HIL Example Guide and Instructions

1. This example design demonstrates a simple Hardware In the Loop (HIL) design. The design moves data over the JTAG connection using the run_hil.tcl Tcl script. That data is written to the test_source_ram RAM block. The source_dma block than reads the data and transmits it to the DUT. The sink_dma block reads the data from the DUT and writes it to the test_sink_ram. The data is than read from the test_sink_ram block over the JTAG connection using the run_hil.tcl Tcl script.
2. Edit Windows PATH environment variable.
   a. Open the Start Search, type in "env", and choose "Edit the system environment variables":
   b. Click the "Environment Variables…" button near the bottom.
   c. In the "User variables for <user>" frame highlight "Path" and then press "Edit..."
   d. Use the "New" or "Edit" button to ensure the following paths have been added.

```
C:\intelFPGA_pro\21.4\quartus\bin64
C:\intelFPGA_pro\21.4\syscon\bin
```

3. Verify correctness of PATH environment variable.
   a. Double click the open_cmd_prompt.bat file to open a Windows Command Prompt in the present working directory.
   b. From the Windows Command Prompt run path_env_check.bat. The output should look something like the following:

```
This script will check to ensure your PATH environment variable
is set correctly.

Testing system-console path...
C:\intelFPGA_pro\21.4\syscon\bin\system-console.exe

Testing quartus path...
C:\intelFPGA_pro\21.4\quartus\bin64\quartus_sh.exe
```

   If you see a "Could not find files..." message you will need to correct your PATH environment variable prior to proceeding.

```
Testing quartus path...
INFO: Could not find files for the given pattern(s).
```

4. Unzip HIL_DMA_Example.zip file.
   a. Below is a description of the files included in the HIL_DMA directory.

Filename	Description
cleanup.bat	Removes all restored and generated files.
compile_design.bat	Compiles the FPGA from a Windows Prompt.
hil_dma_23_2_0_94.qar	Quartus project archive file, including Platform Designer system and Verilog files.
HIL_DMA_Guilde.pdf	Descriptions and instructions for the example design.
open_cmd_prompt.bat	Opens a Windows Command Prompt in the present working directory.
path_env_check.bat	Windows batch file to help check correctness of PATH environment variable.
program_fpga.bat	Windows batch file script to run the program_fpga.tcl and toggle_issp.tcl scripts from a Windows Prompt.
program_fpga.tcl	Tcl script to assist in the programming of the FPGA.
restore_qar.bat	Windows batch file script to restore the Quartus project from the Quartus archive (.qar file)
run_hil.bat	Windows batch file script to run the run_hil.tcl application.
run_hil.tcl	Software application file written in tcl.
toggle_issp.tcl	Tcl script to toggle the internal reset register using In System Sources and Probes (ISSP).

5. Restore and View Quartus project.
    a. From the command line type “restore_qar.bat”
    b. Open the Quartus project and the Platform Designer system (sys.qsys). The system consists of the following:
        i. **clk_in** and **reset_in** – Used to clock and reset the design.
        ii. **jtag_master** - Used to provide a bridge between jtag and an Avalon Memory Mapped interface. Allowing Tcl API commands to communicate with agent/slave devices on the Avalon bus.
        iii. **test_source_ram** – RAM block containing data to be send to DUT.
        iv. **source_dma** – msgDMA engine used to read data from test_source_ram and send the data to DUT.
        v. **dut** – Device under test, in this simple example the DUT is just a FIFO.
        vi. **sink_dma** – msgDMA engine used to receive data from DUT and write it to test_sink_ram.
        vii. **test_sink_ram** – RAM block used to store data read from DUT.

Use	Con...	Name	Description
✔		clock_in	Clock Bridge Intel FPGA IP
✔		reset_in	Reset Bridge Intel FPGA IP
✔		jtag_master	JTAG to Avalon Master Bridge Intel FPGA IP
✔		test_source_ram	On-Chip Memory (RAM or ROM) Intel FPGA IP
✔		source_dma	Modular Scatter-Gather DMA Intel FPGA IP
✔		dut	Avalon Streaming Single Clock FIFO Intel FPGA IP
✔		sink_dma	Modular Scatter-Gather DMA Intel FPGA IP
✔		test_sink_ram	On-Chip Memory (RAM or ROM) Intel FPGA IP

6. Compile Project.
    a. The project is currently targeted towards an Arria 10 development board. You will need to change the target to match your board and update the location of the "clk" pin to match your board.
    b. You can compile the FPGA from within Quartus or from the command line by running the compile_design.bat file.

