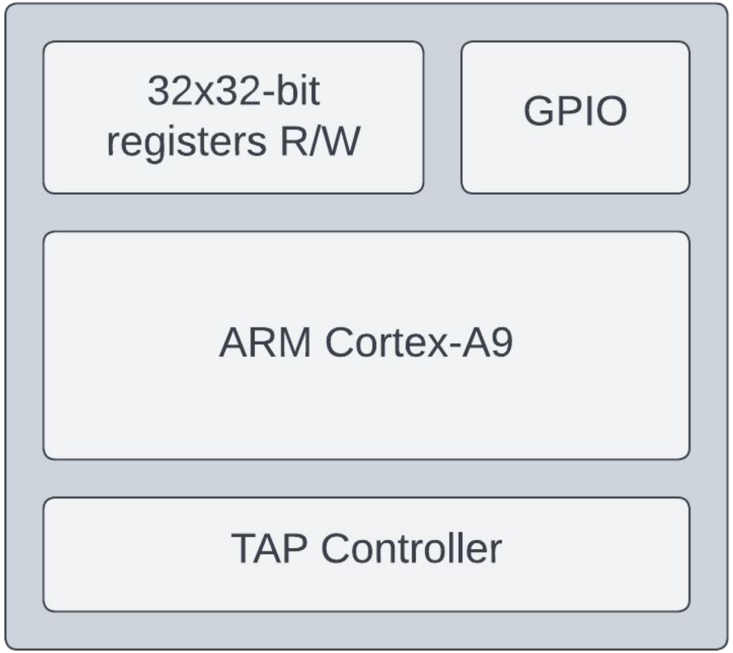
RWU HOCHSCHULE RAVENSBURG-WEINGARTEN UNIVERSITY OF APPLIED SCIENCES

System-on-Chip

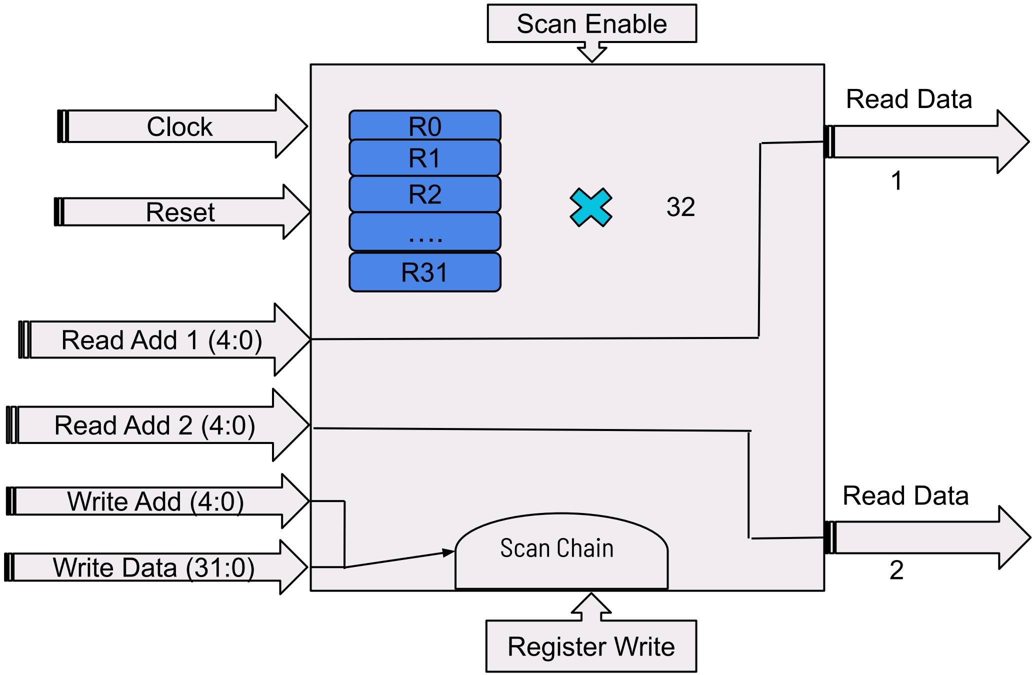
Prepared By :
Jayant Patil
Avanindra Kumar Mishra
Aditya Grewal
Sagar Shreeshailappa Hosmani

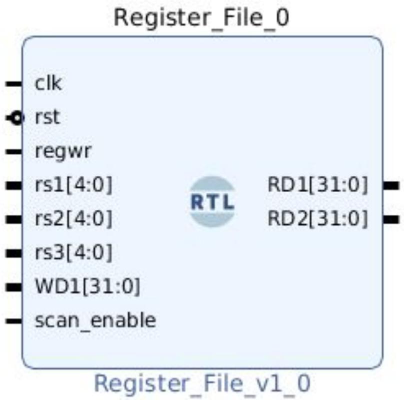
Under Guidance of :
Prof. Dr. A. Siggelkow

