Instructions for Tele-Training

Prior to the start of the class:

- Download the latest PSoC Designer software at http://www.cypress.com/support/link.cfm?sd=4.
- If you have a PSoC ICE, connect it to your computer.
- Visit http://cypress.webex.com, select a training session under the “Today” or “Upcoming” tab, and follow the instructions to register. After you register you’ll receive an email with directions on how to join your session. ***(NOTE: IT IS BEST TO REGISTER AT LEAST ONE DAY IN ADVANCE.)***

If you have questions or need assistance, please call us toll free at 800.669.0557 (425.787.4400 for local calls or calls outside North America).

CYPRESS

Connecting From Last Mile to First Mile.™

Module 2: Designing with the PSoC

PERSONAL ACCESS ENTERPRISE METRO CORE

Objectives

- Discuss PSoC Data Sheet Resources
- Discuss PSoC™ Routing Resources
- Discuss PSoC Design Considerations
- Complete an Example Project

PSoC Data Sheets

Three Level's of PSoC Data Sheets

1. Device Data Sheet
2. Individual User Module Data Sheets
3. User-Defined PSoC Project Configuration Data Sheet

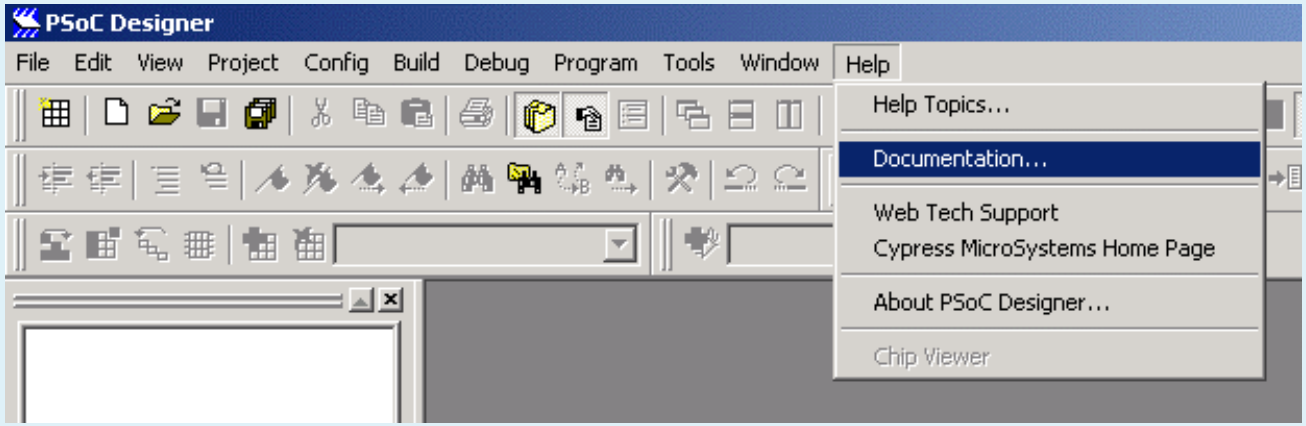
CY8C27X Device Data Sheet

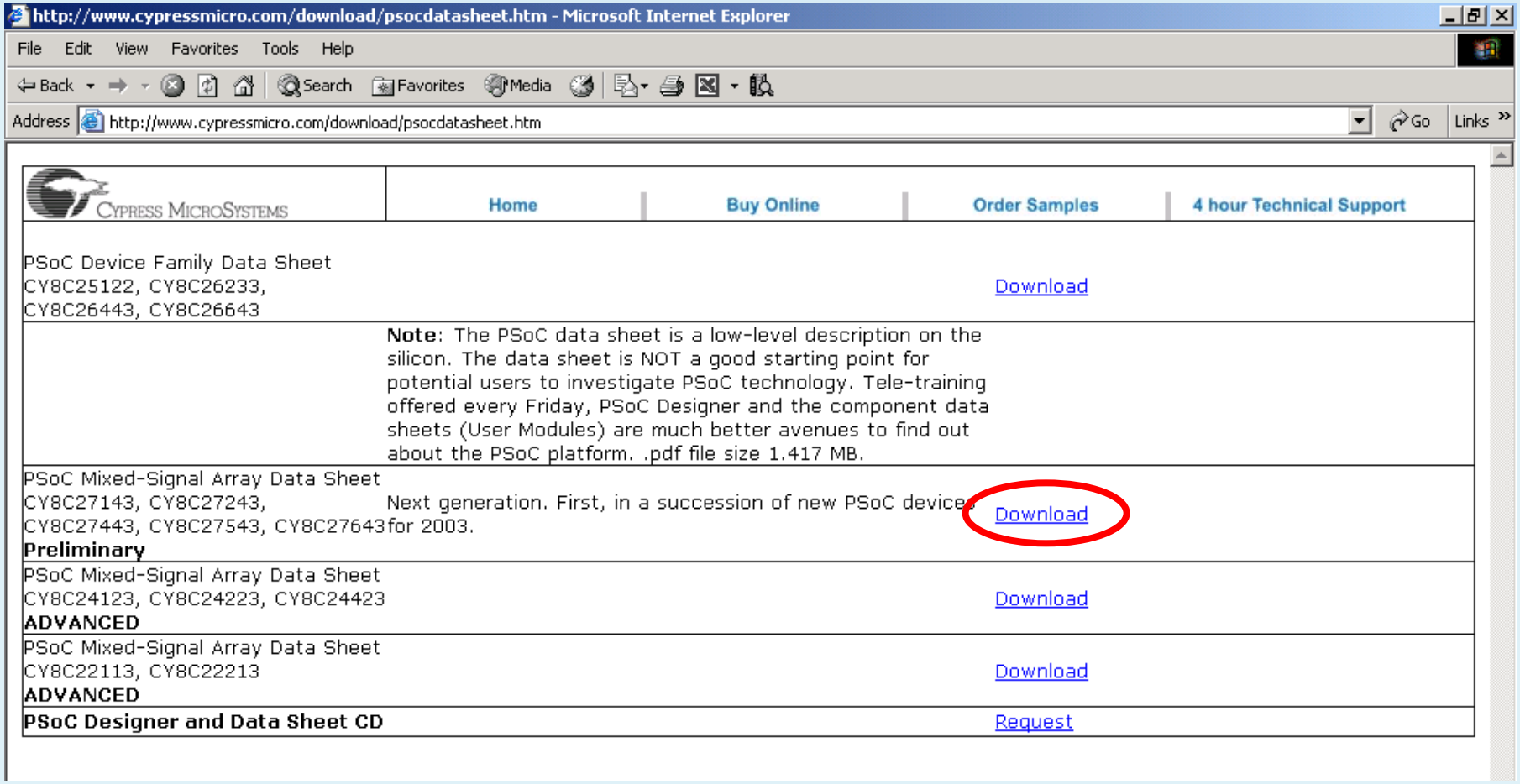
The complete device data sheet for all part types is under Help >> Documentation inside PSoC Designer

DataSheet_CY8C27xxx.pdf

Also, at

http://www.cypress.com/cfuploads/img/products/CY8C7443-24PI.pdf




http://www.cypressmicro.com/download/psocdatasheet.htm - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.cypressmicro.com/download/psocdatasheet.htm
Go Links
CYPRESS MICROSYSTEMS
Home
Buy Online
Order Samples
4 hour Technical Support
PSoC Device Family Data Sheet
CY8C25122, CY8C26233,
CY8C26443, CY8C26643
Download
Note: The PSoC data sheet is a low-level description on the silicon. The data sheet is NOT a good starting point for potential users to investigate PSoC technology. Tele-training offered every Friday, PSoC Designer and the component data sheets (User Modules) are much better avenues to find out about the PSoC platform. .pdf file size 1.417 MB.
PSoC Mixed-Signal Array Data Sheet
CY8C27143, CY8C27243,
CY8C27443, CY8C27543, CY8C27643
Next generation. First, in a succession of new PSoC devices for 2003.
Download
Preliminary
PSoC Mixed-Signal Array Data Sheet
CY8C24123, CY8C24223, CY8C24423
Download
ADVANCED
PSoC Mixed-Signal Array Data Sheet
CY8C22113, CY8C22213
Download
ADVANCED
PSoC Designer and Data Sheet CD
Request

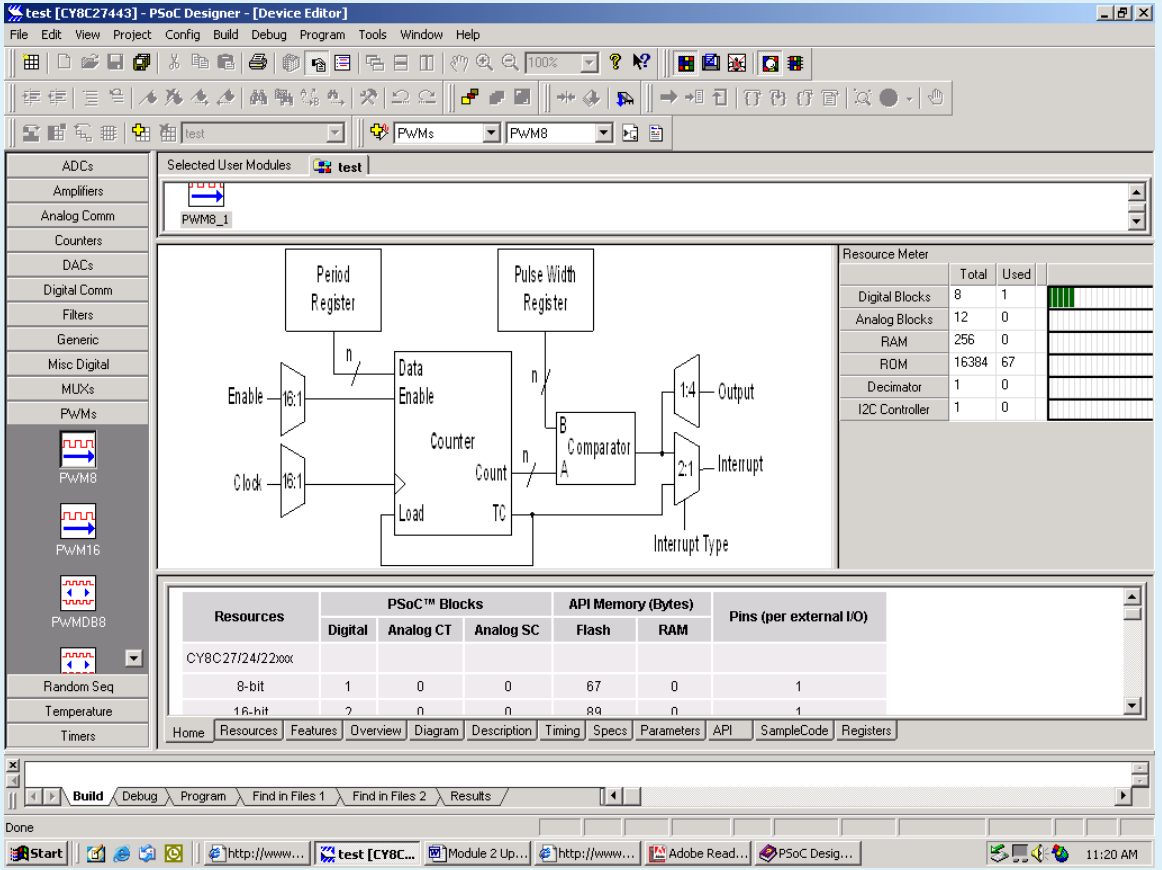
Individual User Module Data Sheets

Each User Module has its own data sheet contained within the PSoC Designer software

- Detailed specifications for the User Modules included in the data sheet
- Placement considerations, and example code

User Module Data Sheet Sections

- Resources
- Features
- Overview
- Diagram
- Description
- Timing
- Specs
- Placement
- Parameters
- API
- Sample Code
- Registers



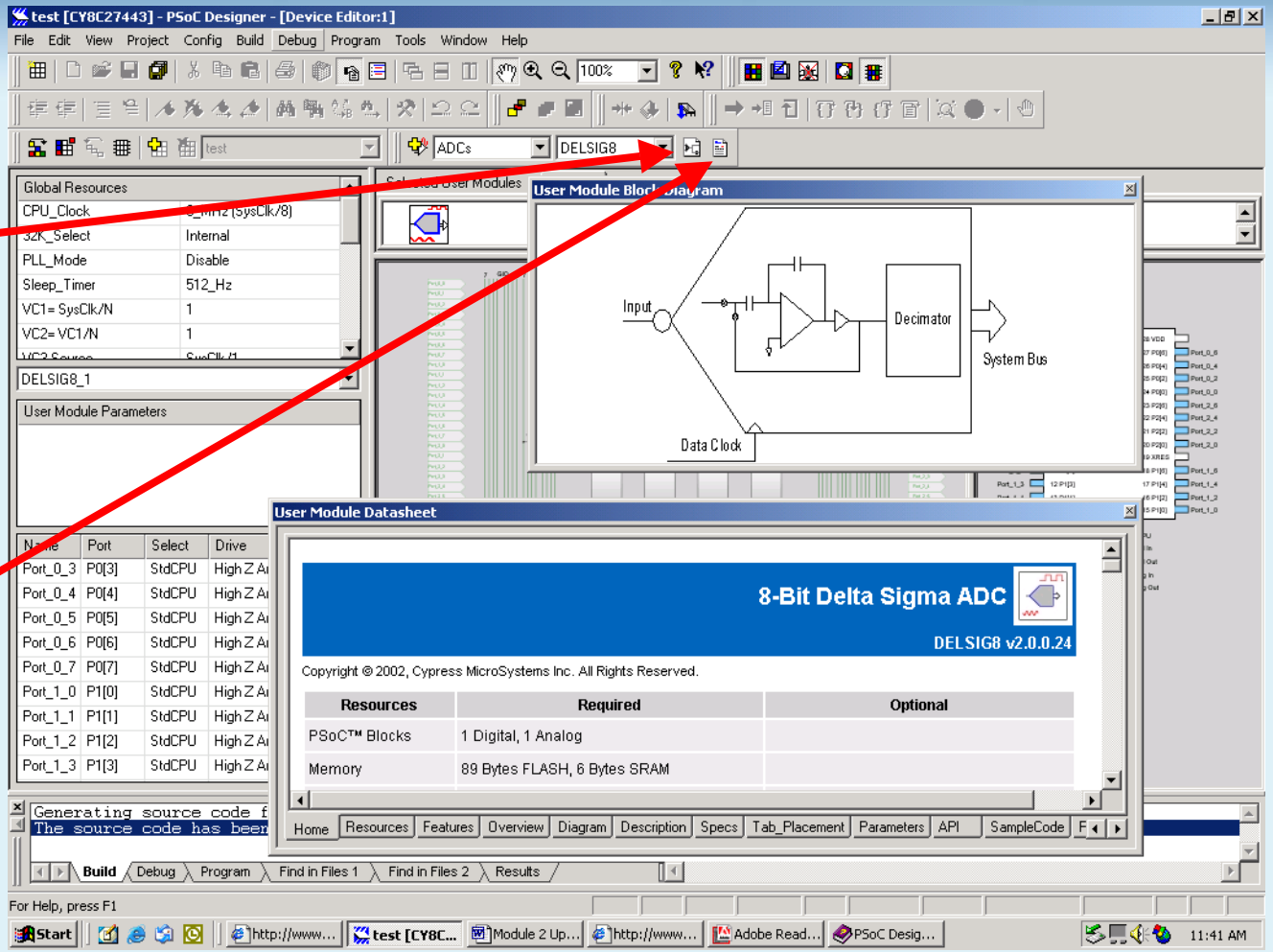
User Module View Options

UM Block Diagram

- Left-click

UM Data Sheet

- Left-click

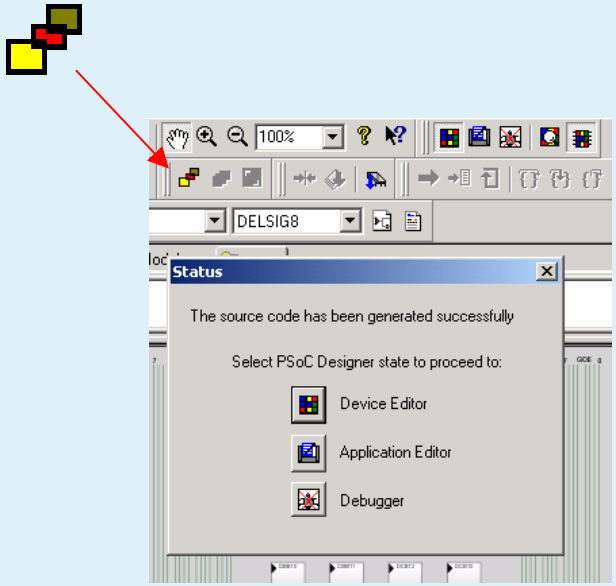


PSoC Project Configuration Data Sheet

When the Generate Application icon is selected the User Defined Configuration Data Sheet is created

This datasheet saves hours of documentation time!

