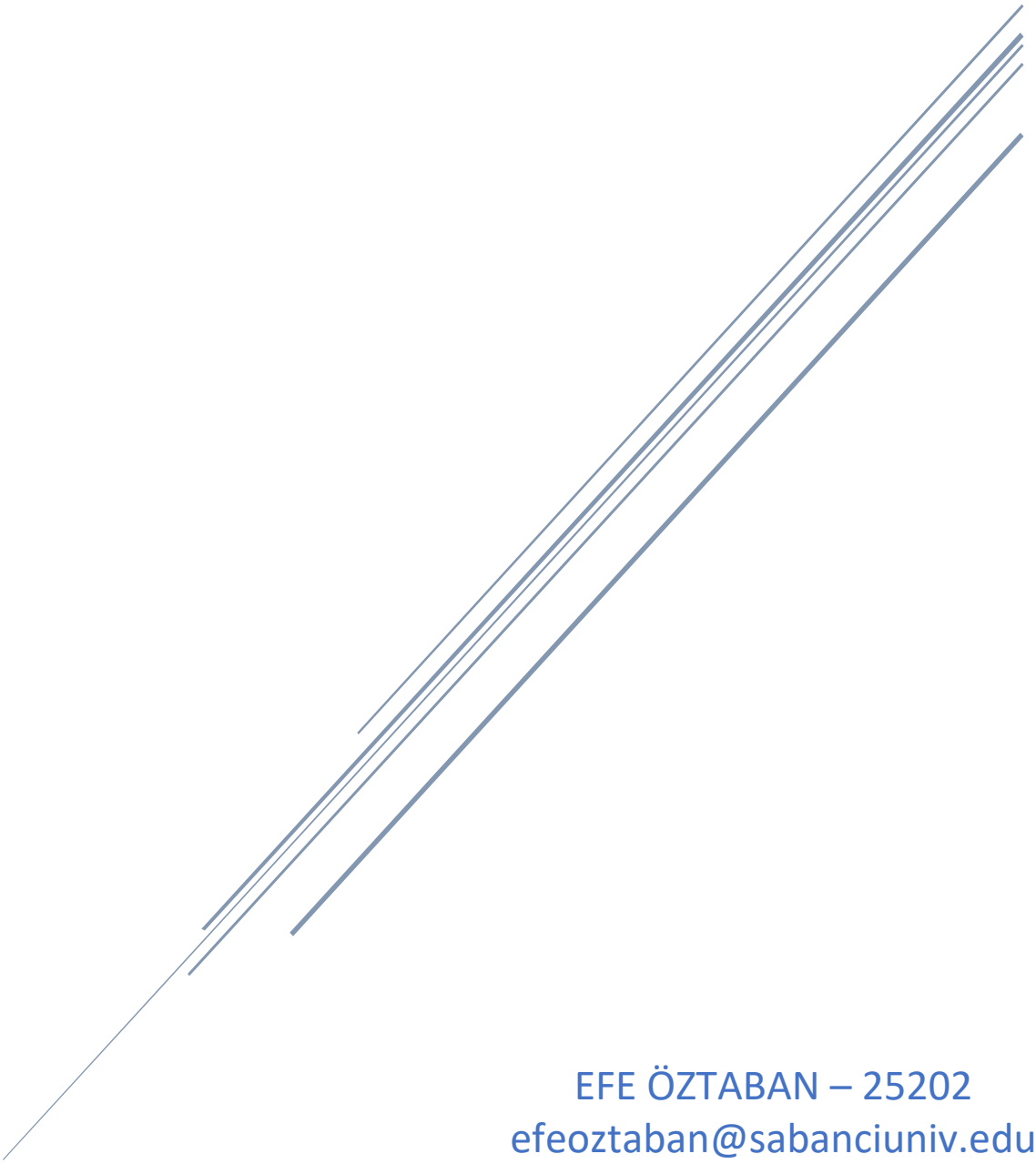
EE 402 VLSI SYSTEM DESIGN II

Lab Report for Lab #1

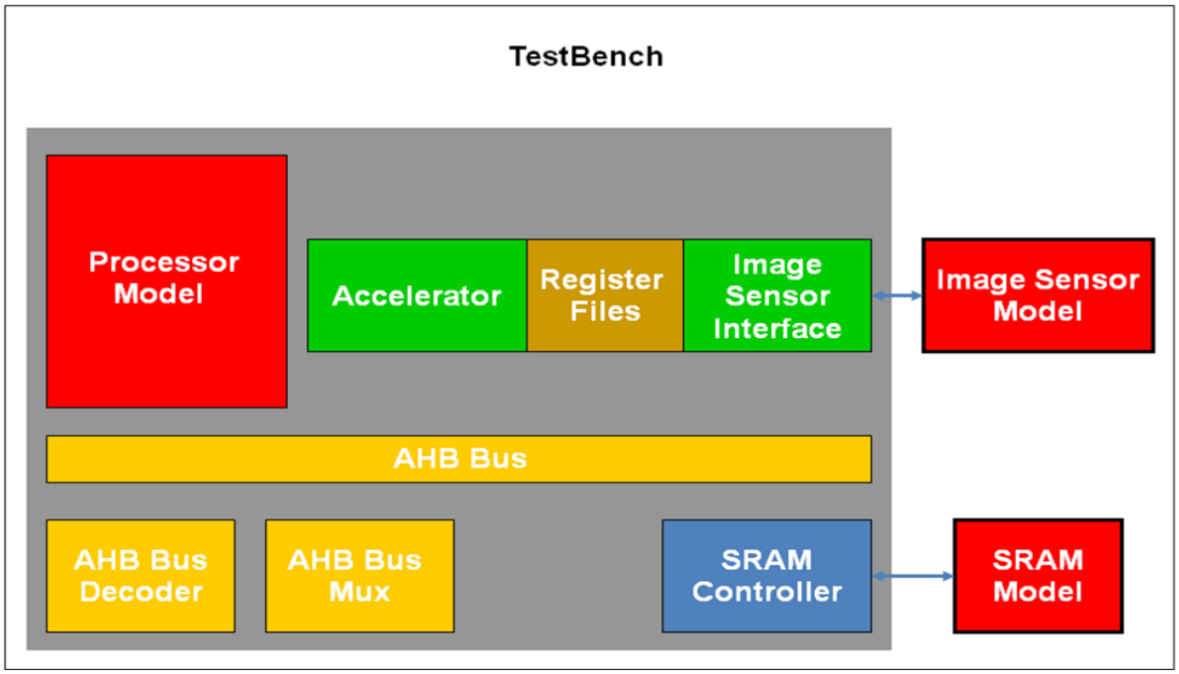


EFE ÖZTABAN – 25202
efeoztaban@sabanciuniv.edu

KAYRA BİLGİN – 25117
kayrabilgin@sabanciuniv.edu

1. Introduction

In this laboratory assignment, a System-on-Chip ASIC prototype is given as the Verilog RTL implementation. The design of the SoC ASIC is given below with a figure. The architecture, module implementations and testbench is analysed in the laboratory assignment. The SoC ASIC implementation is simulated with Mentor Graphics QuestaSim and Xilinx ISE. The simulation results are given in the related section of the report. Module hierarchy, memory map and the interconnect diagram are made according to the design. These diagrams are given in the report under the related sections. In addition to that, operations performed by each module in the design is briefly explained in the report.