Project Overview



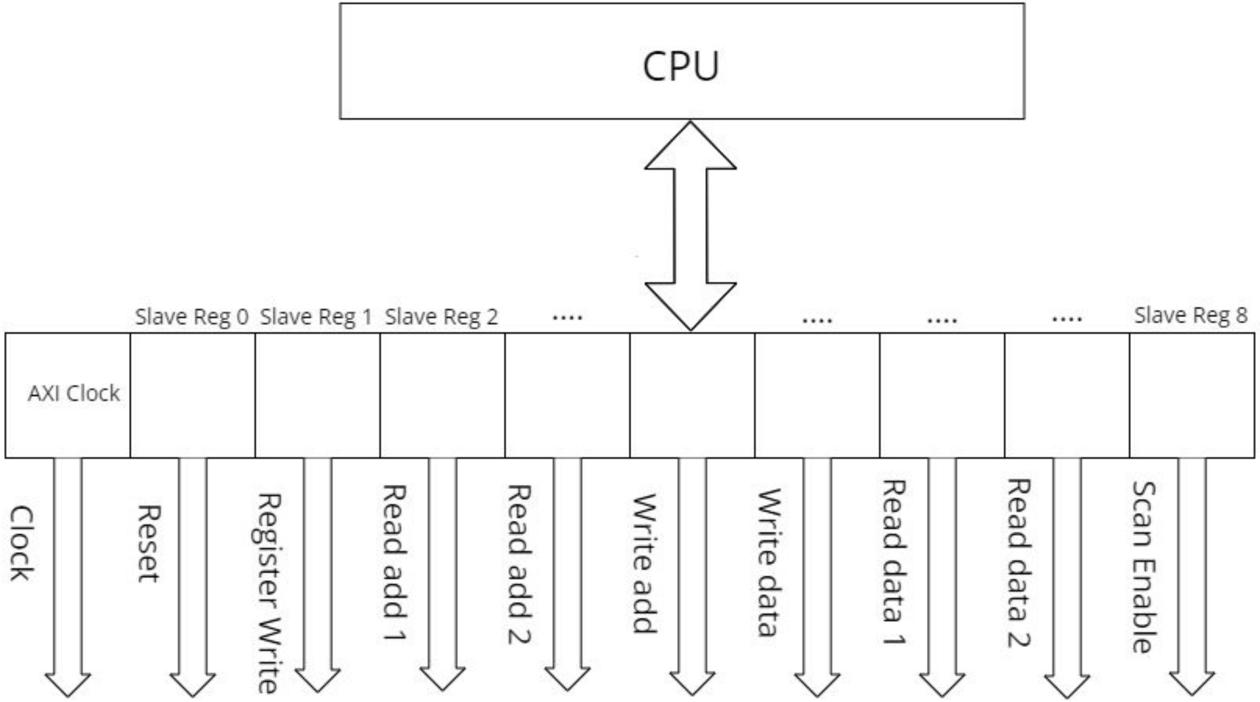
Register File



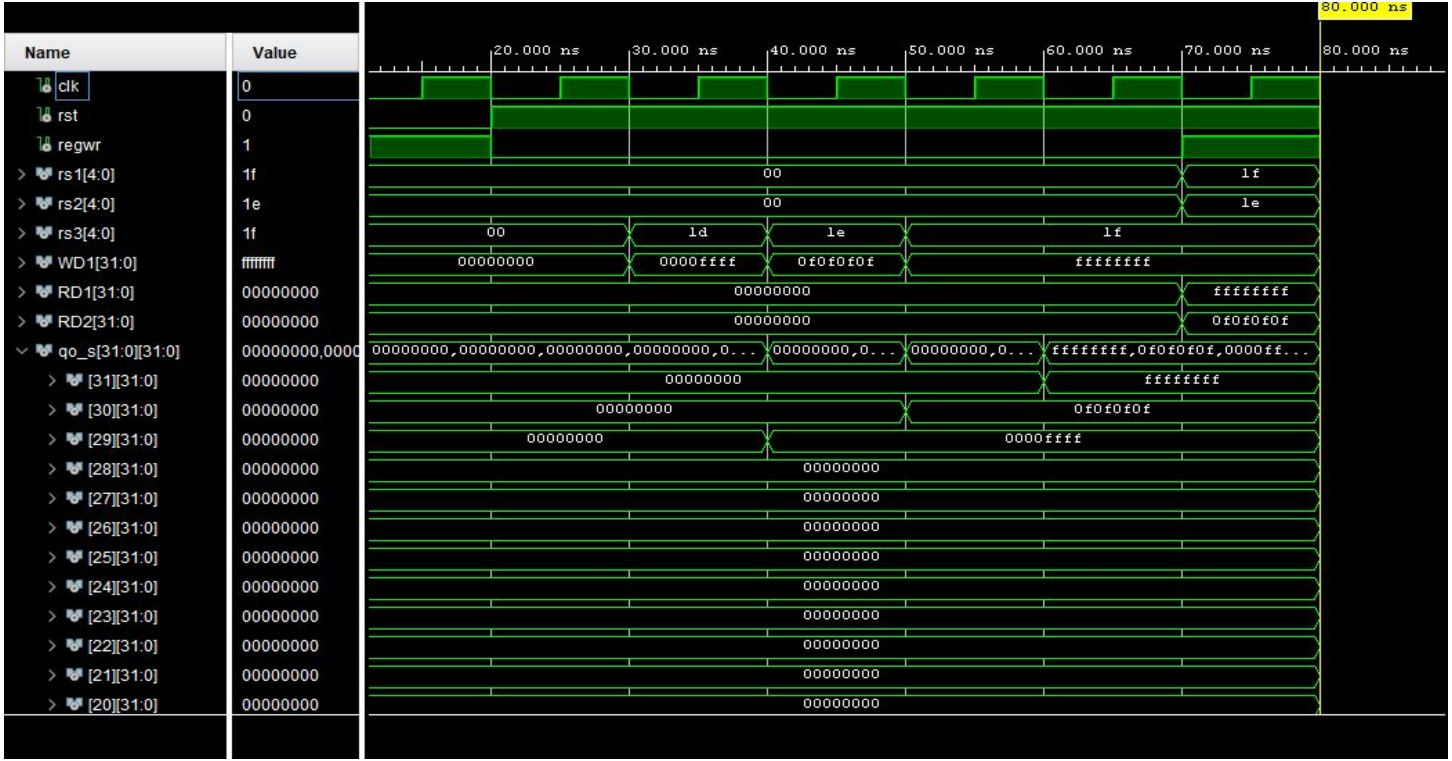


Vivado View

AXI4 - Lite Bus

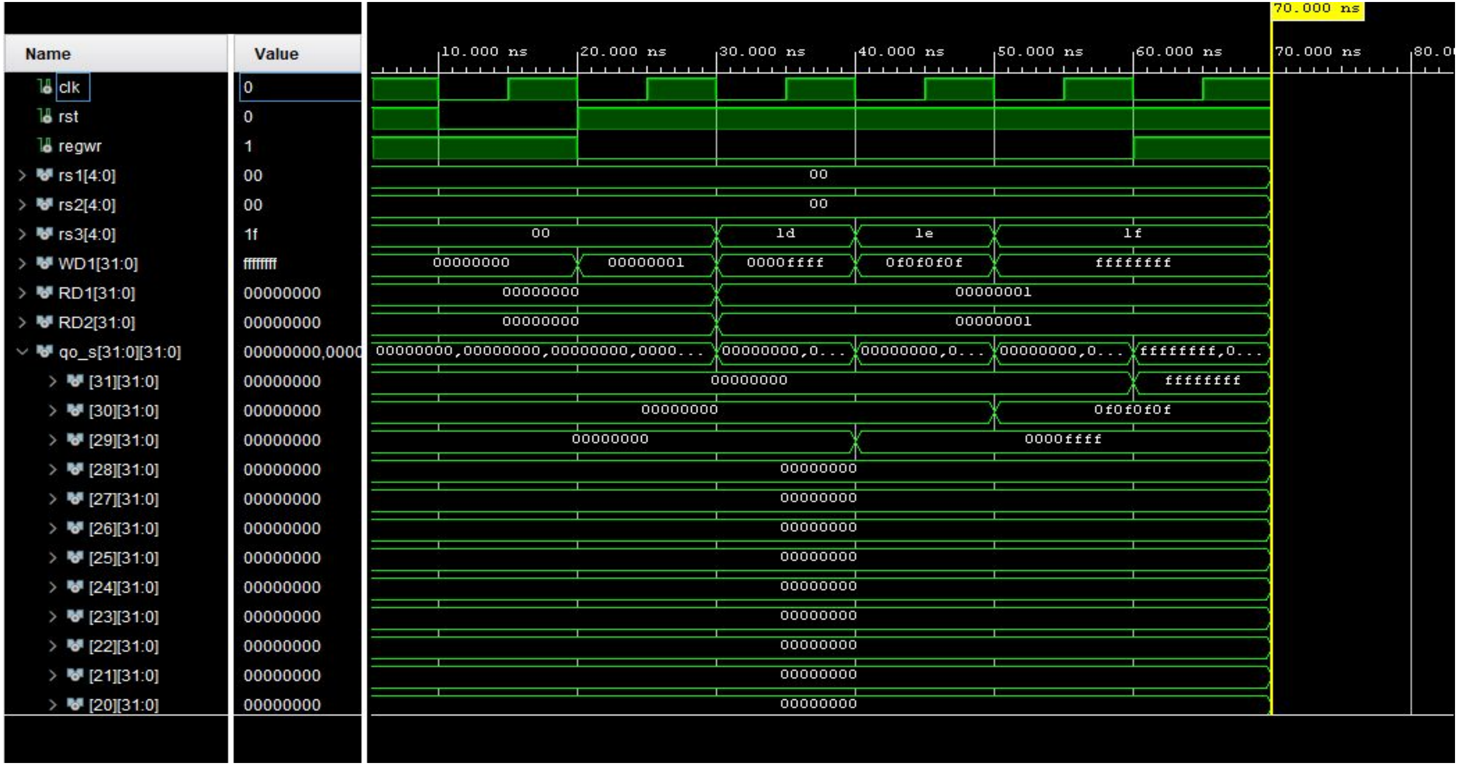


Simulation - Read