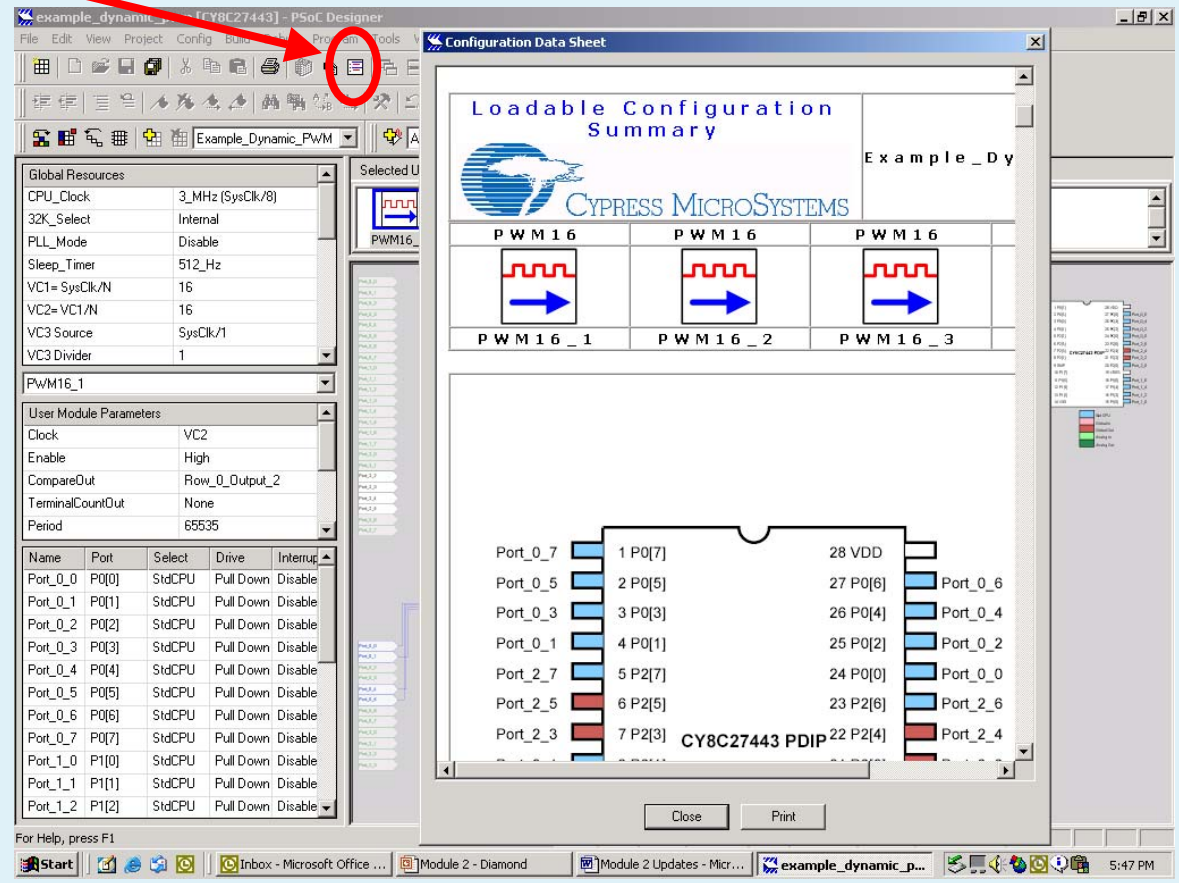
The next several slides show examples of the data sheet



PSoC Project Configuration Data Sheet

The complete data sheet is *updated* as user configuration changes

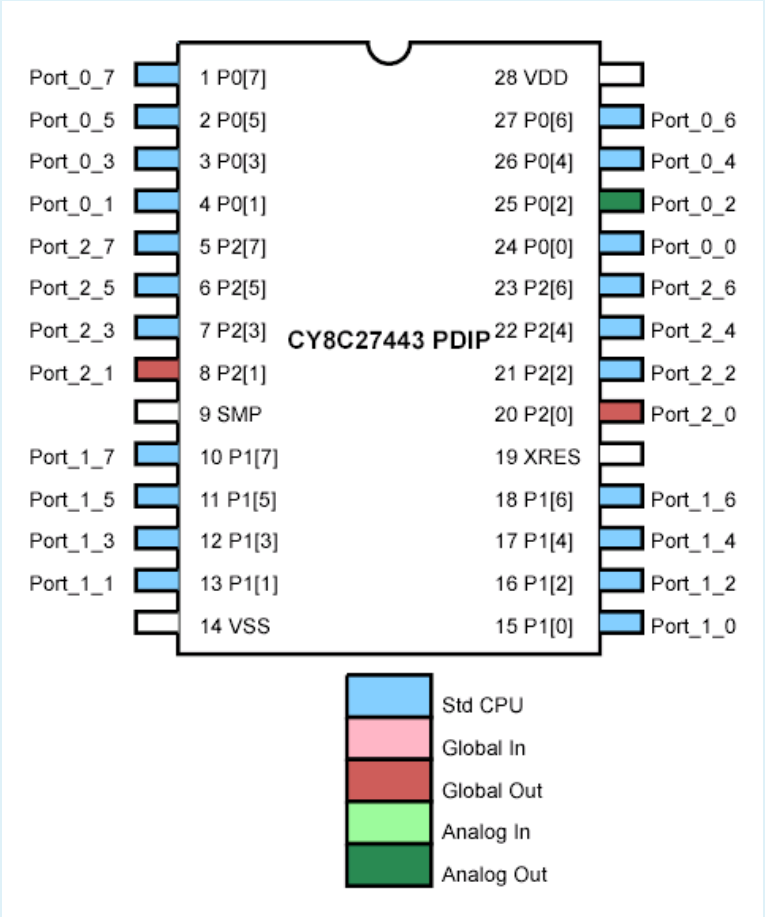
Detailed project documentation



PERSONAL ACCESS ENTERPRISE METRO CORE

PSoC Project Configuration Data Sheet

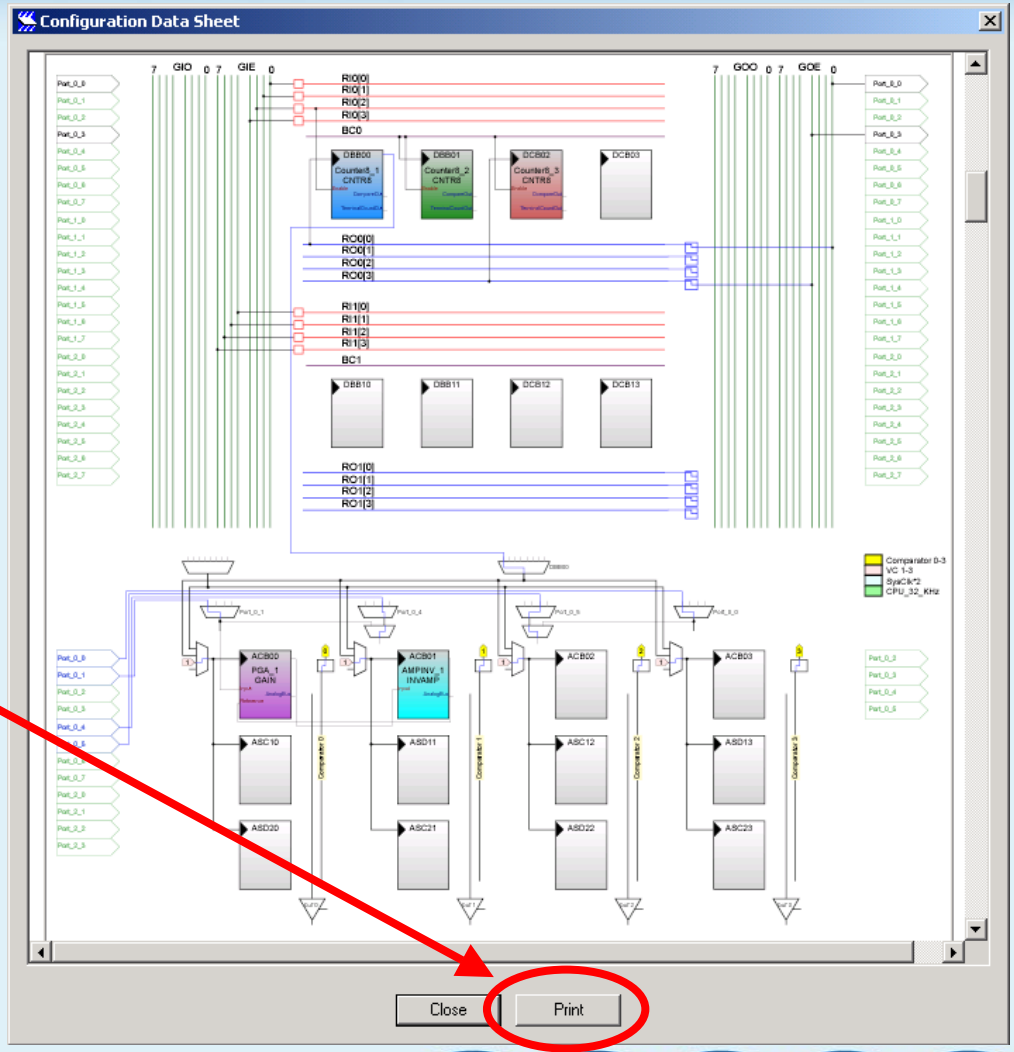
User defined pin-outs are color-coded and detailed in the data sheet.



PSoC Project Configuration Data Sheet

Data sheet contains the placement and routing of User Modules

Placement View Window is Printable



PERSONAL ACCESS ENTERPRISE METRO CORE

PSoC Routing Resources

Digital System

- Digital PSoC Blocks
- Global Digital Interconnect
- Row Digital Interconnect
- Row Broadcast Nets

Analog System

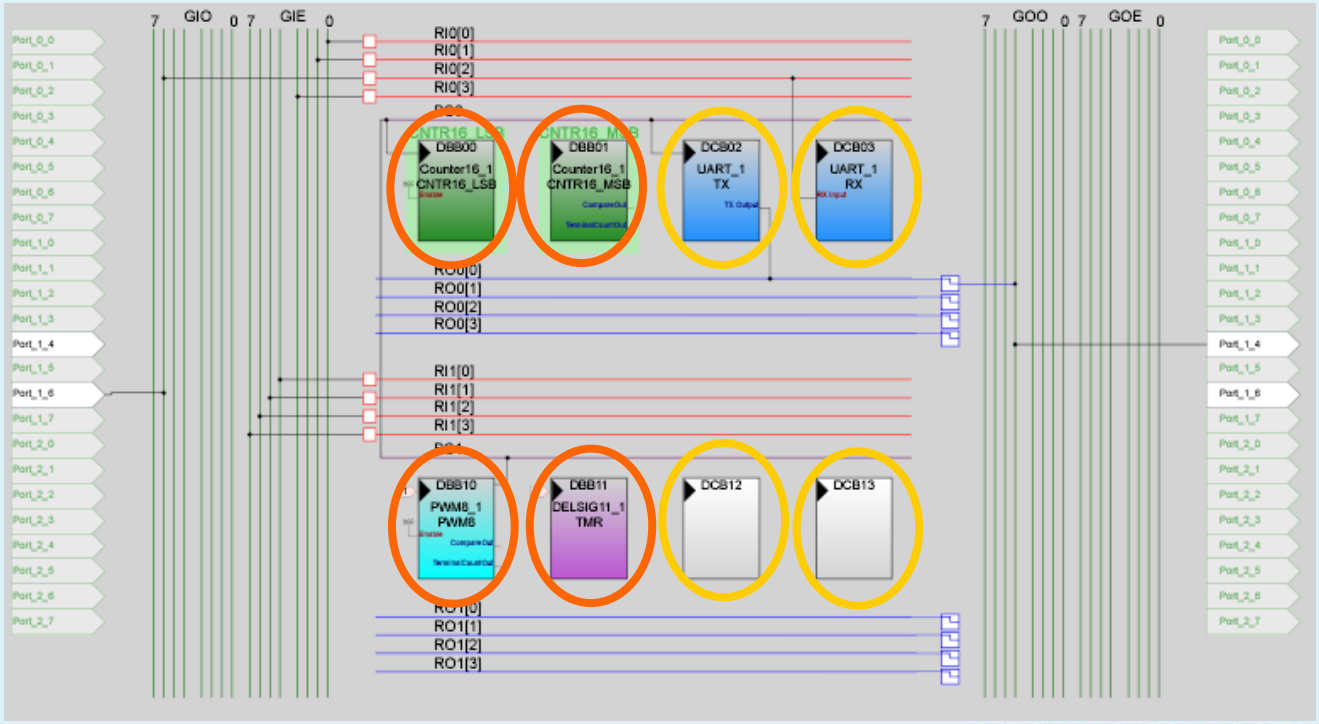
- Analog PSoC Blocks
- Configurable Clock Source
- Configurable Inputs
- Configurable Outputs

Digital Blocks

Two Rows of Digital Blocks

Each Row Contains:

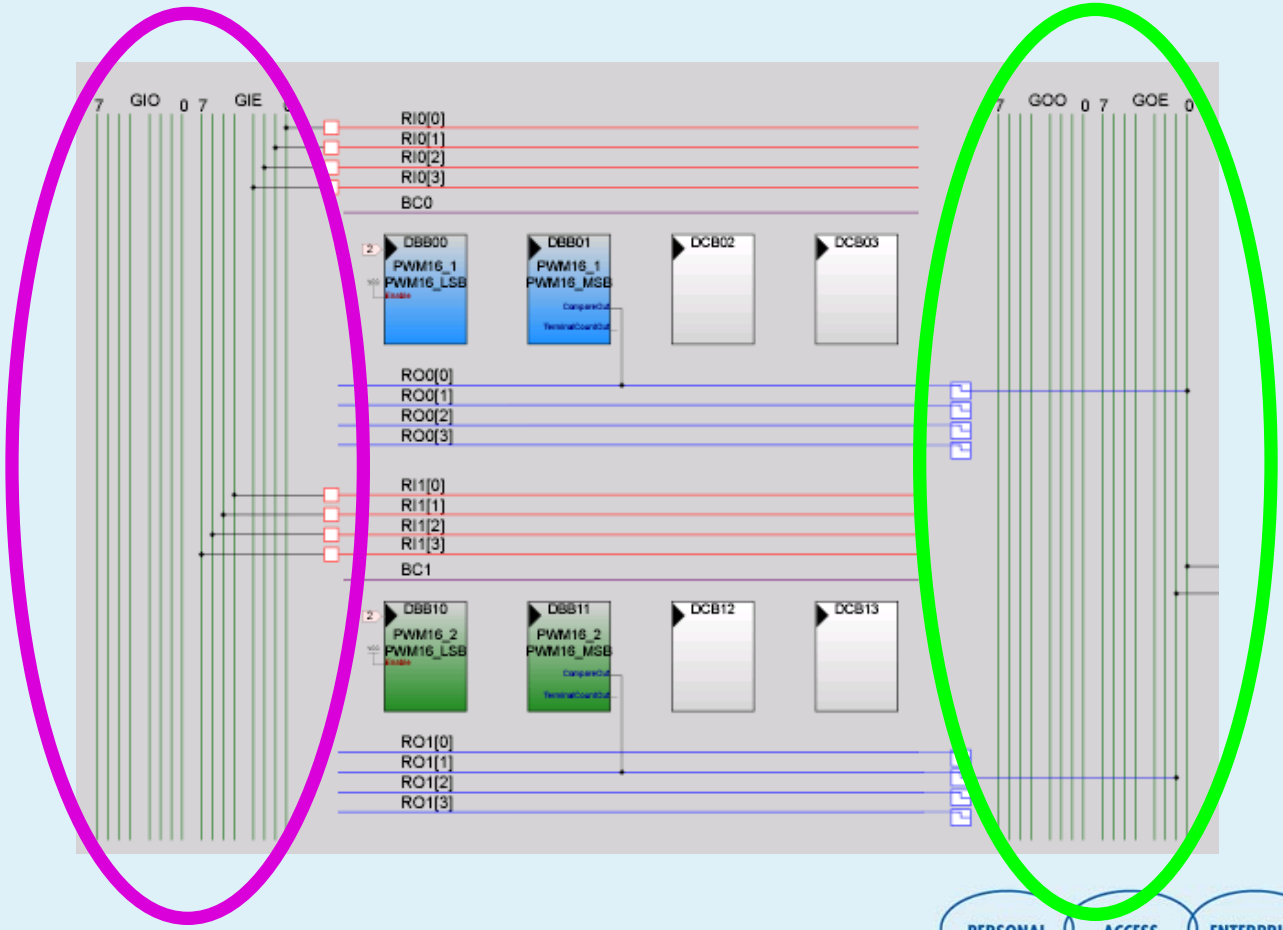
- Two Digital Basic Bocks (DBB)
- Two Digital Communication Blocks (DCB)