7. Program the FPGA.
    a. Connect your computer to your board and run the "program_fpga.bat" from the command line. This script will program the FPGA and toggle the built-in reset register (using In System Sources and Probes).

8. Run the software application.
    a. The run_hil.tcl script perform the following steps:
        i. Write a series of 32 incrementing words to the test_source_ram RAM block. The script will print the following message during this step:
            - *Initializing read DMA buffer…*
        ii. Write 32 words of zeros to the test_sink_ram RAM block. The script will print the following message during this step:
            - *Clearing write DMA buffer…*
        iii. Read and display the status of both the test_source_ram and test_sink_ram blocks. The script will print the following message during this step:
            - *DMA buffers initialized and cleared.*
        iv. Program the write/sink DMA descriptor. This step informs the write DMA engine to write 32 words into the test_sink_ram RAM block. The script will print the following message during this step:
            - *Program write DMA descriptor…*
        v. Program the read/source DMA descriptor. This step informs the read DMA engine to read 32 words from the test_source_ram RAM block. The script will print the following message during this step:
            - *Program read DMA descriptor…*
        vi. Once the read DMA descriptor is sent the data will begin to be read from the test_source_ram block and moved through the device under test (DUT), which in this case is a simple FIFO. The write DMA will collect the data from the output of the DUT and move the data into the test_sink_ram RAM block. The script will print the following message

during this step:

- *Sending test data to Device Under Test*

vii. Read and display the status of both the test_source_ram and test_sink_ram blocks. The script will print the following message during this step:

- *DMA buffers:*

b. From the command line run the “run_hil.bat” script. You should see the following output.

```
Opening JTAG master service path...
Initializing read DMA buffer...
Clearing write DMA buffer...
DMA buffers initialized and cleared.
Reading source RAM data:
0x00000001 0x00000002 0x00000003 0x00000004 0x00000005 0x00000006 0x00000007 0x00000008 0x00000009 0x0000000a 0x0000000b 0x0000000c 0x0
000000d 0x0000000e 0x0000000f 0x00000010 0x00000011 0x00000012 0x00000013 0x00000014 0x00000015 0x00000016 0x00000017 0x00000018 0x0000
0019 0x0000001a 0x0000001b 0x0000001c 0x0000001d 0x0000001e 0x0000001f 0x00000020
Reading sink RAM data:
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x0
0000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x0000
0000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
Reading Sink DMA status...
0x00000002
Program write DMA descriptor...
Reading Sink DMA status...
0x00000003
Program read DMA descriptor...
Sending test data to Device Under Test
DMA buffers:
Reading source RAM data:
0x00000001 0x00000002 0x00000003 0x00000004 0x00000005 0x00000006 0x00000007 0x00000008 0x00000009 0x0000000a 0x0000000b 0x0000000c 0x0
000000d 0x0000000e 0x0000000f 0x00000010 0x00000011 0x00000012 0x00000013 0x00000014 0x00000015 0x00000016 0x00000017 0x00000018 0x0000
0019 0x0000001a 0x0000001b 0x0000001c 0x0000001d 0x0000001e 0x0000001f 0x00000020
Reading sink RAM data:
0x00000001 0x00000002 0x00000003 0x00000004 0x00000005 0x00000006 0x00000007 0x00000008 0x00000009 0x0000000a 0x0000000b 0x0000000c 0x0
000000d 0x0000000e 0x0000000f 0x00000010 0x00000011 0x00000012 0x00000013 0x00000014 0x00000015 0x00000016 0x00000017 0x00000018 0x0000
0019 0x0000001a 0x0000001b 0x0000001c 0x0000001d 0x0000001e 0x0000001f 0x00000020
Close connection to FPGA target
C:\work\HIL_DMA_Example>
```