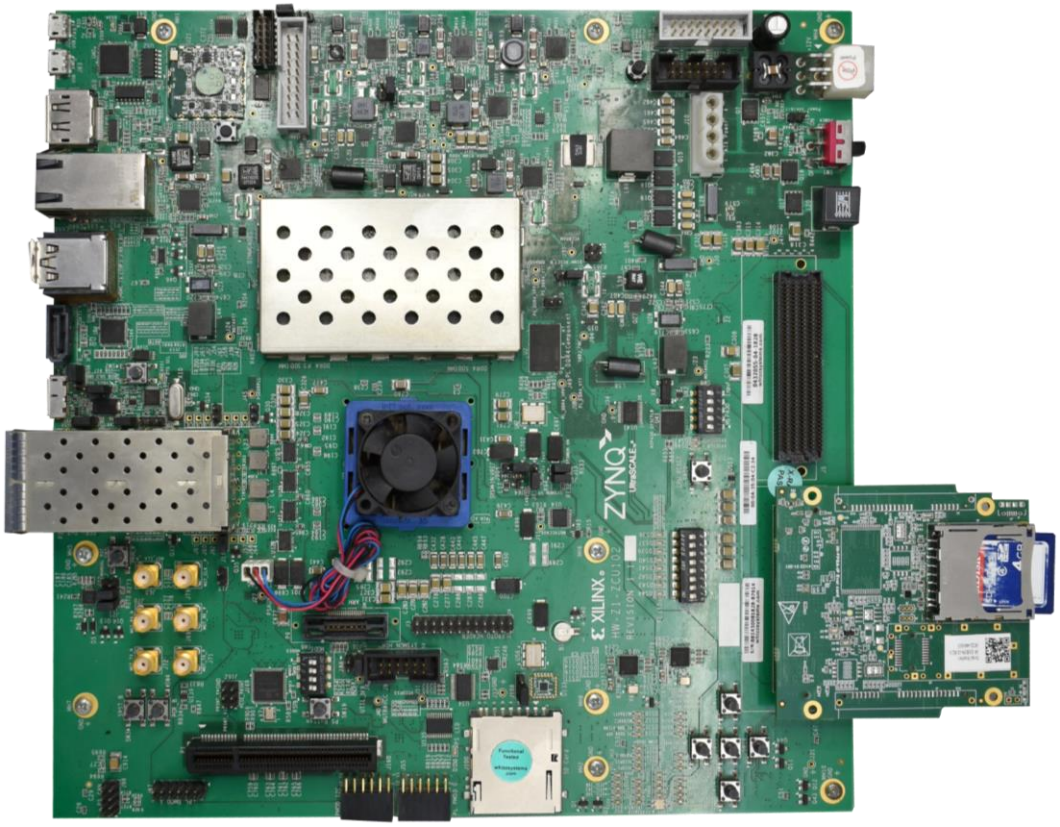
SD/SDIO Host Controller 3.0 IP Integration Manual



iWave

Table of Content

1 INTRODUCTION 5
1.1 PURPOSE 5
1.2 REFERENCE DOCUMENT 5
1.3 OVERVIEW 5
1.4 ACRONYMS AND ABBREVIATIONS 5
2 IP CONFIGURATION AND INSTANTIATION 6
2.1 EXAMPLE DESIGN 6
2.2 SD HOST CCONTROLLER IP CONFIGURATION 6
2.3 STEPS TO ADD SD HOST IP TO THE BLOCK DESIGN 7
3 IMPLEMENTATION DETAILS 10
3.1 CLOCK DOMAIN 10
3.2 CONSTRAINTS 10
4 DESIGN MODIFICATION TO BE DONE FOR CUSTOM BOARD 11
5 RESOURCE UTILIZATION 12

List Of Figures

Figure 1:Design sources where SD host netlist is instantiated 6
Figure 2:Adding SD_Host_controller IP to the IP repository 7
Figure 3:Click on Open Block design 8
Figure 4:SD Host Controller added to the block design 8
Figure 5:Instantiation of block design IO signals in the top module 9
Figure 6:Pin Constraints in example design .xdc file 10

List Of Tables

Table 1: Acronyms & Abbreviations 5
Table 2: Resource Utilization for device for ZCU102 dev. kit 12

1 Introduction

1.1 Purpose

The purpose of this document is to describe SD / SDIO Host Controller 3.0. IP Integration details.