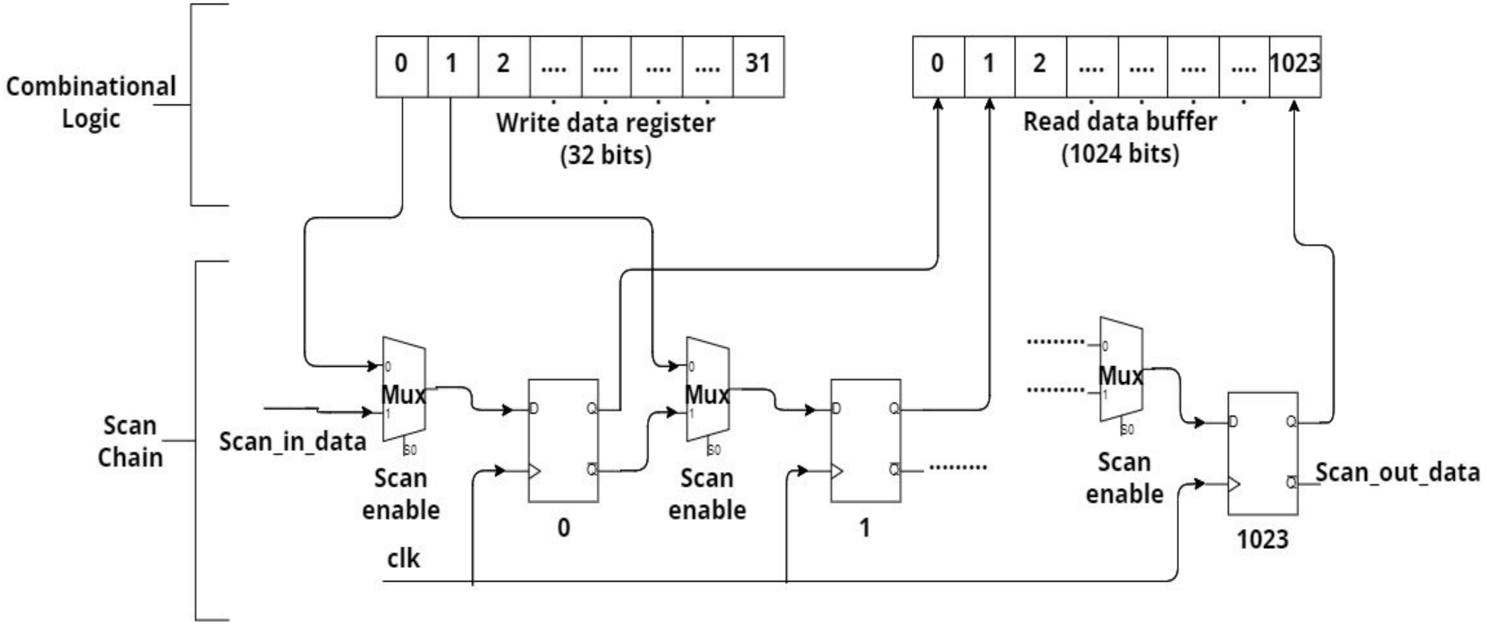
Name	Value
clk	0
rst	0
regwr	1
rs1[4:0]	1f
rs2[4:0]	1e
rs3[4:0]	1f
WD1[31:0]	ffffffff
RD1[31:0]	00000000
RD2[31:0]	00000000
qo_s[31:0][31:0]	00000000,0000
[31][31:0]	00000000
[30][31:0]	00000000
[29][31:0]	00000000
[28][31:0]	00000000
[27][31:0]	00000000
[26][31:0]	00000000
[25][31:0]	00000000
[24][31:0]	00000000
[23][31:0]	00000000
[22][31:0]	00000000
[21][31:0]	00000000
[20][31:0]	00000000