Global Digital Interconnect

32 total nets for Digital Routing

16 Separate Global Inputs



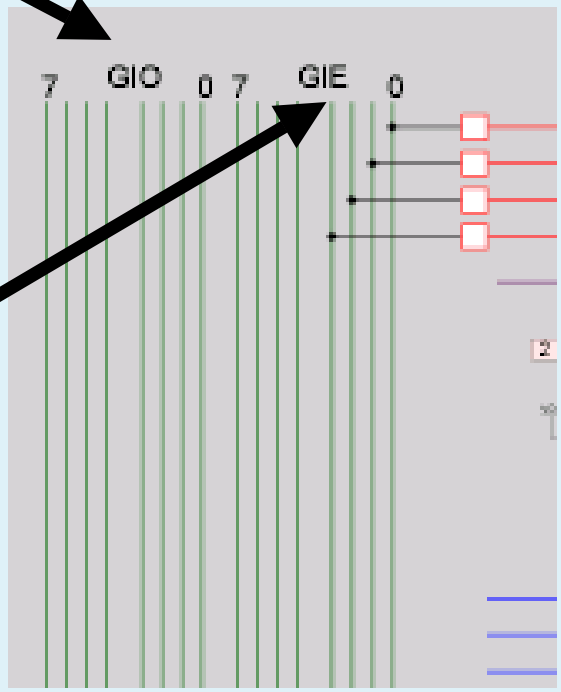
16 Separate Global Outputs

Global Digital Interconnect

Global nets are divided into Odd and Even

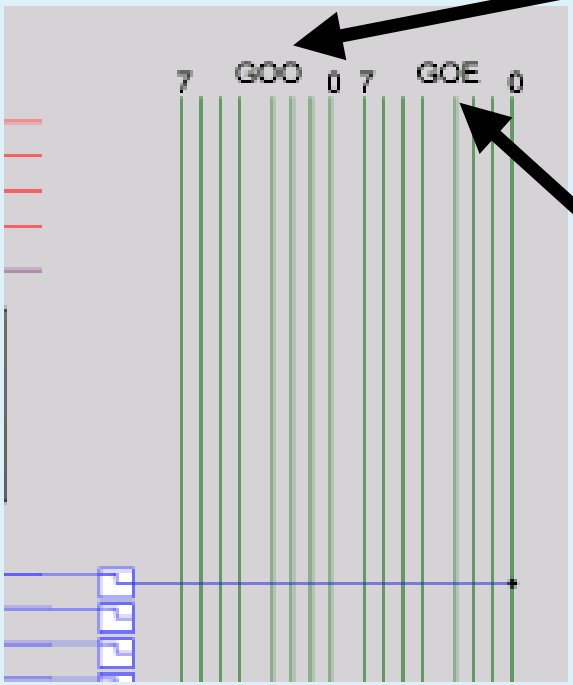
GIO = Global Input Odd

GIE = Global Input Even



GOO = Global Output Odd

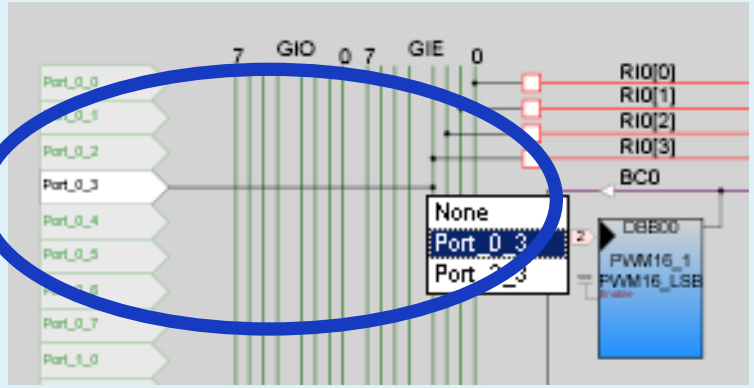
GOE = Global Output Even



Global Digital Interconnect

Global nets can be used to:

- **Connect to other Global nets**
- **Connect to Pins**



Row Digital Interconnect

Each row of Digital Blocks has its own set of Row Interconnects

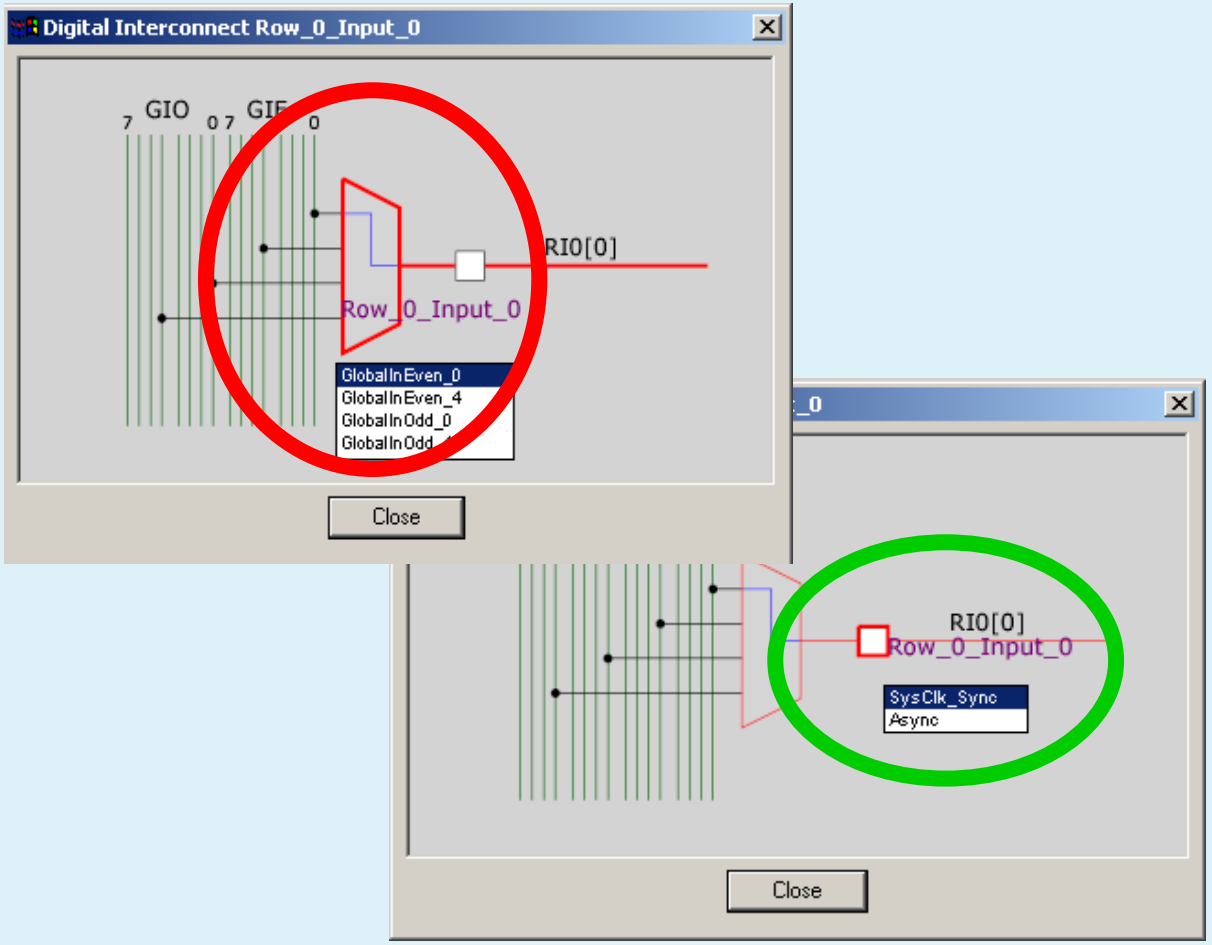
4 Input Rows and 4 Output Rows for the top row of digital blocks

4 Input Rows and 4 Output Rows for the bottom row of digital blocks

Row Inputs For Digital Signals

Row Inputs can be used for:

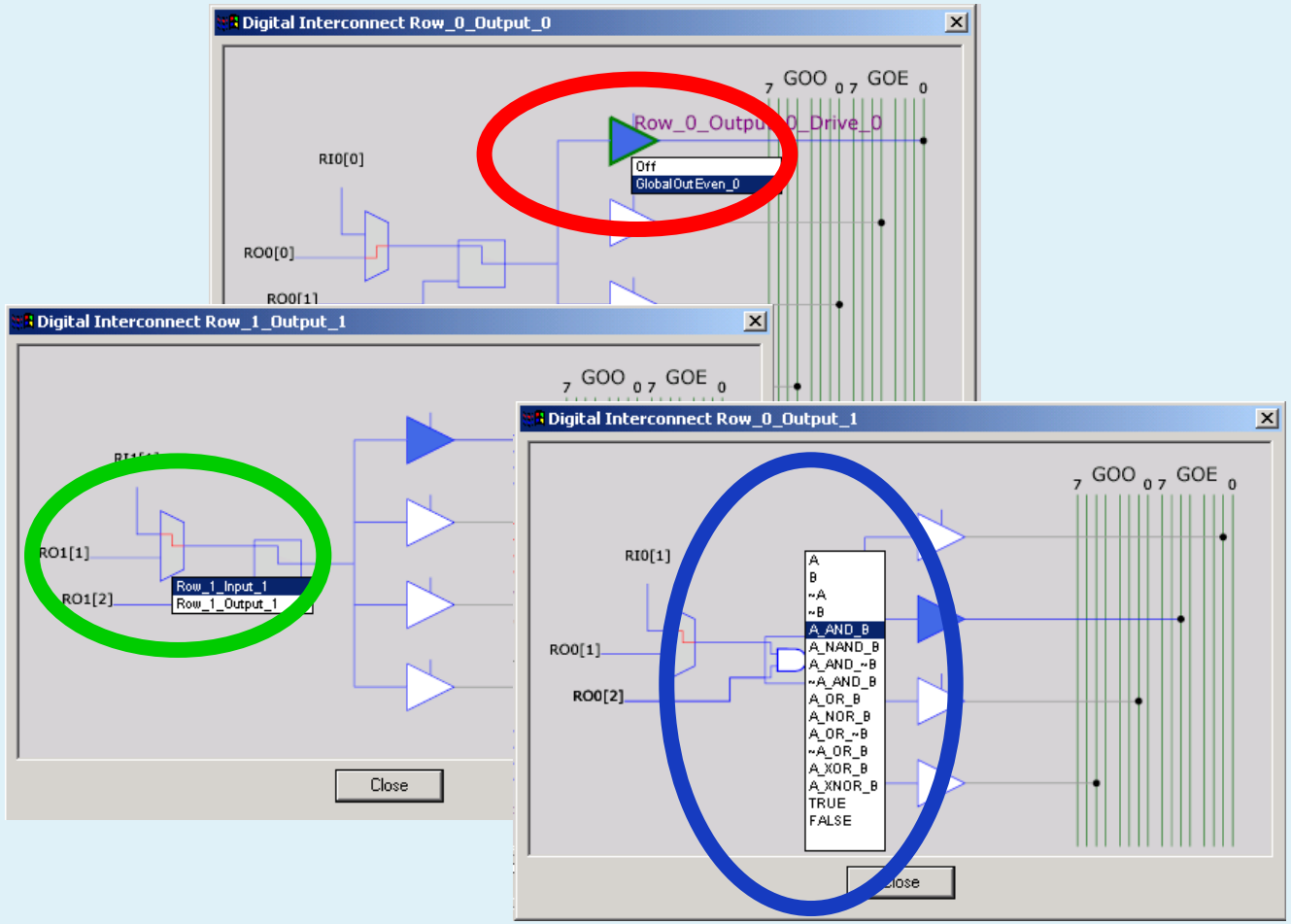
- **Connection to Global Inputs**
- **Clock Synchronization**



Row Outputs For Digital Signals

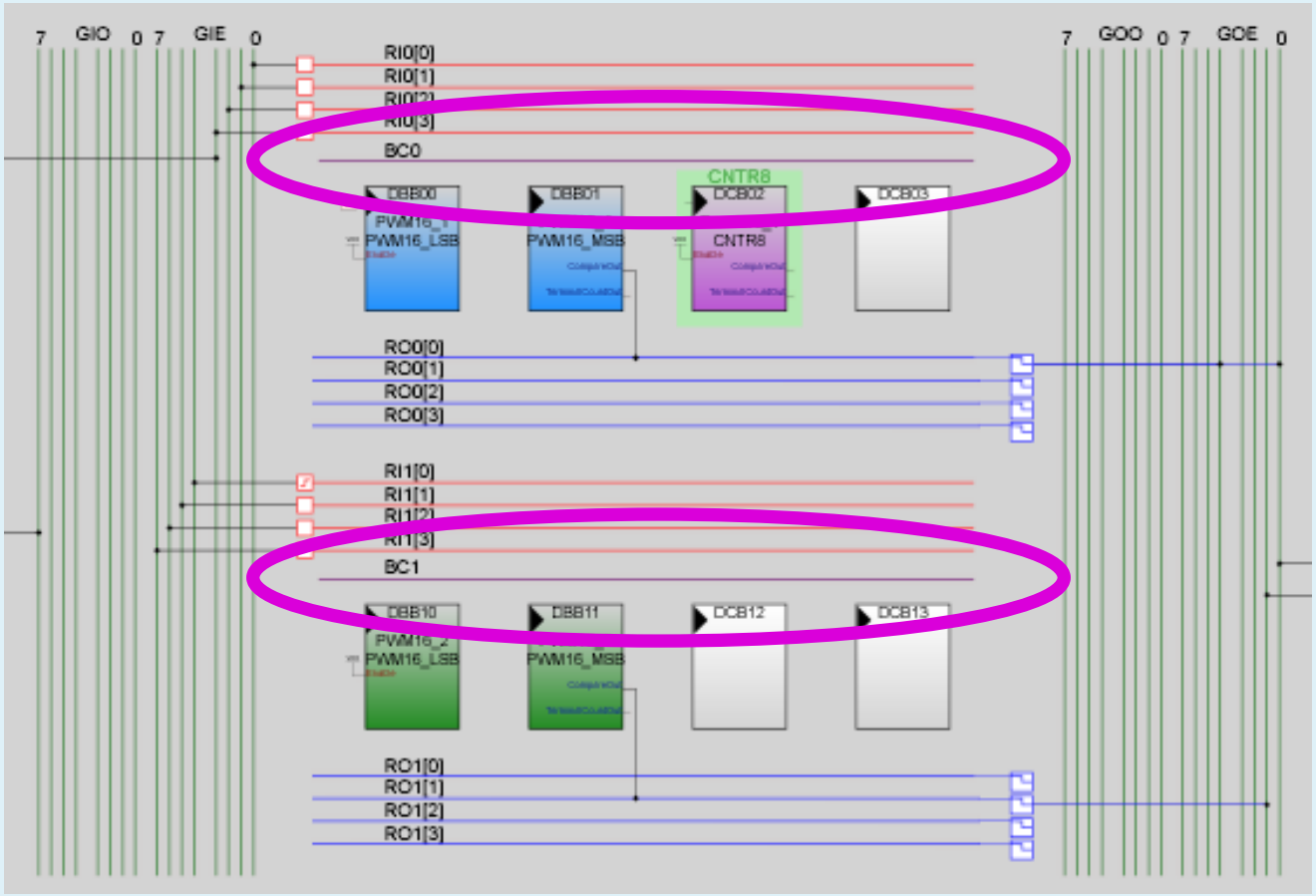
Row Outputs can be used to:

- **Connect to global output nets**
- **Connect a Row Input to global output nets**
- **Perform logical operations on two Row nets.**



