# **REPORT PORTING SIFIVE U500 FREEDOM FROM XILINX’S VC707 TO ALTERA’S TR4**

## Requirements

- Finished tutorial from <https://github.com/thuchoang90/tutorial/tree/VC707>.

- Quartus software (recommend using Quartus Prime 18.0 Standard Edition)

- TR4 FPGA Development Kit

- HSMC Communication Card (for SD Card socket)

- UART-USB module AE-UW232R

## Step by step guide

***Step 1:*** Using **TR4 System builder** to create templated Quartus project with preconfigured settings, constraints for all the needed components from TR4 board.

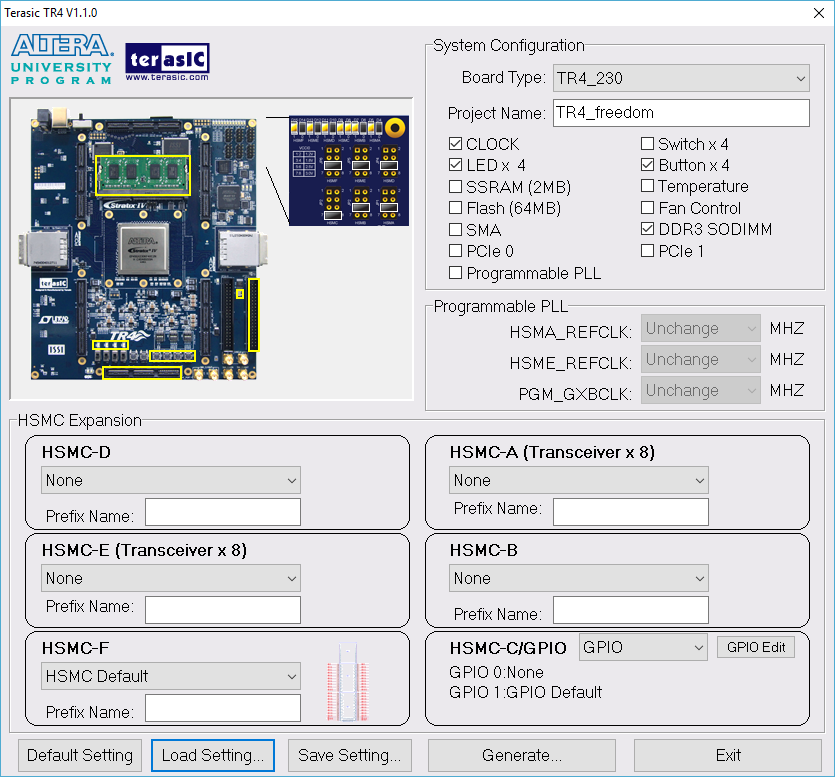


Figure TR4 System Builder setting

Config TR4 System builder as illustrated in Figure 1. Click Generate. Following files will be generated.

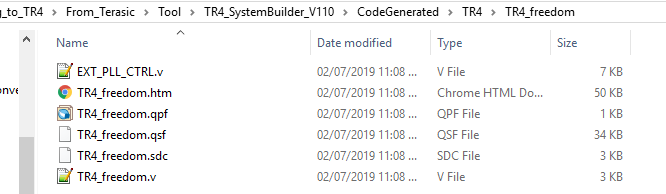


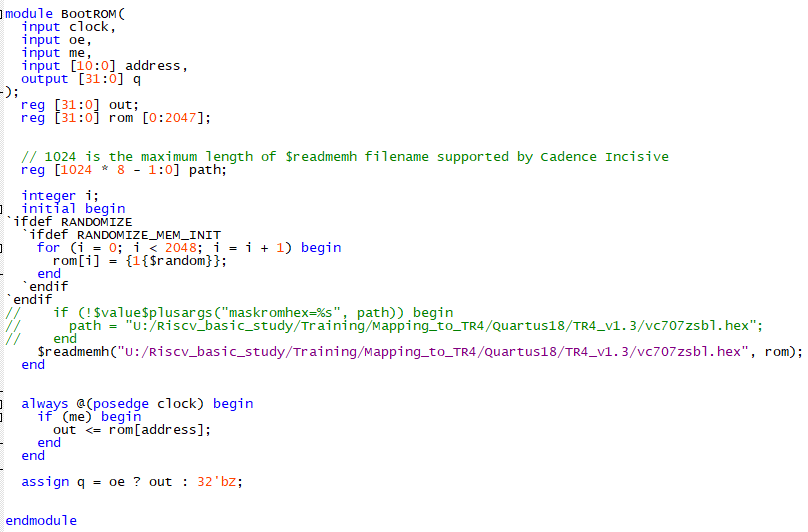
Figure Generated Quartus project

***Step 2:*** Add verilog files which generated in [github.com/thuchoang90/tutorial/tree/VC707](https://github.com/thuchoang90/tutorial/tree/VC707) to Quartus Prime project.

