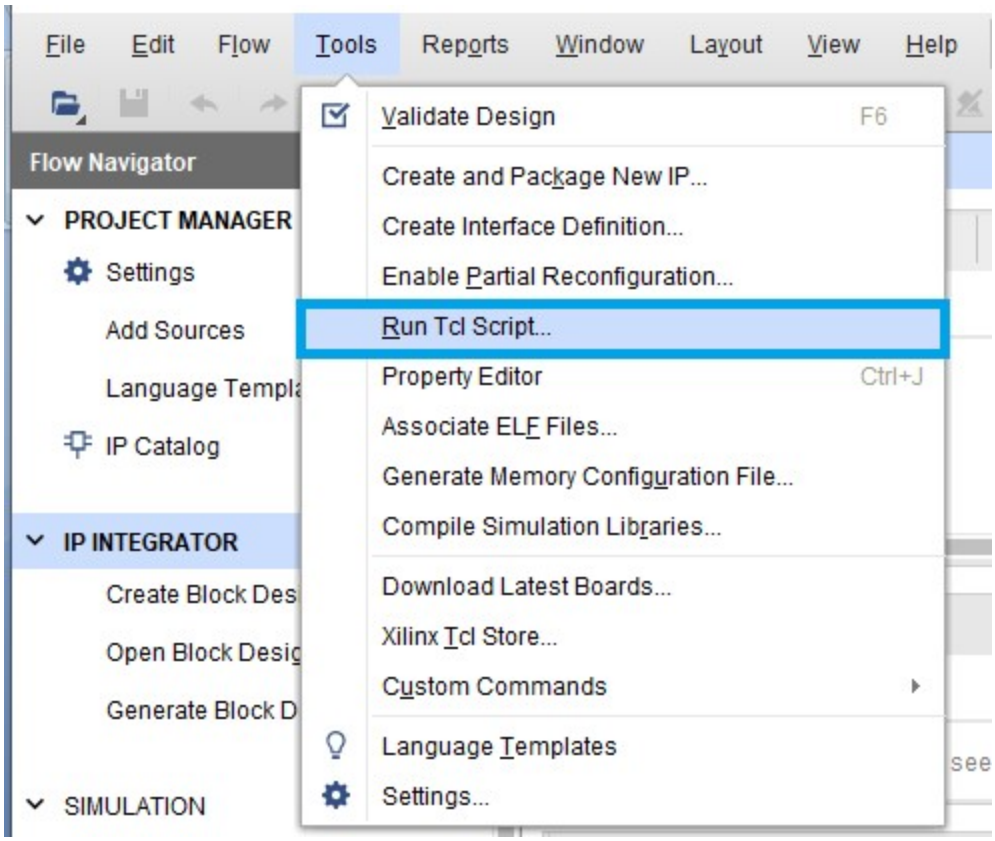
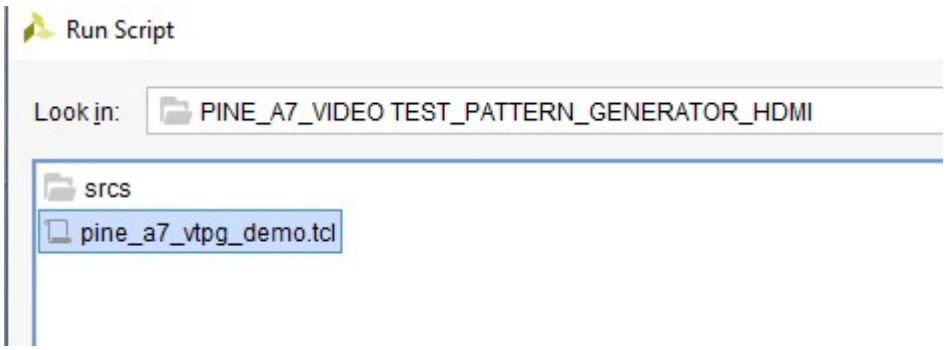
Tutorial on Video Test Pattern Generation on HDMI using Microblaze on Pine A7 board

