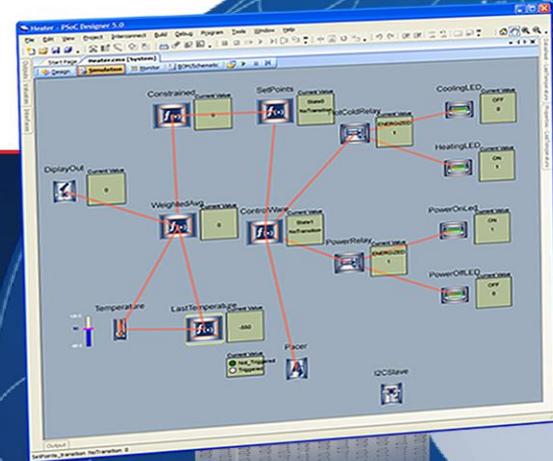




PSoCのアーキテクチャ

PSoC Architecture for PSoC Experiment Lab



Experiment Course Material 2 V2.11
April 9th, 2019
EM2.pptx (23 Slides)
Renji Mikami

ユーザーモジュール

設計済みで、機能・特性評価済みの
デジタルおよびアナログブロック

ハードウェア設計の簡素化

オンチップ周辺回路として利用可能

- ADC's, DAC's, PGA's, Filters
- Timers, Counters, PWM's
- UART, SPI, I2C

ユーザーモジュールの機能仕様設定(パラメタライズ)
はPSoC Designerから容易に可能

ユーザーモジュールが提供するもの

- デバドラ Application Programmer Interfaces (APIs)
- 割り込みサービスルーチン Interrupt Service Routines (ISRs)
- データシート(コーディング・サンプル含む)