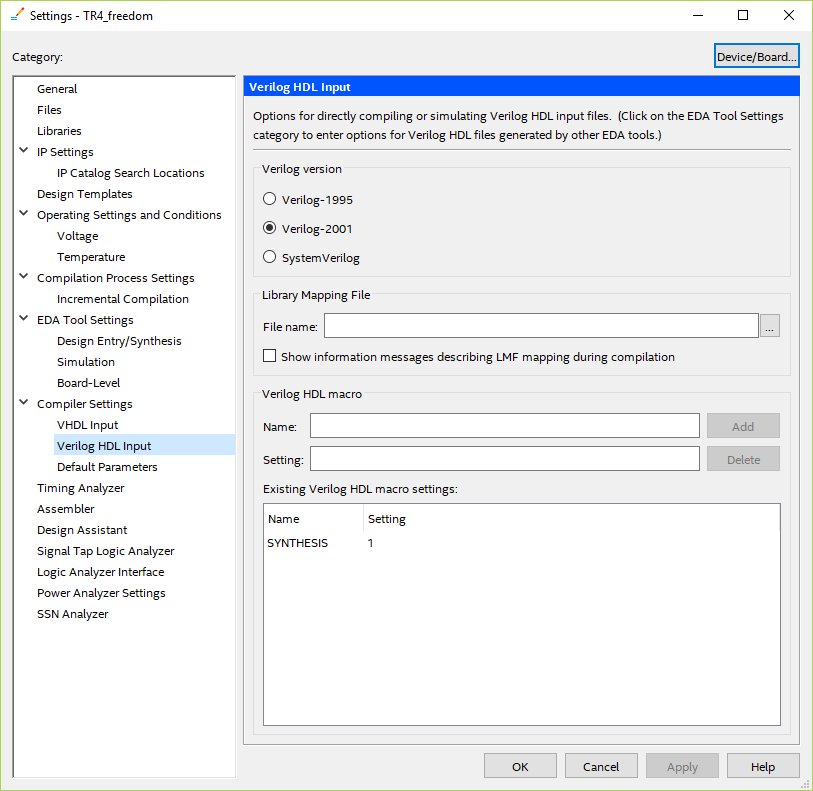
The following files are needed:

|  |  |  |
| --- | --- | --- |
|  | **File name** | **Original Location in freedom folder** |
| 1 | sifive.freedom.unleashed.DevKitU500FPGADesign\_WithDevKit50MHz.v | ../builds/vc707-u500devkit/ |
| 2 | sifive.freedom.unleashed.DevKitU500FPGADesign\_WithDevKit50MHz.rom.v | ../builds/vc707-u500devkit/ |
| 3 | PowerOnResetFPGAOnly.v | ../fpga-shells/Xilinx/common/vsrc |
| 4 | Plusarg\_reader.v | ../rocket-chip/src/main/resources/vsrc |
| 5 | AsyncResetReg.v | ../rocket-chip/src/main/resources/vsrc |
| 6 | vc707zsbl.hex (in case of having Keystone) | ../builds/vc707-u500devkit/ |
| 7 | sdboot.hex (in case of not having Keystone) | ../builds/vc707-u500devkit/ |

Modify module BootROM. (change path that pointed to BootROM hex file)



Define macro SYNTHESIS = 1 (it is used in plusarg\_reader.v)



***Step 3:*** Add Altera’s IP cores.

Three Xilinx IP are needed to be replaced by Altera’s.