1. open Vivado 19.1

1.1) Open up Vivado and click on tools then click on **Run Tcl Script** & navigate **pine_a7_vtpg_demo.tcl,** this process will create full project no need to write code or no need to add IP in block design

Note: you may need valid license for VTPG

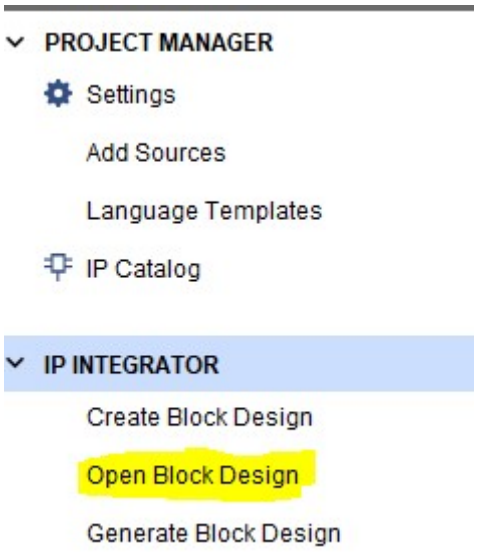




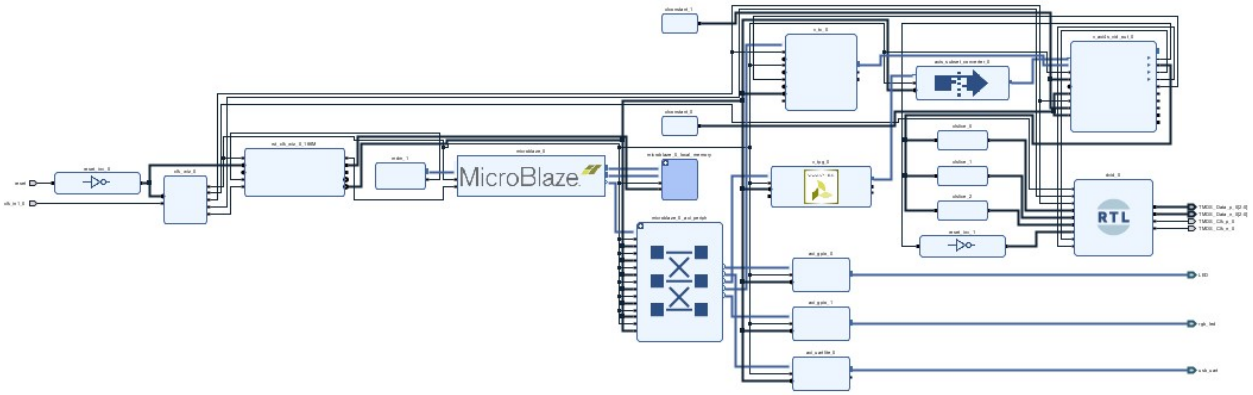
Click on ok

This will generate full project at same location where **pine_a7_vtpg_demo.tcl** file is.

once complete open block design



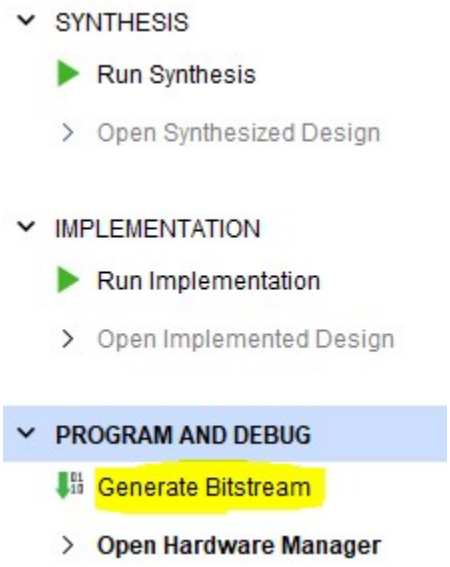
Block design will look like following



This will create a top module in VHDL and will allow you to generate a bitstream.

