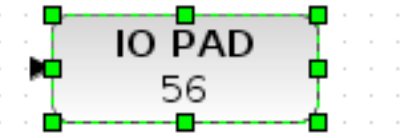
Utilizing the Counter and Resulting Timing Structure on the FPAA device

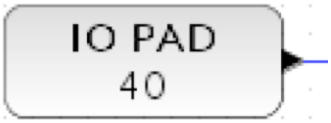
This discussion focuses on utilizing the counter and resulting timing code on the FPAA device. This code can be useful for interfacing a number of circuits, such as the back end of a ramp ADC converter.

A significant amount of the resulting structure is simply abstracted away from the user for nearly transparent operation. Two key blocks are important:

IOpad56: This pin is used to make the counter operational. If the value is high, then the counter proceeds. If the value is low, then the counter stops. Each value is put into SRAM memory, with the data location pointer incrementing after each sample.



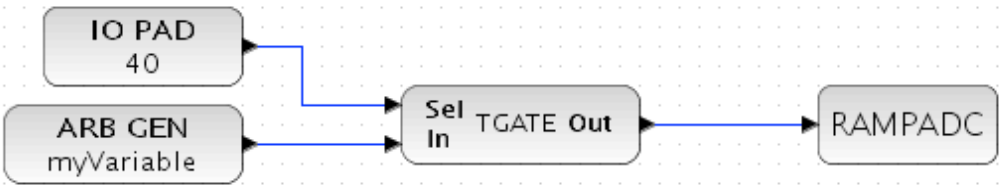
IOpad40: This pin outputs when the counter is not counting or has finished counting. This signal could be used for sample and hold input. The assembly code would hold the value (out is 0) when it is counting, and samples (out is 1) when it is done counting. The approach will enable regular sampling from this pin.



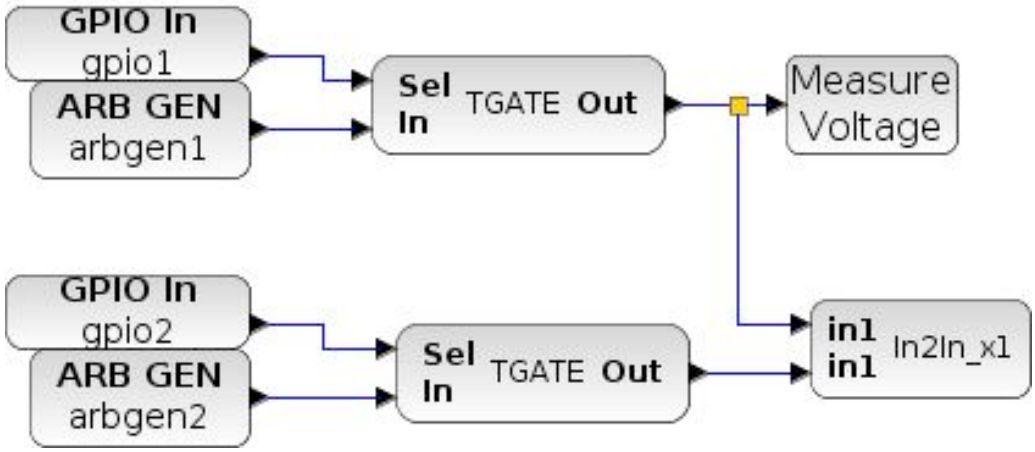
GPIO In: This block enables the input of a vectorized digital variable to be communicated to the hardware fabric.

The example block would take the *vectorized* variable (gpio_in_var; can be choosen by the user; defined like myvariable) of digital values and cycle through these values (stored in SRAM). The core of this block is a similar framework as the arbgen block.

The figure below is a test structure that can be used, utilizing the RAMPADC block (which does not have a S/H block), using a Tgate as a simple S/H block. This block uses the IOpad40 to indicate when the counter is finished / not counting.



We have a number of useful digital structures. A useful test block is shown below:

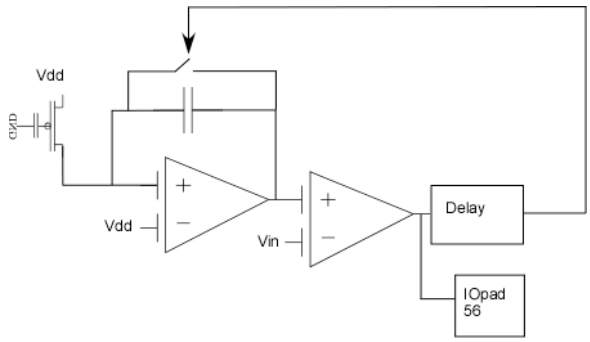


This structure enables testing GPIO (General Purpose I/O, digital lines in the FPAA fabric) with T-gates. This structure has been previously tested (Vin1=1V, Vin2=2V).

The In2in_x1 block creates a connection between two output terminals, not typically allowed in such graphs, but absolutely acceptable for the circuits. If you are using an ADC, such as the ramp ADC block, it might not work for signals near the rail. In fact, the resulting control structure might end up hanging the entire processor system. Yes, a place that we should make things more robust, but something to consider at this point. So, for example, if you take GPIO_In block (0V and 2.5V output) and connect it to the RAMP ADC block, you will get some unusual (and potentially unpredictable) responses. Sometimes it will just hang up the setup. If you use Measure voltage, within the sample rate, you should get predictable results.

Using the Ramp ADC (e.g. to characterize something in the test) while using the counter infrastructure will cause a collision between systems.

For reference, the RAMP ADC block design is shown below, which uses this resulting counter system.



The cap around the amplifier results in a linear capacitor by making the feedback capacitor important, and parasitic capacitances are decreased by the amplifier gain. It also roughly fixes, within the amplifier gain, the drain voltage for the FG pFET current source. It does mean the ramp does not go rail to rail, though, so one should be careful when using that block. This block does not have a S/H.