1.2 Reference Document

- iW- EMFGA-DS-01-R1.0-REL1.0

1.3 Overview

SD/SDIO Host Controller interfaces the Zynq Ultrascale +MpSoC Processor through AXI4 bus enabling the data transfers between each other. The Zynq Processor will send the response to the SD host depending on the command issued.

1.4 Acronyms and Abbreviations

Table 1: Acronyms & Abbreviations

Term	Meaning
FPGA	Field Programmable Gate Array
GPIO	General purpose input output
FMC	FPGA Mezzanine Card
LUT	Look Up Table
IO	Input and Output
FF	Flip Flop

2 IP Configuration and Instantiation

2.1 Example design

The SD/SDIO host controller IP example design mainly consists of

1. **Zynq UltraScale + MpSoC processor:** Zynq Processor is used to configure SD host controller IP mainly access to register set and to read/write to DDR memory
2. **SD/SDIO Host Controller IP:** SD host controller IP takes the command given from the Processor to the register set and with the ADMA the read and write operation will happen from and to the SD Card.

2.2 SD Host Ccontroller IP Configuration

The register set, ADMA configuration , tuning block , command and data path files are pre synthesized. After synthesis ,the post synthesis file i.e., mmc_host.edn file is instantiated in the design as shown in the Figure below . The SD Host controller with other files like the mmc_host.edn file, mmc_host stub file, axi4_lite interface and the AXI memory mapped interface wrapper files are added which makes sd_host_controller IP.

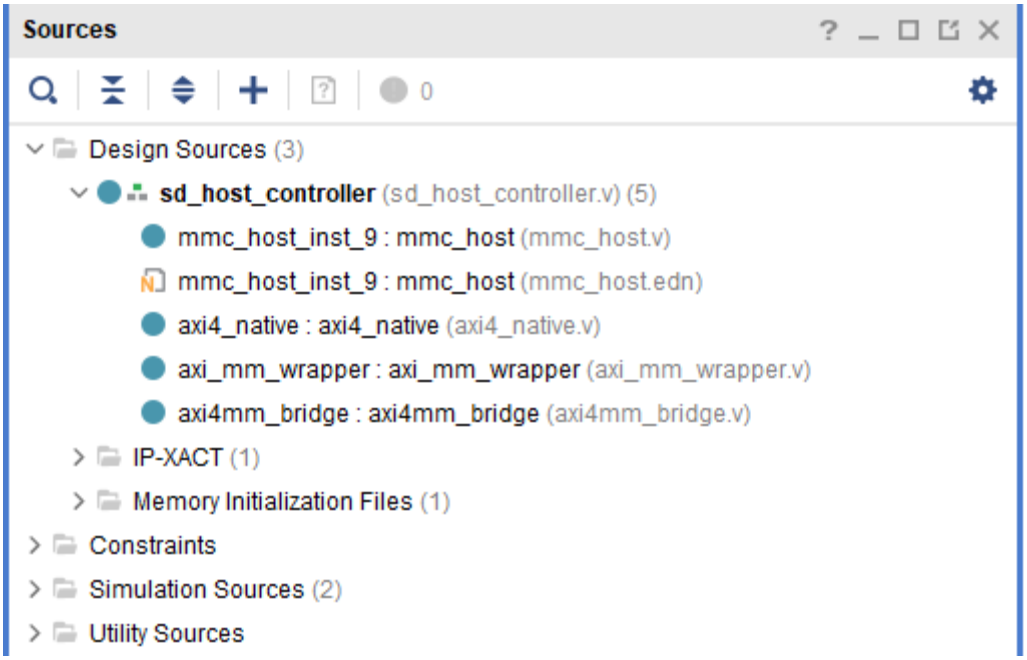


Figure 1:Design sources where SD host netlist is instantiated

Module sd_host_controller is the top module of the project. The sd_host_controller IP needs to be added in the block design of the Vivado project 2021.2.