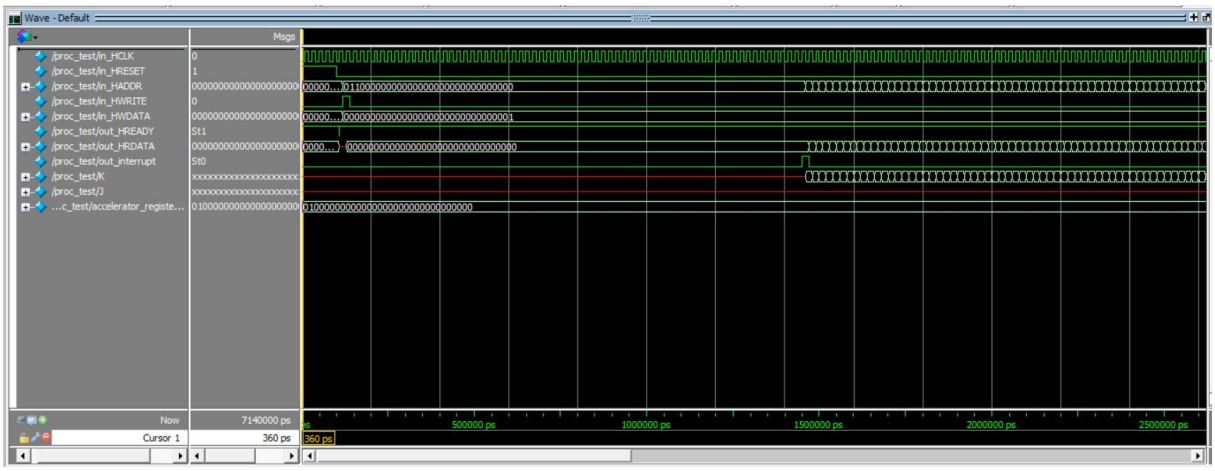
2. Simulation Results

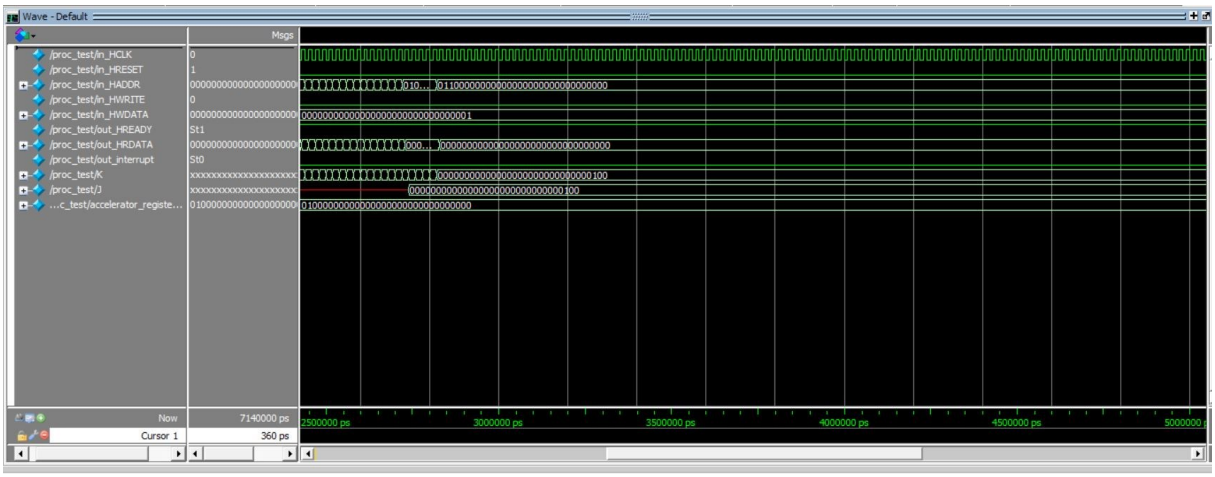
The simulations for the given SoC ASIC are performed with Mentor Graphics QuestaSim and Xilinx ISE.