2. Generating Bit File

2.1) In the top toolbar in Vivado, click  **Generate Bitstream**. This can also be found in the *Flow Navigator* panel on the left, under *Program and Debug*.
If you haven't already saved your design, you will get a prompt to save the block design.



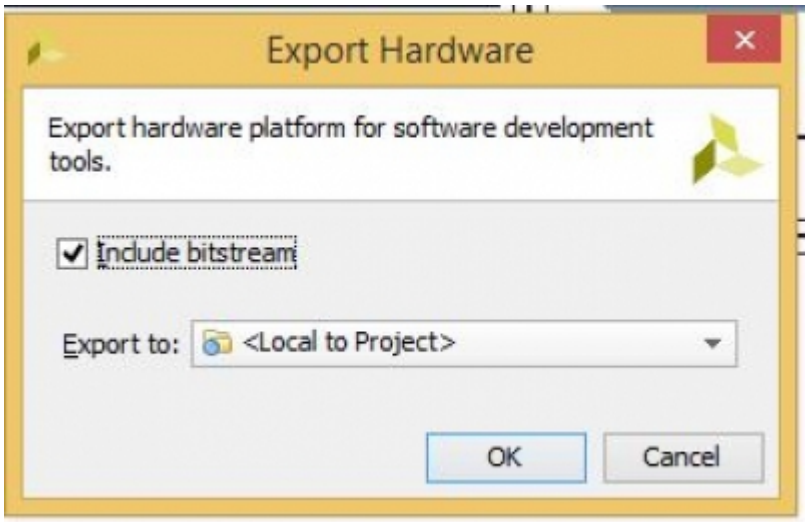
2.2) The bit file generation will begin. The tool will run *Synthesis* and *Implementation*. After both synthesis and implementation have been successfully completed, the bit file will be created. You will find a status bar of Synthesis and Implementation running on the top right corner of the project window.

This process can take anywhere from **2 to 60 minutes** depending on your computer.

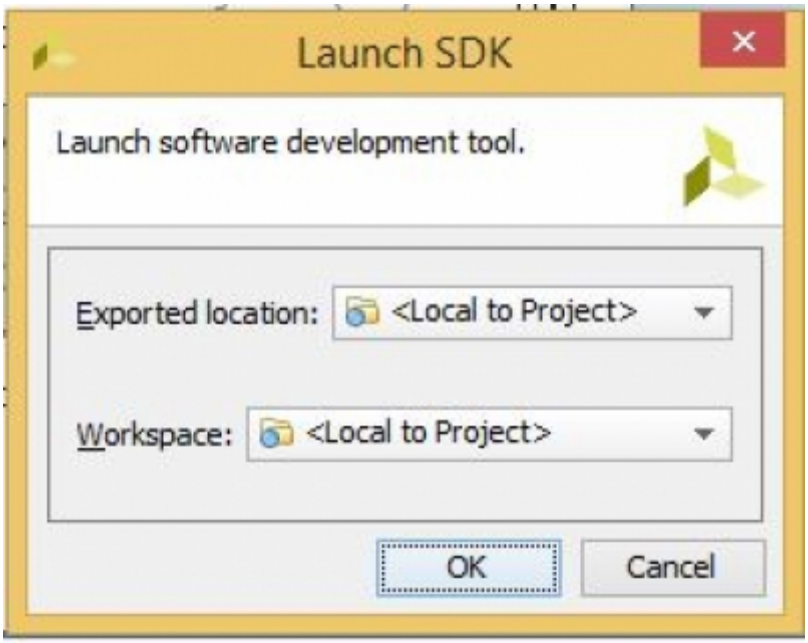
2.3) After the bitstream has been generated, a message prompt will pop-up on the screen. You don't have to open the Implemented Design for this demo. Just click **Cancel**.

3. Exporting Hardware Design to SDK

3.1) On the main toolbar, click **File** and select **Export→Export Hardware**. Check the box to **Include Bitstream** and click **OK**. This will export the hardware design with system wrapper for the Software Development Tool - Vivado SDK.



