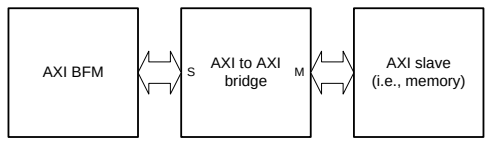
AMBA AXI to AXI Bus Bridge Design

2015 – 2016 - 2017

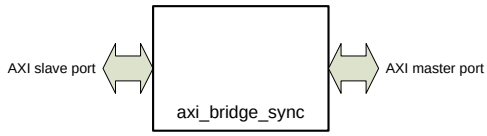
Ando Ki
(adki@future-ds.com)

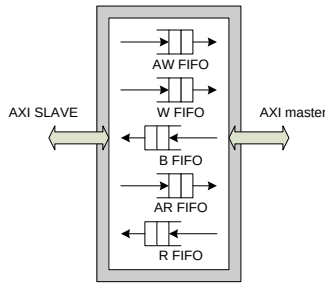
AXI to AXI bus bridge design & verification

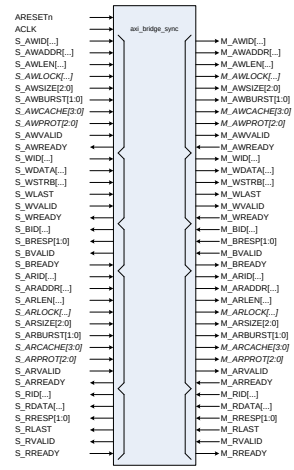


- Test-bench includes
  - Bus bridge
  - BFM: Test pattern generation
  - Memory

AXI to AXI bus bridge







axi_bridge_sync.v

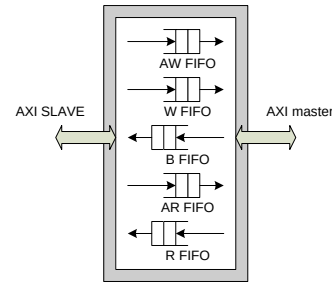
```
`include "axi_bridge_sync_fifo.v"
`include "axi_bridge_core.v"
`timescale 1ns/1ns

module axi_bridge_sync
    #(parameter AXI_WIDTH_CID=4
              , AXI_WIDTH_ID =4
              , AXI_WIDTH_AD =32
              , AXI_WIDTH_DA =32
              , AXI_WIDTH_DS =(AXI_WIDTH_DA/8)
              , AXI_WIDTH_SID=(AXI_WIDTH_CID+AXI_WIDTH_ID)
              )
(
      input  wire                 ARESETn
    , input  wire                 ACLK
    //-------------------------------------------------------
    , input  wire [AXI_WIDTH_SID-1:0] S_AWID
    , input  wire [AXI_WIDTH_AD-1:0]  S_AWADDR
    .....
    , input  wire                 M_RLAST
    , input  wire                 M_RVALID
    , output wire                  M_RREADY
    //-------------------------------------------------------
);
    //-------------------------------------------------------
    axi_bridge_core #(.AXI_WIDTH_CID(AXI_WIDTH_CID)
                    ,.AXI_WIDTH_ID (AXI_WIDTH_ID )
                    ,.AXI_WIDTH_AD (AXI_WIDTH_AD )
                    ,.AXI_WIDTH_DA (AXI_WIDTH_DA )
                    ,.ASYNC        (1'b0         )
                    )
    Uaxi_bridge_core (
                              .ARESETn        (ARESETn     )
                             ,.S_ACLK         (ACLK        )
                             ,.S_AWID         (S_AWID      )
                             ,.S_AWADDR       (S_AWADDR    )
                             ....
                             ,.M_RLAST        (M_RLAST     )
                             ,.M_RVALID       (M_RVALID    )
                             ,.M_RREADY       (M_RREADY    )
    );
    //-------------------------------------------------------
endmodule
```