Row Broadcast For Digital Signals

There are two Row Broadcast nets

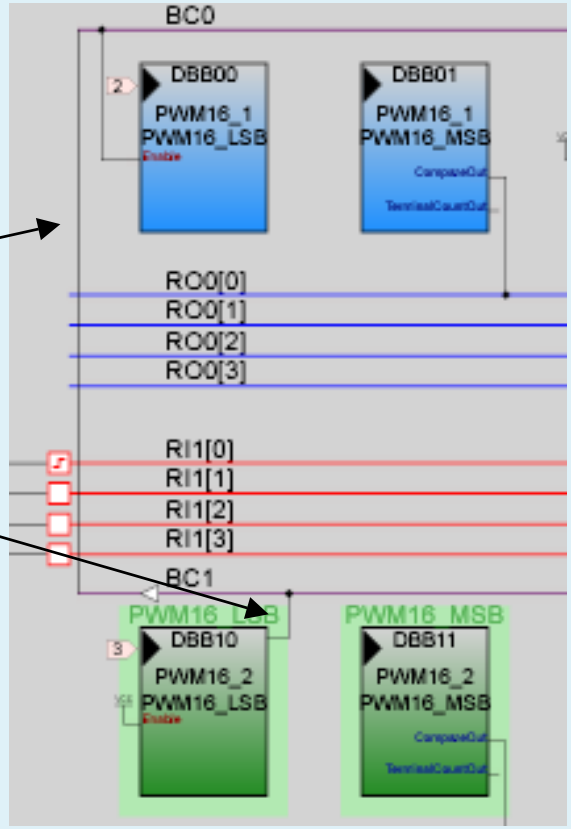


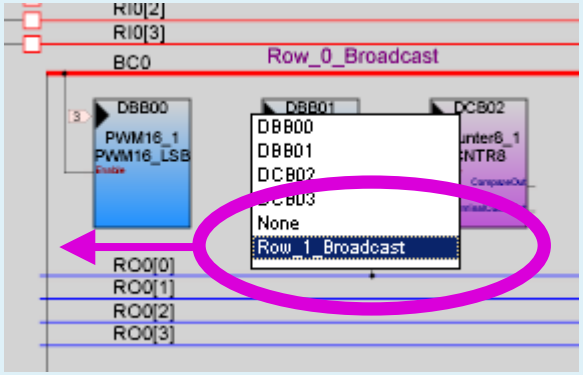
Row Broadcast For Digital Signals

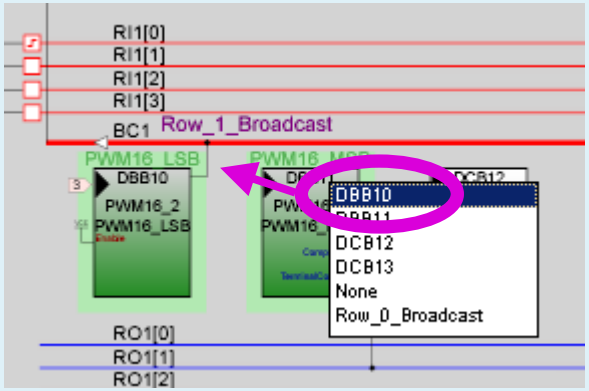
Row Broadcasts can be:

- **Connected to each other**
- **Connected to a block**

Thus any digital block can drive any other digital block or blocks

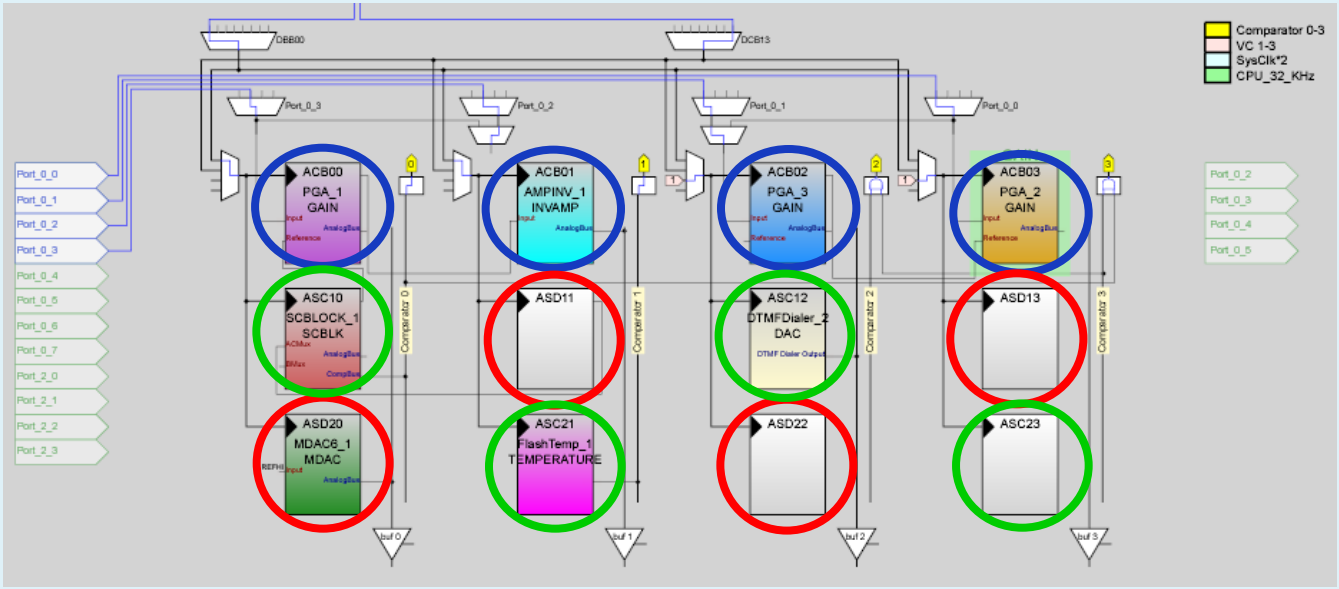






Analog Blocks Arranged in Columns

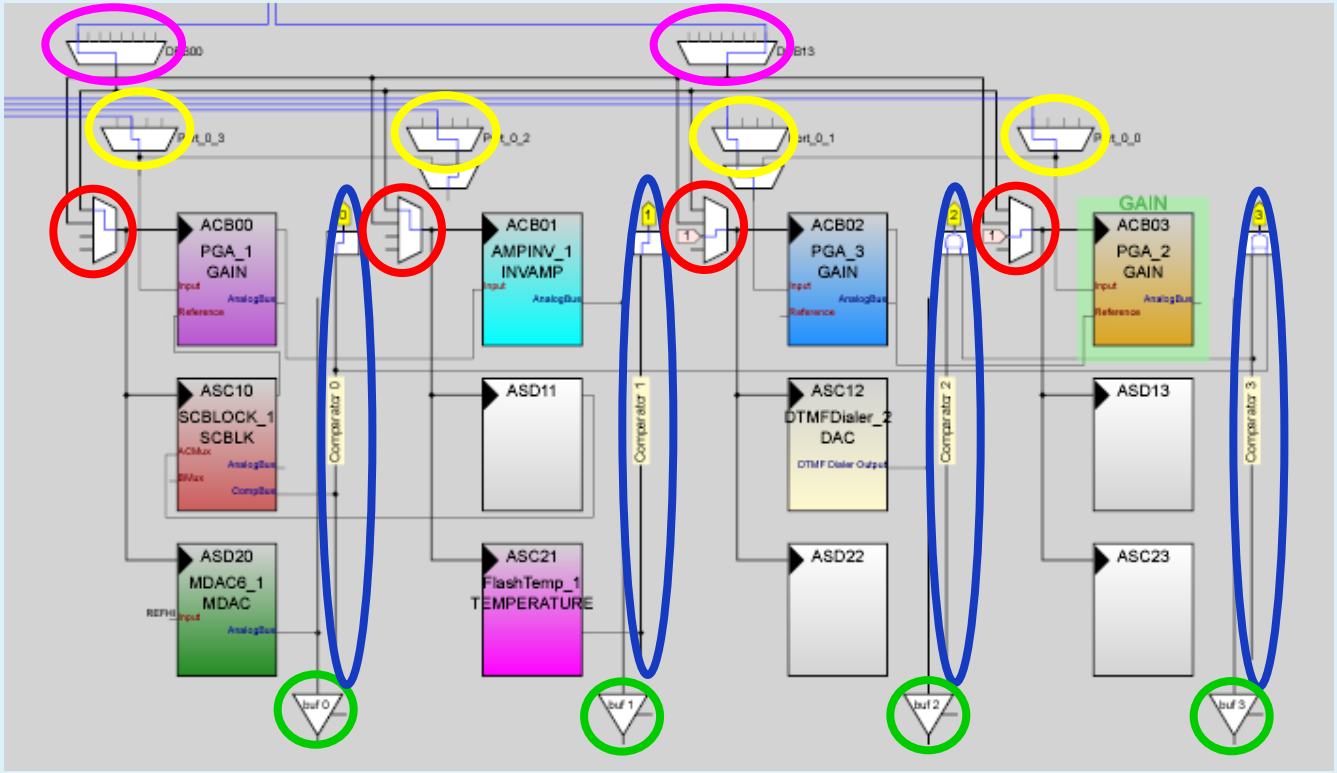
- **Each column contains 3 types of Analog Blocks:**
  - **ACB** (Continuous Time)
  - **ASC** (Switched-Cap C)
  - **ASD** (Switched-Cap D)



Analog Columns Shared Resources

Each Column shares:

- Column Input MUX
- Column Clock Mux
- Output Bus and Buffer
- Comparator Bus



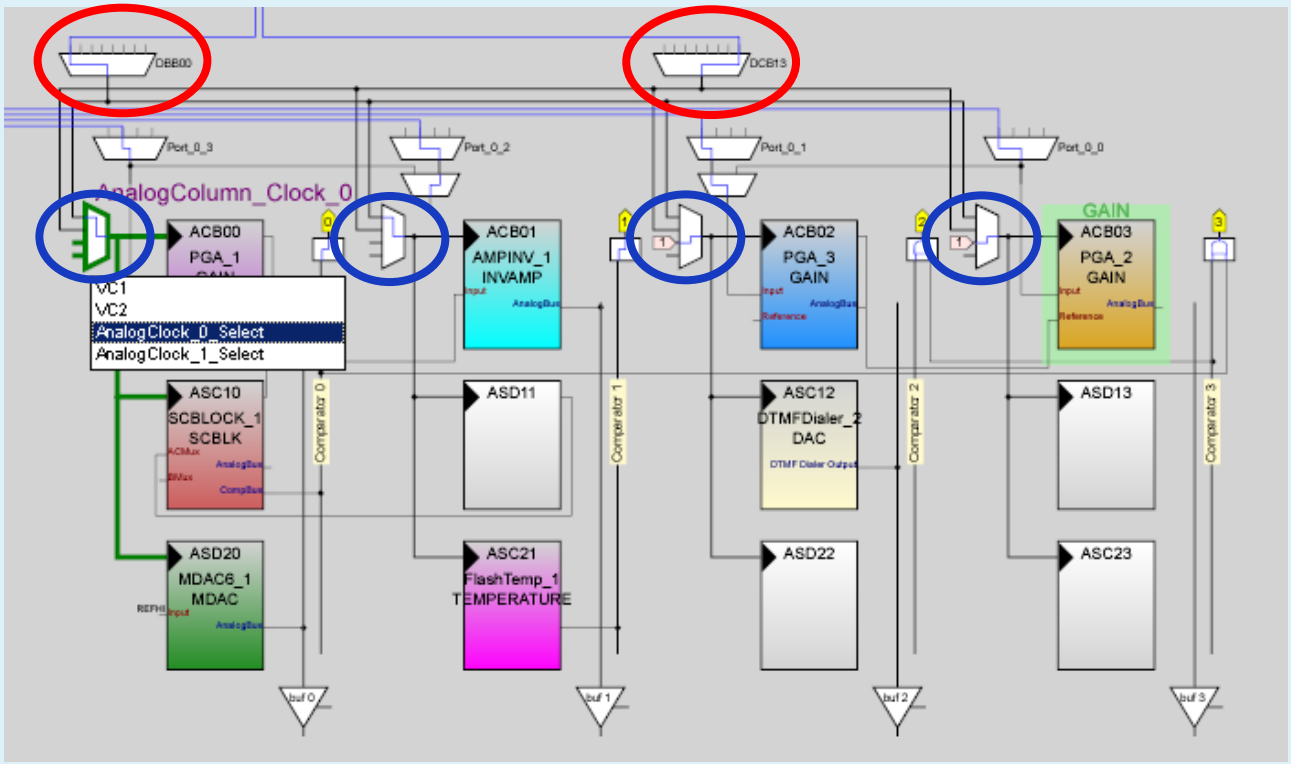
All Columns share:

- Analog Clock Select

Configuring the Clock Source

Clock Select and Column MUXs:

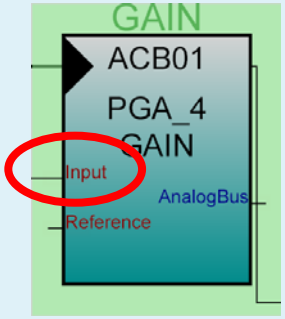
- 4 Column Clock MUXs can connect to VC1, VC2, or connect to Clock Select MUXs
- 2 Clock Select MUXs can connect to any of the 8 digital blocks

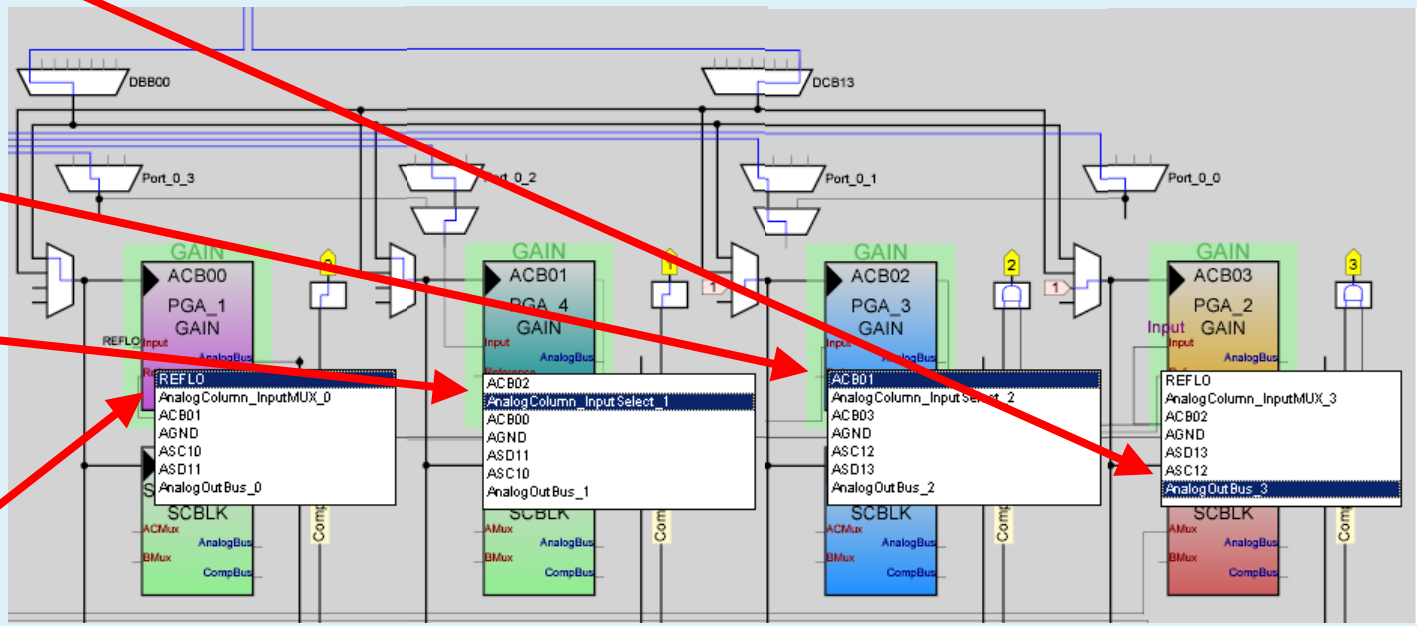


Configuring ACB Inputs

ACB Block Inputs include:

- Analog Out Bus
- Nearby Analog Blocks
- Ports via the Column Input MUX
- Analog Reference

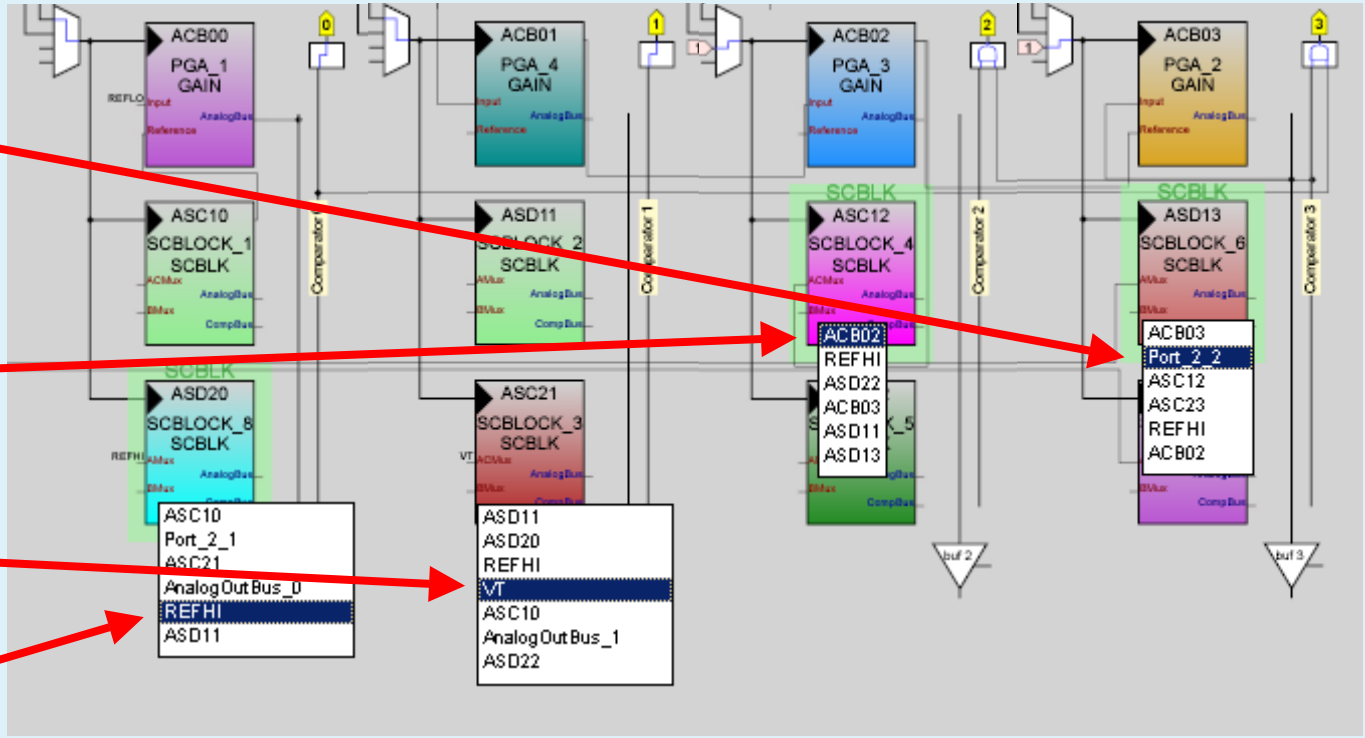




Configuring ASC and ASD Inputs

ASC and ASD Block Inputs include:

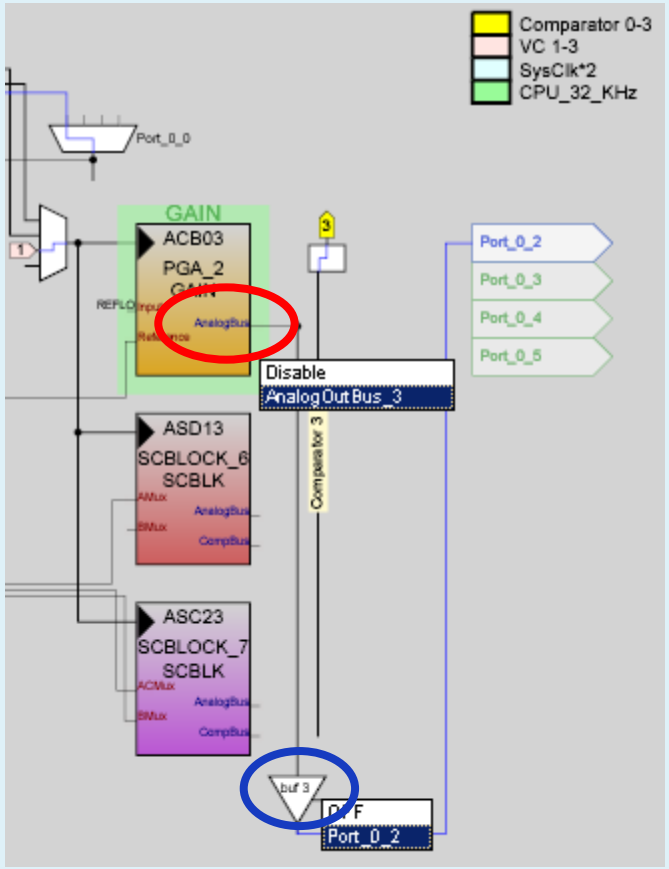
- **Directly from Port 2 (Columns 0 and 3 only)**
- **Nearby Analog Blocks**
- **Vtemp (ASC21 only)**
- **Analog Reference**

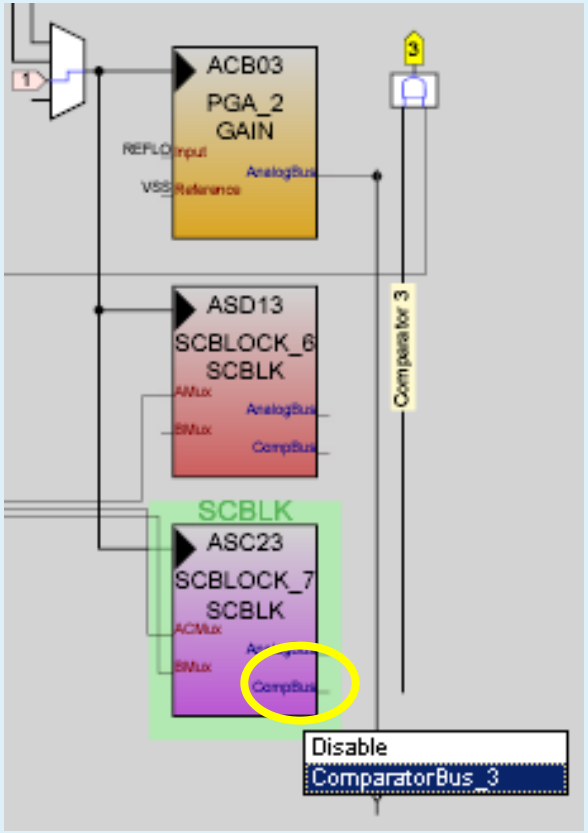


Configuring Outputs

There are two output methods:

- Output to Ports via Analog Out Bus and Buffer
- Output to Comparator Bus

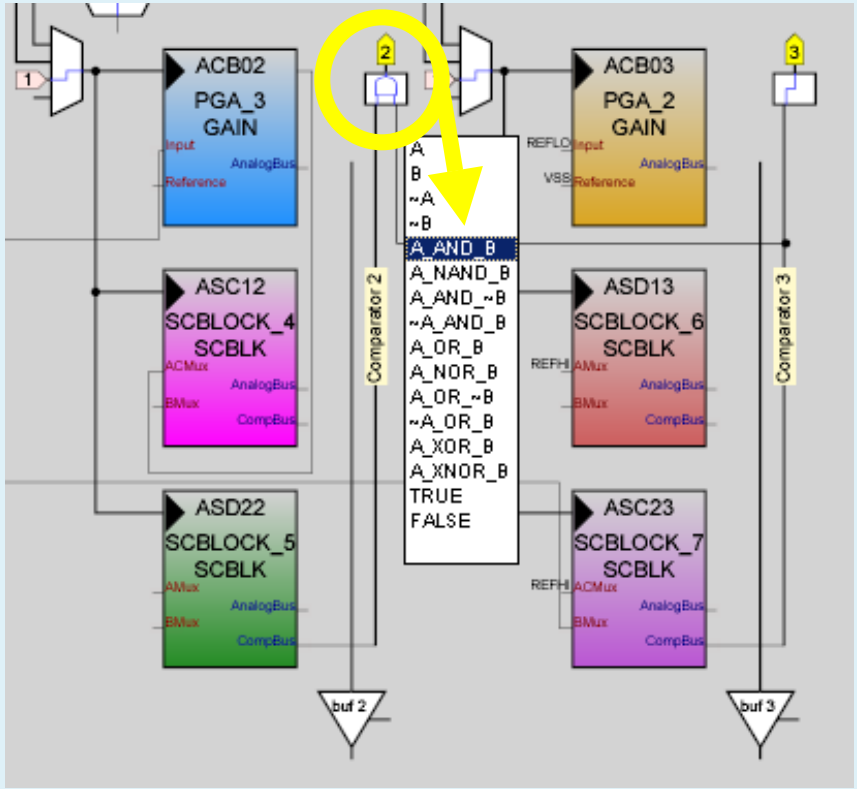


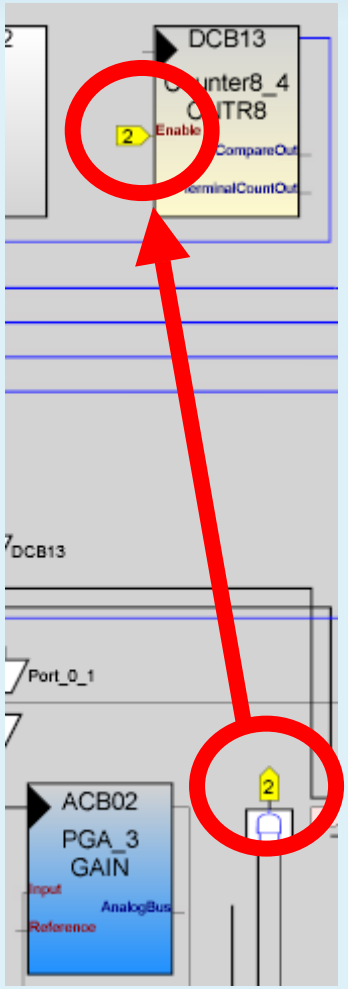


Comparator Bus

The Comparator Bus can:

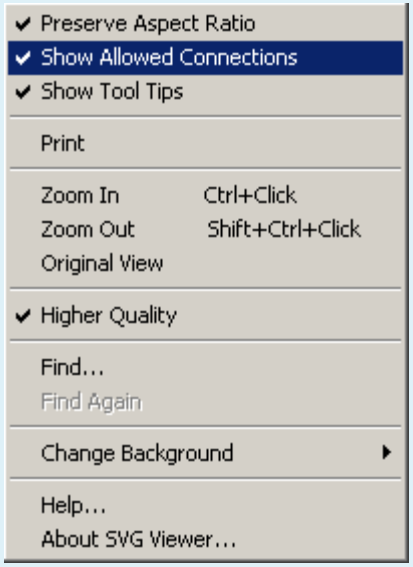
- Perform logical operations on comparator outputs from two adjacent columns using the AnalogLUTs)
- Provide input to digital blocks

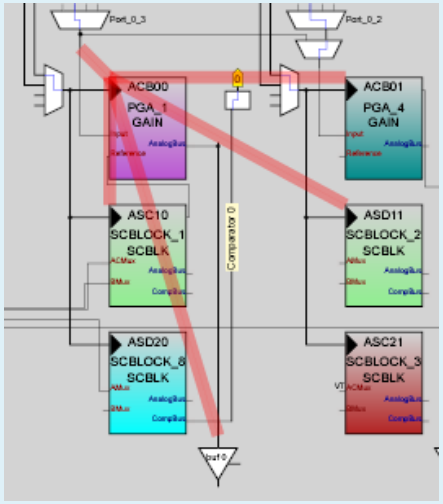


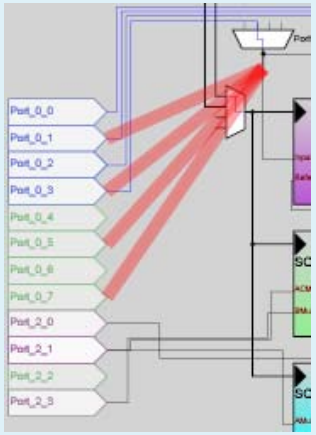


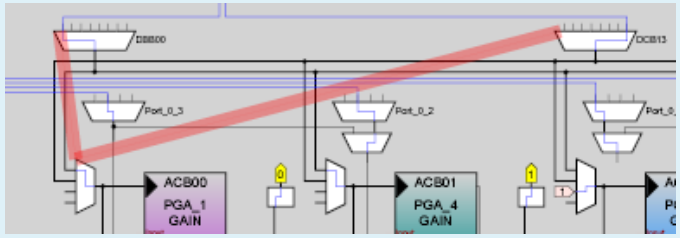
Making a Connection

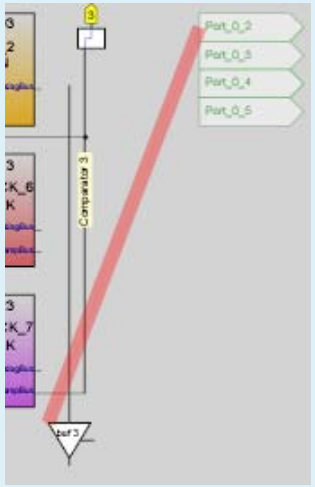
Right Clicking on empty space in the Device Editor Window brings up the following window.











Checking "Show Allowed Connections" enables a feature that highlights all possible connections for any selected block, MUX, bus, or pin.

PSoC design flow

Project Planning
Determine the system requirements

Device Configuration
Choose the User Modules
Place the User Modules
Set Global Resources
Set User Module Parameters
Configure Internal Routing
Define the Pinout for the device

Adding Application Code
Generate Application Code with the touch of a button!

Debugging
Review generated code
Demonstrate Working Configuration

} Covered in detail in Module 3

Example Project

Project Planning

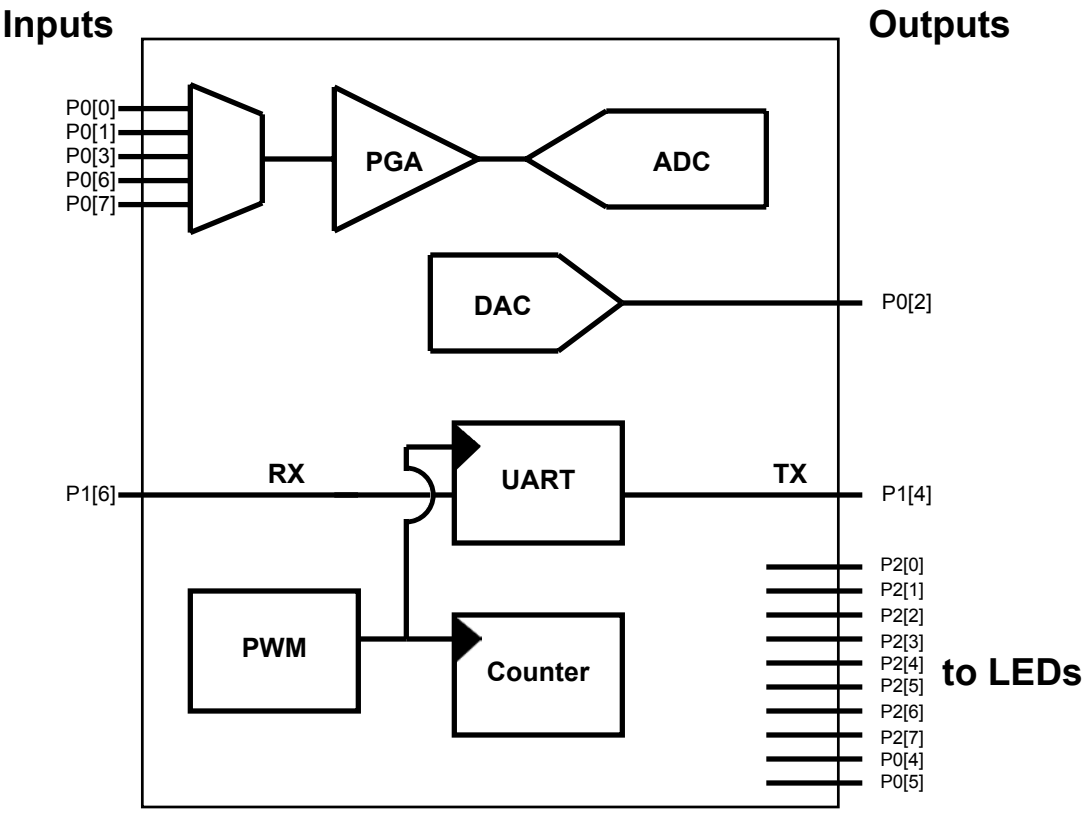
Desired Outcome:

Observe LEDs on Pup board counting down 256 steps at 1 step per ¼ second; these steps correspond to the DAC's output voltage which will decrement from 3.8 V to 1.2 V.