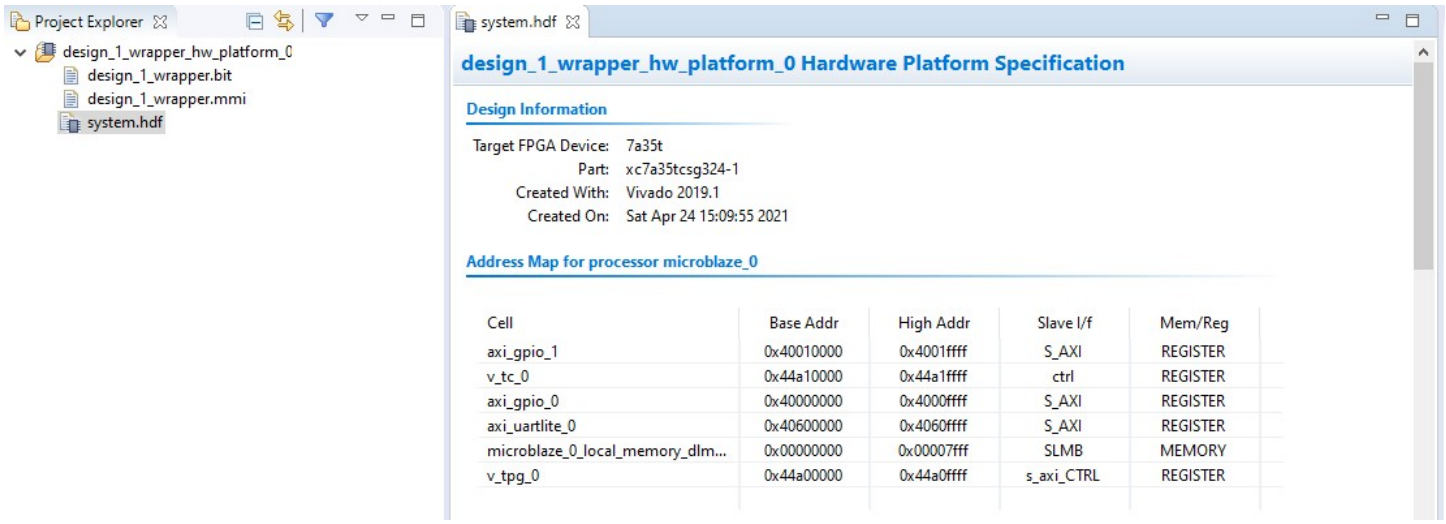
3.2) On the main toolbar, click **File** and then **Launch SDK**. Leave both of the dropdown menus as their default *Local to Project* and click **OK**. This will open Xilinx SDK and import your hardware.



4. Inside Xilinx SDK

4.1) The HW design specification and included IP blocks are displayed in the *system.hdf* file. Xilinx SDK is independent of Vivado, i.e. from this point, you can create your SW project in C/C++ on top of the exported HW design. If necessary, you can also launch SDK directly from the SDK folder created in the main Vivado Project directory.

From this point, if you need to go back to Vivado and make changes to the HW design, then it is recommended to close the SDK window and make the required HW design edits in Vivado. After this you must follow the sequence of creating a new HDL wrapper, save design and bit file generation. This new bit file and system wrapper must then be exported to SDK.

