Final Project

Objective

The purpose of the final project is to demonstrate knowledge of the PSoC 5LP platform by creating a large project that uses a significant amount of hardware in the PSoC.

Criteria

You may choose any project you wish, as long as the project scores at least **12 points** from the following criteria. You may use a combination of any of these, and each bullet point may be used multiple times for different components, unless otherwise indicated.

- **1 point** – Make use of a schematic component covered by any of the labs, other than anything under Digital/Logic and clocks.
- **1 point** (only once) – Make use of at least one flip-flop.
- **1 point** (only once) – Use at least five components under Digital/Logic, excluding digital constant values, flip-flops, and the lookup table.
- **1 point** – Make use of a simple component **not** covered by the labs. Examples:
    - Lookup table
    - Anything under Digital/Utility
    - RTC – Real-time clock for accurate timekeeping
    - EEPROM – non-volatile memory on the PSoC
    - Die temperature – access the internal temperature of the PSoC's CPU
- **2 points** – Make use of a moderately complex component **not** covered by the labs. Examples:
    - Anything under Communications. Examples:
        - SPI – another peripheral bus that's faster than I^{2}C
        - emFile – reading files off the SD card with SPI
    - Any analog components, excluding analog multiplexers
    - Anything under Digital/Functions
- **4 points** – Make use of a complex component **not** covered by the labs. Examples:
    - Digital Filter Block (DFB) and Filter – a simple digital signal processor (DSP) inside the PSoC. See the datasheets for more information.
    - USBFS – Making use of it for anything other than USBUART.
- **4 points** – Create a custom component in PSoC Creator and make use of it in the project. The custom component must not duplicate the functionality of an existing Cypress component. For information on how to do this, go to Help>Documentation>Component Author Guide in PSoC Creator. Cypress also has a few tutorials on their website. Add only one of the following bonuses if they apply:

- **0 point bonus** – only make use of the schematic macro method of making a custom component. Components placed in the schematic macro do not count for any other criteria. Using this can allow for a cleaner design.
  - **2 point deduction** – structure is trivial, e.g. a few logic gates
- **2 point bonus** – make use of the UDB editor to create the custom component
- **4 point bonus** – make use of Verilog to create the custom component

If there is anything not clearly covered by the above criteria, ask your instructor for clarification.

Here are some things you **cannot** do in your project:

- Copy a part of any published PSoC project, including the example, excluding the labs.
- Create a trivial project. The criteria above attempt to avoid trivial projects by requiring a significant amount of schematic components or use of a complex topic not covered in labs.

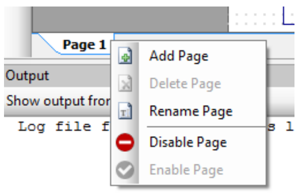
Suggestions

Here are some suggestions for final projects. Remember, you may choose anything as long as it meets the criteria.

- Emulate a hardware device that already exists
- Create a game
- Make use of one or more peripheral devices to create a sensor platform
  - Sparkfun and Adafruit are good websites for obtaining easy to use devices
- Control a robot with the PSoC
- Make the PSoC a peripheral device to another microcontroller
  - For example, use the PSoC to collect data from several forms of input, process it, and act as an I^{2}C or SPI slave to another device

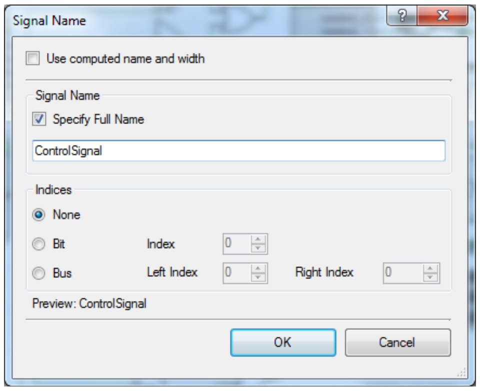
The instructor should provide a list of devices that they have available or are willing to provide for you to use for the project. You can also suggest your own hardware. The main guideline for using external hardware is to **be absolutely sure** those devices work with 3.3 volts for power and signals. You can also use the PSoC with 5 volt power and signals if necessary, but a power supply other than the USB cable is required, as detailed in section B4 in Lab 1.

Because this project will involve a lot of hardware, it should be very difficult to fit everything on one schematic. To create another schematic, right click on the tab at the bottom left of the schematic (named "Page 1" by default) and select "Add Page." There is nothing special required to use multiple schematics.



It is also possible to connect components on one schematic to components on another schematic. To do so:

1. Add a sheet connector to the first schematic by selecting the green diamond near the wire tool or pressing the S key and clicking on the schematic.
2. Connect a wire between the sheet connector and the component that you want to connect to another component.
3. Double click on the wire segment connected to the sheet connector or right click the wire segment and select "Edit Name and Width."
4. Uncheck the "Use computed name and width" box. Specify a name for the wire under the "Specify Full Name" checkbox.



Signal Name

[ ] Use computed name and width

Signal Name

[✓] Specify Full Name

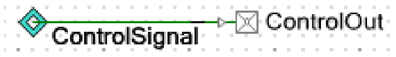
ControlSignal

Indices

(●) None
( ) Bit — Index: 0
( ) Bus — Left Index: 0 — Right Index: 0

Preview: ControlSignal

5. Add another sheet connector near the destination component. Connect a wire from the sheet connector to the component.
6. Edit the name of the wire segment connected to the sheet connector and give it the same name as the origin wire segment.



Note that the sheet connectors simply connect wires to each other; the direction a signal travels depends on where the source and destinations are. You can make multiple sheet connectors for the same named wire segment as well. Sheet connectors can also connect wires on the same schematic, so it can be used to make designs look cleaner by reducing long wires.

The Name and Width editor is also useful if it is necessary to use buses (wires that contain more than one bit of data), especially when wires need to be combined into a bus or a bus needs to be split up into wires. The Output schematic of the example project demonstrates using buses and splitting or combining signals.

Example

An example project has been provided to illustrate a large project that a student could do. This example project emulates the YM2149 programmable sound generator used in the Atari ST. The details of this project are provided in a separate document. Because this project has full implementation details, you may **not** implement the same project, even if you plan to change the implementation. However, you could implement a different programmable sound generator or music synthesizer. This example project has a criteria score of **20**.