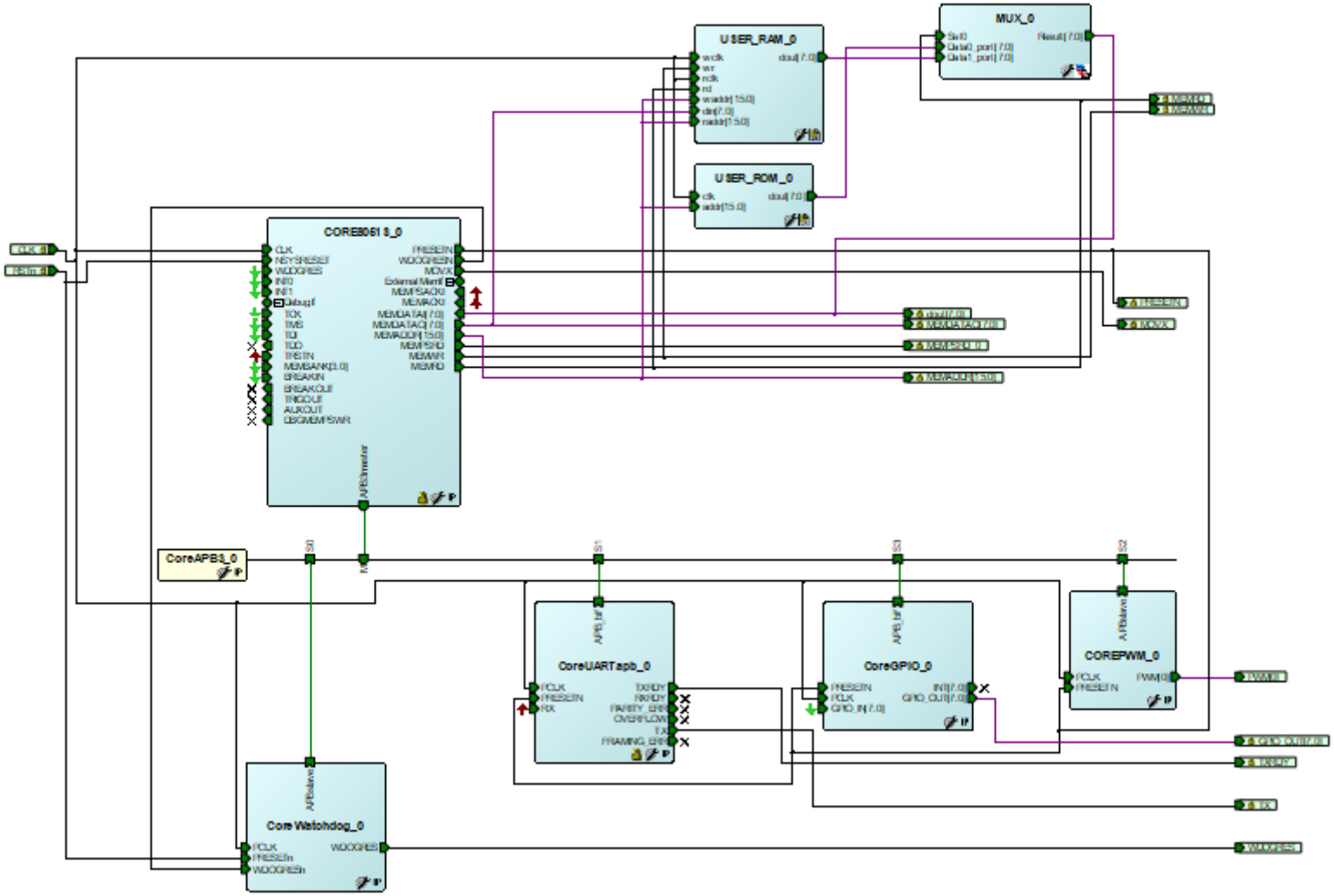
Core8051sのメモリ空間



Core8051sマイクロコントローラはハーバードアーキテクチャを利用し、別々のコードとデータスペースを持っています。Core8051sのメモリ構成は、業界標準8051のものと似ています。3つのメモリ領域があります。