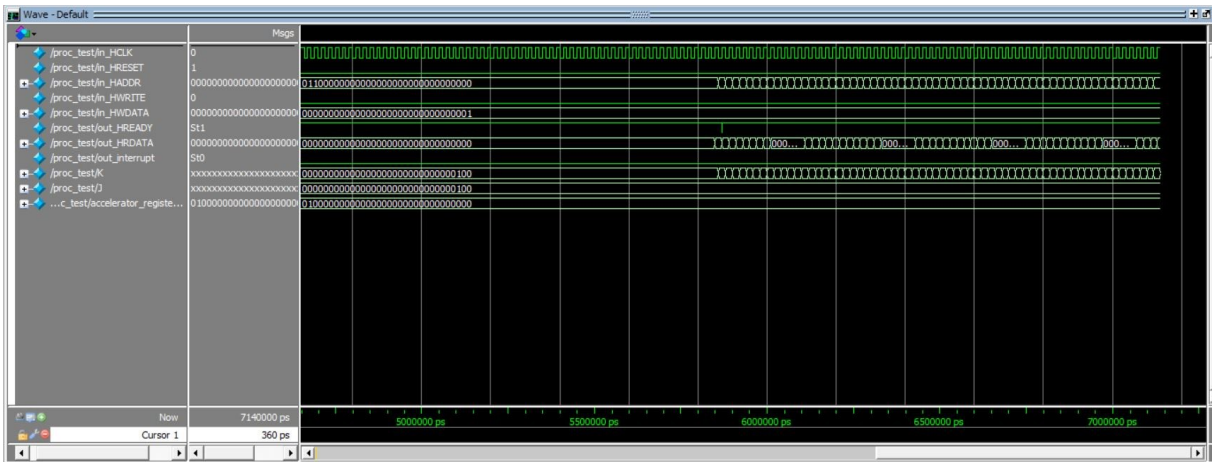
It can be seen that the simulation is runed correctly from the results below:

```
VSIM 3> run -all
#  0 OK
#  1 OK
#  2 OK
#  3 OK
#  4 OK
#  5 OK
#  6 OK
#  7 OK
#  8 OK
#  9 OK
# 10 OK
# 11 OK
# 12 OK
# 13 OK
# 14 OK
# 15 OK
# 16 OK
# 17 OK
# 18 OK
# 19 OK
# 20 OK
# 21 OK
# 22 OK
# 23 OK
# 24 OK
# 25 OK
# 26 OK
# 27 OK
# 28 OK
# 29 OK
# 30 OK
# 31 OK
# 32 OK
# 33 OK
# 34 OK
# 35 OK
# 36 OK
# 37 OK
# 38 OK
# 39 OK
# 40 OK
# 41 OK
# 42 OK
# 43 OK
# 44 OK
# 45 OK
# 46 OK
# 47 OK
# 48 OK
# 49 OK
# 50 OK
# 51 OK
# 52 OK
# 53 OK
# 54 OK
# 55 OK
# 56 OK
# 57 OK
# 58 OK
# 59 OK
# 60 OK
# 61 OK
# 62 OK
# 63 OK
```

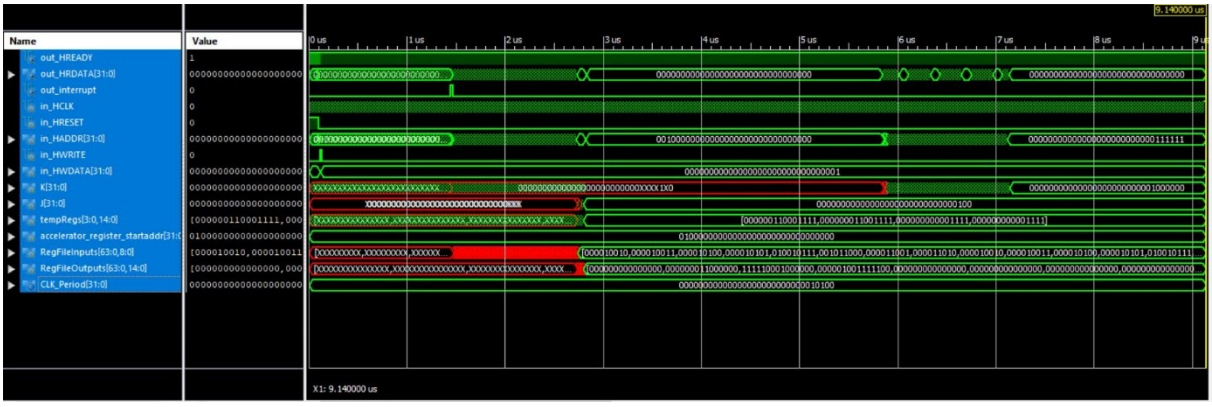
The simulation results from the Mentor Graphics QuestaSim are given with waveform below:





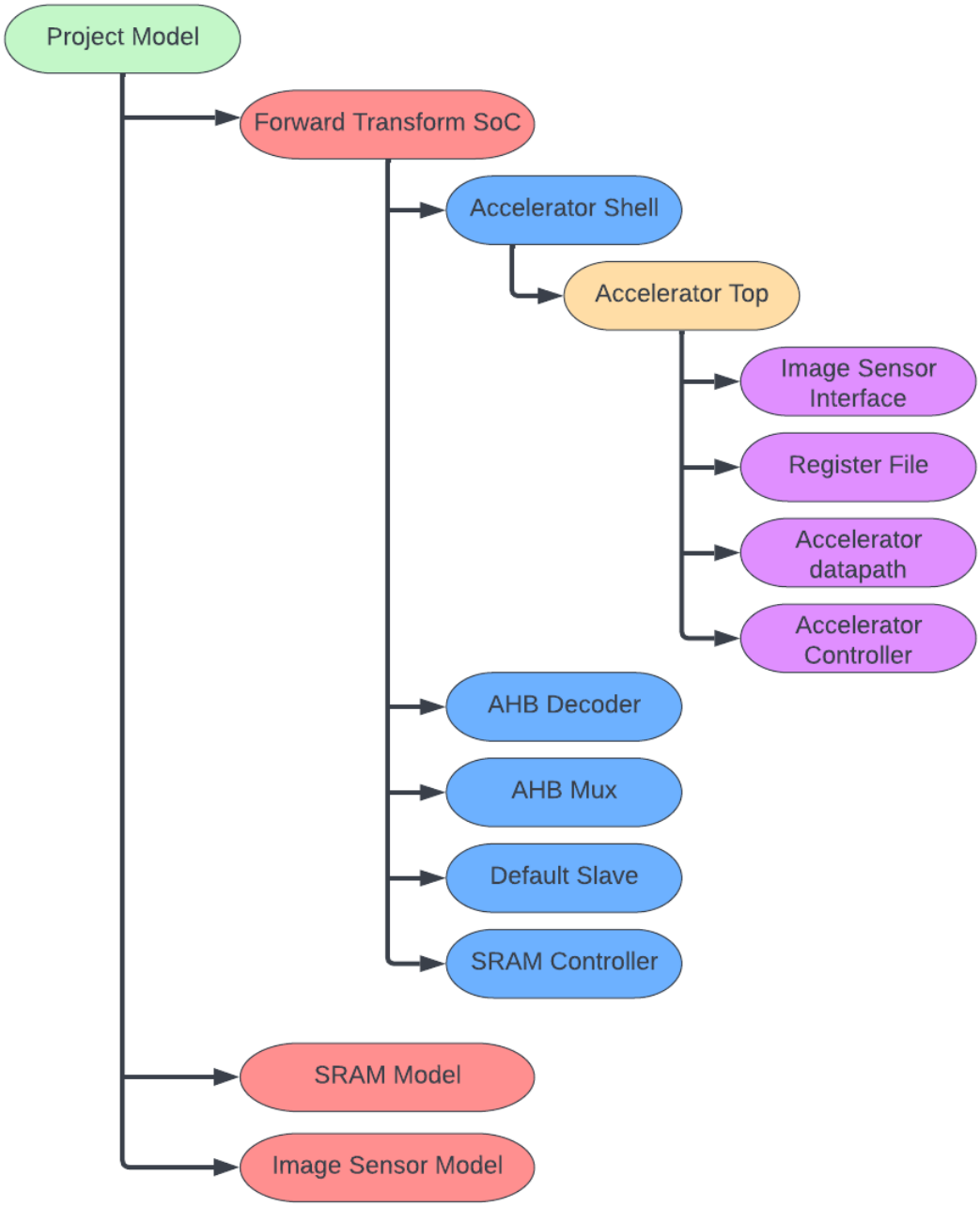


Also, the simulation result of the design in Xilinx ISE is given below:



3. Module Hierarchy

Module hierarchy of the design is given below with a diagram:



4. Operations Performed by each Module

The operations performed by each module is explained in this section.

Project Model

In this module 3 main components are defined. One of the is the Forward Transform SoC design. The other parts are the Image Sensor Model and SRAM Model. Also, the interconnects between these three components are made in this module.

a. Forward Transform SoC

This module defines 5 modules. These modules are Accelerator Shell, AHB Decoder, AHB Mux, Default Slave and SRAM Controller. The interconnections between these 5 modules are done in this module. It receives the main control signals (such as CLK, RESET, READY…) from outside. It also connects the 5 modules inside with the Image Sensor and SRAM model.

i. Accelerator Shell

This module defines Accelerator Top module. It also contains the control registers for image sensor interface and accelerator. It also controls the data registers. This module is the only master module in the AHB bus. It looks at incoming data from the AHB bus and decide on the change control registers. It controls the image sensor interface start, image sensor interface finished, and accelerator done signals. It informs that the image sensor interface and accelerator is done. Then it takes the data from accelerator controller and sends it to SRAM controller to write them into SRAM.