Create the project with:

- **A single 8 bit DAC output**
- **A UART with 9600 baud rate**
- **A rate generator/PWM for the UART baud rate and to provide the input clock**
- **A 16 bit Counter for timing**
- **A programmable gain amplifier/input buffer**
- **An ADC with 5 inputs**

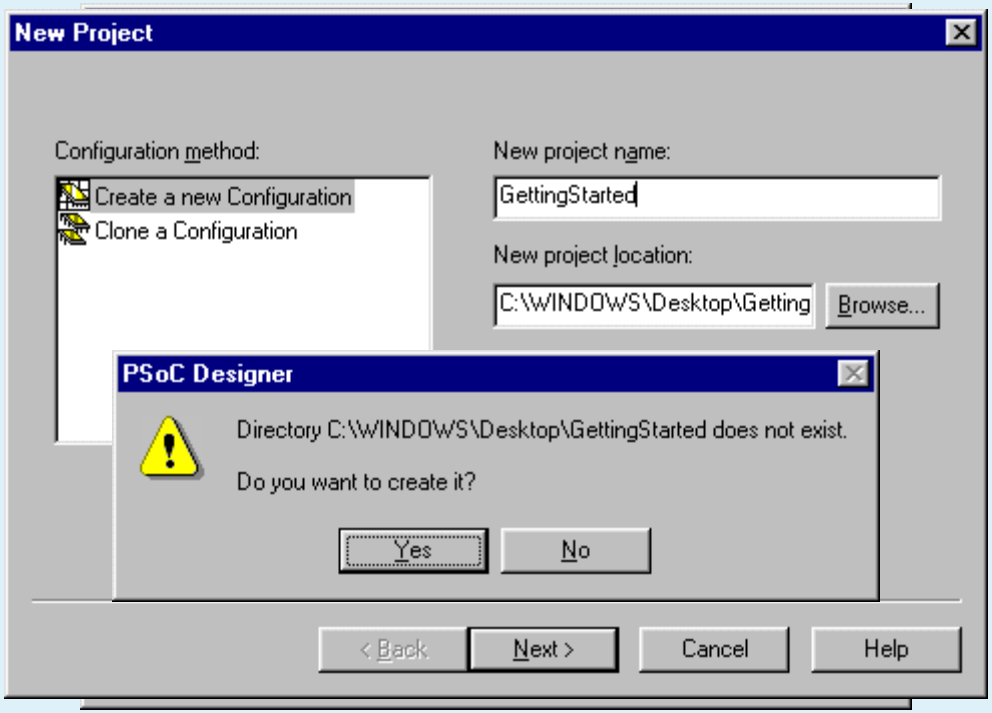
Block Diagram



Let's Create Our Project

Create a New Project Named "GettingStarted"

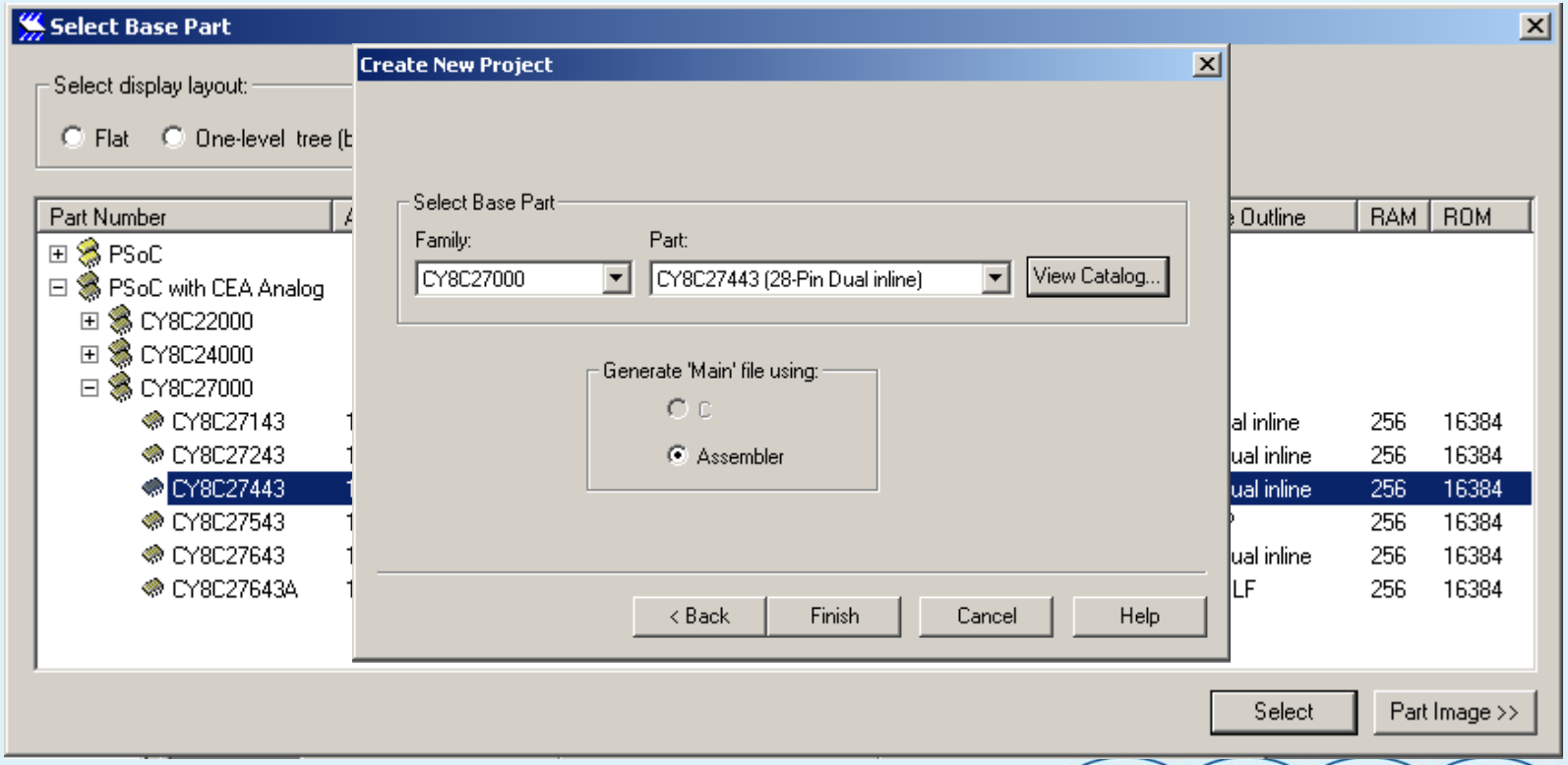
- Set destination directory Desktop/



Let's Create Our Project

Select the Base Part

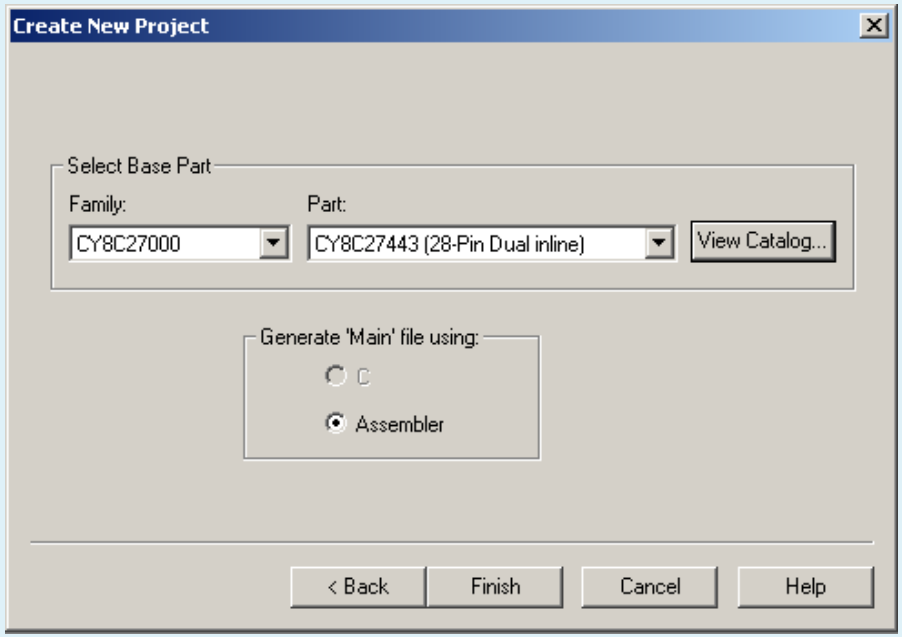
- View the drop-down menu and the catalog
- We'll use CY8C27443 (28-Pin Dual inline)



Let's Create Our Project

Select Project's Language

- Assembly and C (only if enabled) Languages available
- We'll choose assembly

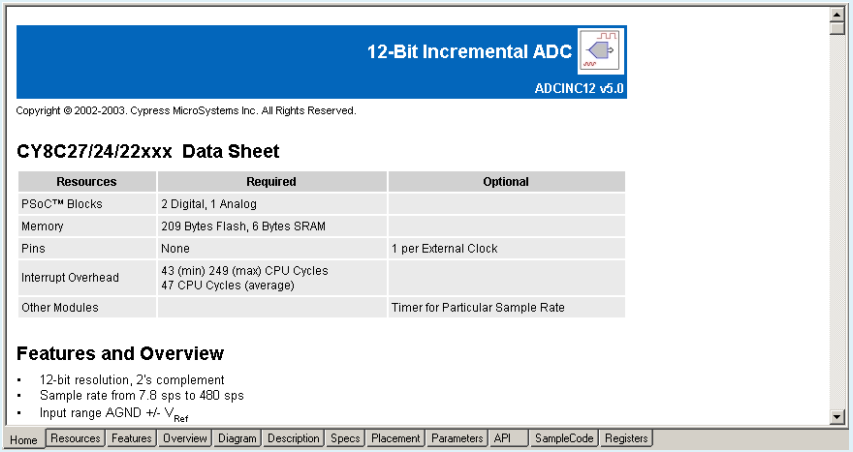


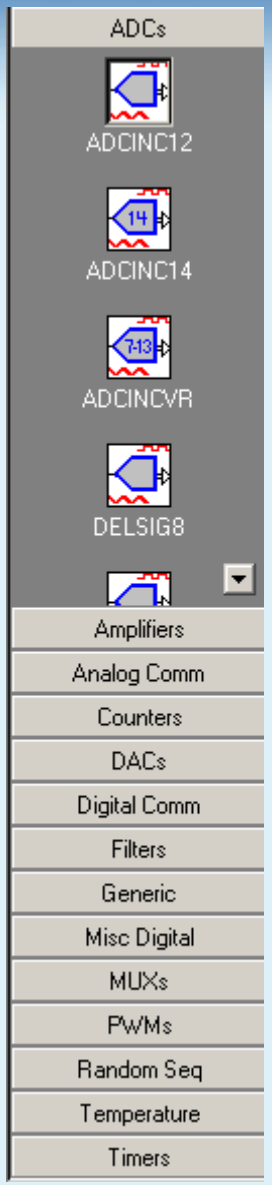
Choosing User Modules

Connecting From Last Mile to First Mile™

Resource Meter

	Total	Used
Digital Blocks	8	0
Analog Blocks	12	0
RAM	256	0
ROM	16384	0
Decimator	1	0
I2C Controller	1	0





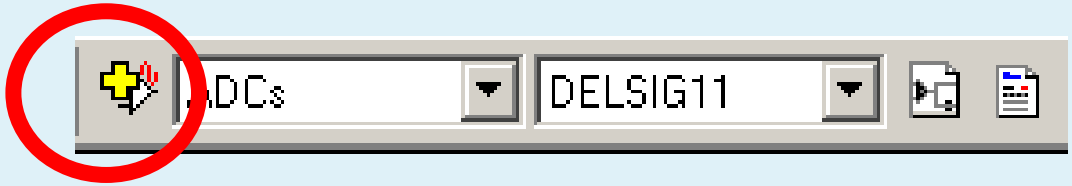
Explore the "Selection View" of Device Editor

- User Module Catalog
- Resource Meter
- User Module Data Sheet Viewer
- Adding, Deleting, Renaming User Module Instances

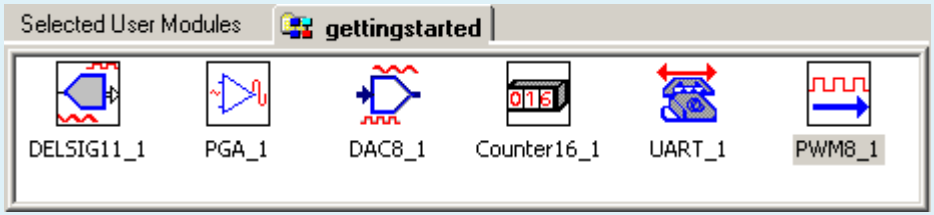
Choosing User Modules

We will select 6 User Modules for this project

Choose User Modules from the pull-down menu, left click on select button to add a chosen User Module

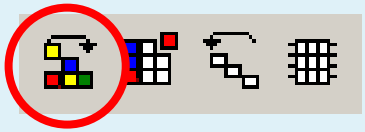
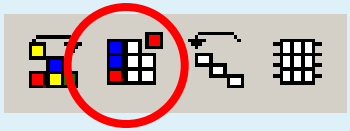
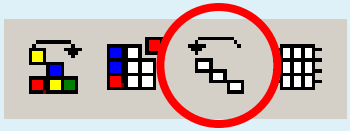


- DELSIG11: A Delta-Sigma ADC with 11 bits of resolution (select DS111)
- PGA_A: A programmable gain amp/input buffer
- DAC8: A single output DAC with 8 bits of resolution
- Counter16: A counter for clocking
- UART
- PWM8: An 8-bit Pulse Width Modulator



Placing User Modules:

Explore the “Interconnect View” of the Device Editor

- Next Position button 
- Selecting the “Active” UM block
- Place Here button 
- Unplace button 

Placing User Modules: Guidlines

There is a great deal of flexibility

- Try-out the modules individually first and see how flexible they are, then undo placements

PSoC Designer will only allow the modules to be placed where the chip can support them

PSoC Designer will not prevent a placement that may create a conflict for resources

- Detailed User Module data sheets describe resource support requirements

Analog Block Columns share Clock settings

- Place User Modules with the same Clock setting in the same column

Placing User Modules: Guidlines

- **Some User Modules only work in Digital Communication blocks, other User Modules will work in either Digital Basic blocks or Digital Communication Blocks**
  - **Place Communication block User Modules first to ensure available placement**
- **The Broadcast Bus allows any one digital block to drive the clock of any other digital block or blocks**
  - **Clock syncing between digital blocks does not impact user module placement (unlike analog blocks)**

Place the User Modules

UART – Blocks DCB02/DCB03

- one block each for RX, TX
- The UART requires digital communication blocks

Counter 16 – Blocks DBB00/DBB01

- The Counter can be placed in either basic or communication blocks

DELSIG11 – Block ASC12 and DBB11

- 5 analog sources — middle columns connect to all analog inputs
- Use a basic (DBA) block rather than communication (DCA)

Place the User Modules

PGA – CT Block ACB02

- In column with the Delta Sigma – same clock

DAC8 – Blocks ASD13/ASC23

- Connect output to the interface pin on the PSoC Pup demo board
- Better to keep blocks in a single column to maximize resources

PWM8 – Digital Block DBB10

- Use a basic (DBA) block rather than communication (DCA) block

User Module Requirement Analysis

UART

- UART datasheet specifies input clock as 8 times baud rate
- 8 times 9600 is 76,800KHz
- Get baud rate clock from PWM8

PWM8

- Generate a clock for UART at 9600 baud

DELSIG11

- Choose data rate and set the clock
- Connect to PGA for inputs

DAC8

- Enable the connection to the output buffer

Counter16

- Get an interrupt every ¼ second

PGA

- Set to unity or adjust gain for inputs to maximize ADC range

Configure the Global Resources

CPU_Clock: Set to 12MHz

32K_Select: Set to Internal

- Not using an external crystal

PLL_MODE: Set to Disable

- PLL can only be enabled when 32K_Select is External (crystal)

Sleep_Timer: Set to the default value of 512_Hz

VC1= SysClk/N: Set to 4

- This signal is used for the DELSIG11, divides 24MHz by 4 (6MHz)

VC2= VC1/N: Set to 12

- This defines the DAC8 clock, divides VC1 by 12 (500Khz)

VC3 Source: SysClk/1 (Default)

VC3 Divider: 1 (Default)

SysClk Source: Internal 24_MHz

SysClk *2 Disable: No (Default)

Configure the Global Resources

Analog Power: Set to SC On/Ref Low
- This is required to power up any of the analog blocks, depending on the number of analog functions, a Ref Med or Ref High may be required (and will increase power consumption)

Ref Mux: Set to (Vdd/2) ±Bandgap (the default)

AGndBypass: Disabled (Default)

Op-Amp Bias: Set to Low (the default)
- This is not recommended as anything but low

A_Buff_Power: Set to Low (default)
- This selects the power level of the analog output buffer
- There is a tradeoff between drive output power and power consumption, Low is adequate for most projects

SwitchModePump: Set to Off

Trip Voltage [LVD (SMP)]: 4.64V (5.00V) (Default)

LVD ThrottleBack: Disable (Default)

Supply Voltage: Set to 5.0V

Watchdog Enable: Disable (Default)

Configure the User Module Parameters

DELSIG11 (Select in the left center drop-down box)

- **TMR Clock set to VC1 (6MHz from global resources)**
- **Input set to ACB02**
- **Leave other parameters as defaults**

(read UM datasheet for more details)

Configure the User Module Parameters

PGA

- Gain: Set to unity or adjust gain for inputs to maximize ADC range by setting Gain to 1.000
- Input: Connect to input muxes by setting Input to AnalogColumn_InputSelect_ 2
- Reference: AGND (Analog Ground)
- AnalogBus: AnalogOutBus_2

Configure the User Module Parameters

DAC8

- AnalogBus: Set to AnalogOutBus_3
- Leave other parameters as defaults

(read UM datasheet for more details why)

Configure the User Module Parameters

PWM8

Generate a clock for UART at 9600 baud

- Clock: VC1 (6MHz)
- Enable: High (to keep the PWM always running)
- CompareOut: None
- TerminalCountOut: None
- Period to 77 (76923/78 = 9615, real close to 9600)
  - 78 – 1 = 77 because PWM counts down through 0
- PulseWidth: 38 (77/2, creates a square wave)
- Compare Type: Less than or Equal
- Interrupt Type: Terminal Count
- ClockSync: Sync to SysClk
- Invert Enable: Normal

Configure the User Module Parameters

UART

Get baud rate clock from PWM8 by connecting to broadcast

- (See slide 55)

Choose global buses for TX/RX (a bit of freedom, you can change your mind as the pinout is defined)

- Set RX Input to Row_0_Input_2
  - Connect Row_0_Input_2 to GlobalInOdd_6 (see slide 56)
- Set TX Output to Row_0_Output_0
  - Connect Row_0_Output_0 to GlobalOutOdd_4 (slide 57)
- TX Interrupt Mode: TXRegEmpty
- Leave other parameters as defaults

Configure the User Module Parameters

Counter 16

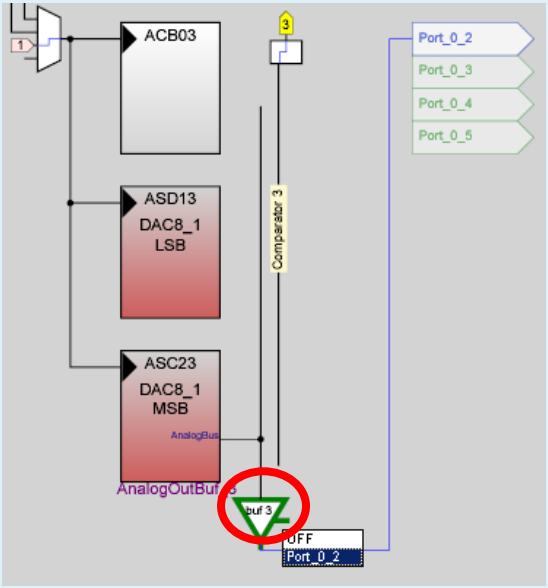
Create an interrupt every ¼ second

- Clock: Row_0_Broadcast (PWM8 output)
- Enable: High to allow Counter16 to always run
- CompareOut: None
- TerminalCountOut: None
- Period: 19230 (19231 – 1 as describe in datasheet)
- CompareValue: 0 (will count down from 19230)
- CompareType: Compare Less Than Or Equal
- InterruptType: Terminal Count
- ClockSync: Sync to SysClk
- Invert Enable: Normal

Interconnect Blocks to Resources

What interconnection possibilities are there?