・プログラムメモリ（内蔵RAM、外付けRAM、外付けROM）
・外部データメモリ（外部RAM）
・内部データメモリ（internal RAM）

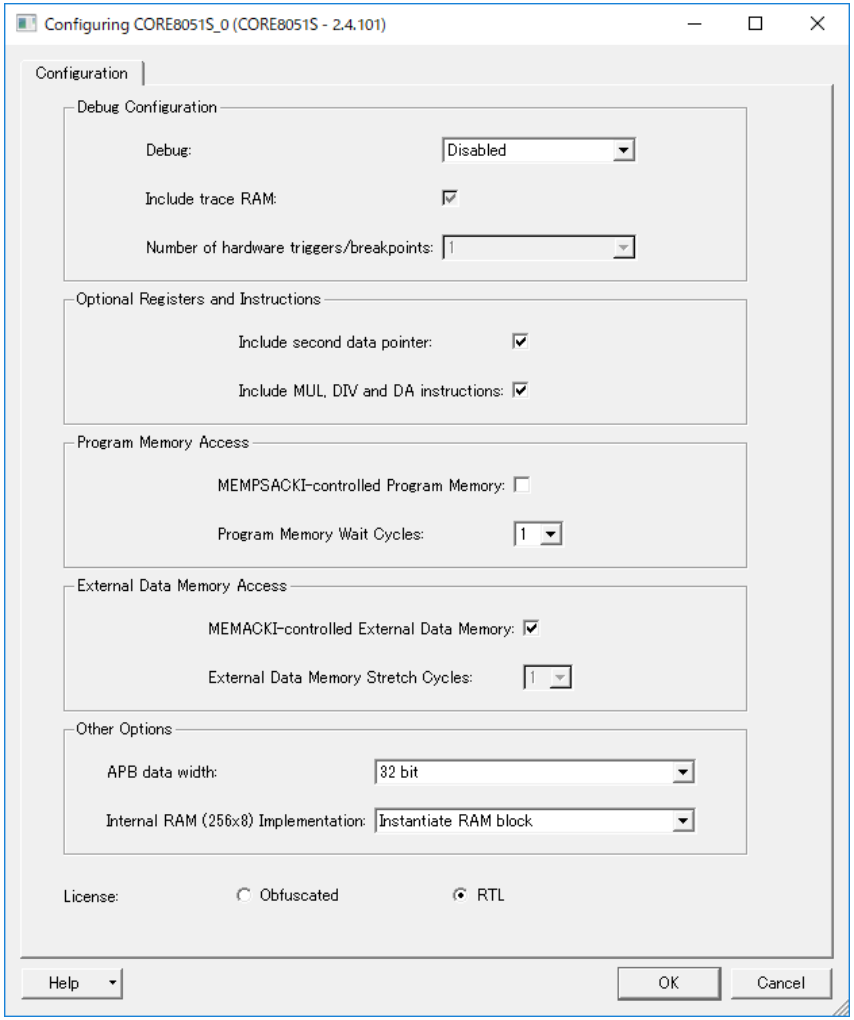
6. Core8051s構成例



ほぼすべてのMicrosemi社製のFPGAで使用することが出来る。

7. 各種設定

Core8051sの設定



-

7. 各種設定

CoreAPB3の設定

Configuring CoreAPB3_0 (CoreAPB3 - 4.1.100)

Configuration

Data Width Configuration

APB Master Data Bus Width: 32-bit / 16-bit / 8-bit

Address Configuration

Number of address bits driven by master: 12

Position in slave address of upper 4 bits of master address: [15:12] (Ignored if master address width >= 20 bits)

Indirect Addressing: Not in use

Allocate memory space to combined region slave

Slot 0:	Slot 1:	Slot 2:	Slot 3:
Slot 4:	Slot 5:	Slot 6:	Slot 7:
Slot 8:	Slot 9:	Slot 10:	Slot 11:
Slot 12:	Slot 13:	Slot 14:	Slot 15:

Enabled APB Slave Slots

Slot 0: ☑	Slot 1: ☑	Slot 2: ☑	Slot 3: ☑
Slot 4:	Slot 5:	Slot 6:	Slot 7:
Slot 8:	Slot 9:	Slot 10:	Slot 11:
Slot 12:	Slot 13:	Slot 14:	Slot 15:

Testbench: User

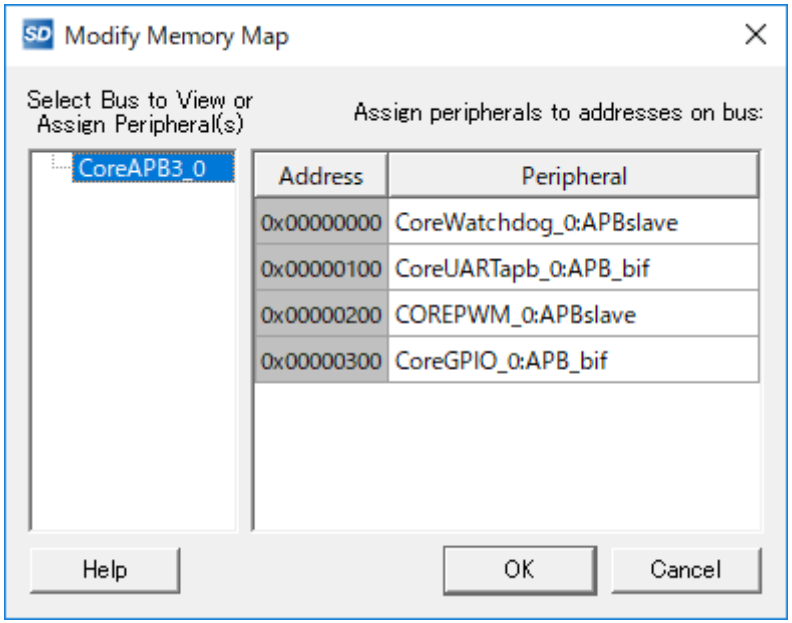
License: Obfuscated / RTL

Help | OK | Cancel

-

7. 各種設定

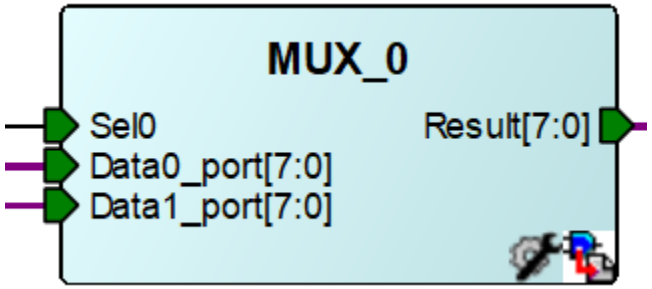
CoreAPB3のMemoryMap設定

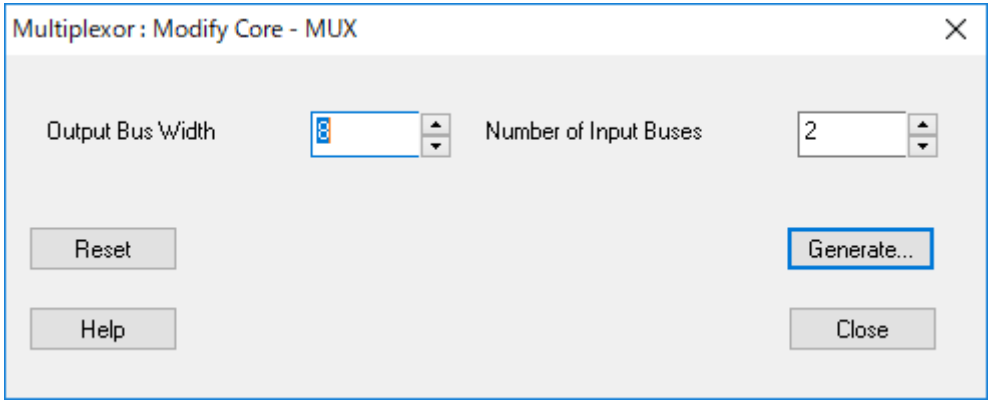


CoreAPB3モジュール上で右クリック、「Modify Memory Map」を選択すると、上記画面が表示される。

7. 各種設定

Multiplexorの設定

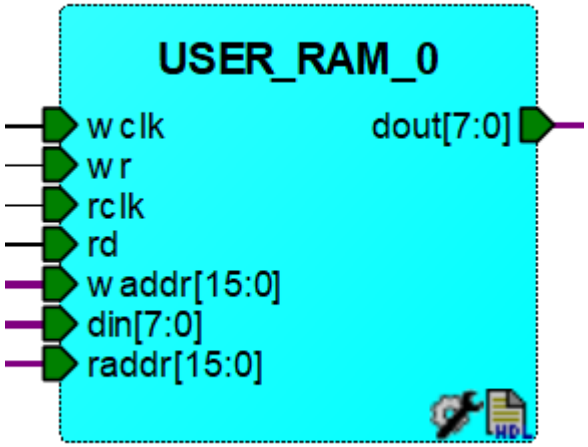




IPカタログのBasic Blocksの項にあるMultiplexorを使用する。

7. 各種設定

USER_RAMの設定



ロジックを使用してRAMを作るRTL（MicrosemiのIPカタログにはないモジュール）

VHDLのgenericで値を以下のように変更する。