1. CorePLL ----> altPLL
2. IBUF, IBUFDS -----> altiobuf (actually we don’t need IBUFDS since it is used to connect with the differential clock input and TR4 only has single clock input)
3. Vc707mig blackbox -----> altera DDR3 controller with UniPHY and AXI4 translator (instantiate using Platform Designer or Qsys and then modify the clock connection)

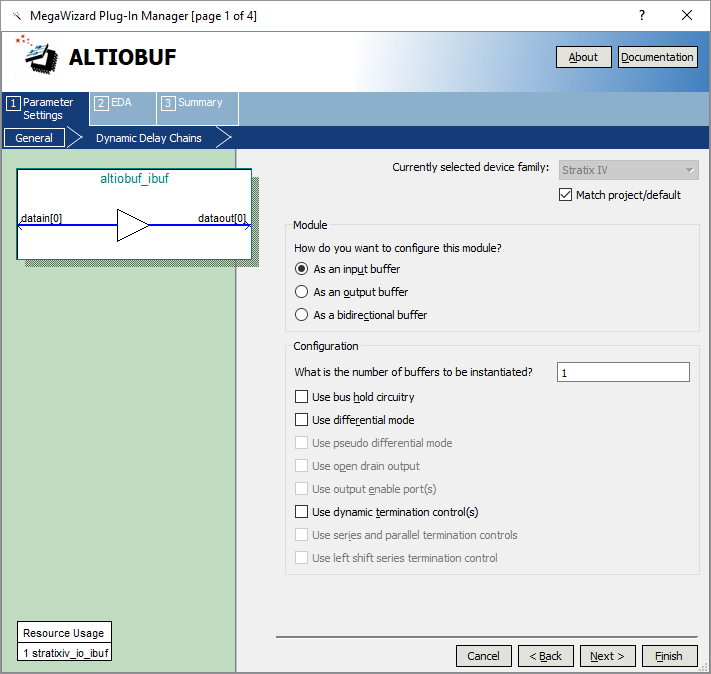


Figure Setting for ALTIOBUF

Create ALTIOBUF IP, config it as an single input buffer and add into Quartus Prime project.

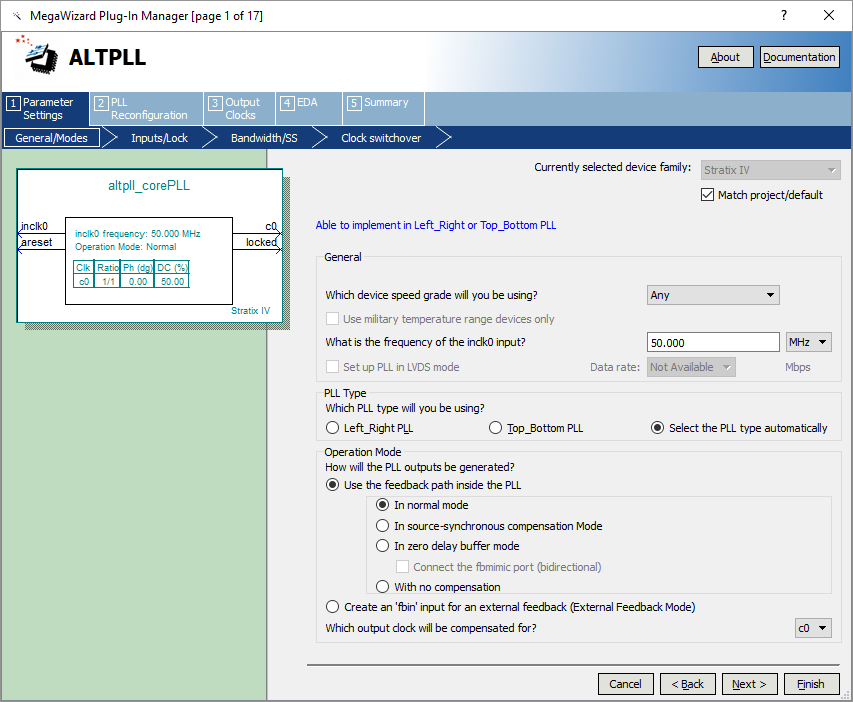
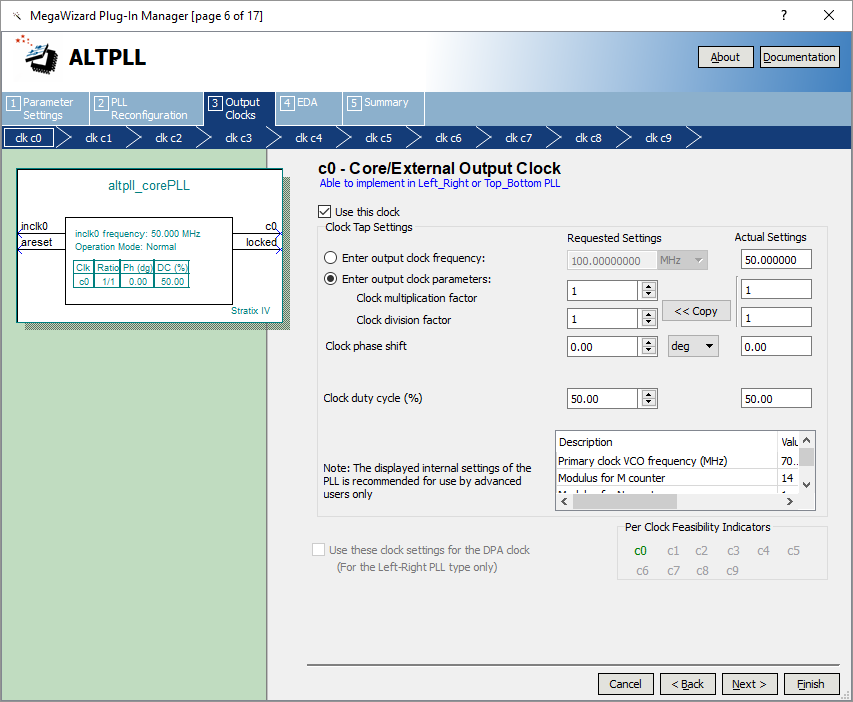
 

