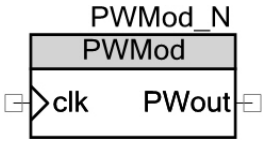
Pulse Width Modulated Density Generator

PWMod v1.0

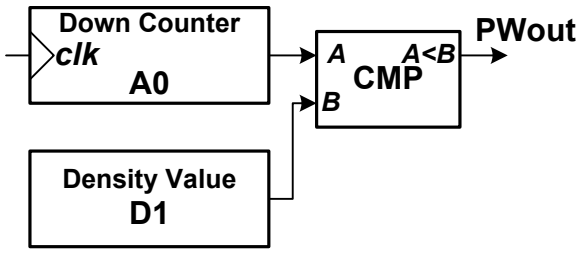


Features

- Synchronous operation
- Positive-edge triggered clock
- Produces a pulse width modulated density stream
- Parameterized density value
- Simple to deconstruct

General Description

A pulse width modulator produces the lowest possible output frequency for a given density. This implementation is a down counter and an 8-bit wide digital comparator. The output is HIGH whenever the counter value (A0) is less than the density value (D1).



For a period of 256 cycles the output will be HIGH for a continuous number of cycles equal to the density value.

This component was built as a teaching tool. This classic component's operation is well understood and this datasheet's function is to help understand how the component was built through its deconstruction. It is recommended that you review the DSMod component before attempting this one.

Pin Description

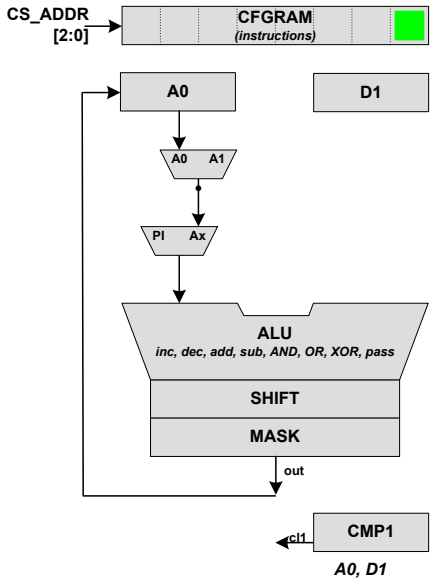
Pin	Type	Function
clk	input	clock input
DSout	output	pulse width density stream

Function Table

Operating Modes	Input	Register	Output
	clk	A0	PWout
count	↑	A0-1	A0 < D1

Deconstructing the Component

This component's purpose is to show how to connect an output from one of the datapath compare registers. For more information about datapaths, first review the DSMod component. This component uses the following pieces of a datapath.



The single operation to be performed is that on each clock cycle, A0 is fed to the ALU were it is decremented and fed back to A0. CMP1 is used to compare A0 and D0. The comparator's output is HIGH whenever A0 is less than D0. This flag (cl1) is brought out.

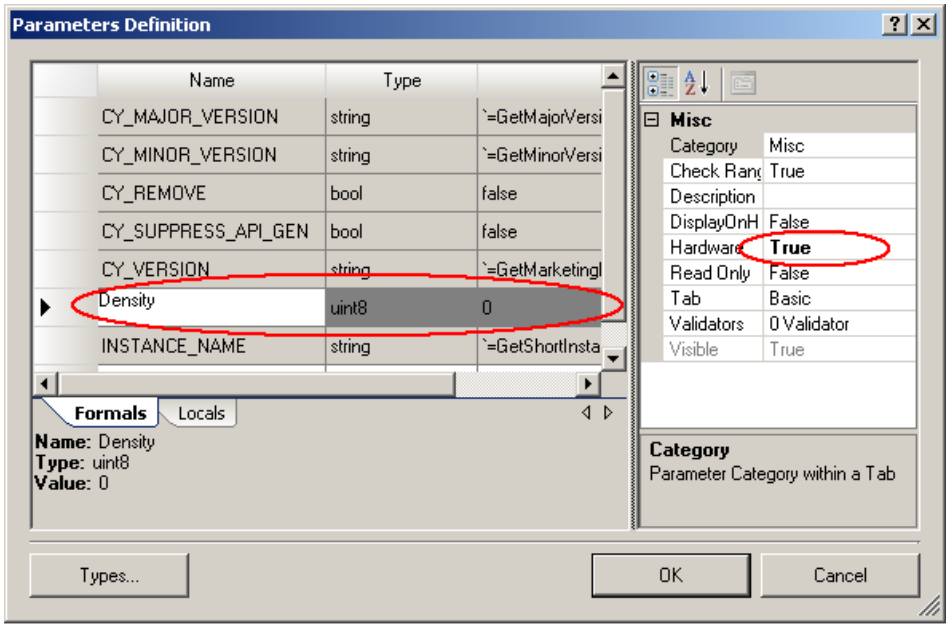
Using PSoC Creator, open the PWMod example project to see the project schematic (*TopDesign.cysch*). It has a PWMod component connected to an output LED and a clock.

In the Workspace Explorer, click the **Components** tab. Then, right-click on the project and select **Import Component**. Navigate to where the CYCC_SimpleComponentLibrary project is, and select the CYCC_PWMod_v1_0 component. Click **OK** and the following files are shown for the component:

- Symbol file (cysym)
- Datasheet (pdf)
- Verilog file (v)

Open the symbol file and you will find a symbol with one input and one output. It looks like the symbol shown on the first page.

