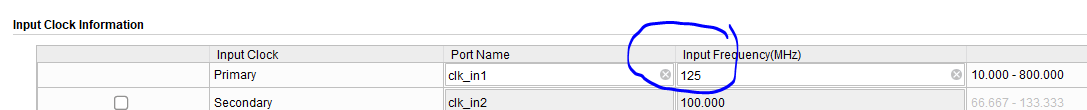
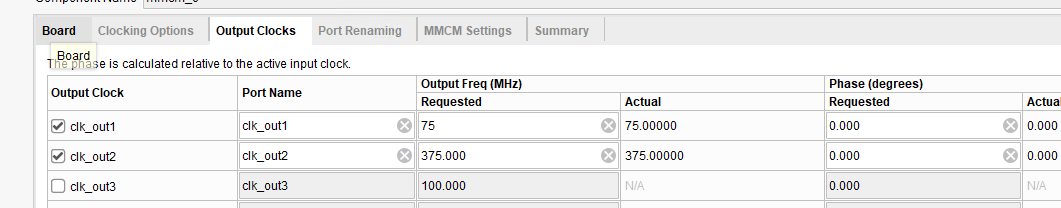
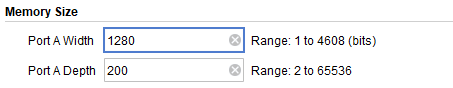
**Steps**:   
  
  
**Please remember that inboard 125 MHz PL clock does not work without connecting an Ethernet cable in Pynq Z2 board. The TUL company made a mistake, as it always unlocks the clock after some time if the Ethernet IC is not active. Therefore, please connect the Ethernet cable to avoid any clocking issues. The Pynq Z1 from Digilent does not have this problem.**  
  
1. Create a new RTL project in Vivado and set the Pynq-Z2 board as the default board. Set the verilog language as target language. If the board is not shown in the list, please download the files from the TUL website and copy them to the vivado boards file location folder. (Note that after 2021.2 and newer versions of vivado they have changed the location of board files folder. Please check the AMD forum if any problem arises).

2. Download all the Verilog source files from the Smart System Lab GitHub folder ([https://github.com/smartsystemslab-uf/SES-Lab-tutorials/tree/master/Xilinx-FPGA tutorials/pong\_pynqz2](https://github.com/smartsystemslab-uf/SES-Lab-tutorials/tree/master/Xilinx-FPGA-tutorials/pong_pynqz2)) and add them to the project. The files are **a) hdmi\_transmit.sv b) object.sv c) paddle.sv d) tmds\_encode.v e) tmds\_oserdes.v and f) top.sv.**

3. From the IP catalog, add a new Clocking Wizard IP and provide the component name as **mmcm\_0**. In the Clocking Options window Tab, select MMCM option for primitive. For the primary input clock, set the frequency to **125** MHz (Board PL clock).



4. From Output Clocks tab select two clock output and request 75 Mhz and 375 Mhz as the clock frequency. Checkout the **locked** checkbox in optional inputs (bottom of the page).   
  
5. From the IP catalog window, add a new Video Timing Controller IP provided by Xilinx. Rename the component name as "**video\_timing**." Uncheck the "Include AXI-4 Lite Interface" and "Enable Detection" options in the first tab. In the Default tab, set the Video Mode as 720p. All other settings should remain the same.

6. Add a block memory generator IP from the IP catalog. Select the interface type to **Native** and Memory Type as **Single Port ROM.** In “**Port A Options” page,** set the memory size as shown in the picture:   
7. From **Other Options** Tab, checkout the Load Init File option and browse the location of the **gameover\_bitmap.coe** (same Github source folder).  
8. Rename the block memory generator IP name as **gameover\_bitmap.** Add the **master.xdc** constraint file from the github folder to the project.   
9.

10. Synthesize the design and generate the bitstream. The game output should be like this:   