```
WIDTH  :  integer := 8;      -- data width
DEPTH  :  integer := 256;    -- RAM depth
ASIZE  :  integer := 16      -- address width
```

7. 各種設定

USER_ROMの設定

ロジックを使用してROMを作るRTL（MicrosemiのIPカタログにはないモジュール）

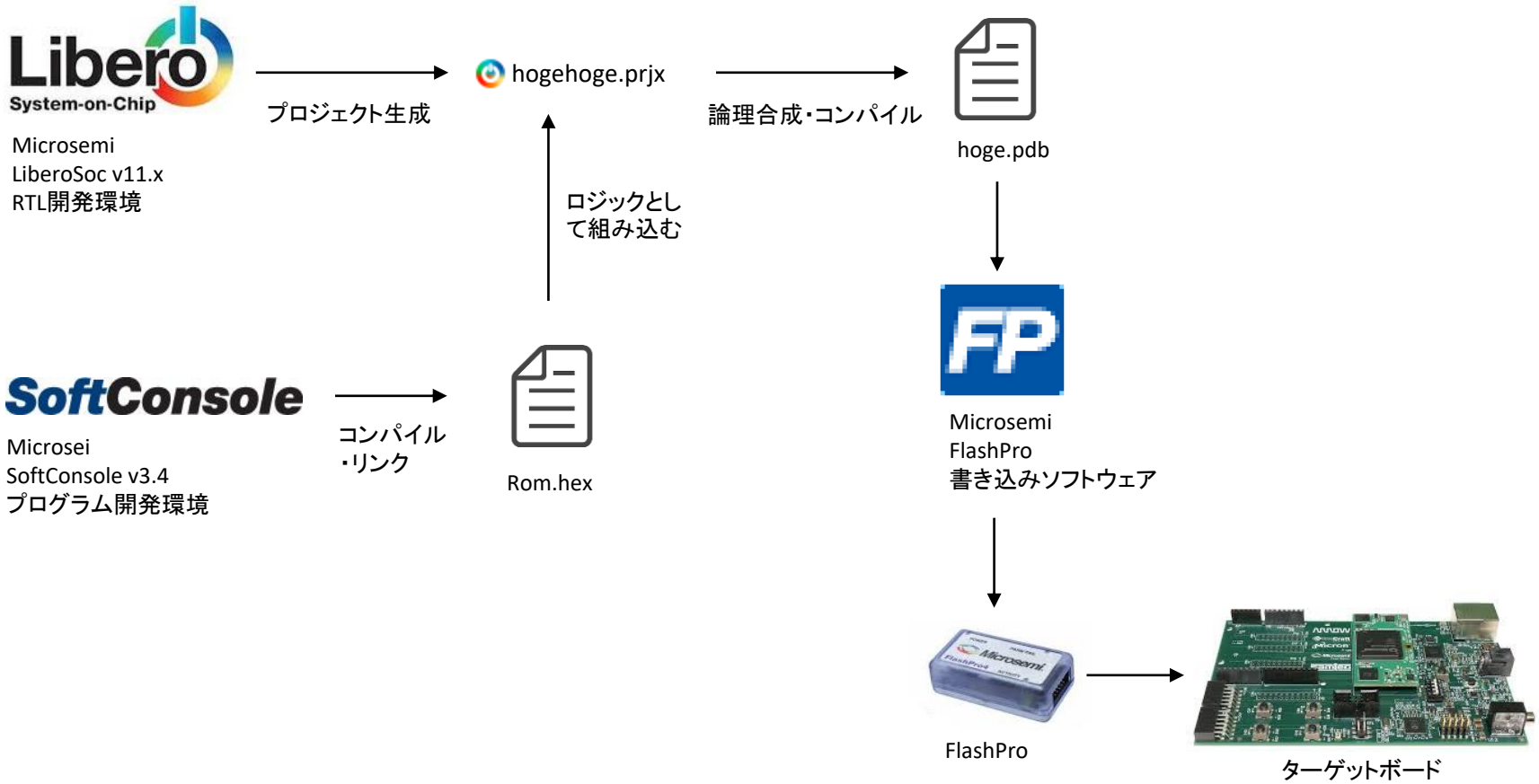
VHDLのgenericで値を以下のように変更する。

```
WIDTH    :  integer:= 8;          -- data width
DEPTH    :  integer:= 2048;       -- ROM depth def:4096/8192/
ASIZE    :  integer:= 16;         -- address width
ROMFILE  :  string:= "rom.hex“
```