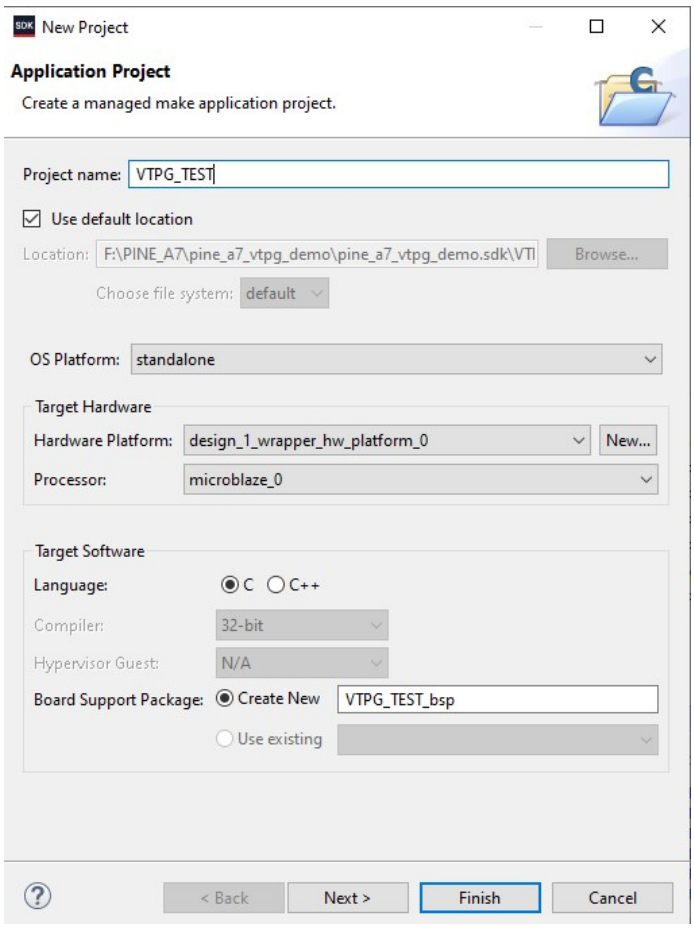


4.2) Within the *Project Explorer* tab on the left, you can see your hardware platform.

system is the name of your block design created in Vivado. This hardware platform has all the HW design definitions, IP interfaces that have been added, external output signal information and local memory address information.

5. Creating New Application Project in SDK

5.1) Click the  **New** dropdown arrow and select **Xilinx→Application Project**.



Give your project a name that has no empty spaces and click **Next**.

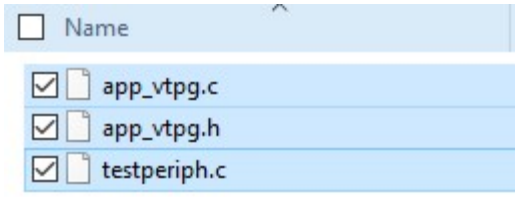
5.2) Select **Peripheral Test** from the list of templates and click **Finish**.

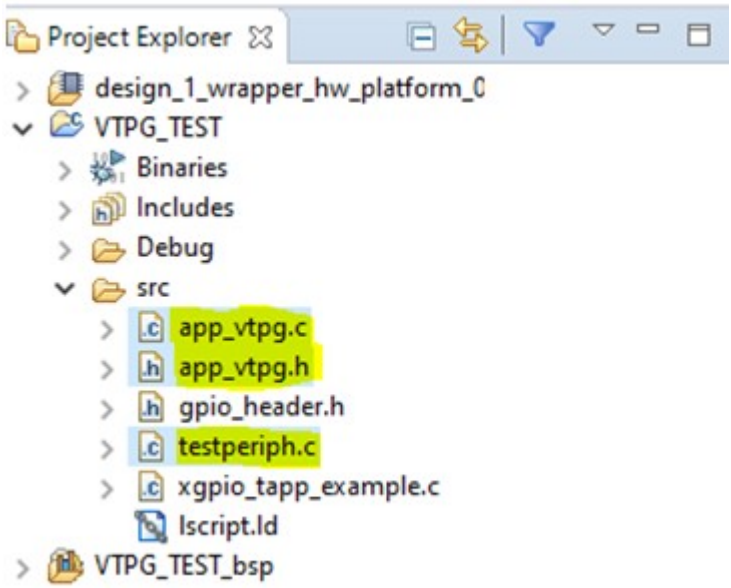
You will see two new folders in the **Project Explorer** panel.

- **VTPG_TEST** which contains all the binaries, .C and .H (Header) files
- **VTPG_TEST_bsp** which is the board support folder

VTPG_TEST is our main working source folder. This also contains an important file shown here which is the "lscript.ld". This is a Xilinx auto generated linker script file. Double click on this file to open.