ユーザーモジュールの例

Digital Comm

- CRC16
- EzI2Cs
- I2CHW
- I2Cm
- IrDARX
- IrDATX
- RX8
- SPIM
- SPIS
- TX8
- UART

Timers

- Timer8
- Timer16
- Timer24
- Timer32

Counters

- Counter8
- Counter16
- Counter24
- Counter32

PWMs

- PWM8
- PWM16
- PWMDB8
- PWMDB16

Random Seq

- PRS8
- PRS16
- PRS24
- PRS32

Misc Digital

- DigBuf
- DigInv
- E2PROM
- LCD
- Sleep Timer

Protocols

- USB
- USBFS

ADCs

- ADCINC
- ADCINC12
- ADCINC14
- ADCINCVR
- DELSIG8
- DELSIG11
- DUALADC
- DUALADC8
- DelSig
- SAR6
- TRIADC
- TRIADC8

MUXs

- AMUX4
- AMUX8
- RefMux
- CSR

Analog Comm

- DTMFDialer

Temperature

- Flash Temp

Amplifiers

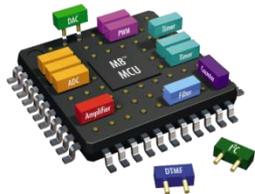
- AMPINV
- CMPPRG
- INSAMP
- PGA

Filters

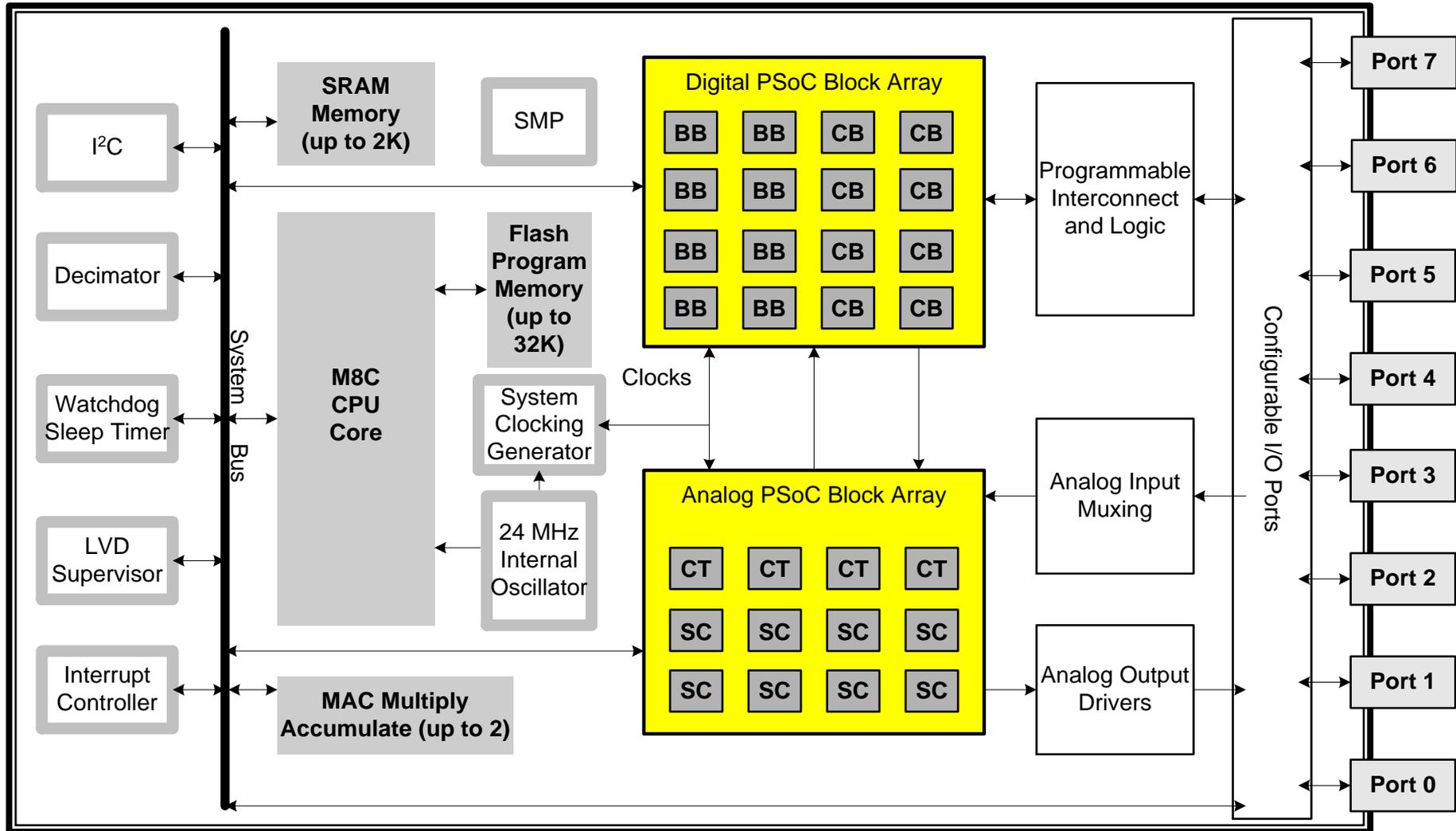
- BPF2
- LPF2

DACs

- DAC6
- DAC8
- DAC9
- MDAC6
- MDAC8



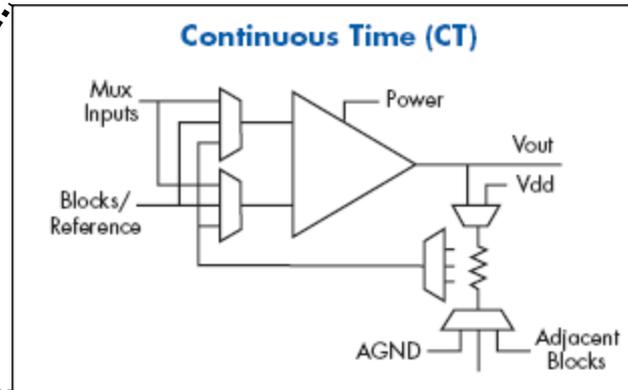
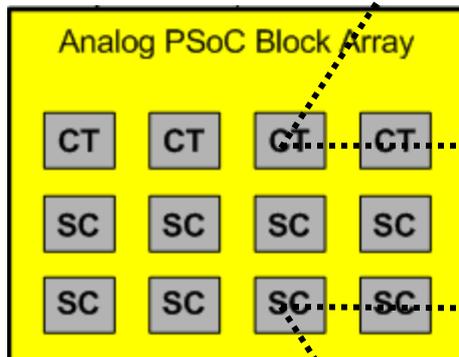
PSoC内部ブロック図(各ブロックはさまざまなUMに変身)



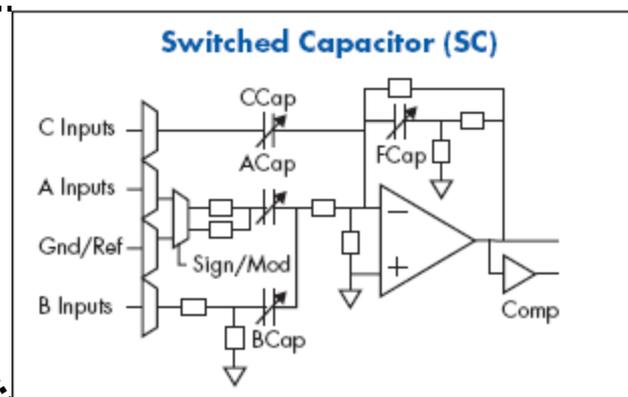
アナログブロック

実現できるユーザーモジュール

ブロック内部



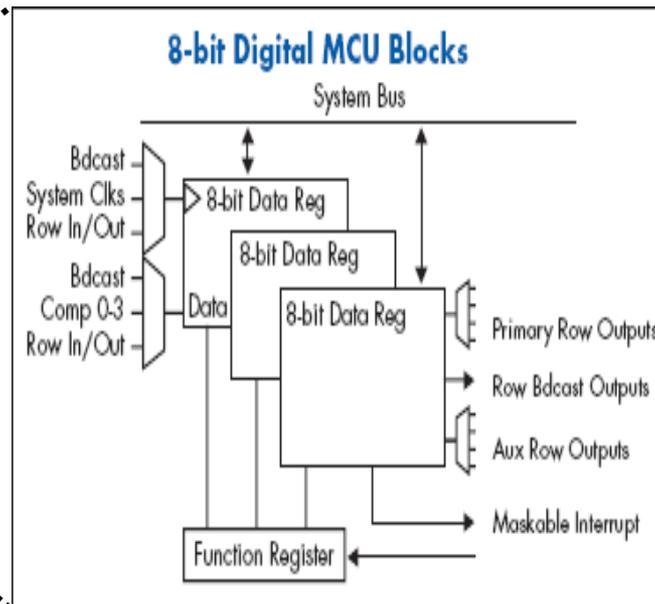
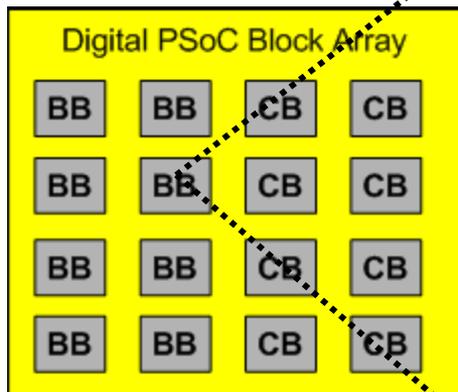
(CTは抵抗タップ型OPアンプ)



(SCはスイッチトキャパシタ型OPアンプ)

- ADC
 - Incremental 6-14 bits
 - Delta Sigma 6-13 bits
- DAC
 - 6, 8, and 9 bit
 - 6 and 8 bit multiplying
- Filters
 - 2-pole Low-pass
 - 2-pole Band-pass
- DTMF Dialer
- Modulator
- Peak Detector
- V to I Converter
- Amplifiers
 - Programmable Gain
 - Instrumental
 - Inverting
- Comparators
 - Programmable
 - Hysteresis
 - Zero-Crossing
- CapSense