ROM化したいプログラムファイルをhdlフォルダにrom.hexという名前で置く。

8. 開発環境

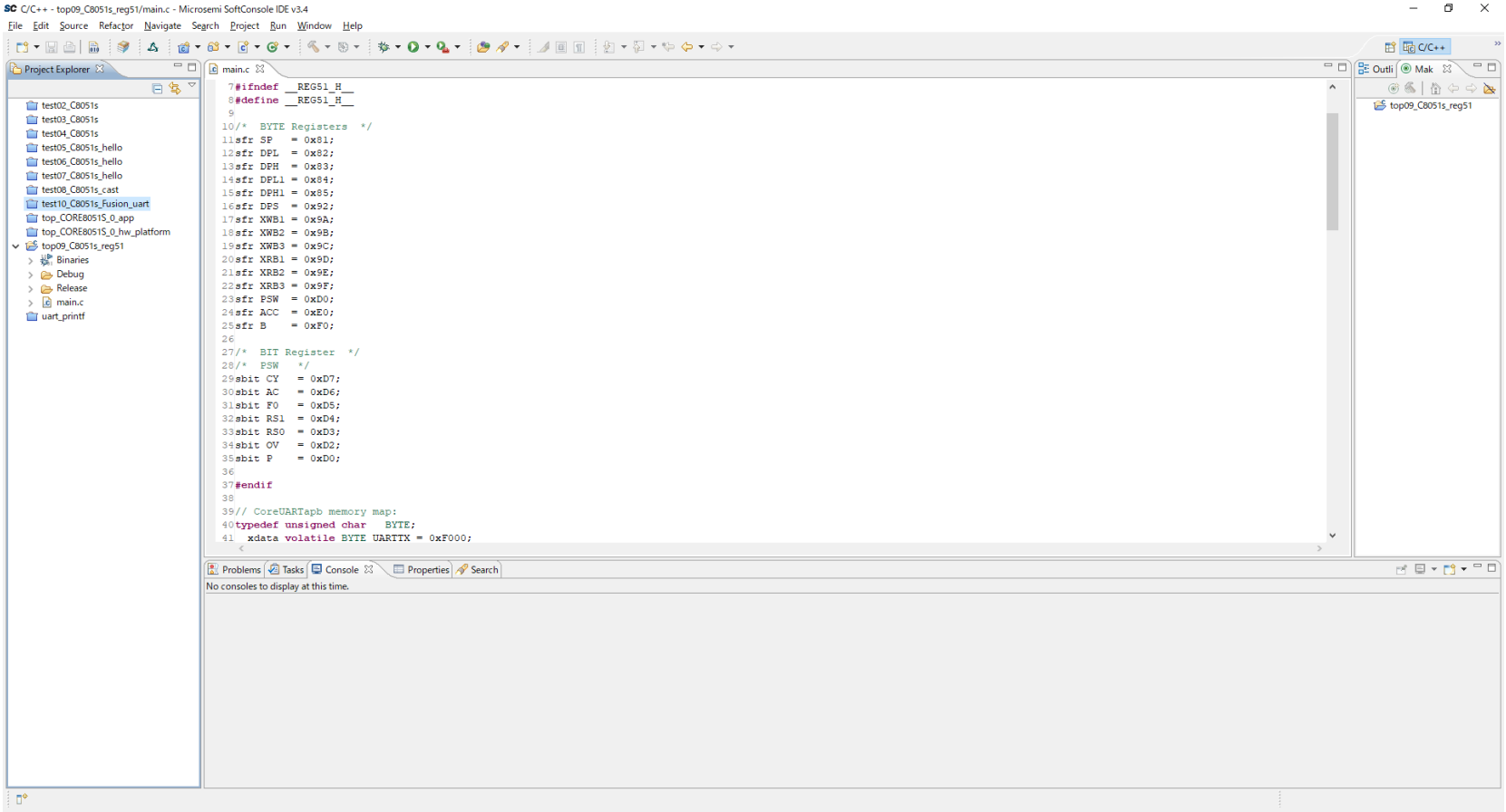
当該プロジェクトの開発環境



現状は、プログラムをRTLとして組み込む形でしか成功していない。
RTLはそのままでプログラムだけ書き換えできると良いのだが、、、

9. プログラム開発

SoftConsoleの開発画面



名前はSoftConsoleだが、中身はEclipse
細かい設定はされているので、基本的にはそのまま使用できる。

10. プログラム開発

APB3バス経由でのUARTモジュールの使い方

```
#define base_wdt((0x100 *  0) + 0xF000)
#define base_uart((0x100 *  1) + 0xF000)
#define base_GPIO((0x100 *  2) + 0xF000)
```

```
// CoreUARTapb memory map:
__xdata atbase_uart+ 0x00 unsigned char TxData;
__xdata atbase_uart+ 0x04 unsigned char RxData;
__xdata atbase_uart+ 0x08 unsigned char Scon1;
__xdata atbase_uart+ 0x0C unsigned char Scon2;
__xdata atbase_uart+ 0x10 unsigned char SStat;
```

・base_***は、MEMMAPのアドレスを記述する。これはユーザのRTLの設計によって変わる。
・__xdata hogehogeの方は、CoreUARTapb内のアドレスであり、これはモジュールによって決まっている固定値（データシート参照）である。

11. プログラム開発

APB3バス経由でのUARTモジュールの使い方

```
void write_uart (char c)
{
        while (!( SStat  & 0x01));
        /* wait until transmitter ready */
        TxData = c;       /* output character */
        return;
}
```

・1Byteの文字を送信する関数を定義
While文で送信READYがOKになるまで待つ。その後、TxDataアドレスに送信する文字を書き込む。
・返り値は無し。