Figure Setting for ALTPLL

Create ALTPLL IP, config it to provide 50MHz clk signal at clk\_c0 (since the vc707-u500 are currently using 50MHz clk, if other clk frequency are selected, the ALTPLL should be configured accordingly)

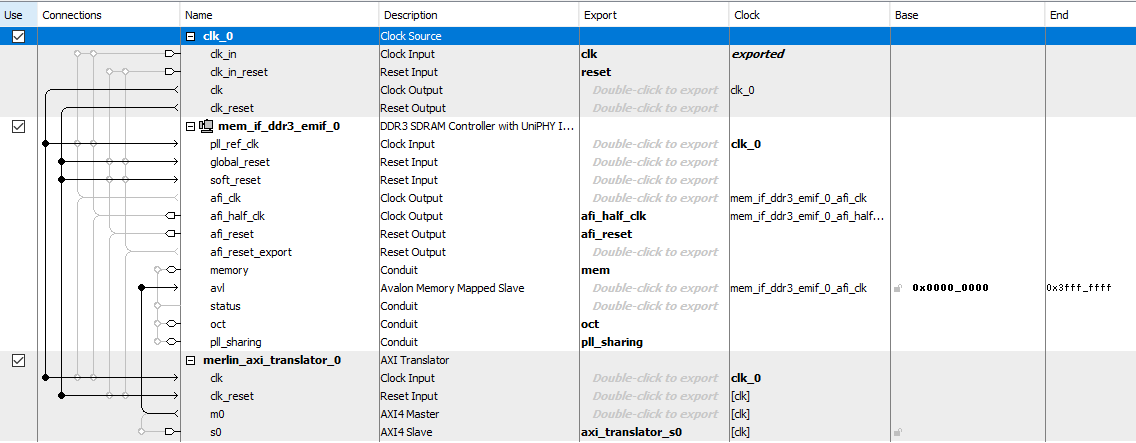


Figure Platform Designer

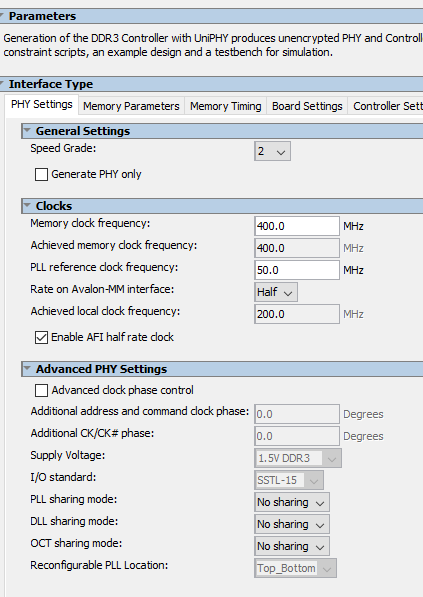
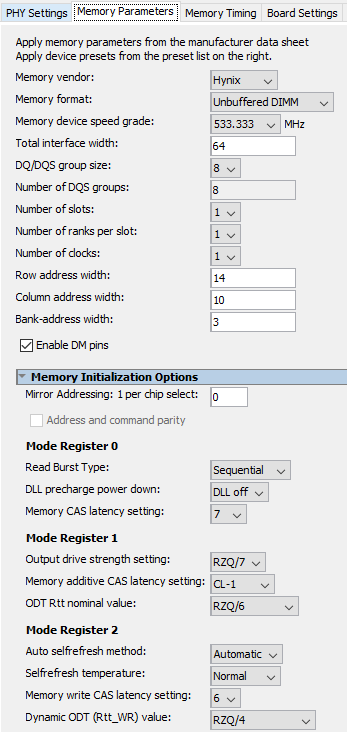
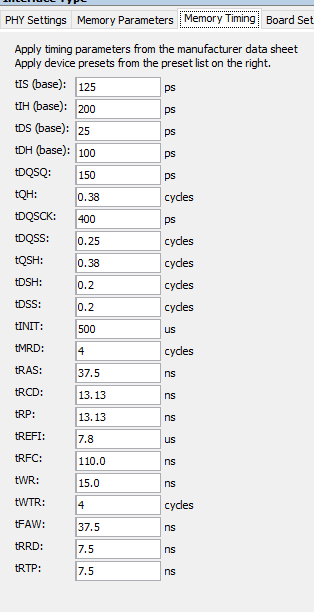
  