5.3) Back in the **Project Explorer**, copy all three files shown below which you download from GIT and pest in **src** folder.





This is the testperiph .C file which will control VTPG IP & send which pattern is being displayed

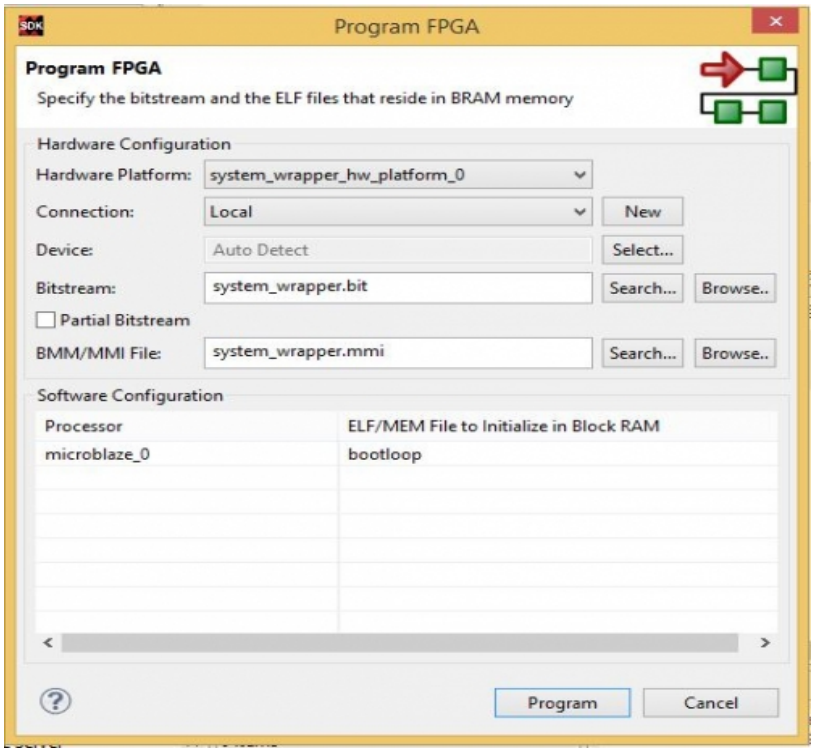
```
app_conf_tpg(&tpg_inst, 600, 800, 0x2, XTPG_BKGND_H_RAMP );
xil_printf("Check HDMI video XTPG_BKGND_H_RAMP \r\n");
MSDelay(del_val);

app_conf_tpg(&tpg_inst, 600, 800, 0x2, XTPG_BKGND_V_RAMP );
xil_printf("Check HDMI video XTPG_BKGND_V_RAMP \r\n");
MSDelay(del_val);

app_conf_tpg(&tpg_inst, 600, 800, 0x2, XTPG_BKGND_TEMPORAL_RAMP );
xil_printf("Check HDMI video XTPG_BKGND_TEMPORAL_RAMP \r\n");
MSDelay(del_val);

app_conf_tpg(&tpg_inst, 600, 800, 0x2, XTPG_BKGND_SOLID_RED );
xil_printf("Check HDMI video XTPG_BKGND_SOLID_RED \r\n");
MSDelay(del_val);

app_conf_tpg(&tpg_inst, 600, 800, 0x2, XTPG_BKGND_SOLID_GREEN );
xil_printf("Check HDMI video XTPG_BKGND_SOLID_GREEN \r\n");
MSDelay(del_val);

app_conf_tpg(&tpg_inst, 600, 800, 0x2, XTPG_BKGND_SOLID_BLUE );
xil_printf("Check HDMI video XTPG_BKGND_SOLID_BLUE \r\n");
MSDelay(del_val);

app_conf_tpg(&tpg_inst, 600, 800, 0x2, XTPG_BKGND_SOLID_BLACK );
xil_printf("Check HDMI video XTPG_BKGND_SOLID_BLACK \r\n");
MSDelay(del_val);

app_conf_tpg(&tpg_inst, 600, 800, 0x2, XTPG_BKGND_SOLID_WHITE );
```

6. Programming FPGA with Bit File

6.1) Make sure that the Arty is turned on and connected to the host PC via the USB-JTAG port - this port will serve dual purpose as the USB-UART connection to the Microblaze.