デジタルブロック

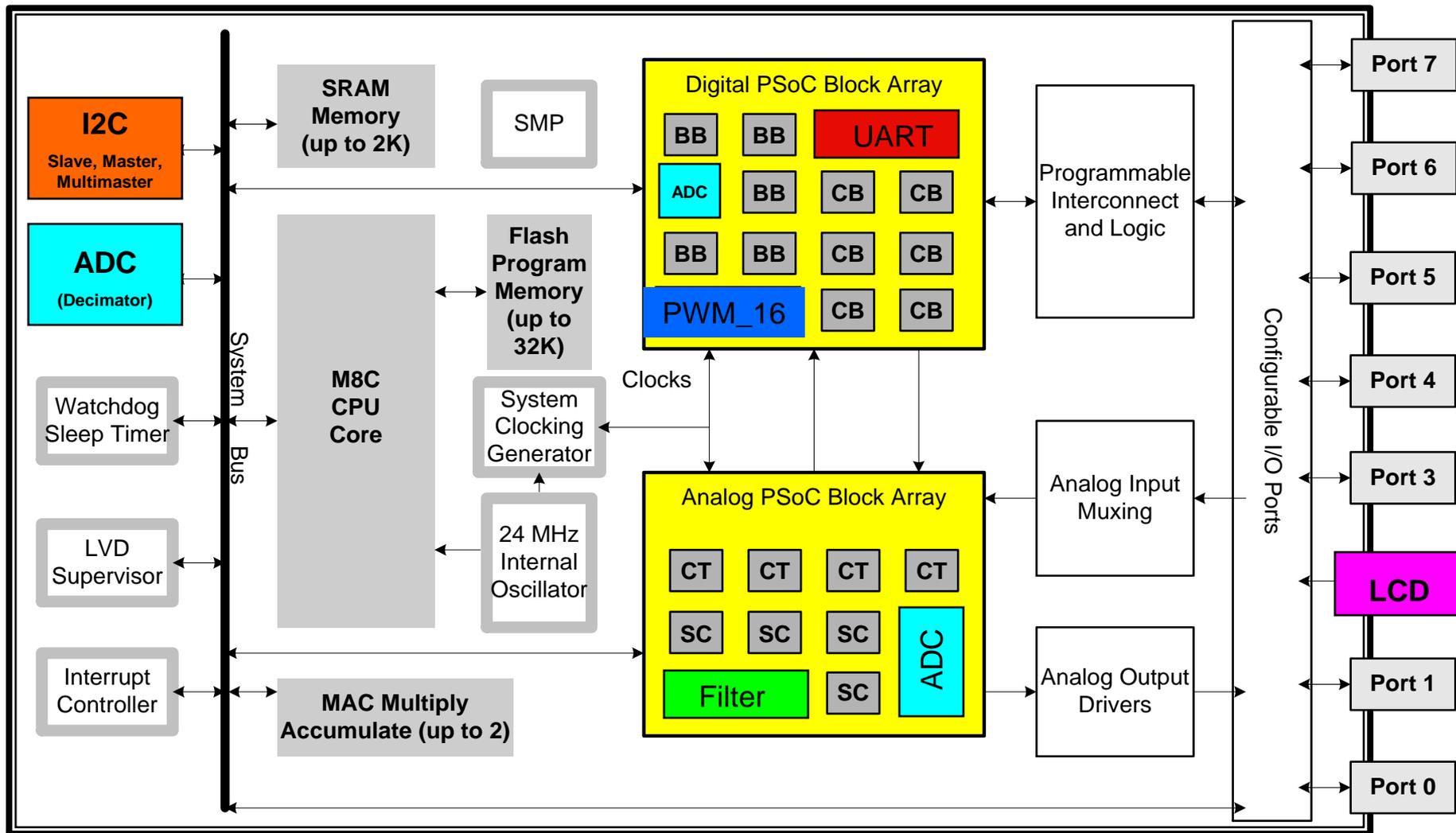


- Timer
8, 16, 24, 32 bit
- Counter
8, 16, 24, 32 bit
- PWM
8, 16, 24, 32 bit
- Dead Band Generator
8, 16, 24, 32 bit
- Pseudo Random Source
- Cyclic Redundancy Check
- Communication Interface
 - I2C Master
 - I2C Slave
 - SPI Master
 - SPI Slave
 - Full Duplex UART
 - Tx, Rx
 - Full Speed USB v2.0

BB:Basic Block は8ビットのカウンタがベース
CB:Communication Block はシフトレジスタにより
シリアル・コミュニケーションに使用できる
どちらも連結して多ビット化(16,24,32Bit化)できる

内部回路の配置

ユーザーモジュール単位で設計
UMは内部リソースで自動構成



ユーザーモジュール データシート

- ユーザーモジュールデータシートに含まれる情報
 - Resources
 - Features
 - Overview
 - Diagram
 - Description
 - Timing
 - Specs
 - Placement
 - Parameters
 - API
 - Sample Code
 - Registers

Datasheet - PWM8

Home Resources Features Overview Diagram Description Timing Specifications Placement Parameters API SampleCode

8-Bit Pulse Width Modulator Data Sheet

PWM8 vX.Y

Copyright © 2000-2007, Cypress Semiconductor. All Rights Reserved.

リソース	PSoC [®] ブロック			APIメモリ(バイト)		ピン(外部入出力ごと)
	デジタル	アナログCT	アナログSC	フラッシュ	RAM	
CY8C29/27/24/22/21xxx, CY7C64215/603xx, CYWUSB6953, CY8CLED02/04/08/16						
8ビット	1	0	0	67	0	1
16ビット	2	0	0	89	0	1

リソース	PSoC [®] ブロック			APIメモリ(バイト)		ピン(外部入出力ごと)
	デジタル	アナログCT	アナログSC	フラッシュ	RAM	
CY8C26/25xxx						
8ビット	1	0	0	103	0	1
16ビット	2	0	0	138	0	1

このユーザーモジュールを使用した機能プロジェクトの実現例については、www.cypress.com/psocexampleprojectsを参照してください。

機能と概要

- 8および16ビットの汎用パルス幅変調器は、それぞれ1個または2個のPSoCブロックを使用します。
- 最大48 MHzのソース クロック速度。
- 各パルス サイクルに対する周期の自動再ロード。
- パルス幅をプログラム可能。
- 入力信号による連続カウンタ動作のイネーブル/ディスエーブル。
- 出力の立ち上がりエッジまたはタイマー カウントでの割り込みオプション。

