Coverage process

The code coverage is based on GCC capability to instrument the code running on the target, with code coverage counters. Every block of code is allocated a counter and when this block is executed the counter is incremented.
GCC creates a **gcno** file during compilation for every instrumented module. The **gcno** file contains information that enables the matching of the run-time counters (**gcda** file) with the source code.

Downloading the counters to a **gcda** file and matching them to the **gcno**, enables us to create coverage report.

The **gcno** and **gcda** contains checksum and version information in order to insure compatibility.

The process involves three steps:

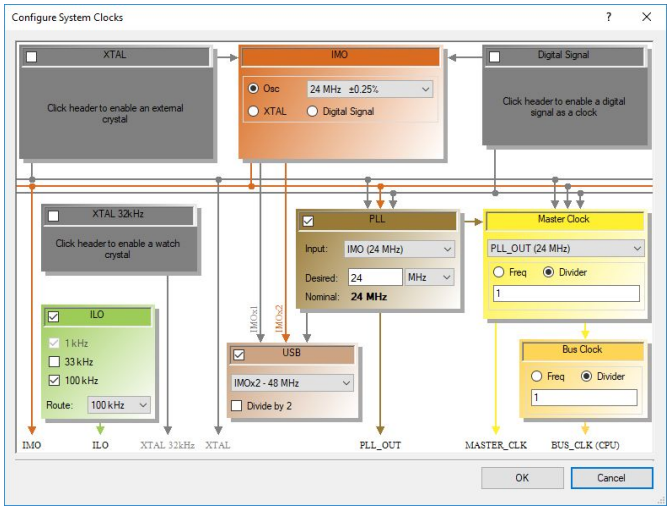
1. Target Instrumentation
2. Download coverage counters
3. Create Report

Target Instrumentation

USBUART

We shall use a USBUART to communicate with the host.

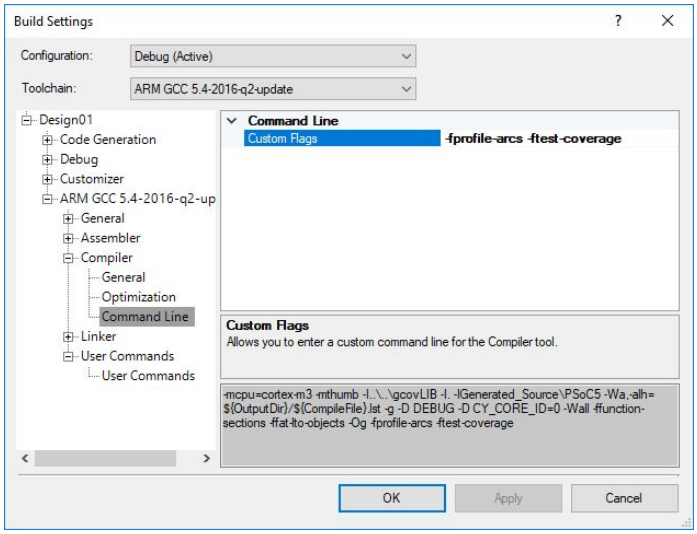
- Add a USBUART component to your design.
- Rename the component to embUART (just to be consistent with my example)
- Make sure the clocks are properly defined to support USB



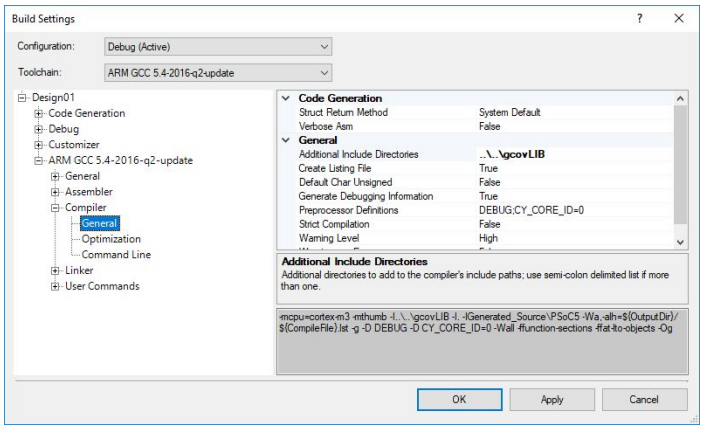
Setup build tools

Right-Click the project and select **Build Settings…**

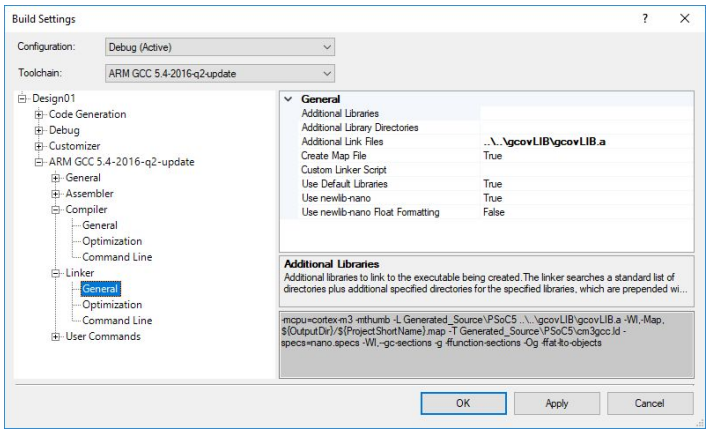
Add coverage flags



Add include directory to point to **psocGcov.h**



Link with **gcovLIB.a** library



Activating gcov

The following code is a main.c file example that initializes the gcovLIB.a and waits for the host communication

```
#include <project.h>
#include "psocGcov.h"

// add these flags to files to be instrumented :
// -fprofile-arcs -ftest-coverage

int main()
{
  USB_DEVICE usb;            /* create a gcov usb device */

  /* fill the embUART interface structure */
  usb.CDC_Init               = embUART_CDC_Init;
  usb.CDCIsReady             = embUART_CDCIsReady;
  usb.DataIsReady            = embUART_DataIsReady;
  usb.GetAll                 = embUART_GetAll;
  usb.GetConfiguration       = embUART_GetConfiguration;
  usb.IsConfigurationChanged = embUART_IsConfigurationChanged;
  usb.PutData                = embUART_PutData;
  usb.Start                  = embUART_Start;

  psocGcov_Init(&usb);      /* initialize gcov */

  CyGlobalIntEnable;
  for(;;)
  {
    psocGcov_Loop();        /* wait for host communication */
  }
}
```

The code does the followings:

- Include **psocGcov.h**
- Create a usb as USB_DEVICE and fill it with embUSB components API
- Call **psocGcov_Init(&usb)** once to initialize the coverage library
- Call **psocGcov_Loop()** periodically to process host packets

That's it for the smartSIM!

Download coverage counters

The utility gcovGET.exe shall be used to communicate with the target and download the coverage counters.

usage

gcovGET <COM port> <Baud>

Example

gcovGET COM9 115200

The files shall be downloaded into the same directory of the gcno files.
gcovGET shall create one gcda file per module (exactly like the gcno file).
The extension of the downloaded files shall be module_name.gcda_xxxx while the xxx is the time in ms.

Create report

The utility gcovREPORT.exe shall be used to create a text report from the gcno, gcda and c file.
The utility shall create one report per module in the same directory of the gcno file.
The extension of the report files shall be module_name.gcov.

usage

gcovREPORT<source_path\module_name.c> <build_path\module_name.gcno> <build_path\module_name.gcda>

Example

gcovREPORT C:\MyProj\source\src.c C:\MyProj\build\src.gcno C:\MyProj\build\src.gcda_567876