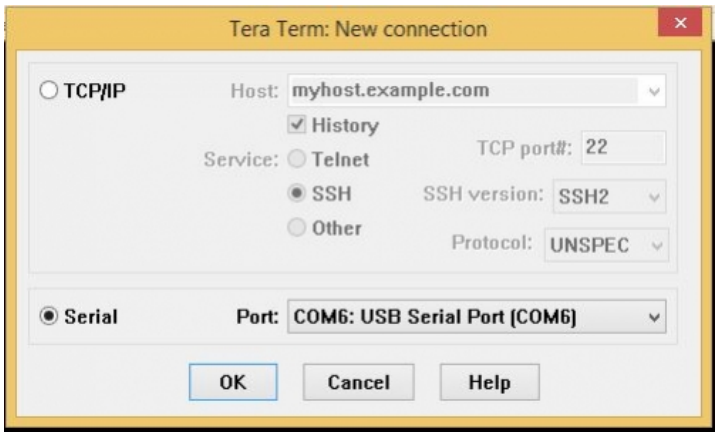
On the top toolbar, click the **Program FPGA** button

6.2) Click **Program** to program your FPGA with your hardware design.



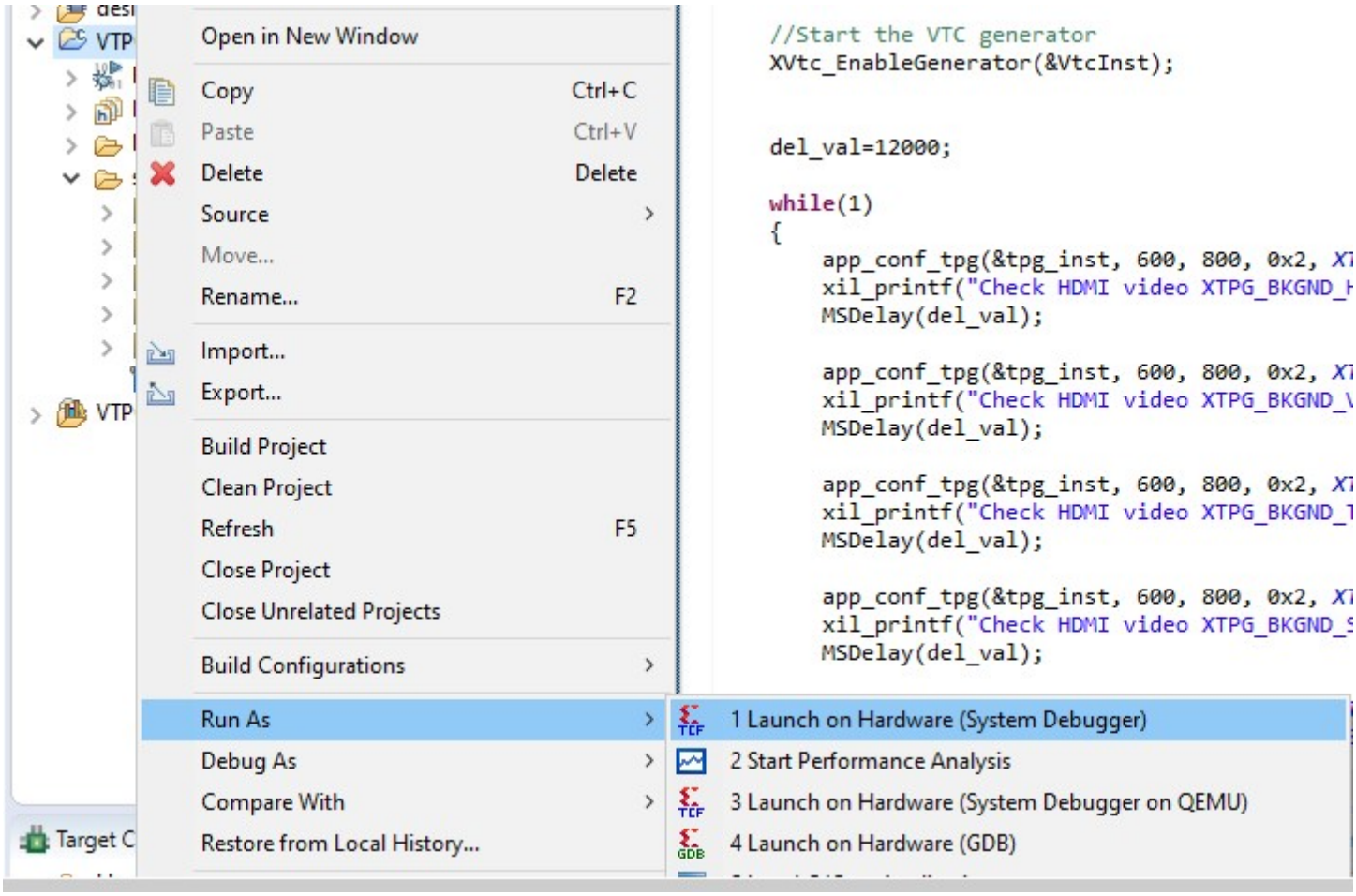
7. Setting up UART Terminal

7.1) Open up a Serial Terminal application (Tera Term). Connect to the Arty UART port with a baud rate of **9600**. This baud rate can be altered in your block design by double clicking the Uartlite block.



8. Program the Microblaze Processor

8.1) Back in SDK, select your **VTPG_TEST** project and click the  **Run As…** button. Select **Launch on Hardware (System Debugger)** and click **OK**.



8.2) Your program will run and you should see following output inside of your Serial Terminal. Hooray!

HDMI will display same pattern as per **VTPG_OUTPUT.mp4** video

```
**************************************************
*          PINE-A7 ARTIX7 BOARD                  *
*       Video Test Pattern Generator + HDMI      *
**************************************************
  TPG application on PINE-A7 using on-board HDMI