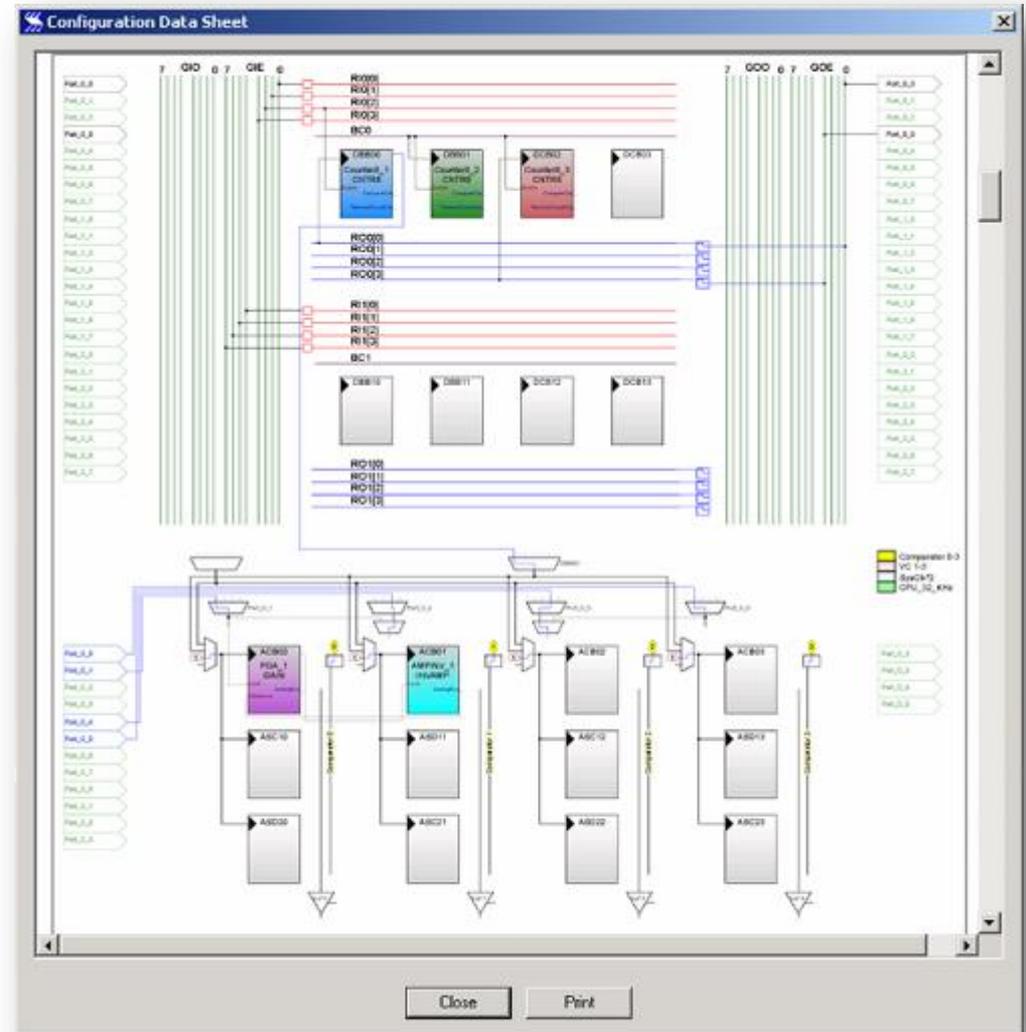
8ビットおよび16ビットのPWMユーザーモジュールは、周期とパルス幅をプログラム可能なパルス幅変調器です。クロックおよびイネーブル信号は、各種信号源から選択できます。出力信号はピンに配線するか、グローバル出力バスのいずれか1つに配線して、チップ内部の他のユーザーモジュールで使用できます。出力の立ち上がりエッジ、またはカウンタがタイマー カウント条件に達した時点での割り込みをプログラム可能です。

Period Register Pulse Width Register

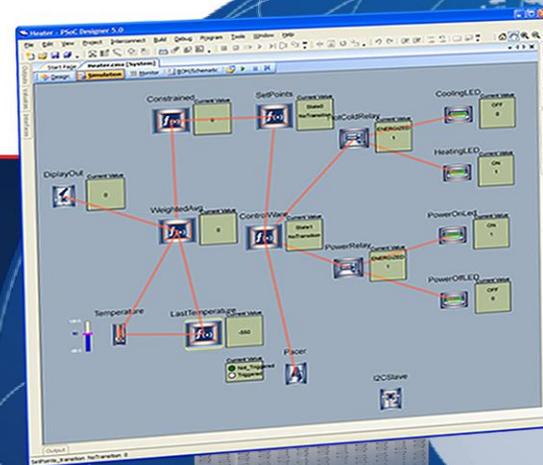
バージョン
により英文の
場合もあります

Project コンフィグレーションデータシート

- プロジェクトコンフィグレーションのデータシートには内蔵ハードウェアの配置配線情報が記載される
- 配置画面は印刷可能
- ユーザー定義のピンにはマーキングすることができ、データシート内で詳細情報が記載される
- レポートに有用



PSoC 開発環境とフロー



開発ソフトウェア

PSoC Designer

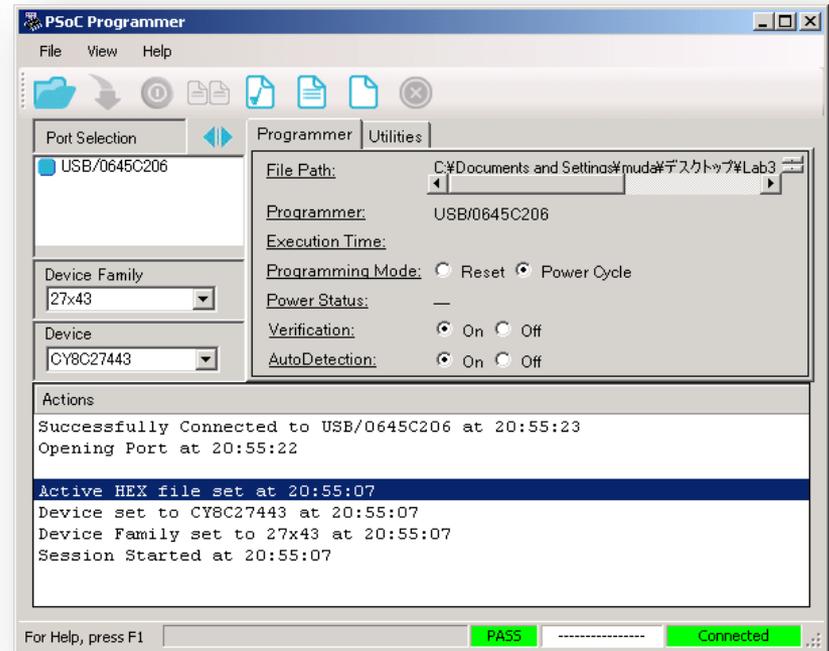
統合開発環境

C言語/アセンブリ言語



PSoC Programmer

プログラム書き込みソフト



Cコンパイラ

Hi-Tech C Pro US \$1495

Hi-Tech C Lite **Free** Image Craft ICCM8C \$249

設計フロー

1. ユーザーモジュールの設置
2. 回路の配線
3. Global Resource, GPIO, UM パラメータの入力
GC:Generate Config の実行
(PSoCレジスタのコンフィギュレーションデータ生成)
4. コーディング
Build の実行
(プログラムのコンパイルとBuildを実行,HEX ファイル生成)
5. インシステム・プログラミング
(PSoC MiniProgで書き込み)

ハードウェア初期設定

ソフトウェアの記述

このプログラム内でレジスタ値を書き換えることでハードウェア機能を自由に更新,変更できる

1. ユーザーモジュールの配置

User Modules

- ADCs
 - ADGINC
 - ADGINC12
 - ADGINC14
 - ADGINCVR
 - DELSIG11
 - DELSIG8
 - DUALADC
 - DUALADC8
 - SAR6
 - TRIADC
 - TRIADC8
- Amplifiers
 - AMPINV
 - CmpLP
 - CMPPRG
- INSAMP
 - Three OpA
 - Two OpA
- PGA
- Analog Comm
 - DTMDialer
- Counters
- DACs
 - DAC6
 - DAC8
 - DAC9
 - MDAC6
 - MDAC8
- Digital Comm
- Filters
 - BPF2
 - BPF2A
 - BPF2B
 - BPF2V
 - BPF2VA
 - LPF2
 - LPF2A
 - LPF2B
 - LPF2V
 - LPF2VA
- Misc Digital
- MUXs
- Protocols
- BootLdrI2C
- PSOC_GENERIC

Datashheet - LCD

Home Resources Features Overview Diagram Description Parameters

LCD Tool Box Data Sheet

LCD vX.Y

Copyright © 2002-2007, Cypress Semiconductor. All Rights Reserved.