Figure DDR3\_EMIF setting

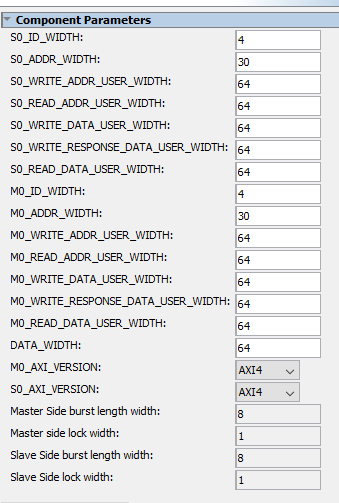
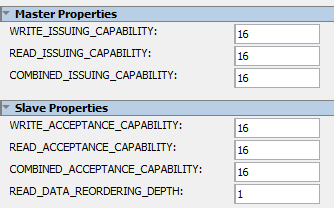
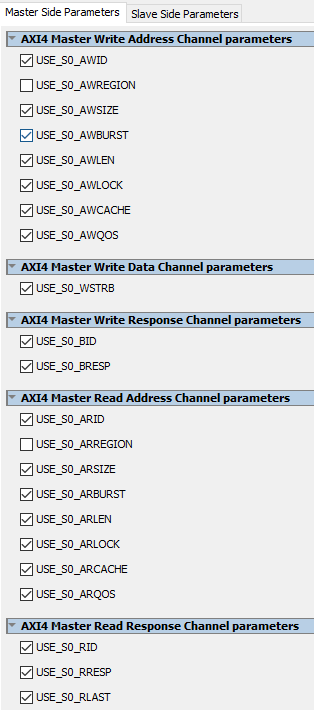
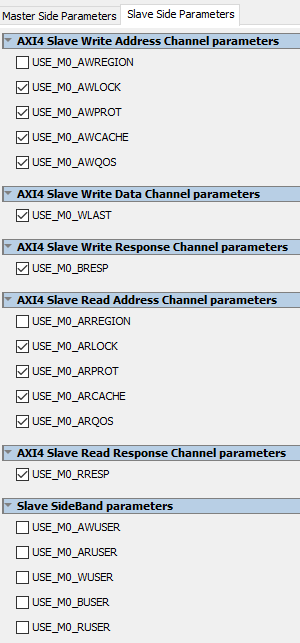
  

Figure AXI4 translator setting

Open Platform Designer, create an system as described in Figure 5 and choose the setting as shown in Figure 6 and Figure 7. Save the system and generate HDL. (generally, for DDR3 Controller IP setting, we can select appropriate option by reading datasheet and using FPGA board’s demo project as references)

Open the top module file of generated Qsys platform (here is the DDR3AXI4.v) and make the modification

Change **.aclk(clk\_clk)** of **merlin\_axi\_translator\_0** and **.clk\_0\_clk\_clk(clk\_clk)** of **mm\_interconnect\_0** into **.aclk(afi\_half\_clk\_clk)** and **.clk\_0\_clk\_clk(afi\_half\_clk\_clk)**.

\*\*\*\* remember to run .tcl generated by Platform designer (after analysis & synthesis) \*\*\*\*

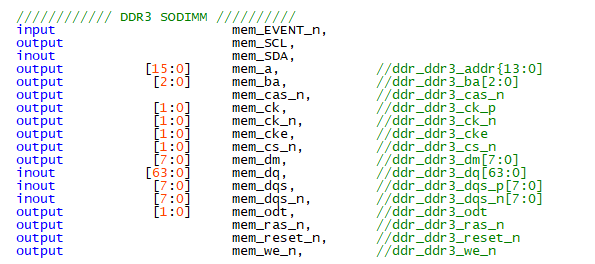
***Step 4*:** Connect RTL design to TR4 Port.