********** Visit us www.fpgatechsolution.com *****
HDMI Setup Complete!
Check HDMI VIDEO FOR XTPG_BKGND_TEMPORAL_RAMP
TPG started!
Check HDMI video XTPG_BKGND_H_RAMP
Check HDMI video XTPG_BKGND_V_RAMP
Check HDMI video XTPG_BKGND_TEMPORAL_RAMP
Check HDMI video XTPG_BKGND_SOLID_RED
Check HDMI video XTPG_BKGND_SOLID_GREEN
Check HDMI video XTPG_BKGND_SOLID_BLUE
Check HDMI video XTPG_BKGND_SOLID_BLACK
Check HDMI video XTPG_BKGND_SOLID_WHITE
Check HDMI video XTPG_BKGND_COLOR_BARS
Check HDMI video XTPG_BKGND_ZONE_PLATE
Check HDMI video XTPG_BKGND_TARTAN_COLOR_BARS
Check HDMI video XTPG_BKGND_CROSS_HATCH
Check HDMI video XTPG_BKGND_RAINBOW_COLOR
Check HDMI video XTPG_BKGND_HV_RAMP
Check HDMI video XTPG_BKGND_CHECKER_BOARD
Check HDMI video XTPG_BKGND_PBRS
Check HDMI video XTPG_BKGND_DP_COLOR_RAMP
Check HDMI video XTPG_BKGND_DP_BW_VERTICAL_LINE
Check HDMI video XTPG_BKGND_DP_COLOR_SQUARE
Check HDMI video XTPG_BKGND_LAST
Check HDMI video XTPG_BKGND_H_RAMP
```

Note: you can just see output on UART & HDMI by sampling programming download.bit