**FPGA Implementation document**

**1. Higher level overview and Connections:**

FPGA\_Random\_Walk\_Module can are defnied in bram\_random\_walk.v file (Verilog to be found in github repository), and it can be instantiated into multiple instances depending on how many processing is needed for the application (the following picture shows the 2 FPGA\_Random\_Walk\_Module instantiation)

Diagram

Description automatically generated

The connections to ZCU102 PS is described in section 4.

**2. What and how Data are stored in each BRAM Block:**

The following diagram shows what stores in the each BRAM\_Block

A picture containing text

Description automatically generated

**3. Verilog files and modules:**

Under <https://github.com/lixali/FPGA_PPR/tree/master/FPGA/ultra96_random_walk/>, the following Verilog files are created and used to constructed the above circuits.

*BRAM.v (module definition of BRAM block)*

*Clock\_divider.v (clock divider modulethat make main clocks into various clocks with various frequency)*

*LFSR.v (linear feedback shift register to mimic generating random numbers in FPGA)*

*BRAM\_random\_walk.v (random walk module that read data from BRAM and calculate the counter table)*

**4. Simulations Waveform**

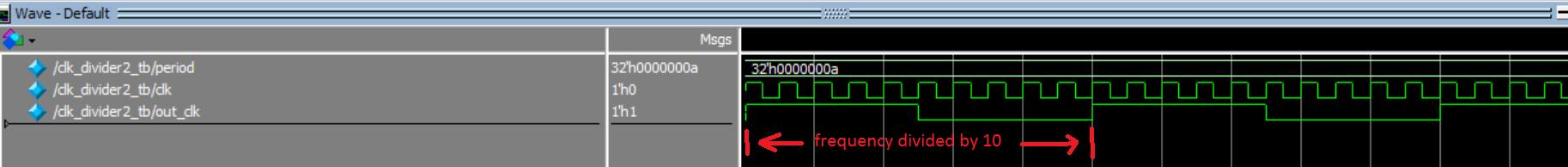
The following waveforms the simulations result from various block (all test bench files are committed to the same directory as the verilog module files)

BRAM basic read and write simulation

A picture containing graphical user interface

Description automatically generated

Clock frequency divider ()



Linear Feedback Shift Register

**4. Connections to zcu102’s PS**