リソース	PSoC [®] ブロック			APIメモリ(バイト)	
	デジタル	アナログCT	アナログSC	フラッシュ	RAM
CY8C29/27/26/25/24/22/21xxx, CY7C603xx/64215, CYWUSB953, CY8C20x34, CY8CCL					
横グラフあり	0	0	0	646	0
横グラフなし	0	0	0	434	0

このユーザ モジュールを使用した機能プロジェクトの実現例については、www.cypress.com/psocexampleprojectsを参照してください。

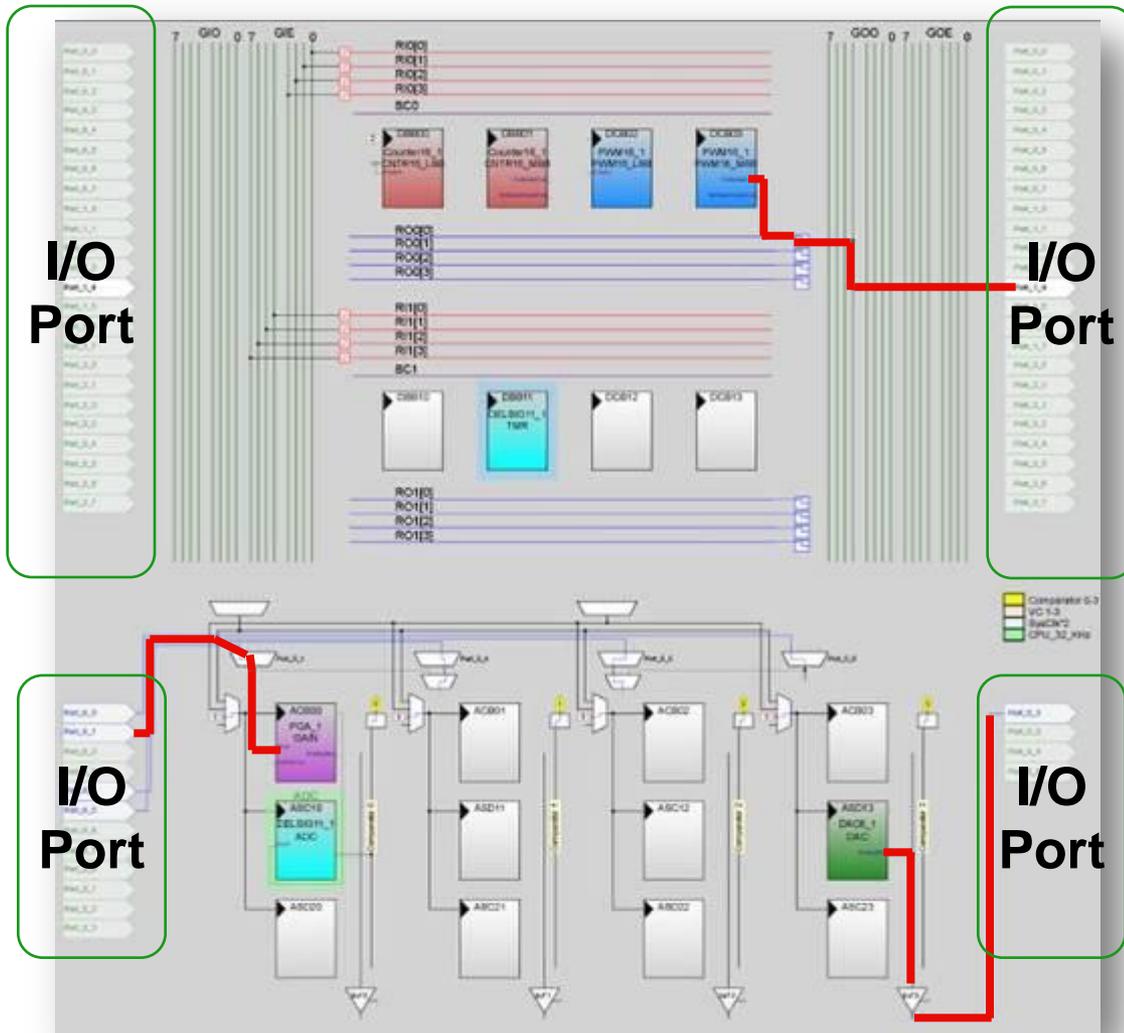
機能と概要

- 業界標準の日立HD44780 LCDディスプレイドライバチッププロトコルを使用。
- I/Oピンは7本のみ必要。
- RAMまたはROMの文字列を出力するルーチン。
- 数値を出力するルーチン。
- 水平および垂直横グラフを表示するルーチン。
- 1つのI/Oポートを使用。

LCDツールボックス ユーザ モジュールは、テキスト文字列と書式付きの数値を共通の2行または4行のLCDモジュールに書き込むライブラリ ルーチンのセットです。これらのLCDモジュールの文字グラフィック機能を使用して、水平および垂直横グラフがサポートされています。このモジュールは、業界標準の日立HD44780 2行x16文字LCDディスプレイドライバチップ用に特別に開発されたものですが、その他の多数の4行ディスプレイにも対応します。このライブラリは、4ビット インタフェース モードを使用する必要があるI/Oピンの数を少なくしています。