2.3 Steps to add SD HOST IP to the Block design

- Install the required Vivado Design Suite 2021.2 for the host PC adding the license path. Downloads (xilinx.com)
- Copy the project from "….\EMFGA_Release1.0_SD_3.0_Host\iW-EMFGA-PF-01-R1.0-REL1.0\iW-EMFGA-FF-01-R1.0-REL1.0\iW-EMFGA-ED-01-R1.0-REL1.0\ iW_EMFGA_REL_1.0\project_1" to your project directory in the host PC.
- Open the Xilinx Vivado tool 2021.2 and click on "Open Project" and select the required project from the folder by clicking ok.
- Now the project selected will be opened in Vivado Design Suite 2021.2.
- Go to settings of Vivado 2021.2 and add the IP to the Project Settings → IP → Repository→then add the IP path .

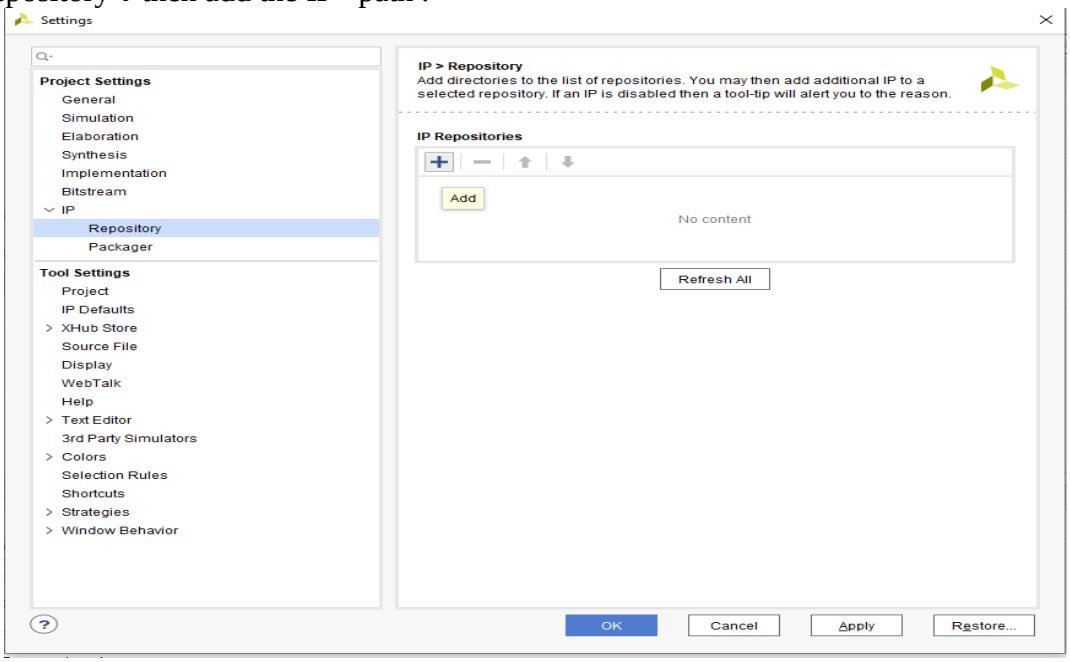


Figure 2:Adding SD_Host_controller IP to the IP repository

IP location is ..\EMFGA_Release1.0_SD_3.0_Host\iW-EMFGA-PF-01-R1.0-REL1.0\iW-EMFGA-FF-01-R1.0-REL1.0\iW-EMFGA-ED-01-R1.0-REL1.0\ iW_EMFGA_REL_1.0\ project_1.srcs \sources_1 \IP .Add IP by clicking on the + sign as shown in Figure above. After the IP is added click on Apply and close the dialogue box

Once the IP is added in the project repository open the block design by clicking on mpsoc_preset under top_module → mpsoc_preset_wrapper present in the source tab.

