

Vexriscv SoC with UART & RAM

Axi Ram application code running on a Vexriscv. This design contains a Vexriscv processor, ON chip axi ram and UART.

Instructions:

You can follow the below steps to generate the designs and simulate the application on Verilator.

Generate Verilog for the LiteX design (No Simulation)

```
litex_sim --cpu-type vexriscv --axiram --no-compile-gateway
```

Generate and Simulate the verilog for the LiteX design

Here We simulate the Axi Ram example using litex_sim_rs script provided in the example design directory.

The following command generates your SoC:

```
~/litex_instll/litex_rs/raptor_example_designs/Vexriscv_axi_ram/litex_sim_r  
s.py --integrated-main-ram-size=0x10000 --cpu-type=vexriscv --axiram --no-  
compile-gateway --sim-debug
```

Generate binary for the application code

Run the following command to generate .bin file:

```
python3  
~/litex_instll/litex_rs/raptor_example_designs/Vexriscv_axi_ram/test/demo/d  
emo.py --build-path=build/sim
```

Simulating the application using Verilator

Run the following command to execute your application code onto the processor:

```
~/litex_instll/litex_rs/raptor_example_designs/Vexriscv_axi_ram/litex_sim_r  
s.py --integrated-main-ram-size=0x10000 --axiram --cpu-type vexriscv --ram-  
init=demo.bin --sim-debug
```

Output:

```
=====
-----TEST-RESULT-----
=====

TEST-1: PASSED

TEST-2: PASSED

TEST-3: PASSED

TEST-4: PASSED

TEST-5: PASSED

TEST-6: PASSED

TEST-7: PASSED

TEST-8: PASSED

=====
-----END-----
=====
```

Application

This application code does multiple tests onto the axiram, these tests do multiple write and read sequences on the ram to test the integration of the IP.