Substitute VC707MIG1GB with DDR3AXI4. Use following table for port connections

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Group** | **Port name in VC707** | **Direction** | **Wire** | **Port name in TR4** | **Description** |
| 1 |  | device\_temp | Output | blackbox\_device\_temp |  | Commented, not use |
| 2 | AXI4 read data Port | s\_axi\_rvalid | Output | blackbox\_s\_axi\_rvalid | axi\_translator\_s0\_rvalid |  |
| 3 | AXI4 read data Port | s\_axi\_rlast | Output | blackbox\_s\_axi\_rlast | axi\_translator\_s0\_rlast |  |
| 4 | AXI4 read data Port | s\_axi\_rresp | Output | blackbox\_s\_axi\_rresp | axi\_translator\_s0\_rresp |  |
| 5 | AXI4 read data Port | s\_axi\_rdata | Output | blackbox\_s\_axi\_rdata | axi\_translator\_s0\_rdata |  |
| 6 | AXI4 read data Port | s\_axi\_rid | Output | blackbox\_s\_axi\_rid | axi\_translator\_s0\_rid |  |
| 7 | AXI4 read data Port | s\_axi\_rready | Input | blackbox\_s\_axi\_rready | axi\_translator\_s0\_rready |  |
| 8 | AXI4 read address Port | s\_axi\_arready | Output | blackbox\_s\_axi\_arready | axi\_translator\_s0\_arready |  |
| 9 | AXI4 read address Port | s\_axi\_arvalid | Input | blackbox\_s\_axi\_arvalid | axi\_translator\_s0\_arvalid |  |
| 10 | AXI4 read address Port | s\_axi\_arqos | Input | blackbox\_s\_axi\_arqos | axi\_translator\_s0\_arqos |  |
| 11 | AXI4 read address Port | s\_axi\_arprot | Input | blackbox\_s\_axi\_arprot | axi\_translator\_s0\_arprot |  |
| 12 | AXI4 read address Port | s\_axi\_arcache | Input | blackbox\_s\_axi\_arcache | axi\_translator\_s0\_arcache |  |
| 13 | AXI4 read address Port | s\_axi\_arlock | Input | blackbox\_s\_axi\_arlock | axi\_translator\_s0\_arlock |  |
| 14 | AXI4 read address Port | s\_axi\_arburst | Input | blackbox\_s\_axi\_arburst | axi\_translator\_s0\_arburst |  |
| 15 | AXI4 read address Port | s\_axi\_arsize | Input | blackbox\_s\_axi\_arsize | axi\_translator\_s0\_arsize |  |
| 16 | AXI4 read address Port | s\_axi\_arlen | Input | blackbox\_s\_axi\_arlen | axi\_translator\_s0\_arlen |  |
| 17 | AXI4 read address Port | s\_axi\_araddr | Input | blackbox\_s\_axi\_araddr | axi\_translator\_s0\_araddr |  |
| 18 | AXI4 read address Port | s\_axi\_arid | Input | blackbox\_s\_axi\_arid | axi\_translator\_s0\_arid |  |
| 19 | AXI4 write response Port | s\_axi\_bvalid | Output | blackbox\_s\_axi\_bvalid | axi\_translator\_s0\_bvalid |  |
| 20 | AXI4 write response Port | s\_axi\_bresp | Output | blackbox\_s\_axi\_bresp | axi\_translator\_s0\_bresp |  |
| 21 | AXI4 write response Port | s\_axi\_bid | Output | blackbox\_s\_axi\_bid | axi\_translator\_s0\_bid |  |
| 22 | AXI4 write response Port | s\_axi\_bready | Input | blackbox\_s\_axi\_bready | axi\_translator\_s0\_bready |  |
| 23 | AXI4 write data Port | s\_axi\_wready | Output | blackbox\_s\_axi\_wready | axi\_translator\_s0\_wready |  |
| 24 | AXI4 write data Port | s\_axi\_wvalid | Input | blackbox\_s\_axi\_wvalid | axi\_translator\_s0\_wvalid |  |
| 25 | AXI4 write data Port | s\_axi\_wlast | Input | blackbox\_s\_axi\_wlast | axi\_translator\_s0\_wlast |  |
| 26 | AXI4 write data Port | s\_axi\_wstrb | Input | blackbox\_s\_axi\_wstrb | axi\_translator\_s0\_wstrb |  |
| 27 | AXI4 write data Port | s\_axi\_wdata | Input | blackbox\_s\_axi\_wdata | axi\_translator\_s0\_wdata |  |
| 28 | AXI4 write address Port | s\_axi\_awready | Output | blackbox\_s\_axi\_awready | axi\_translator\_s0\_awready |  |
| 29 | AXI4 write address Port | s\_axi\_awvalid | Input | blackbox\_s\_axi\_awvalid | axi\_translator\_s0\_awvalid |  |
| 30 | AXI4 write address Port | s\_axi\_awqos | Input | blackbox\_s\_axi\_awqos | axi\_translator\_s0\_awqos |  |
| 31 | AXI4 write address Port | s\_axi\_awprot | Input | blackbox\_s\_axi\_awprot | axi\_translator\_s0\_awprot |  |
| 32 | AXI4 write address Port | s\_axi\_awcache | Input | blackbox\_s\_axi\_awcache | axi\_translator\_s0\_awcache |  |
| 33 | AXI4 write address Port | s\_axi\_awlock | Input | blackbox\_s\_axi\_awlock | axi\_translator\_s0\_awlock |  |
