

Cntr8

Introduction

Cntr8 is a simple 8-bit up counter that uses a UDB Datapath element to minimize logic resources. It uses exactly a single Datapath, 3 Macro-cells, and 4 P-terms. The Macro-cells and P-terms make up a 3-state state-machine that controls the Datapath. No status or control registers are used.

Normally the counter runs with its default period of 255, counting starting from 0. There is no way to make this particular design work with a full period of 256, unfortunately.

The count always restarts immediately after reaching a count of 255. Periods shorter than 255 are implemented by changing the initial count value. That makes the design simpler, by using the built-in Datapath capability of generating a signal when the register value is 255.

Terminals:

Input:

- Clock This will normally be the bus clock
- Reset A single pulse of at least one clock period will reset the state machine and the counter
- A The signal that is being counted. The counter will increment at each rising edge

Output:

- TC Terminal count signal. This goes high for one clock period at the end of each period.

The TC output signal can be used to trigger an interrupt request.

The internal count itself is not exposed in hardware but can be read by the CPU.

API

WritePeriod

`void Cntr8_WritePeriod(uint8 period)` Sets the counter starting value to obtain the desired period

Note: the Cntr8 default is to start counting from 0. This function sets the initial count to $255 - \text{period}$. The maximum period is 255. Setting the period to zero would not be a great idea, but this is not checked.

ReadCount

`uint8 Cntr8_ReadCount()` Reads the current count value directly from the counter register

Note: it is not clear what the result will be if the count is read during a counter transition. Ideally, one should disable the input signal before reading the counter.