日本語化された
ユーザーモジュール
データシート

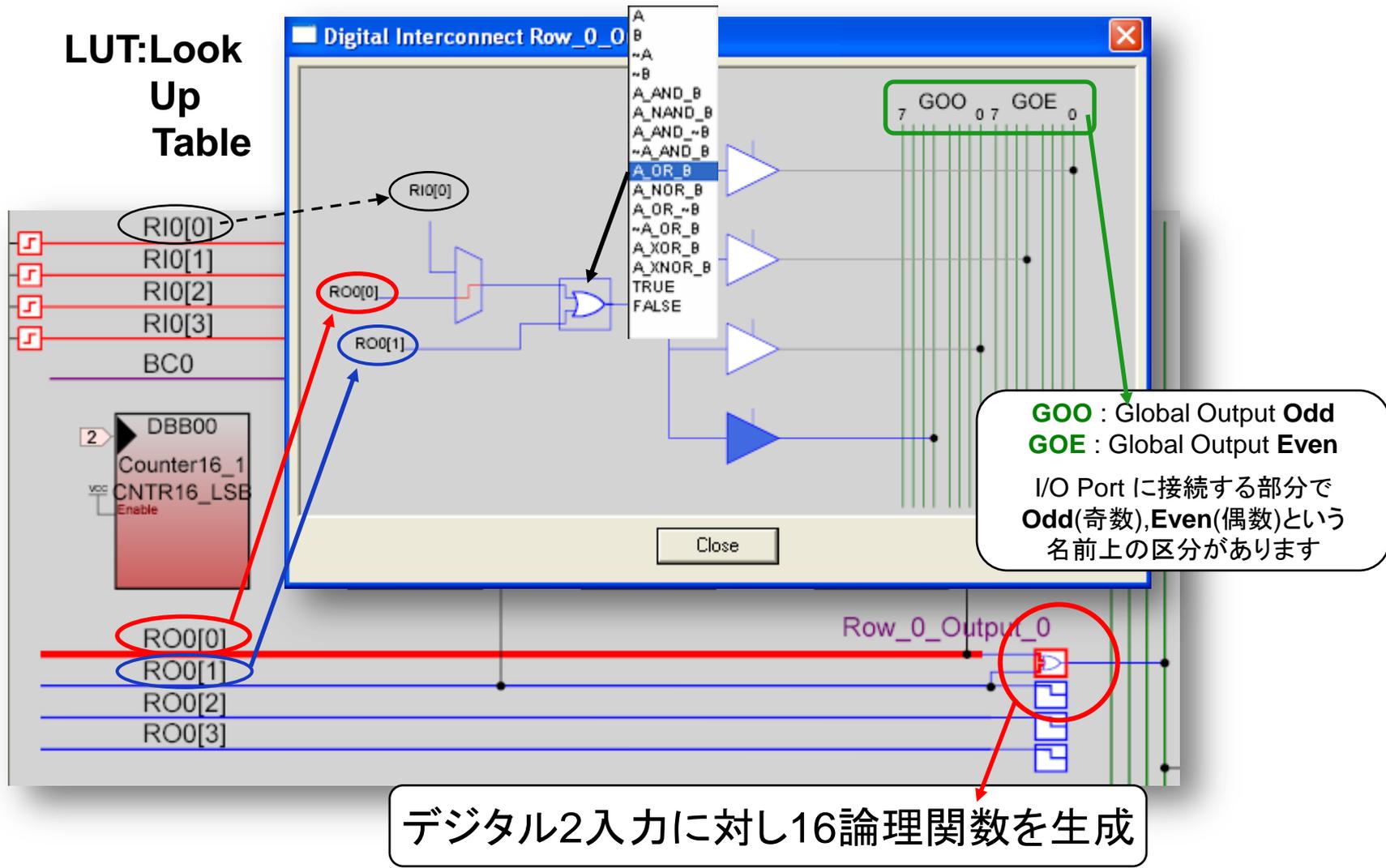
2.0 回路の配線



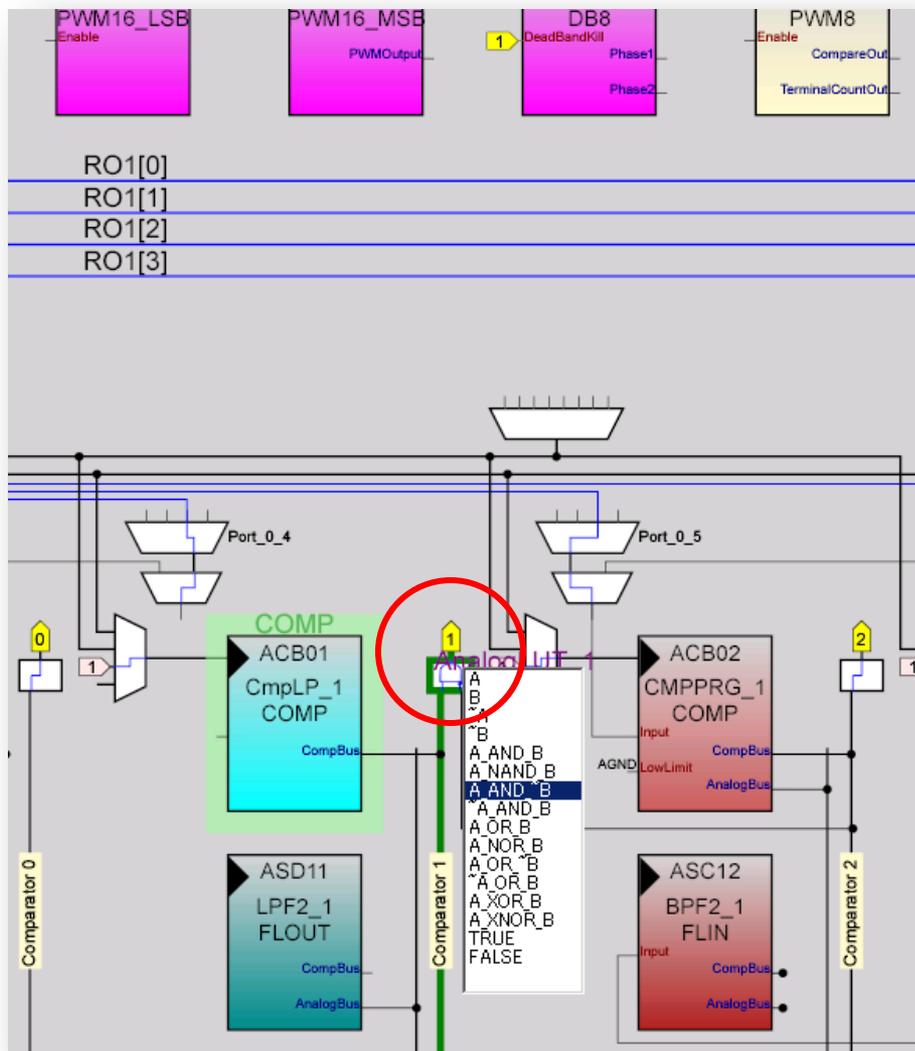
配線は、ユーザー
モジュールから
I/Oポート方向に
向けてつないで
いくとやりやすい

Port2は、
3210キットでは
LCDユニットに
接続されている。

2.1 ロジックLUTの設定



2.2 アナログコンパレータLUTの設定



アナログコンパレータ
出力を入力として2入力
16論理関数を生成します

3.0 ユーザーモジュールプロパティへの パラメータの入力

Name	CmpLP_1
User Module	CmpLP
Version	1.0
CompBus	ComparatorBus_1
RefValue	0.250 Vdd

Name	PWM8_2
User Module	PWM8
Version	2.5
Clock	VC2
Enable	High
CompareOut	Row_0_Output_1
TerminalCountOut	None
Period	250
PulseWidth	150
CompareType	Less Than Or Equal
InterruptType	Terminal Count
ClockSync	Sync to SysClk
InvertEnable	Normal