1. Accelerator Top

This module is defining 4 different modules. These modules are Image Sensor Interface, Register File, Accelerator Datapath and Accelerator Controller. The interconnections between these four modules are done in this module. It also connects the Accelerator Shell top these 4 modules.

a. Image Sensor Interface

This module is an interface module for Image sensor. Accelerator Shell Module starts the Image Sensor interface by setting the related control register as 1. After the Image Sensor interface is started, it sends a signal to the image sensor to start capturing image. After image sensor is started, Image sensor interface started to receive the data from the image sensor. It receives 8 bits of data at a time and continue to receive data for 64 clock cycles. In each clock cycle, it writes this data into the Register File. After all the data is received, it sets the out_frame_capture to 0 for stopping the Image sensor. Then it sends a signal to accelerator to inform that the reading and writing process is done.

b. Register File

This module includes a register file in the size of 8x64. Image Sensor Interface Module can write to this register. When Image Sensor Interface module receives a data from Image Sensor, this data is kept in this register file. Also, processor and accelerator can directly access to this register file and read the data.

c. Accelerator Datapath

This module performs the H.264 forward transform algorithm. It is controlled by the Accelerator Controller module. It receives 16 parallel inputs from the Accelerator Controller and gives 16 parallel outputs. It uses a state machine to do the calculations on the data accordingly to the H.264 forward transform algorithm.

d. Accelerator Controller

This module is the control module of Accelerator Datapath. When Accelerator Controller receives the start signal from Accelerator Shell, it starts. It reads the data from Register file and sends it to the Accelerator Datapath. It gives 16x8 bit data to the accelerator Datapath in parallel. Then it starts the Accelerator Datapath by a signal. After the Accelerator Datapath is finished the calculations, it takes the result data from Accelerator Datapath and sends it to Accelerator Shell.

ii. AHB Decoder

This module takes a 32-bit data as input. Then it checks the last two bits to decide on the selection of the slave. It selects the SRAM controller if “00”, it selects the Accelerator if “01” and it selects the Default Slave if “1?”. AHB Mux module uses this selection bits as input.

iii. AHB Mux

This module takes input from the AHB decoder about the selection of the slave which can be Default Slave, Accelerator or SRAM Controller. It uses this selection to choose the data input from slaves. It chooses the data input and sends it as output. It also gives a ready signal when the data output is ready.

iv. Default Slave

This module is also a slave module on bus. If an undefined area of the memory map is accessed, or an invalid address is driven onto the address bus, the default slave outputs are selected and passed to the current bus master.

v. SRAM Controller

This module controls the SRAM module. It sends the write and read enable signals to control SRAM. Also, it sends the read address, write address, and write data to the SRAM which are coming from Accelerator Shell.

b. SRAM Model

This module is a off-chip SRAM memory design. There is a register which is in the size of 32x32k in the module. It performs read and write operations. It uses the global clock signal for the operations. It is controlled by the SRAM controller in the Forward Transform SoC. When the read enable pin is activated, it returns the data from the register according to the given read address. When the write enable pin is activated, it receives the new data from in_write_data and store it in the register according to the given write address.

c. Image Sensor Model

This module is a behavioural module. It is simulating an image sensor by reading an input file. It reads the “input.txt” as an input and store in a register array which is 8x64. When it receives a signal to start the frame capture, it sends 8 bits for every clock cycle to Image Sensor Interface. It continues to send 8 bits of information for 64 clock cycles. After the information is send in 64 clock cycle, it makes the out_done flag as 1 to inform that all the information is sent.

Testbench

This is a testbench module. It simulates the model as giving signals like CLK, RESET… It also acts as the processor in the design as a behavioural model. It calculates the result of the h.264 transform algorithm and compares it with the result done by the accelerator.