Simulation - Write

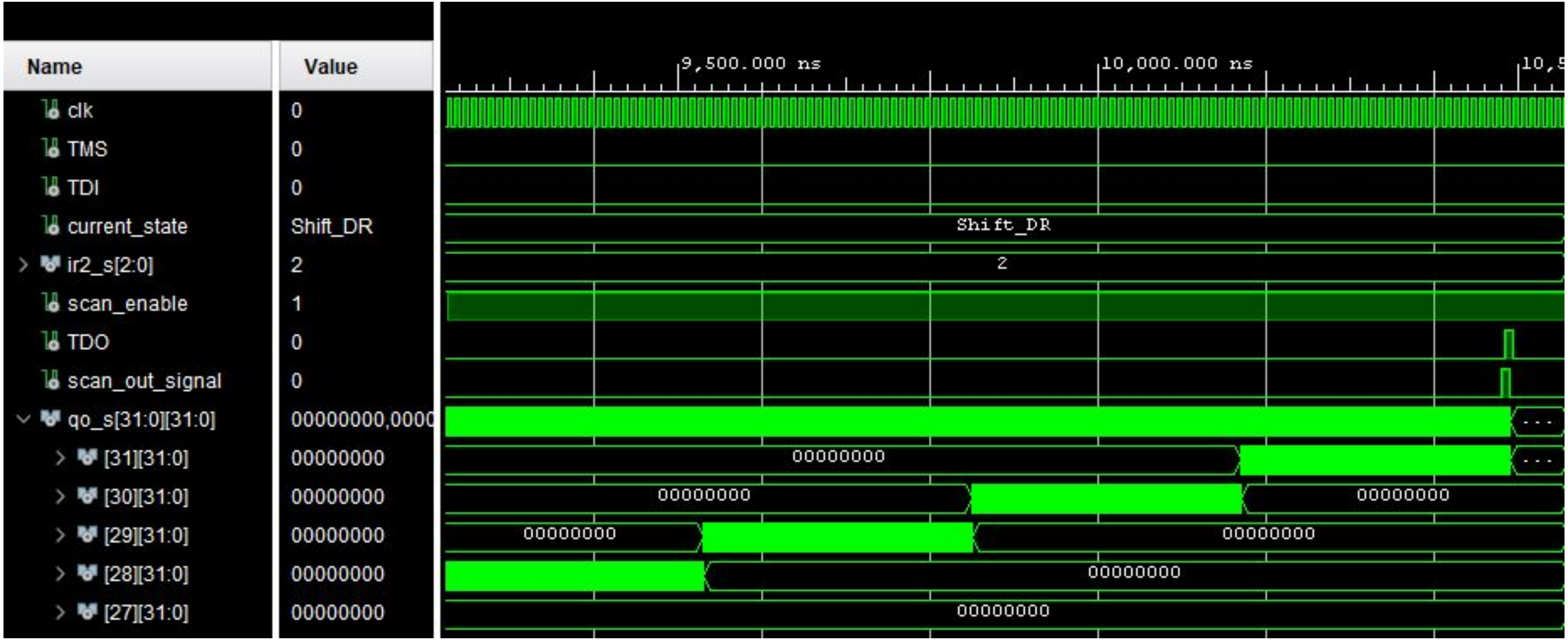


Name	Value
clk	0
rst	0
regwr	1
rs1[4:0]	00
rs2[4:0]	00
rs3[4:0]	1f
WD1[31:0]	ffffffff
RD1[31:0]	00000000
RD2[31:0]	00000000
qo_s[31:0][31:0]	00000000,0000
[31][31:0]	00000000
[30][31:0]	00000000
[29][31:0]	00000000
[28][31:0]	00000000
[27][31:0]	00000000
[26][31:0]	00000000
[25][31:0]	00000000
[24][31:0]	00000000
[23][31:0]	00000000
[22][31:0]	00000000
[21][31:0]	00000000
[20][31:0]	00000000

Scan chain


Combinational Logic
0
1
2
....
....
....
....
31
Write data register (32 bits)
0
1
2
....
....
....
....
1023
Read data buffer (1024 bits)
Scan Chain
Scan_in_data
Mux
Scan enable
0
Mux
Scan enable
1
....
....
....
Mux
Scan enable
1023
Scan_out_data
clk
D
Q
Q

SImulation - Scan Chain

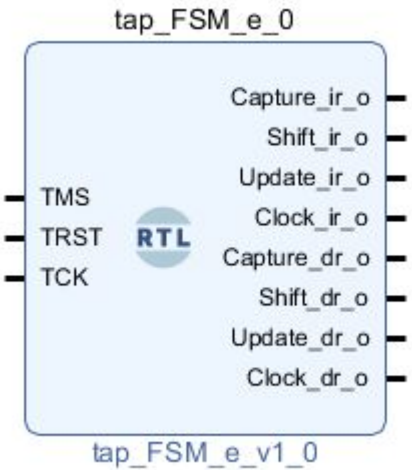


Name	Value
clk	0
TMS	0
TDI	0
current_state	Shift_DR
ir2_s[2:0]	2
scan_enable	1
TDO	0
scan_out_signal	0
qo_s[31:0][31:0]	00000000,0000
[31][31:0]	00000000
[30][31:0]	00000000
[29][31:0]	00000000
[28][31:0]	00000000
[27][31:0]	00000000