12. プログラム開発

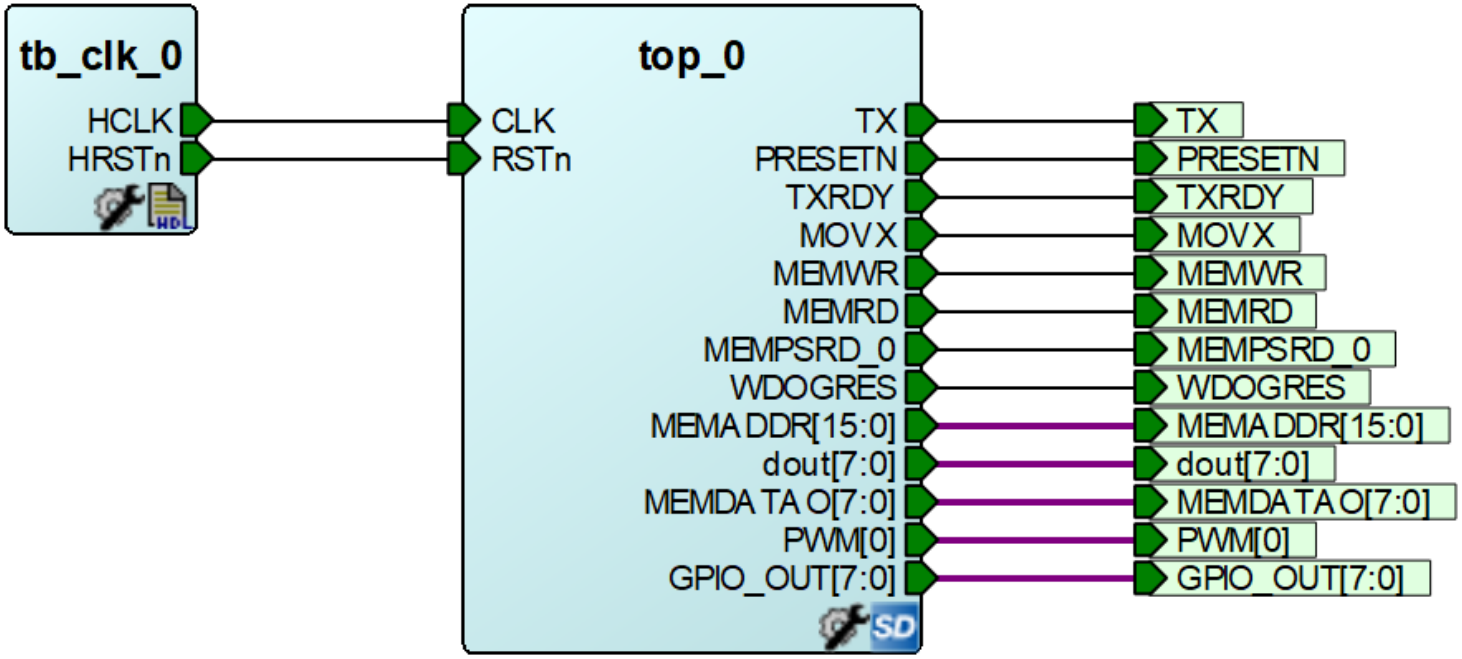
コンパイルの生成ファイル

名前	更新日時	種類	サイズ
main.adb	2019/05/26 9:22	Actel Designer De...	3 KB
main.asm	2019/05/26 9:22	Assembler Source	12 KB
main.d	2019/05/26 9:22	D ファイル	1 KB
main.lst	2019/05/26 9:22	C/ASM File	28 KB
main.rel	2019/05/26 9:22	REL ファイル	9 KB
main.rst	2019/05/26 9:22	RST ファイル	28 KB
main.sym	2019/05/26 9:22	C Symbols File	44 KB
makefile	2019/05/26 9:22	ファイル	3 KB
memory-map.xml	2019/05/26 9:22	XML ドキュメント	1 KB
objects.mk	2019/05/26 9:22	Makefile	1 KB
sources.mk	2019/05/26 9:22	Makefile	1 KB
subdir.mk	2019/05/26 9:22	Makefile	2 KB
top09_C8051s_reg51	2019/05/26 9:22	ファイル	2 KB
top09_C8051s_reg51.cdb	2019/05/26 9:22	CDB ファイル	6 KB
top09_C8051s_reg51.elf	2019/05/26 9:22	ELF ファイル	4 KB
top09_C8051s_reg51.hex	2019/05/26 9:22	HEX File	2 KB
top09_C8051s_reg51	2019/05/26 9:22	ショートカット	1 KB
top09_C8051s_reg51.lst	2019/05/26 9:22	C/ASM File	5 KB
top09_C8051s_reg51.map	2019/05/26 9:22	Linker Address Map	17 KB
top09_C8051s_reg51.mem	2019/05/26 9:22	MEM ファイル	2 KB
top09_C8051s_reg51.srec	2019/05/26 9:22	SREC ファイル	1 KB