axi_bridge_core.v

```
`timescale 1ns/1ns

module axi_bridge_core
    #(parameter AXI_WIDTH_CID=4        // Channel ID
              , AXI_WIDTH_ID =4        // ID width in bits
              , AXI_WIDTH_AD =32       // address width
              , AXI_WIDTH_DA =32       // data width
              , AXI_WIDTH_DS =(AXI_WIDTH_DA/8)
              , AXI_WIDTH_SID=(AXI_WIDTH_CID+AXI_WIDTH_ID)
              , ASYNC        =1'b0
              )
(
      input  wire                  ARESETn
    //------------------------------------------------------
    , input  wire                  S_ACLK
    , input  wire [AXI_WIDTH_SID-1:0] S_AWID
    , input  wire [AXI_WIDTH_AD-1:0]  S_AWADDR
     .........
    , input  wire                  M_RLAST
    , input  wire                  M_RVALID
    , output wire                   M_RREADY
    //------------------------------------------------------
);
... fifo instances
endmodule
```



Simulation with ModelSim (1/4)

```
# Makefile
SHELL                   = /bin/sh
MAKEFILE        = Makefile
#----------------------------------------------------
VLIB            = $(shell which vlib)
VLOG            = $(shell which vlog)
VSIM            = $(shell which vsim)
WORK            = work
#----------------------------------------------------
TOP             = top
#----------------------------------------------------
all: vlib compile simulate

vlib:
                if [ -d $(WORK) ]; then /bin/rm -rf $(WORK); fi
                $(VLIB) $(WORK)

compile:
                $(VLOG) -lint -work $(WORK)  -f modelsim.args

simulate: compile
                $(VSIM) -novopt -c -do "run -all; quit" $(WORK).$(TOP)
```

Modelsim commands

Specify where to store compile results

Compilation

Simulation

```
@ECHO OFF
REM RunMe.bat
SET MODELSIMWORK=work
SET MODELSIMVLIB=vlib
SET MODELSIMVSIM=vsim
SET MODELSIMVCOM=vcom
SET MODELSIMVLOG=vlog

SET DESIGNTOP=top

IF EXIST %MODELSIMWORK% RMDIR /S/Q %MODELSIMWORK%

%MODELSIMVLIB% %MODELSIMWORK%
%MODELSIMVLOG% -work %MODELSIMWORK% -lint^
            -f modelsim.args
%MODELSIMVSIM% -novopt -c -do "run -all; quit"^
          %MODELSIMWORK%.%DESIGNTOP%
```

Simulation with ModelSim (2/4)

modelsim.args

```
//-----------------------------------------
+define+RIGOR
+define+VCD
//-----------------------------------------
./sim_define.v
//-----------------------------------------
+incdir+../../rtl/verilog
        ../../rtl/verilog/axi_bridge_sync.v
//-----------------------------------------
+incdir+../../bench.sync/verilog
        ../../bench.sync/verilog/bfm_axi.v
+incdir+../../bench.sync/verilog
        ../../bench.sync/verilog/axi_slave.v
+incdir+../../bench.sync/verilog
        ../../bench.sync/verilog/top.v
//-----------------------------------------
```

sim_define.v

```
`ifndef _SIM_DEFINE_V_
`define _SIM_DEFNE_V_
//-----------------------------------------------------
// Copyright (c) 2013 by Dynalith Systems Co., Ltd.
// All rights reserved.
//-----------------------------------------------------
`define SIM
`undef  SYN
//-----------------------------------------------------
`ifdef SIM
`define RIGOR
`define VCD
`endif
//-----------------------------------------------------
`endif
```


Example: AXI bridge

- This example shows how to use BFM with tasks
  - Step 1: go to your project directory
    - [user@host] cd $(PROJECT)/codes/axi_to_axi
  - Step 2: see the codes
    - [user@host] cd $(PROJECT)/codes/axi_to_axi/desing/verilog
  - Step 3: compile and run
    - [user@host] cd $(PROJECT)/codes/axi_to_axi/sim.sync/modelsim
    - [user@host] make
  - Step 4: waveform view
    - [user@host] gtkwave wave.vcd &

```
[user@host] cd $(PROJECT)/codes/axi_to_axi/sim.sync/modelsim
[user@host] make
[user@host] gtkwave wave.vcd &
```