| 34 | AXI4 write address Port | s\_axi\_awburst | Input | blackbox\_s\_axi\_awburst | axi\_translator\_s0\_awburst |  |
| 35 | AXI4 write address Port | s\_axi\_awsize | Input | blackbox\_s\_axi\_awsize | axi\_translator\_s0\_awsize |  |
| 36 | AXI4 write address Port | s\_axi\_awlen | Input | blackbox\_s\_axi\_awlen | axi\_translator\_s0\_awlen |  |
| 37 | AXI4 write address Port | s\_axi\_awaddr | Input | blackbox\_s\_axi\_awaddr | axi\_translator\_s0\_awaddr |  |
| 38 | AXI4 write address Port | s\_axi\_awid | Input | blackbox\_s\_axi\_awid | axi\_translator\_s0\_awid |  |
| 39 | Application Interface Port | app\_zq\_ack | Output | blackbox\_app\_zq\_ack |  | NC |
| 40 | Application Interface Port | app\_ref\_ack | Output | blackbox\_app\_ref\_ack |  | NC |
| 41 | Application Interface Port | app\_sr\_active | Output | blackbox\_app\_sr\_active |  | NC |
| 42 | Application Interface Port | app\_zq\_req | Input | blackbox\_app\_zq\_req |  | Const 0 |
| 43 | Application Interface Port | app\_ref\_req | Input | blackbox\_app\_ref\_req |  | Const 0 |
| 44 | Application Interface Port | app\_sr\_req | Input | blackbox\_app\_sr\_req |  | Const 0 |
| 45 | System Input | sys\_rst | Input | blackbox\_sys\_rst | reset\_reset\_n | **Inverted**  (assign blackbox\_sys\_rst = ~ io\_port\_sys\_rst) |
| 46 | Memory Interface Ports | init\_calib\_complete | Output | blackbox\_init\_calib\_complete |  | Commented |
| 47 | Application Interface Port | aresetn | Input | blackbox\_aresetn |  | Commented |
| 48 | Application Interface Port | mmcm\_locked | Output | blackbox\_mmcm\_locked | pll\_sharing\_pll\_locked |  |
| 49 | Application Interface Port | ui\_clk\_sync\_rst | Output | blackbox\_ui\_clk\_sync\_rst | afi\_reset\_reset\_n | **Inverted**  (assign io\_port\_ui\_sync\_rs t= ~blackbox\_ui\_clk\_sync\_rst) |
| 50 | Application Interface Port | ui\_clk | Output | blackbox\_ui\_clk | afi\_half\_clk\_clk | (for SocKit, use afi\_clk\_clk instead) |
| 51 | System Input | sys\_clk\_i | Input | blackbox\_sys\_clk\_i | clk\_clk |  |
| 52 | Memory Interface Ports | ddr3\_dqs\_p | In-Out | io\_port\_ddr3\_dqs\_p | mem\_mem\_dqs |  |
| 53 | Memory Interface Ports | ddr3\_dqs\_n | In-Out | io\_port\_ddr3\_dqs\_n | mem\_mem\_dqs\_n |  |
| 54 | Memory Interface Ports | ddr3\_dq | In-Out | io\_port\_ddr3\_dq | mem\_mem\_dq |  |
| 55 | Memory Interface Ports | ddr3\_odt | Output | blackbox\_ddr3\_odt | mem\_mem\_odt |  |
| 56 | Memory Interface Ports | ddr3\_dm | Output | blackbox\_ddr3\_dm | mem\_mem\_dm |  |
| 57 | Memory Interface Ports | ddr3\_cs\_n | Output | blackbox\_ddr3\_cs\_n | mem\_mem\_cs\_n |  |
| 58 | Memory Interface Ports | ddr3\_cke | Output | blackbox\_ddr3\_cke | mem\_mem\_cke |  |
| 59 | Memory Interface Ports | ddr3\_ck\_n | Output | blackbox\_ddr3\_ck\_n | mem\_mem\_ck\_n |  |
| 60 | Memory Interface Ports | ddr3\_ck\_p | Output | blackbox\_ddr3\_ck\_p | mem\_mem\_ck |  |
| 61 | Memory Interface Ports | ddr3\_reset\_n | Output | blackbox\_ddr3\_reset\_n | mem\_mem\_reset\_n |  |
| 62 | Memory Interface Ports | ddr3\_we\_n | Output | blackbox\_ddr3\_we\_n | mem\_mem\_we\_n |  |
| 63 | Memory Interface Ports | ddr3\_cas\_n | Output | blackbox\_ddr3\_cas\_n | mem\_mem\_cas\_n |  |
| 64 | Memory Interface Ports | ddr3\_ras\_n | Output | blackbox\_ddr3\_ras\_n | mem\_mem\_ras\_n |  |
| 65 | Memory Interface Ports | ddr3\_ba | Output | blackbox\_ddr3\_ba | mem\_mem\_ba |  |
| 66 | Memory Interface Ports | ddr3\_addr | Output | blackbox\_ddr3\_addr | mem\_mem\_a |  |
|  |  |  | Input |  | oct\_rdn | **Pull up to top module** |
|  |  |  | Input |  | oct\_rup | **Pull up to top module** |
|  |  |  | Output |  | pll\_sharing\_pll\_mem\_clk | NC |
|  |  |  | Output |  | pll\_sharing\_pll\_write\_clk | NC |
|  |  |  | Output |  | pll\_sharing\_pll\_write\_clk\_pre\_phy\_clk | NC |
|  |  |  | Output |  | pll\_sharing\_pll\_addr\_cmd\_clk | NC |
|  |  |  | Output |  | pll\_sharing\_pll\_avl\_clk | NC |
|  |  |  | Output |  | pll\_sharing\_pll\_config\_clk | NC |