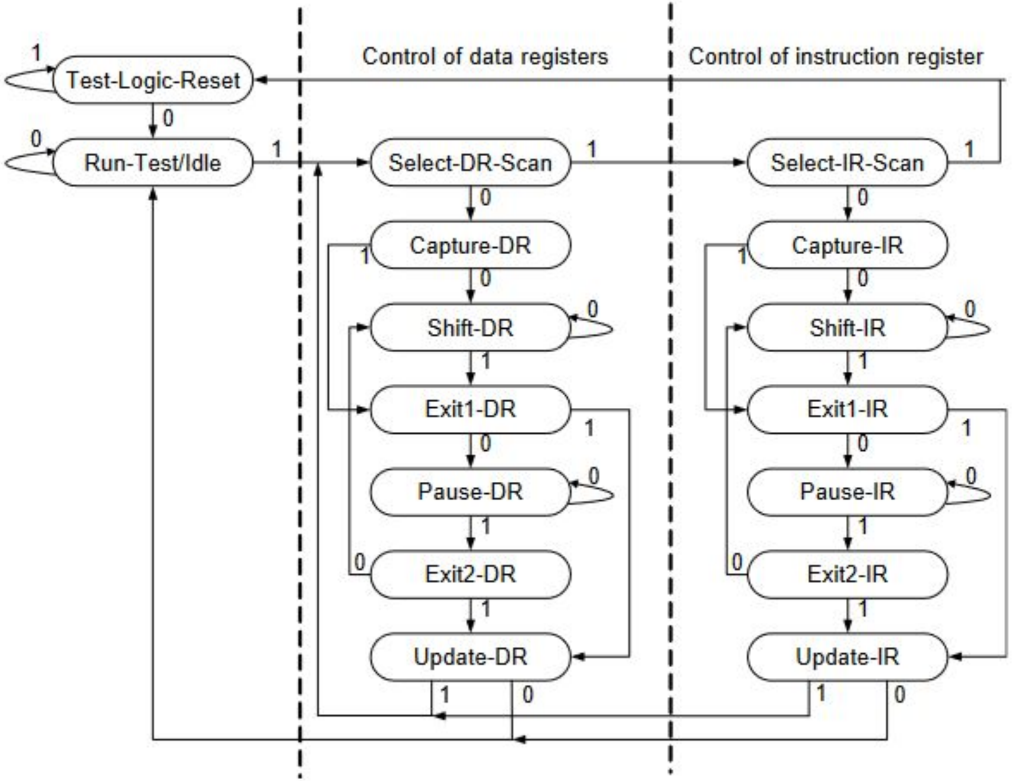
TAP FSM

- **Input**
  1. Test Reset
  2. Test mode select
  3. Test clock

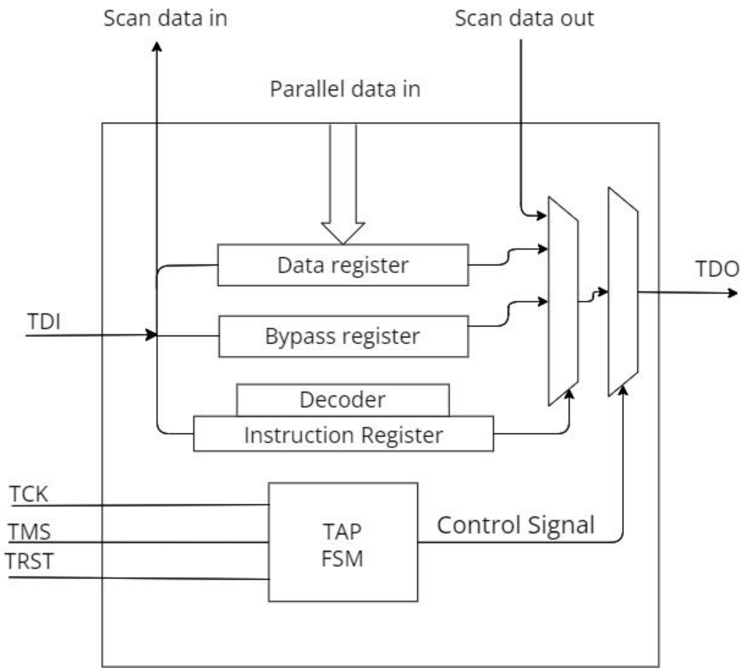
- **Output**
  1. Capture IR Signal
  2. Shift IR Signal
  3. Update IR Signal
  4. Clock IR Signal
  5. Shift DR Signal
  6. Update DR Signal
  7. Capture_DR
  8. Clock DR Signal

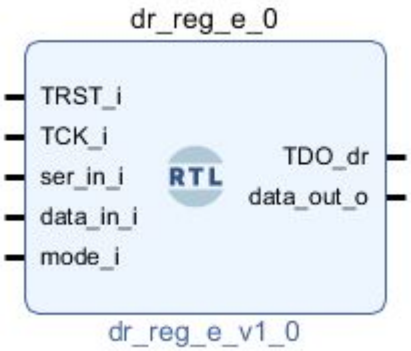


State diagram of TAP controller

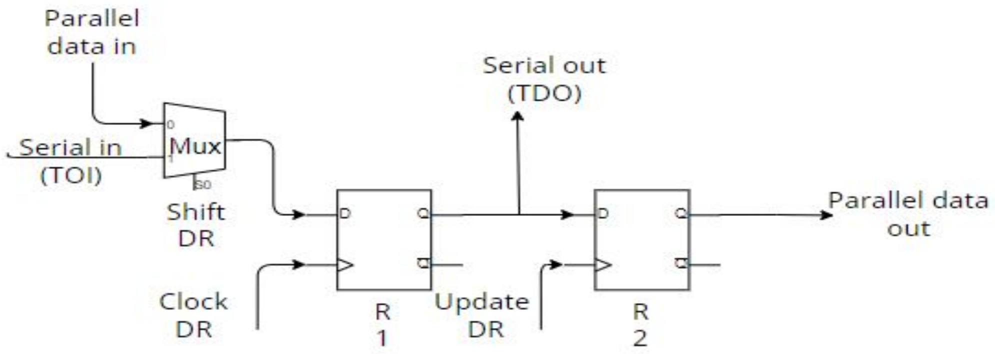


TAP Controller Architecture:





Data/Instruction Register



- Test Operations:
  - Capture
  - Shift
  - Update

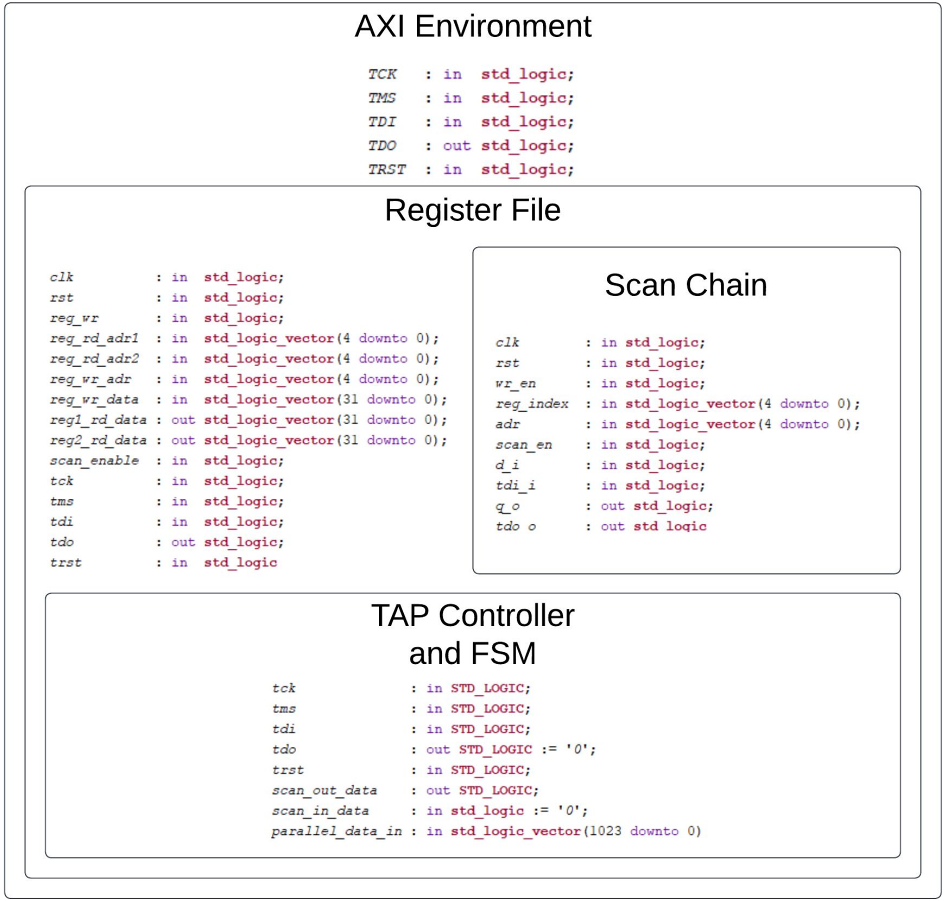
Instruction Register

- Design Same as Data register.
- TAP Controller Tests:
  - Scan Chain
  - Bypass
  - Sample
  - Preload

Parallel and Serial Data Handling

- Parallel data (parallel data in, parallel data out)
- Serial data (TDI, TDO)

Top Level Hierarchy



AXI Environment

```
TCK   : in  std_logic;
TMS   : in  std_logic;
TDI   : in  std_logic;
TDO   : out std_logic;
TRST  : in  std_logic;
```

Register File

```
clk           : in  std_logic;
rst           : in  std_logic;
reg_wr        : in  std_logic;
reg_rd_adr1   : in  std_logic_vector(4 downto 0);
reg_rd_adr2   : in  std_logic_vector(4 downto 0);
reg_wr_adr    : in  std_logic_vector(4 downto 0);
reg_wr_data   : in  std_logic_vector(31 downto 0);
reg1_rd_data  : out std_logic_vector(31 downto 0);
reg2_rd_data  : out std_logic_vector(31 downto 0);
scan_enable   : in  std_logic;
tck           : in  std_logic;
tms           : in  std_logic;
tdi           : in  std_logic;
tdo           : out std_logic;
trst          : in  std_logic
```

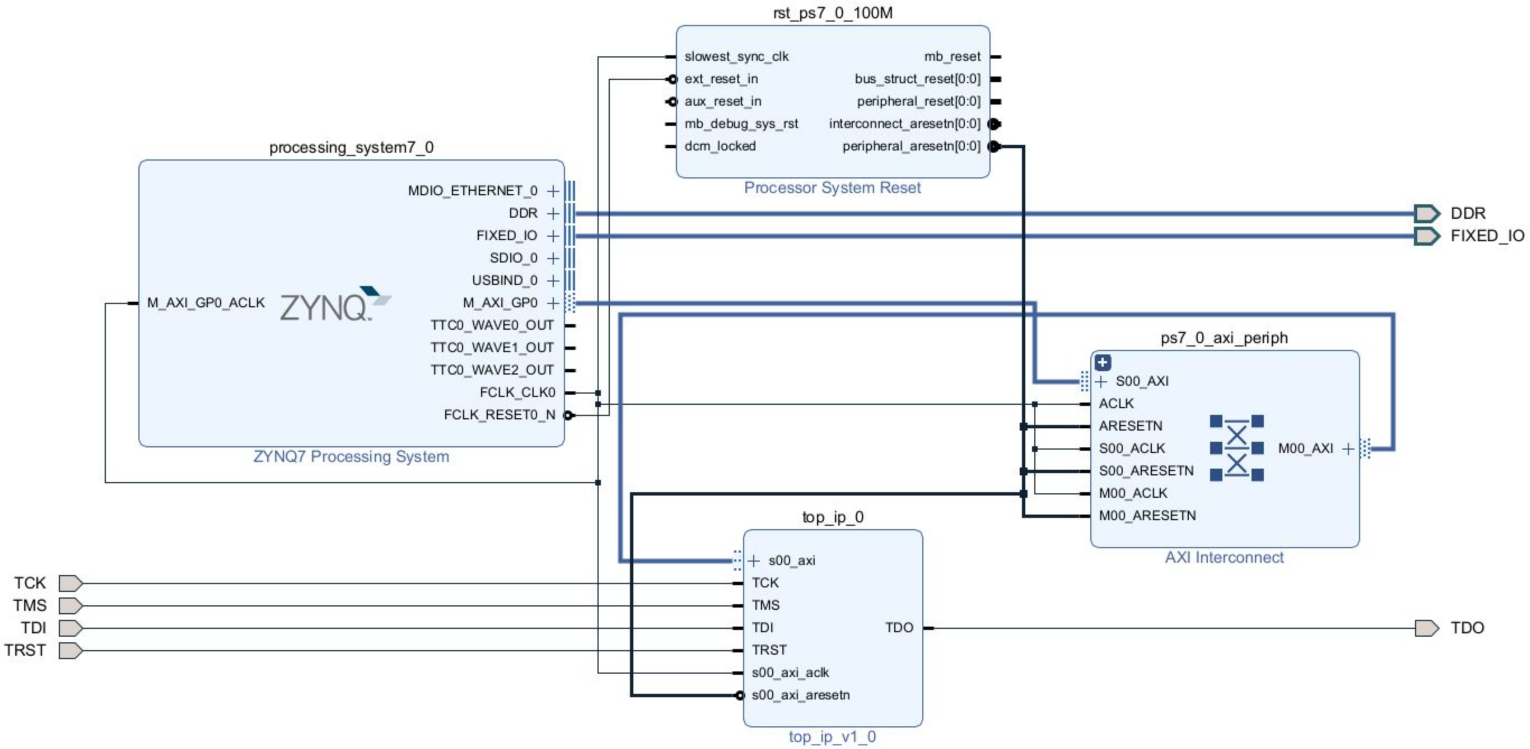
Scan Chain

```
clk         : in std_logic;
rst         : in std_logic;
wr_en       : in std_logic;
reg_index   : in std_logic_vector(4 downto 0);
adr         : in std_logic_vector(4 downto 0);
scan_en     : in std_logic;
d_i         : in std_logic;
tdi_i       : in std_logic;
q_o         : out std_logic;
tdo_o       : out std_logic
```