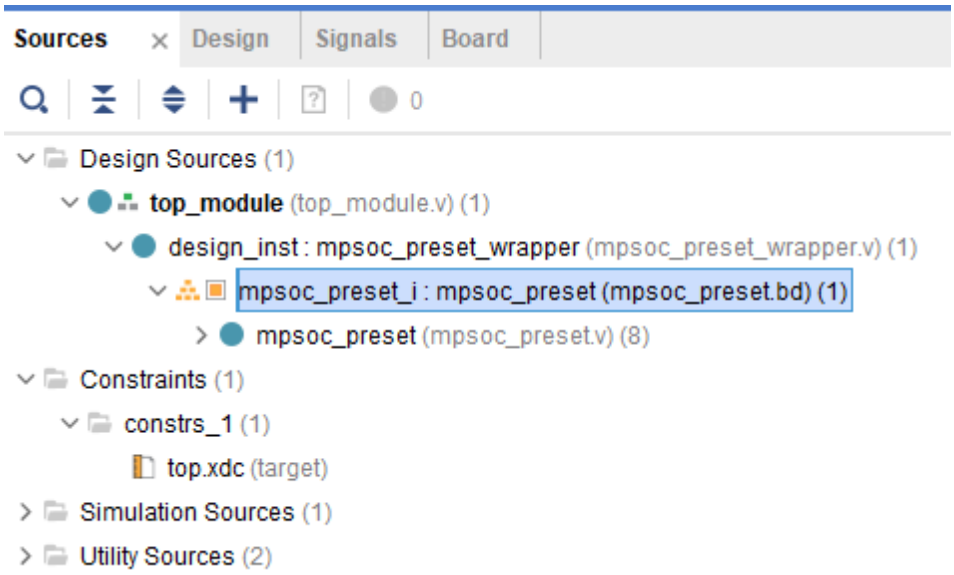


Figure 3:Click on Open Block design

1. After adding the other IPs like Zynq Ultrascale + MpSoC ,UART ,MIG Controller and Clocking Wizard, add the SD host controller IP to the block design by clicking on the + sign as shown in the Figure below

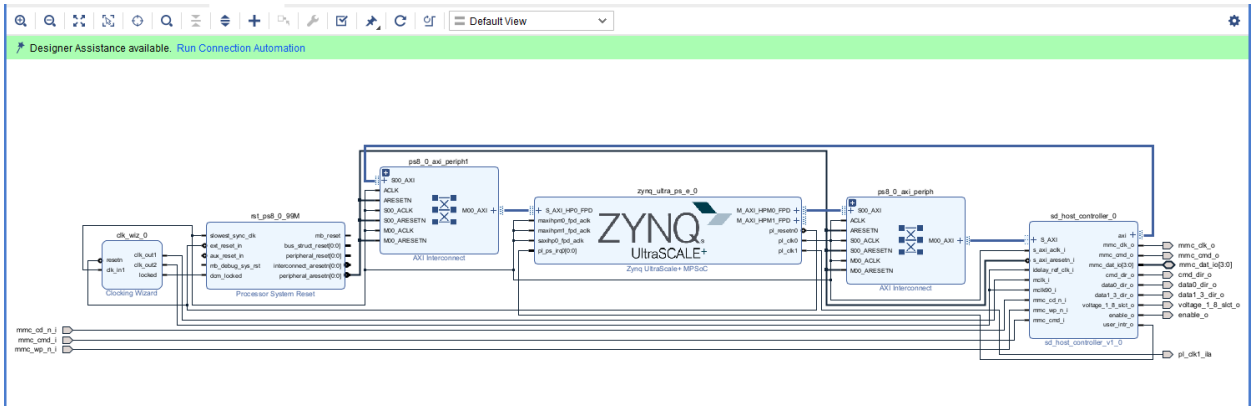


Figure 4:SD Host Controller added to the block design

2. Synthesize the block design first by validating the design(F6) and generate the output products by right clicking on mpsoc_preset under top_module → mpsoc _preset_wrapper present in the source tab and create HDL Wrapper
3. Instantiate mpsoc_preset_wrapper in the top module with the ports and signals of block design, in the top_module.v file as shown in the Figure below

```
mpsoc_preset_wrapper design_inst (
    .cmd_dir_o              ( cmd_dir_o           ) ,
    .mmc_cmd_i              ( cmd_in              ) ,
    .mmc_cmd_o              ( cmd_out             ) ,
    .data0_dir_o            ( data0_dir_o         ) ,
    .data1_3_dir_o          ( data1_3_dir_o       ) ,
    .pl_clk1_ila            ( pl_clk1_ila         ) ,
    .enable_o               ( enable_o            ) ,
    .mmc_cd_n_i             ( mmc_cd_n_i          ) ,
    .mmc_clk_o              ( mmc_clk_o           ) ,
    .mmc_dat_io             ( mmc_dat_io          ) ,
    .mmc_wp_n_i             ( mmc_wp_n_i          ) ,
    .voltage_1_8_slct_o     ( voltage_1_8_slct_o  )
);
```

Figure 5:Instantiation of block design IO signals in the top module

4. The Constraint file (.xdc) provided in the design is for Xilinx ZCU102 development Board and should be changed for custom boards
5. Give the required clock, Pin/IO constraints for SD host controller in the .xdc file and compile the custom design with SD host IP.