5. Memory Map

Memory map of the Soc ASIC design is given below with a figure and explained in the pseudo code below:

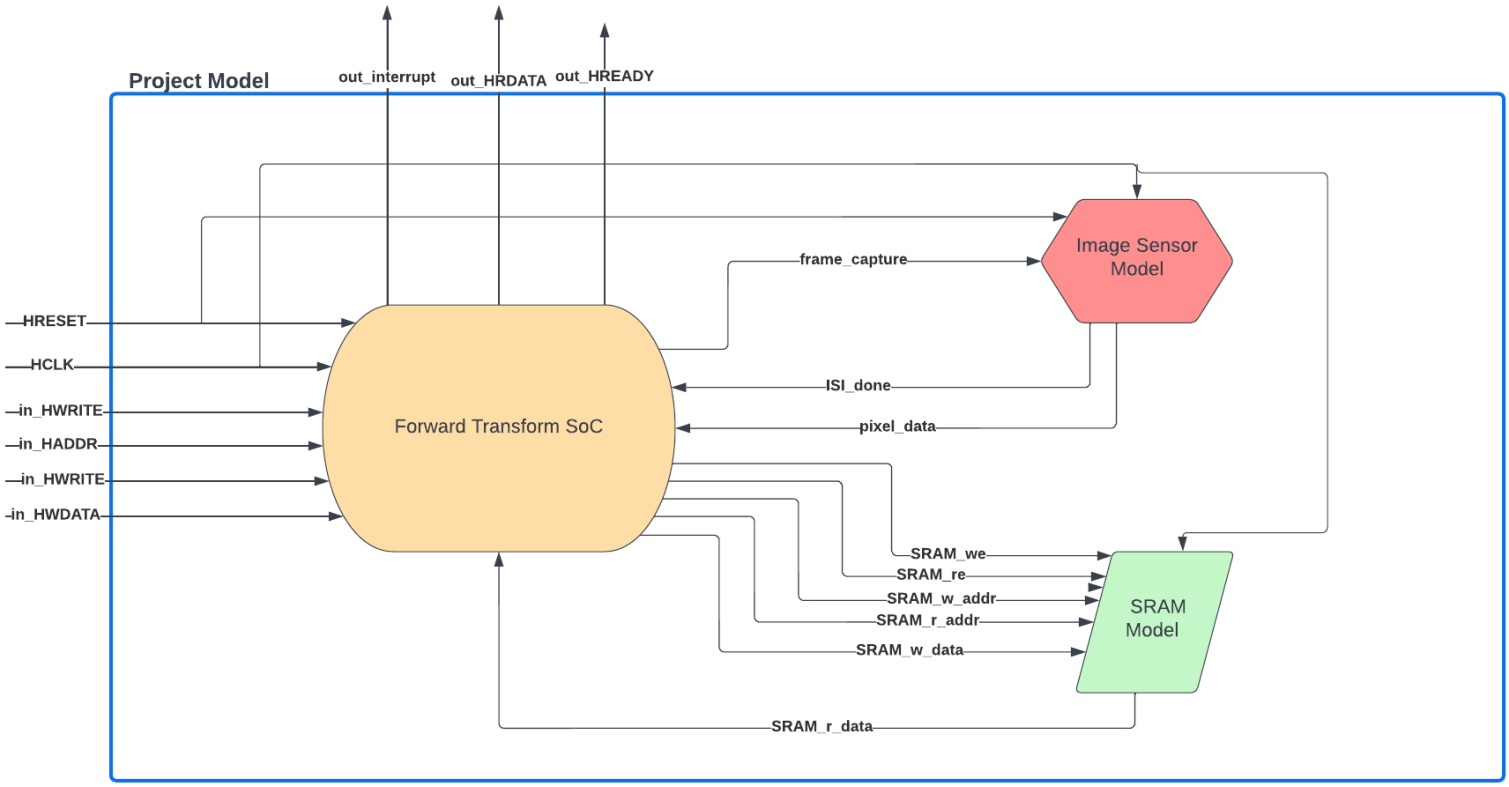
```
select_bits = In_HADDR[31:30]
If (select_bits == 00)
        SRAM Controller
If (select_bits == 01)
        Accelerator
If (select_bits == 10 or select_bits == 11)
        Default Slave
```