ユーザーモジュールの
機能パラメータは、
ダイアログボックスで
自由に設定できます
M8CMPUはこの設定に基づいて
起動コンフィギュレーション時に
機能設定レジスタに
データを書き込みます
これによってハードウェアの
機能が設定されます

つまり

すべてのハードウェアは
レジスタの値でまります

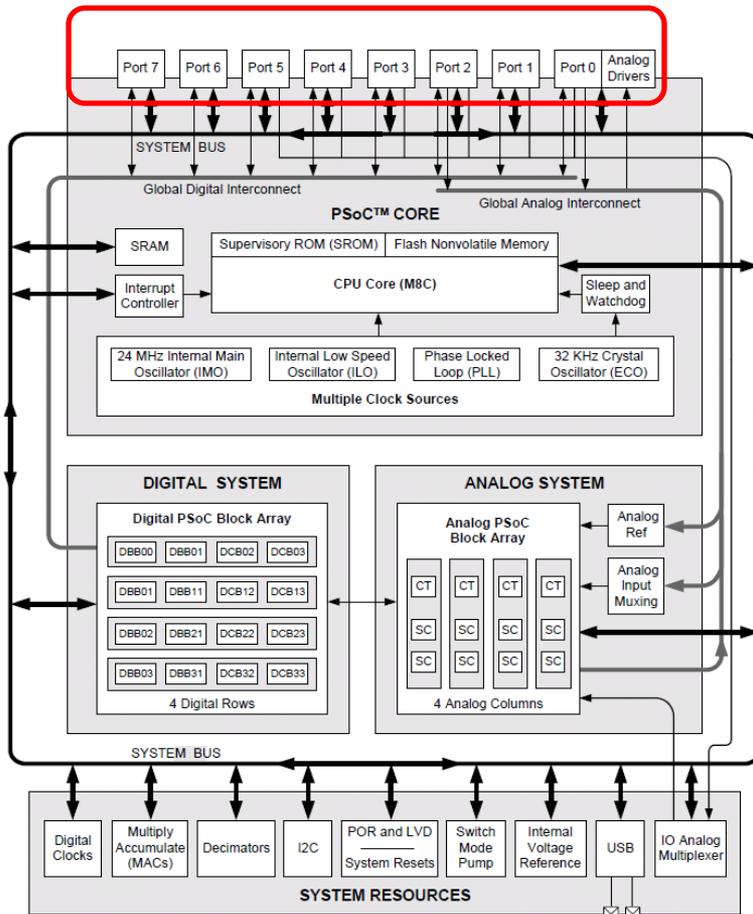
とすれば

ソフトウェアのプログラムでも
書き換えができてしまいます

3.1 GPIO(Pinout)の設定

Port 0 から Port 7

P0[4] : Port 0 の4bit目



<input type="checkbox"/>	P0[0]	Port_0_0, GlobalOutEven_0, Strong, DisableInt
<input type="checkbox"/>	P0[1]	Port_0_1, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P0[2]	Port_0_2, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P0[3]	Port_0_3, StdCPU, High Z Analog, DisableInt
<input checked="" type="checkbox"/>	P0[4]	Sw1, StdCPU, Pull Up, FallingEdge
Name		Sw1
Port		P0[4]
Select		StdCPU
Drive		Pull Up
Interrupt		FallingEdge
<input type="checkbox"/>	P0[5]	Port_0_5, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P0[6]	Port_0_6, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P0[7]	Port_0_7, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P1[0]	Port_1_0, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P1[1]	Port_1_1, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P1[2]	Port_1_2, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P1[3]	Port_1_3, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P1[4]	Port_1_4, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P1[5]	Port_1_5, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P1[6]	Port_1_6, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P1[7]	Port_1_7, StdCPU, High Z Analog, DisableInt
<input type="checkbox"/>	P2[0]	LCDD4, StdCPU, Strong, DisableInt
<input type="checkbox"/>	P2[1]	LCDD5, StdCPU, Strong, DisableInt
<input type="checkbox"/>	P2[2]	LCDD6, StdCPU, Strong, DisableInt
<input type="checkbox"/>	P2[3]	LCDD7, StdCPU, Strong, DisableInt
<input type="checkbox"/>	P2[4]	LCDE, StdCPU, Strong, DisableInt
<input type="checkbox"/>	P2[5]	LCDRS, StdCPU, Strong, DisableInt
<input type="checkbox"/>	P2[6]	LCDRW, StdCPU, Strong, DisableInt
<input type="checkbox"/>	P2[7]	Port_2_7, StdCPU, High Z Analog, DisableInt