- Input pins
- Output pins
- Clocks

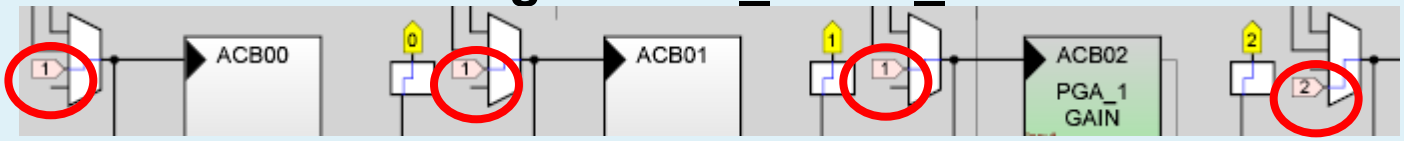


Connect DAC8 to Analog Buffer

- Go to the bottom right corner – “buf3”
- Select Port_0_2
- Output is already enabled from UM parameter definitions

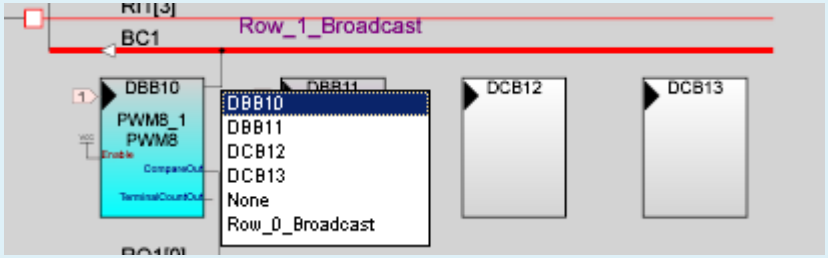
Check Set Up for all Clocks

- VC1 for AnalogColumn_Clock 0, 1, and 2
- VC2 for AnalogColumn_Clock_3

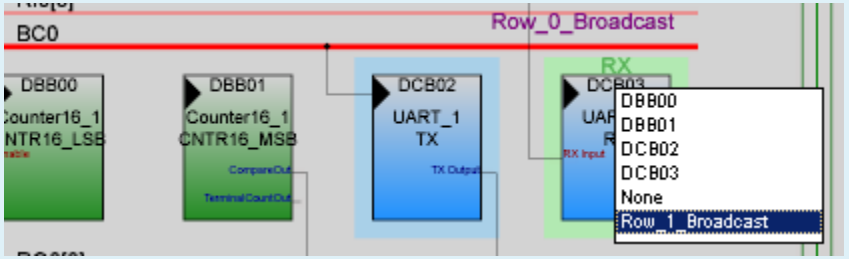


Routing PWM clock to UART

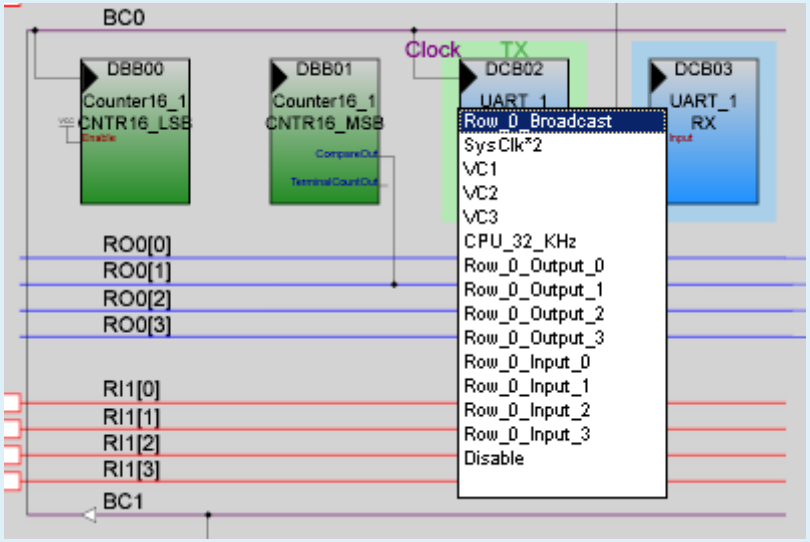
Connecting to Broadcast



First, Left Click BC1 and Select DBB10



Second, Left Click on BC0 and Select Row_1_Broadcast

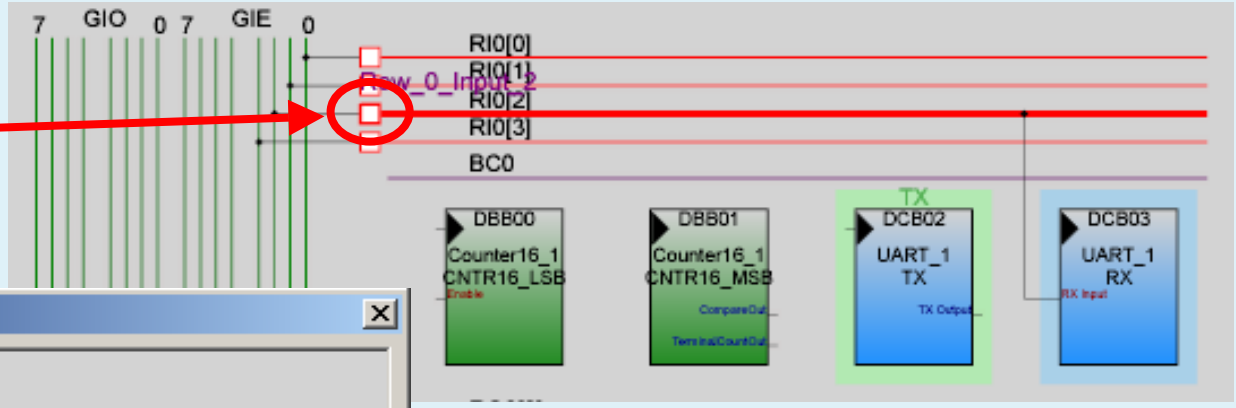


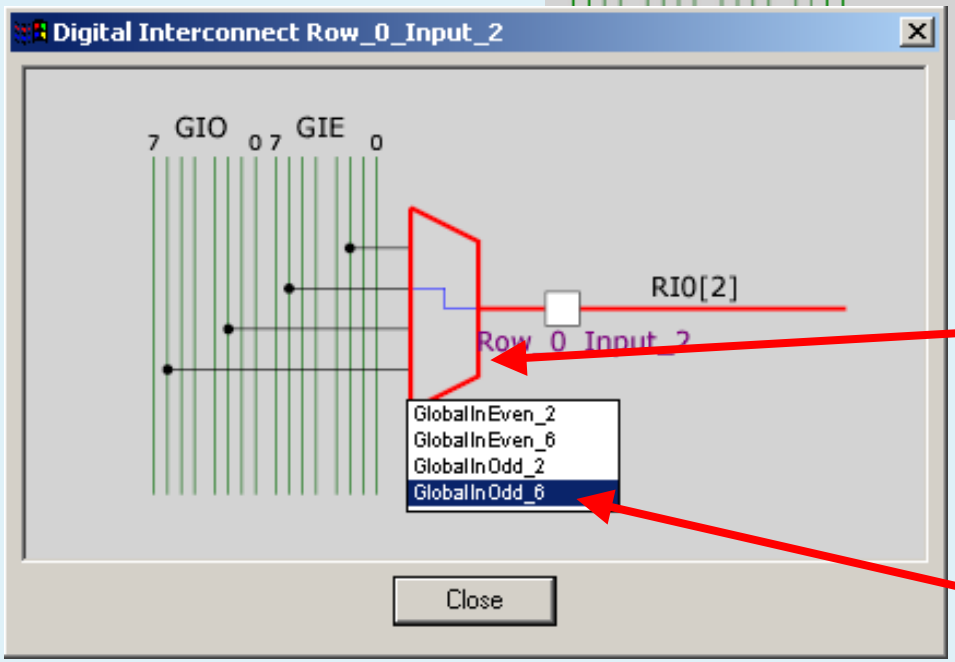
Third, Left Click on DCB02 Clock input and Select Row_0_Broadcast

Routing UART RX to Global In

Connect Row_0_Input_2 to GlobalInOdd_6

Left Click Here



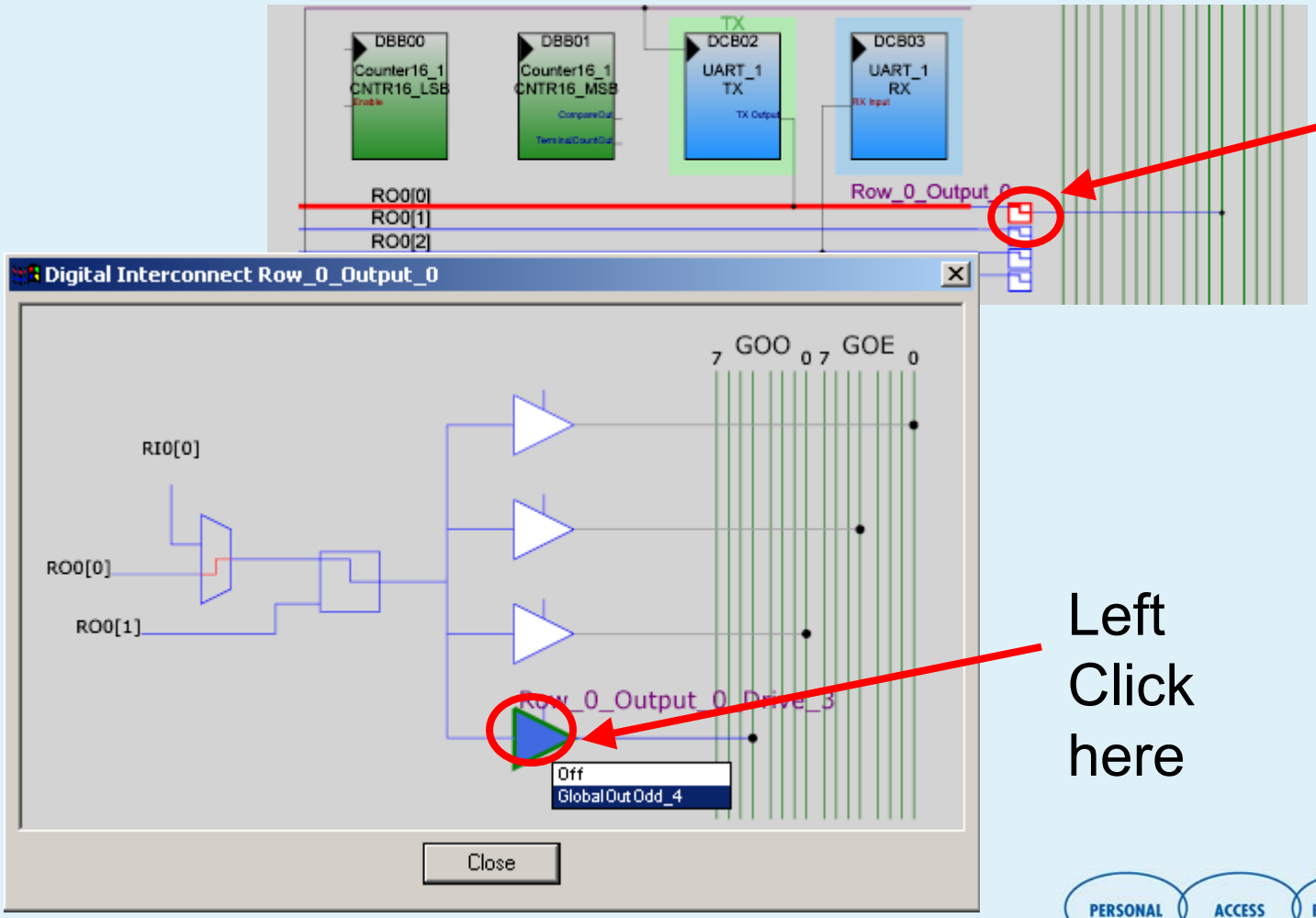


Left Click on MUX

Select GlobalInOdd_6

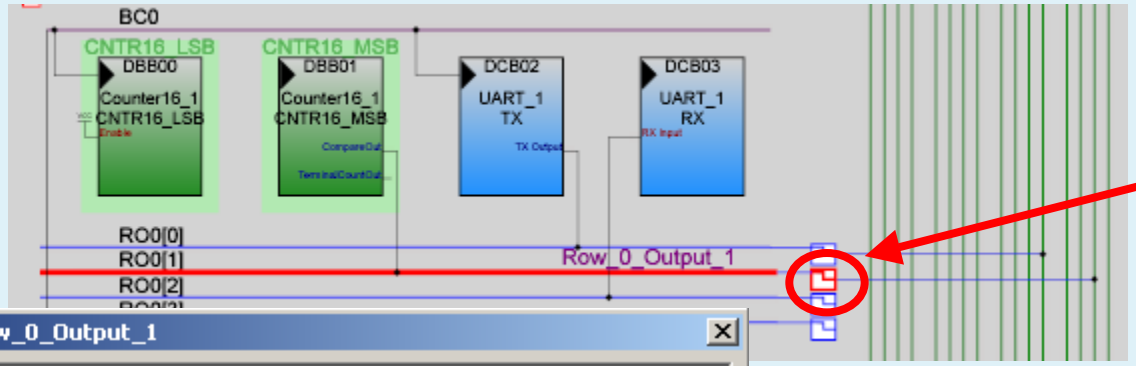
Routing UART TX to Global Out

Connect Row_0_Output_0 to GlobalOutOdd_4

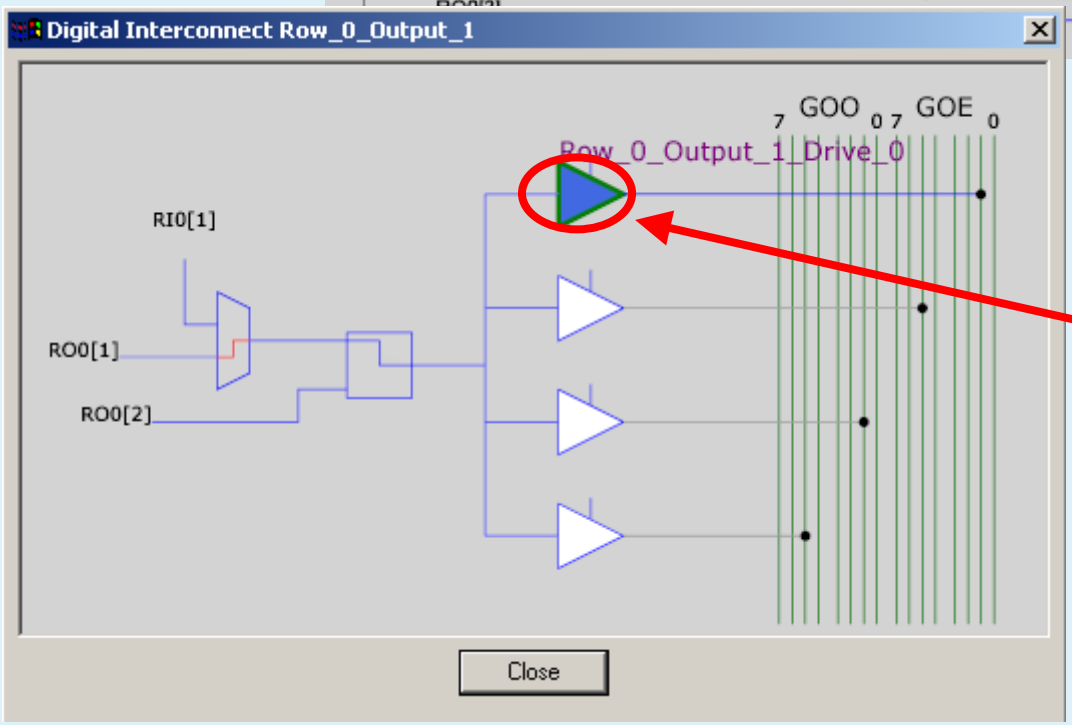


Routing Counter to Global Out

Connect Row_0_Output_1 to GlobalOutEven_1



Left Click here



Left Click here

Define the Pin-out

What pins need to be defined?