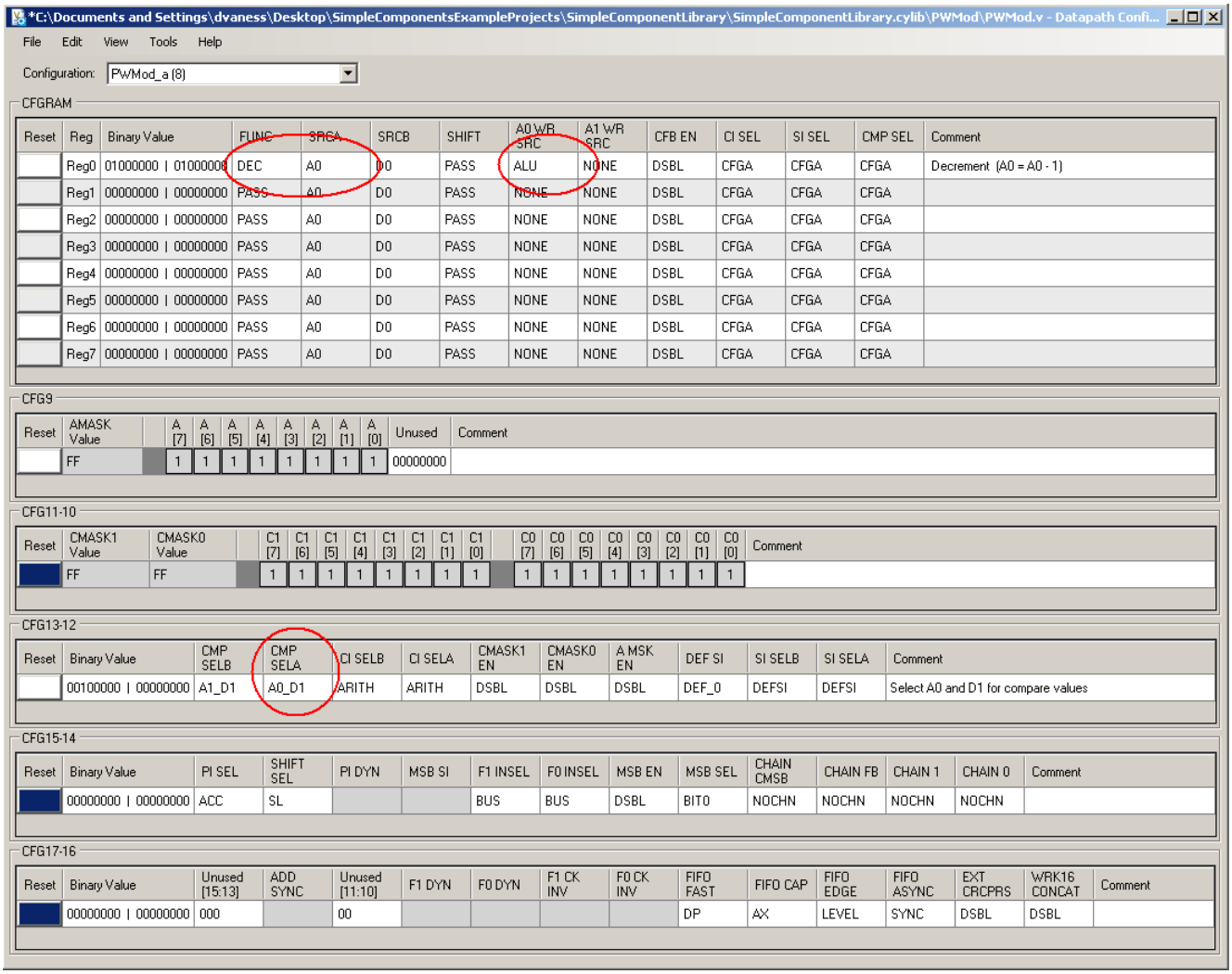
While viewing the symbol file, right-click on the canvas and select **Symbol Parameters**.



Note that **Density** is included as a parameter. It is an unsigned 8-bit value that has a default value of zero. Its hardware flag has been enabled.

Open the Verilog file and notice that at lines 22-23, these definitions were passed from the symbol to this Verilog file when it was created. Line 25 is where the symbol Density parameter got passed. The first 16 lines of this header list register usage and the datapath instruction definitions.

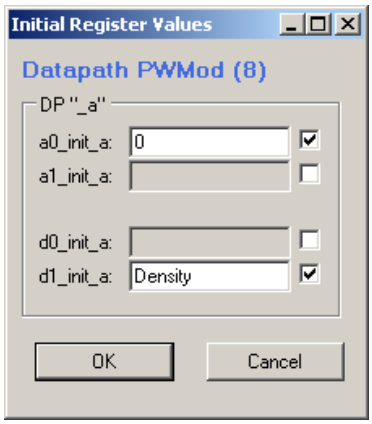
What follows is the datapath module definition. It was created and inserted by the Datapath Configuration Tool. This information is backward compatible so opening up the Datapath Configuration Tool for this Verilog file results in the following.



Note that only one operation is needed to be implemented and that CMP_SELA is set so A0 and D1 are compared.

For the dec instruction A0 is decremented and the result is fed back to A0.

These two registers have been initialized as shown below.



When saved, this tool inserts the updated configuration data back into the Verilog file along with an instance of the datapath interface. Just pass the correct signals into and out of it and you are done.

This counter parameter list may look ominous but only a few parameters have to be entered as shown below.

```
    /* input           */  .clk(clk),           // Connect a clock to the datapath
    /* input [02:00]   */  .cs_addr(3'b00),     // Only Reg0 instruction is used
    /* output          */  .cl1(PWout),         // Get compare flag from datapath
```

These parameters do the following:

- Connect a clock signal to the datapath clock circuitry
- Only a single datapath instruction (REG0) is used
- Bring the CMP1 flag to an output

Support

PSoC Creator Community Components are developed and supported by the Cypress Developer Community. Go to www.cypress.com/CommunityComponents to discuss this and other Community Components.