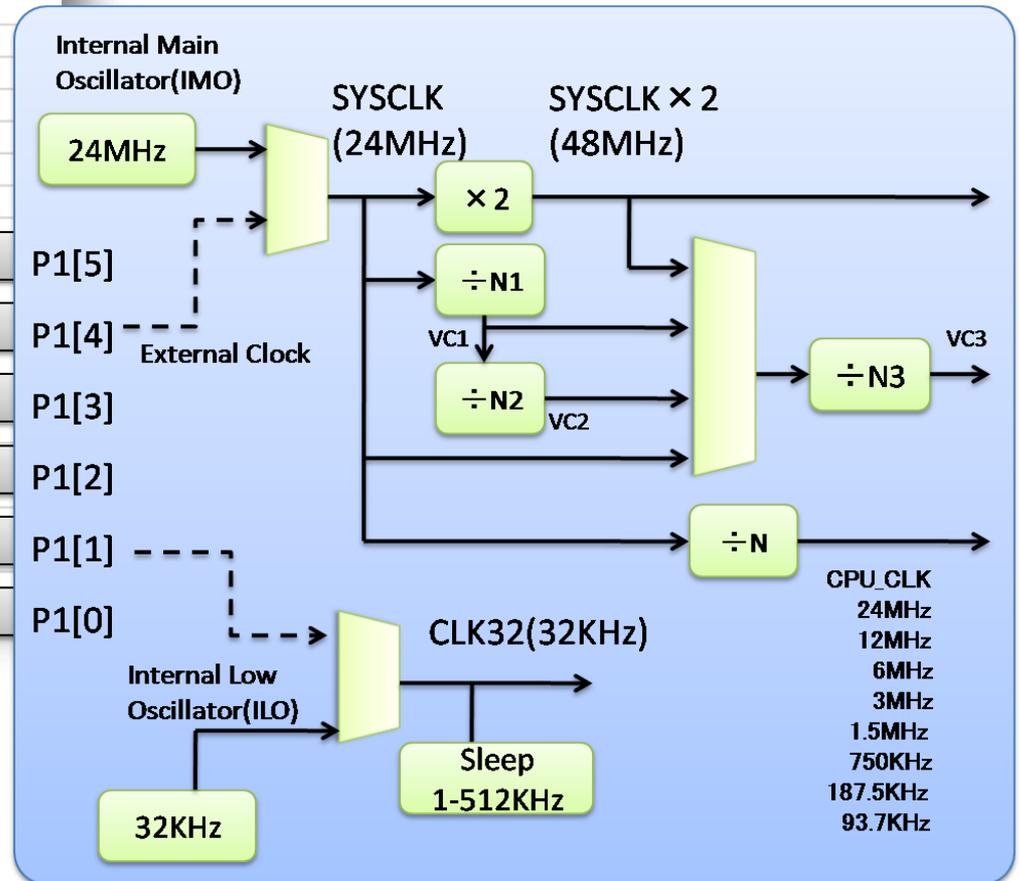
GPIOは極めて高機能

3.2 グローバルリソース-クロック系

Global Resources - lab2_pwm_lcd	
CPU_Clock	3_MHz (SysClk/8)
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	16
VC2= VC1/N	16
VC3 Source	VC2
VC3 Divider	250
SysClk Source	Internal 24_MHz
SysClk*2 Disable	No
Analog Power	SC On/Ref Low
Ref Mux	(Vdd/2)+/-BandGap
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
SwitchModePump	OFF
Trip Voltage [LVD (SMP)]	4.81V (5.00V)
LVDThrottleBack	Disable
Supply Voltage	5.0V
Watchdog Enable	Disable

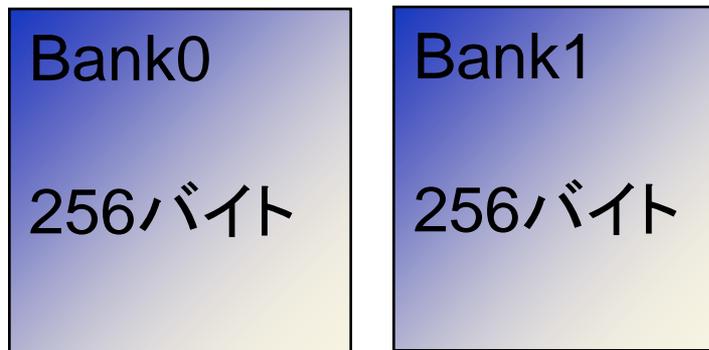
外部水晶
(オプション)

グローバル・リソースの
パラメータ設定



この機能設定もすべてレジスタ値に対応しますから
ソフトウェアプログラムから変更できます

3.3 PSoCレジスタとM8 MPU



PSoC機能設定レジスタは2バンク構成で各256バイトのアドレスを持ちます。以下の機能設定を行います

M8レジスタ(Fレジスタ)RAMページング
割り込みコントローラ設定
デジタルブロック設定
アナログブロック設定
GPIO設定
オシレータ・PLL設定

M8 MPU のレジスタ構成



バンクの切り替えはFレジスタで行います。

全てのインターナルレジスタは8ビット。ただし、PCは16ビット

A,X,PCレジスタはリセット後、0クリアされます

SPは次のスタック位置を指します。0XFFを指している状態でPUSH命令を実行すると0X00へロールします。

4. Cソースのコーディング(main.c)

```
Start Page lab3_adc [Chip] main.c
1 //-----
2 // C main line
3 //-----
4
5 #include <m8c.h> // part specific constants and macro
6 #include "PSoCAPI.h" // PSoC API definitions for all User
7
8 |
9 void main()
10 {
11     unsigned int adc_data;
12     PGA_Start(PGA_HIGHPOWER);
13     LCD_Start();
14     LCD_InitBG(LCD_SOLID_BG);
15     M8C_EnableGInt;
16     ADCINC_Start(ADCINC_HIGHPOWER);
17     ADCINC_GetSamples(0);
18     while(1){
19         while(ADCINC_fIsDataAvailable() == 0);
20         adc_data = ADCINC_wClearFlagGetData();
21         LCD_Position(0,0);
22         LCD_PrHexInt(adc_data);
23         LCD_DrawBG(1,0,16,(adc_data/50));
24     }
25 }
26
```

M8Cは最初に
ハードウェア機能
設定をします
(Bank0/Bank1
レジスタに初期データを
書き込みます)

このデータはGUIで
パラメータ入力した
ものです

実際のデータ生成はC
ソースのコンパイル前
のGCで行われます

GC:Generate Configuration

5. 書き込み(ISP)とデバッグ

The screenshot displays the PSoC Designer 5.0 IDE interface. The main window shows assembly code for `startup.as` with the following content:

```
43  __stack_start__:
44  psect  bss0, class=RAM, space=1
45  psect  nvram0, class=RAM, space=1
46  psect  rbit0, bit, class=RAM, space=1
47  psect  nvbit0, bit, class=RAM, space=1
48  psect  ramdata0, class=RAM, space=1
49  psect  romdata0, class=BANKROM, space=0
50
51  ;Declare areas defined in usermodules and other assembler code
52  psect  InterruptRAM, class=RAM, space=1
53
54  global  start, startup, _main
55  global  reset_vec, intlevel0, intlevel1, intlevel2
56  intlevel0:
57  intlevel1:
58  intlevel2:      ; for C funcs called from assembler
59
60  fnconf  fnauto, ??, ?
61  fnroot  _main
62  TMP_DR0 equ 108
63  TMP_DR1 equ 109
64  TMP_DR2 equ 110
65  TMP_DR3 equ 111
66  CUR_PP  equ 208
67  STK_PP  equ 209
68  IDX_PP  equ 211
69  MVR_PP  equ 212
70  MVU_PP  equ 213
71  CPU_F   equ 247
72  psect  vectors
73  reset_vec:
74  start:
75      ljmp  startup
76
77  psect  init
78  startup:
79      M8C_ClearWDT
80      mov  a, low __Lstackps
81      swap a, sp
82
83      ; falls through to _main
84
85  end start
```

The interface also includes several panels: **User Modules** (listing ADCs and Amplifiers), **Memory** (showing RAM contents), **CPU Registers** (showing A: 0x00, PC: 0x0001, etc.), **Watch Variables**, and **Break Points**. A central callout box contains the text "デバッグ画面" (Debug Screen).

ISP: In System Programming

Memo

- フォローアップURL
- <http://mikami.a.la9.jp/meiji/MEIJI.HTM>

- 担当講師
- 三上廉司(みかみれんじ)
- Renji_Mikami(at_mark)nifty.com (Default - Recommended)
- mikami(at_mark)meiji.ac.jp (Alternative)
- http://mikami.a.la9.jp/_edu.htm