- UM Inputs
- UM Outputs
- General Purpose IO

What pin-out options are there?

- Permanent versus test/debug

What happens as pins are defined?

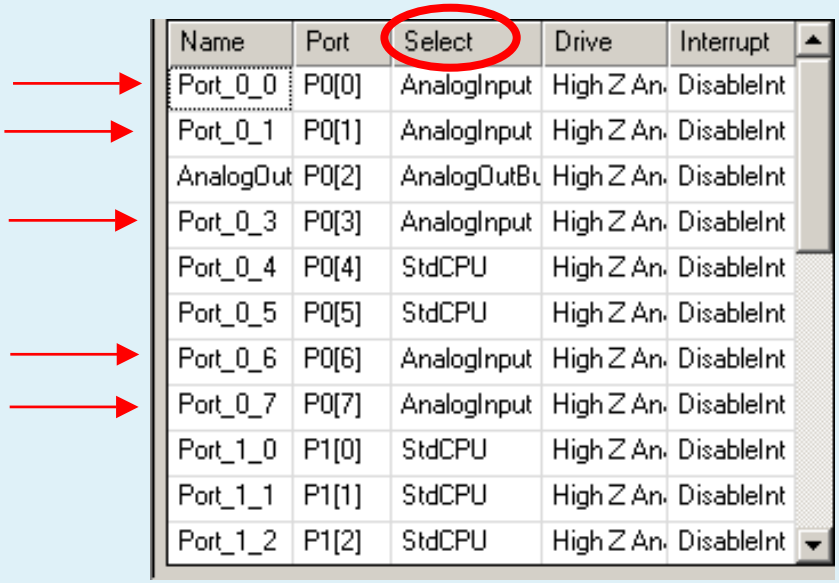
Pin-out our project

- LEDs
- SignalOut
- Power and Ground

PIN Out details
(to match Pup Board)

Port 0: Define the Analog Inputs (to Mux)

- P0[0], P0[1], P0[3], P0[6], P0[7]



Name	Port	Select	Drive	Interrupt
Port_0_0	P0[0]	AnalogInput	High Z An.	DisableInt
Port_0_1	P0[1]	AnalogInput	High Z An.	DisableInt
AnalogOut	P0[2]	AnalogOutBu	High Z An.	DisableInt
Port_0_3	P0[3]	AnalogInput	High Z An.	DisableInt
Port_0_4	P0[4]	StdCPU	High Z An.	DisableInt
Port_0_5	P0[5]	StdCPU	High Z An.	DisableInt
Port_0_6	P0[6]	AnalogInput	High Z An.	DisableInt
Port_0_7	P0[7]	AnalogInput	High Z An.	DisableInt
Port_1_0	P1[0]	StdCPU	High Z An.	DisableInt
Port_1_1	P1[1]	StdCPU	High Z An.	DisableInt
Port_1_2	P1[2]	StdCPU	High Z An.	DisableInt

Analog Output (Level is displayed to LEDs)

- Port 0 [2] – AnalogOutBuf3

PIN Out details
(to match Pup Board)

LED Display

- **Port 2 [0..7] and Port 0[4], Port 0[5] set to StdCPU/Strong**

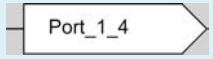
Name	Port	Select	Drive	Interrupt
Port_1_6	P1[6]	GlobalInOdd	High Z	DisableInt
Port_1_7	P1[7]	StdCPU	High Z An	DisableInt
Port_2_0	P2[0]	StdCPU	Strong	DisableInt
Port_2_1	P2[1]	StdCPU	Strong	DisableInt
Port_2_2	P2[2]	StdCPU	Strong	DisableInt
Port_2_3	P2[3]	StdCPU	Strong	DisableInt
Port_2_4	P2[4]	StdCPU	Strong	DisableInt
Port_2_5	P2[5]	StdCPU	Strong	DisableInt
Port_2_6	P2[6]	StdCPU	Strong	DisableInt
Port_2_7	P2[7]	StdCPU	Strong	DisableInt

Global Input: UART RX

- **Port 1 [6] – UART Rx (GlobalInOdd_6)**

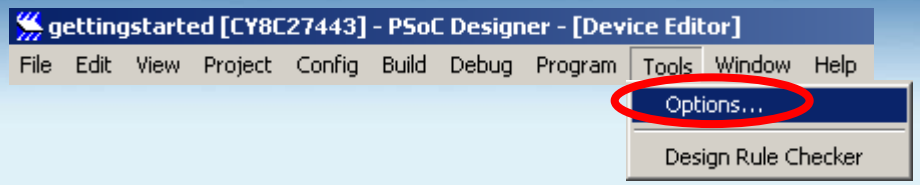
Global Output: UART TX

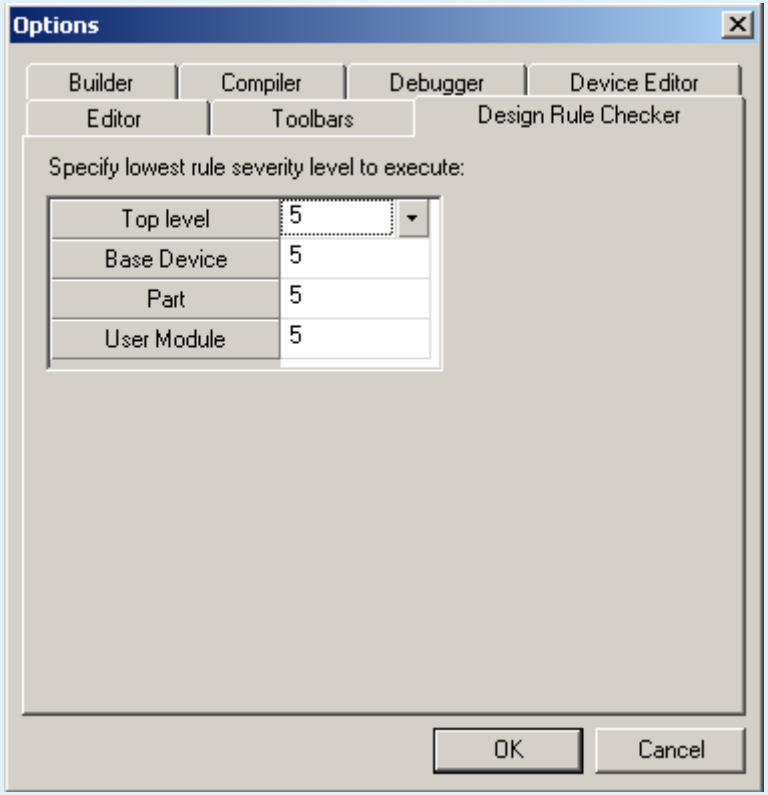
- **Port 1 [4] – UART Tx (GlobalOutOdd_4 (strong))**



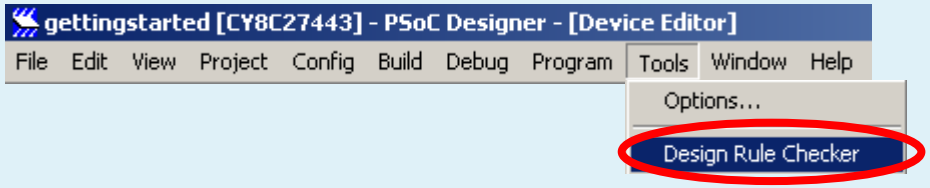
Design Rule Checker

First, setup the Design Rule Checker severity levels





Second, run the Design Rule Checker



PERSONAL ACCESS ENTERPRISE METRO CORE

Design Rule Checker

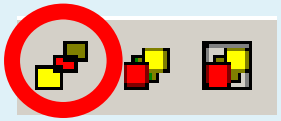
Rules Include:

- The CPU clock should not be set to 24MHz with 3.3V supply voltage
- The Drive should be set to High Z Analog for an Analog Input
- The ICE can only supply 5V, not 3.3V
- No placed User Modules should have undefined parameters
- The SC Blocks should most likely be powered on
- All User Modules should be placed
- Port_1_0 and Port_1_1 should be set to High Z for an external crystal
- Clock and ClockSync parameters of any placed User Modules must conform to the guidelines set in 17.1.2 of the CY8C27X data sheet

Configuration Complete!

Time to Save your project (Go to File→Save Project)

Time to Generate Application



- All settings combined by PSoC Designer to create the boot-up code to configure the registers at reset
- ISRs are created (but not updated)
- APIs are created or updated
- Device Data Sheet generated

You must do this whenever changes are made to the configuration

- But be careful with Interrupt changes

Time to add the Application Code

Cut and Paste code into Application

- The file “Code_for_Mod2.txt” contains ASM code to complete the project
- Insert as directed into main.asm and Counter16_1int.asm
- Note: The file contains both code excerpts.

Time to add the Application Code

main.asm should contain the following

```
include "m8c.inc"      ;include m8c specific declarations and
    macros
include "DAC8_1.inc"

export _main
export outputV

area  bss  (RAM)       ;inform assembler of variables to follow
outputV:      blk  1   ;declare global variable to hold output
    voltage

area  text (ROM, REL)          ;inform assembler of code to
    follow
```

Time to add the Application Code

```
_main:
    M8C_EnableGInt    ;macro to enable global interrupts m8c.inc
    file
    call        PWM8_1_Start    ;start PWM8_1 to provide clock
    call        Counter16_1_EnableInt ;enable Counter 16
    interrupt
    call        Counter16_1_Start ;start Counter 16

    mov         A,DAC8_1_LOWPOWER  ;specify low power for
    DAC
    call        DAC8_1_Start              ;starts DAC operation

    mov         [outputV],255             ;setup variable for
    maximum voltage value

mainloop:
    jmp         mainloop
    ret
```

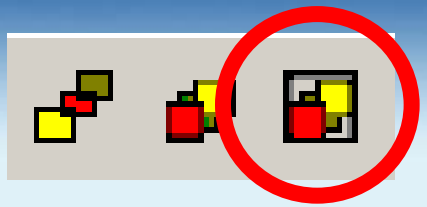
Time to add the Application Code

Counter16_1int.asm should contain the following:

```
_Counter16_1_ISR:

    push A          ;save A on stack
    dec             [outputV]           ;Decrement voltage variable and
    place on port 2
    mov             A,[outputV]
    mov             reg[PRT2DR],A
    call   DAC8_1_WriteStall            ;write new voltage variable to DAC8
    ;and wait for optimum write time
    mov             A,[outputV]         ;if voltage variable reaches 0 reset
    to maximum value
    jnz             branch1
    mov             [outputV],254
branch1:
    pop             A           ;restore A from stack
reti
```

Build Project



- Assembles code, links and locates
- Can individually assemble files as well
- Explore the Application Editor Features
  - Project file management (view/add/delete files)
  - Finding compilation errors

Code Reduction

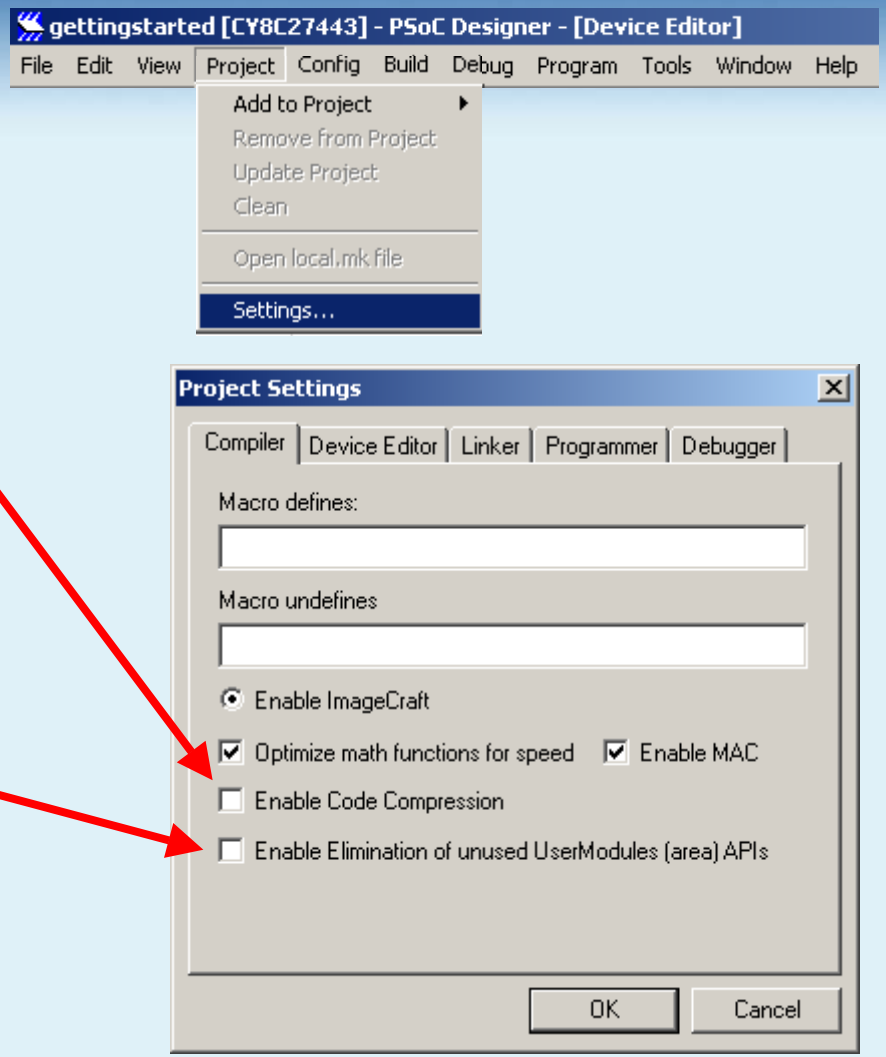
Code Reduction:

Enable Code Compression

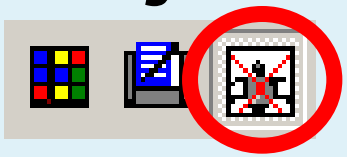
- Optimization for C compiler

Enable Elimination of unused User Modules (area) APIs

- linker will only pull in functions that are actually called while unused subroutines are ignored



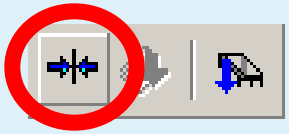
Execute Project Within Debugger



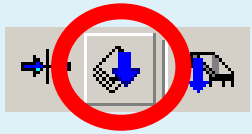
Switch to the Debugger – What's Different?

- **Looks like the Application Editor, but files are read-only**

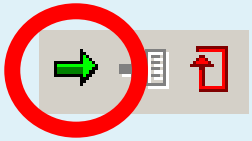
Connect to the ICE



Download the project to ICE



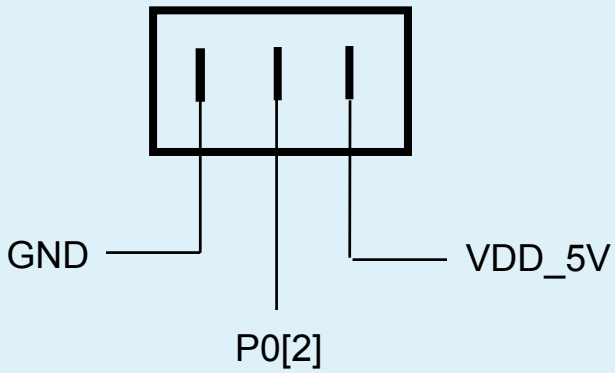
Run It!



Watch it work!

The DAC outputs to P0[2].

The Pup board has three pins that are as follows:



Our project was set up to output the voltage to P0[2]. So if you attach the multi-meter to the middle pin and the left pin on the Pub board, you will be able to watch the voltage decrement from 3.8 to 1.2 Volts as the LEDs count down the 256 steps.

The LEDs should show the value stepping down, corresponding to the level which is being generated by the DAC8.

Thank You!