コンパイラには、SDCCが用いられており、生成物としてはIntel-Elf形式の***.hexファイルである。

13. シミュレーション結果

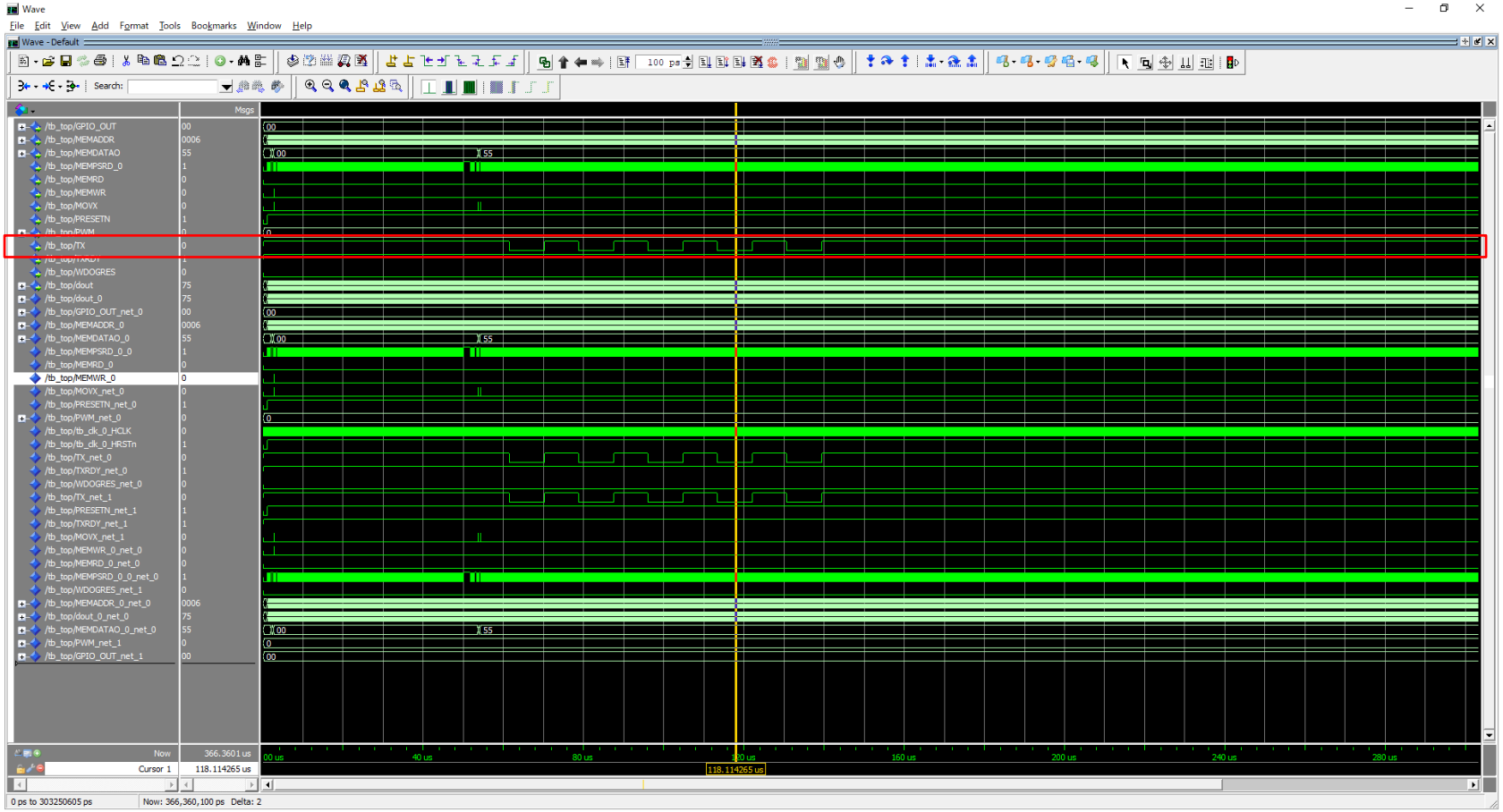
シミュレーション用のTESTBENCHを以下に示す。



tb_clkは、CLK信号（20MHz）とRSTn信号（負論理、最初0しばらくした後1となる）を出力するRTLである。

13. シミュレーション

生成したhexファイルをRTLに組み込み、シミュレーション実施



Txから出力されていることが確認できる。

Microsemi®