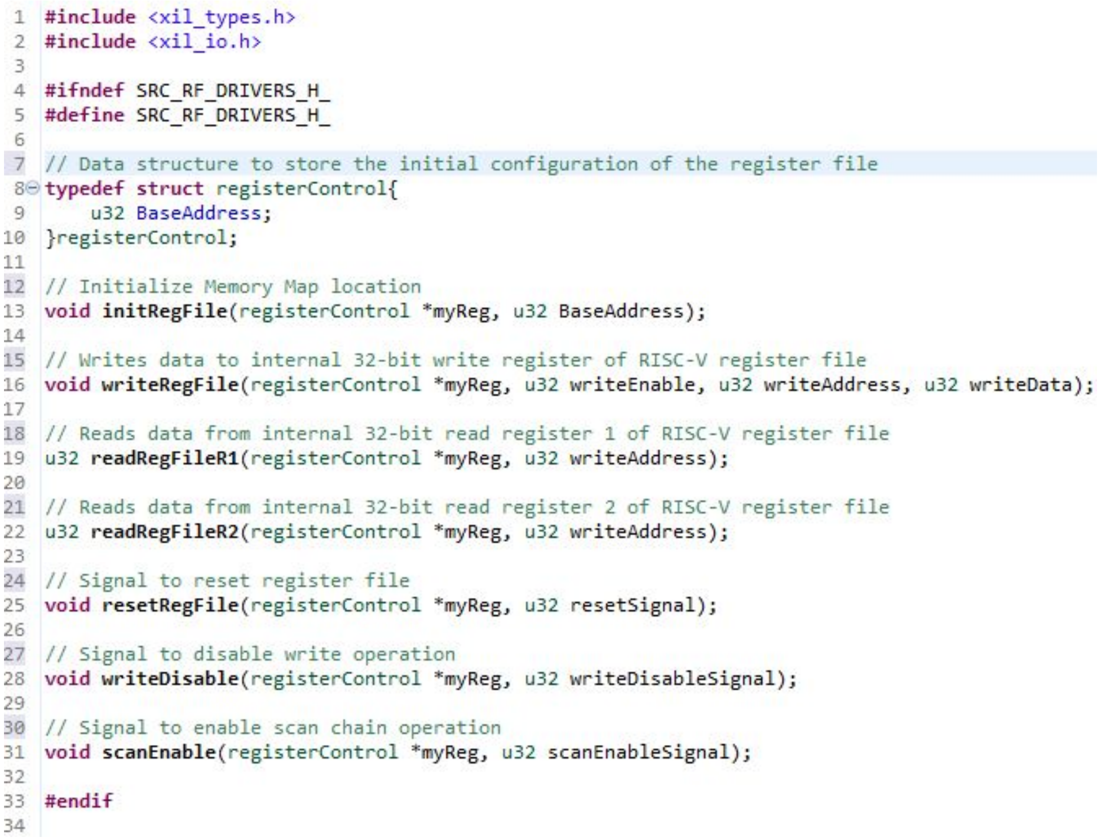
TAP Controller
and FSM

```
tck              : in STD_LOGIC;
tms              : in STD_LOGIC;
tdi              : in STD_LOGIC;
tdo              : out STD_LOGIC := '0';
trst             : in STD_LOGIC;
scan_out_data    : out STD_LOGIC;
scan_in_data     : in std_logic := '0';
parallel_data_in : in std_logic_vector(1023 downto 0)
```

Custom SoC Block Diagram



Board Support Package



```c
#include <xil_types.h>
#include <xil_io.h>

#ifndef SRC_RF_DRIVERS_H_
#define SRC_RF_DRIVERS_H_

// Data structure to store the initial configuration of the register file
typedef struct registerControl{
    u32 BaseAddress;
}registerControl;

// Initialize Memory Map location
void initRegFile(registerControl *myReg, u32 BaseAddress);

// Writes data to internal 32-bit write register of RISC-V register file
void writeRegFile(registerControl *myReg, u32 writeEnable, u32 writeAddress, u32 writeData);

// Reads data from internal 32-bit read register 1 of RISC-V register file
u32 readRegFileR1(registerControl *myReg, u32 writeAddress);

// Reads data from internal 32-bit read register 2 of RISC-V register file
u32 readRegFileR2(registerControl *myReg, u32 writeAddress);

// Signal to reset register file
void resetRegFile(registerControl *myReg, u32 resetSignal);

// Signal to disable write operation
void writeDisable(registerControl *myReg, u32 writeDisableSignal);

// Signal to enable scan chain operation
void scanEnable(registerControl *myReg, u32 scanEnableSignal);

#endif

```

THANK YOU!