Impulse C Instrumentation Library

A design note that describes the usage and operation of the Impulse C instrumentation library

Introduction

This Impulse C profiling library is a simple macro-based interface that can be used to gather timer-based profiling information for Impulse C applications compiled for the Xilinx ML507 development platform. It requires that a timer IP component has been added to the system and synthesized for use.

Functions

The library consists of two very primitive pragma directives that can be inserted anywhere inside the Impulse C application code. A brief description and usage example for each is shown below.

#pragma START_TIME(startCycles)

This directive is an indication that the timer module should be reset and the new clock cycle value should be stored in a variable called ‘startCycles’.

#pragma END_TIME(endCycles)

This directive is an indication that the current clock value should be stored in a variable called ‘endCycles’.

NOTE: Each START_TIME directive should be accompanied with an appropriate END_TIME directive. Otherwise, the timing logic will not be correct and this will produce erroneous results.

The code below illustrates the use of these directives to clock the amount of time it takes to stream data to the FPGA.

```
#pragma START_TIME(cycles)
        co_stream_open(input, O_WRONLY, INT_TYPE(STREAMWIDTH));
        {
                for (index = STREAMDEPTH; index >= 0; index--)
                {
                        co_stream_write(input, &index, sizeof(co_int32));
                }
        }
#pragma END_TIME(cycles)
```