3 Implementation Details

3.1 Clock Domain

SD Host Controller 3.0 IP works on the SDR104 mode and the base clock mclk_i drives the output clock to the SD card. User may need to adjust this clock between 104 MHz to 208 MHz based on the hardware setup. In other words, if user gets CRC error with say 200 MHz, then user can reduce the clock to say 190 MHz or below than to make sure write and read works fine with that base clock.

mclk90_i is a phase shifted version of the mclk_i which is 200MHz and this clock is phase shifted based on the response from the SD card. Phase shift of 45 degree is added.
Idelay_ref_clk_i is the fixed 200 MHz clock which will be provided for the IDELAY primitive to be used for the tuning implementation block.
s_axi_clk_i which is 100 MHz which will be used for the communication between the processor and IP using the AXI4-lite interface for register access and AXI4-MM for DMA access.

3.2 Constraints

Figure 8 shows the pin constraints in the .xdc file of example design.

```
#constraints ZCU102 for SD3.0 HOST COntroller##HPC1
set_property PACKAGE_PIN AH1 [get_ports {mmc_dat_io[3]}]
set_property PACKAGE_PIN AE3 [get_ports {mmc_dat_io[2]}]
set_property PACKAGE_PIN AF2 [get_ports {mmc_dat_io[1]}]
set_property PACKAGE_PIN AH4 [get_ports {mmc_dat_io[0]}]
set_property PACKAGE_PIN AD2 [get_ports mmc_clk_o]
set_property PACKAGE_PIN AD4 [get_ports mmc_cmd_io]
set_property PACKAGE_PIN AE8 [get_ports data0_dir_o]
set_property PACKAGE_PIN AD7 [get_ports data1_3_dir_o]
set_property PACKAGE_PIN AG3 [get_ports cmd_dir_o]
set_property PACKAGE_PIN AE2 [get_ports voltage_1_8_slct_o]
set_property PACKAGE_PIN AH2 [get_ports mmc_cd_n_i]
set_property PACKAGE_PIN AJ2 [get_ports mmc_wp_n_i]

set_property IOSTANDARD LVCMOS18 [get_ports mmc_clk_o]
set_property IOSTANDARD LVCMOS18 [get_ports {mmc_dat_io[3]}]
set_property IOSTANDARD LVCMOS18 [get_ports {mmc_dat_io[2]}]
set_property IOSTANDARD LVCMOS18 [get_ports {mmc_dat_io[1]}]
set_property IOSTANDARD LVCMOS18 [get_ports {mmc_dat_io[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports cmd_dir_o]
set_property IOSTANDARD LVCMOS18 [get_ports data0_dir_o]
set_property IOSTANDARD LVCMOS18 [get_ports data1_3_dir_o]
set_property IOSTANDARD LVCMOS18 [get_ports mmc_cd_n_i]
set_property IOSTANDARD LVCMOS18 [get_ports mmc_cmd_io]
set_property IOSTANDARD LVCMOS18 [get_ports mmc_wp_n_i]
set_property IOSTANDARD LVCMOS18 [get_ports voltage_1_8_slct_o]
```

Figure 6:Pin Constraints in example design .xdc file

NOTE: These constraints are in accordance to example design created for the ZCU102 Evaluation kit and there is no need of adding the MIG/ DDR IO, UART IO, clock and reset pins .
Define the constraints for clock, reset and UART pins accordingly for any other custom board.

4 Design modification to be done for Custom Board

- Update the FPGA part number/board according to the FPGA device used
- Update the complete design for the selected FPGA device
- Update the pin constraints for SD interface, clock, reset and UART pins
- Update the clock constraint according to the input clock frequency for the selected FPGA device
- Recompile the design to generate the new binaries and use the XSA file to create the new application project in Vitis

5 Resource Utilization

The table below shows the resource utilization summary for ZCU102 dev. kit. for SD Host Controller IP.

Table 2: Resource Utilization for device for ZCU102 dev. kit .

Resource	Utilization	Available
LUT	2530	274080
FF	2216	548160