Memory Map

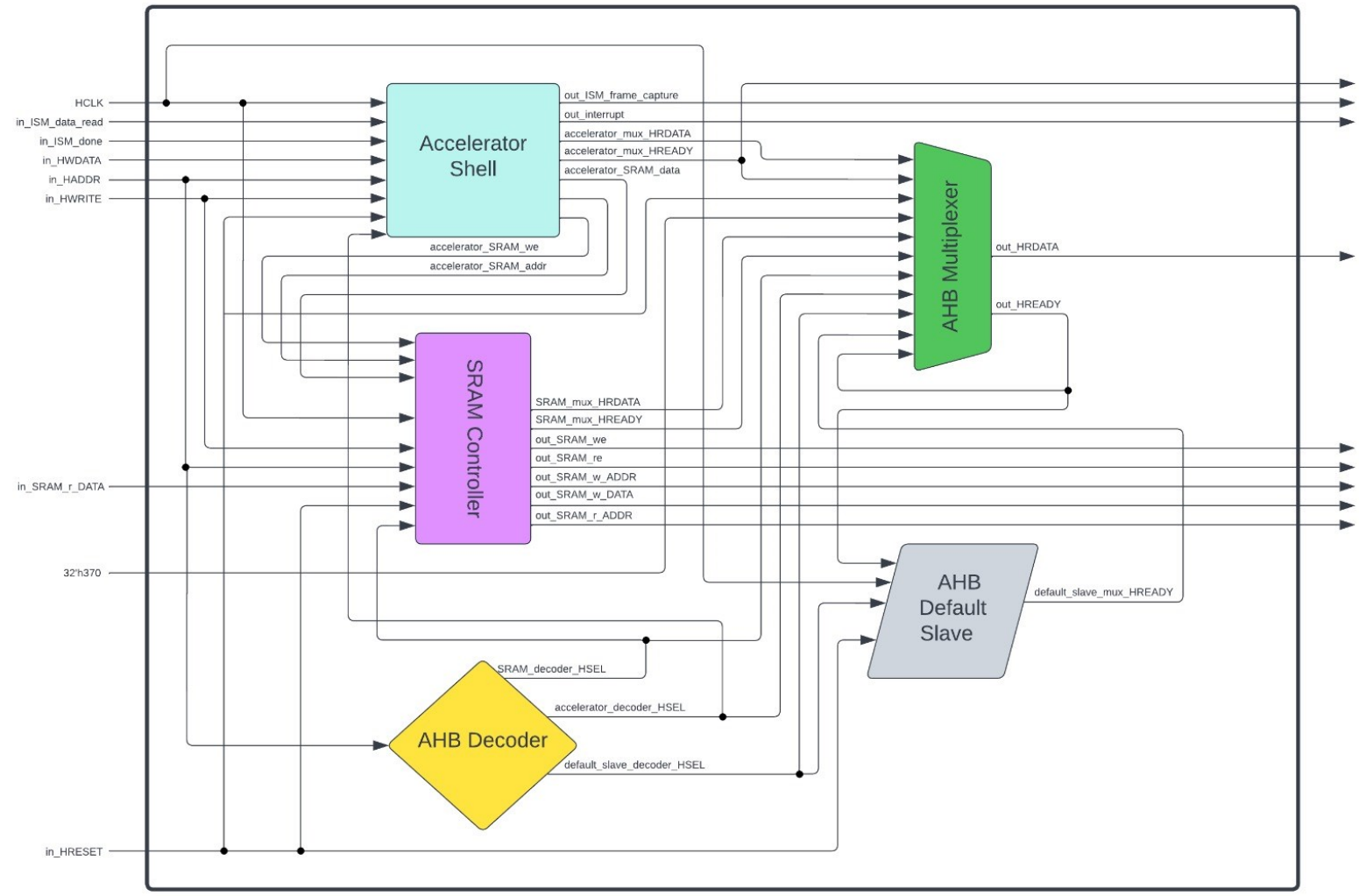
0x00000000	SRAM Controller
.....	
0x3FFFFFFF	
0x40000000	Accelerator
.....	
0x7FFFFFFF	
0x80000000	Default Slave
.....	

6. Modules and Main Interconnects

Modules and their interconnects of the Soc ASIC design are given with a diagram below:



Forward Transform SoC



Accelerator Shell