***Step 5:*** integrate the top module to Quartus project (VC707Shell ---> TR4\_freedom)

1. Replace “ddr\_ddr3\_\*” ports by “mem\_\*” ports



1. Replace uart by GPIO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RTL Ports** | **TR4 FPGA Pin** | **AE-UM232R module** | **TR4 GPIO pin (JP10)** | **HSMC Comm Card Pin** |
| Txd | AD29 | RxD | 40 | PIO\_0\_OUT0 (2) |
| Rxd | AE29 | TxD | 36 | PIO\_0\_IN (1) |
| Rtsn | AE31 | CTS# | 32 | PIO\_0\_OUT2 |
| Ctsn | AE30 | RTS# | 28 | PIO\_0\_OUT1 |
| Gnd |  | GND | 30 | GND |

1. Replace SDIO by HSM\_\*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Vivado ports** | **Xilinx FPGA Pin** | **SD Card Pin** | **HSM Comm Card** | **Altera Pin** | **Quartus Port** |
| sdio\_clk | AN30 | 5 | 95 | AN21 | HSM\_CLKOUT\_p1 |
| sdio\_cmd | AP30 | 2 | 144 | AE20 | HSM\_RX\_p15 |
| sdio\_dat\_0 | AR30 | 7 | 140 | AG22 | HSM\_RX\_n14 |
| sdio\_dat\_1 | AU31 | 8 | 125 | AF23 | HSM\_TX\_p12 |
| sdio\_dat\_2 | AV31 | 9 | 120 | AN18 | HSM\_RX\_p11 |
| sdio\_dat\_3 | AT30 | 1 | 108 | AU17 | HSM\_RX\_p9 |

1. Replace sys\_clock\_p, sys\_clock\_n by OSC\_50\_BANK
2. Reset Port replaced by ~BUTTON[0] (remember to invert BUTTON[0] since Altera use active low reset while Xilinx use active high reset)

***Step 6:*** Full compile then program TR4 board. Done!