# PORTLAND STATE UNIVERSITY

## ECE571- Introduction to System Verilog

## PROJECT

## Verification Plan of Intel 8237A DMA Controller


Group Members

Saket Swami

Kaavya Thrilochan

Manoj Reddy

## Verification Requirements:

1. Description of Verification Levels:

   The Verification starts with a basic Sanity Test which checks the basic functionality of the Controller. Then to verify and maximize the coverage, the verification is divided into two levels, Unit Level and System Level.

   a. Sanity Test:

      Perform one read and one write transaction (Memory to IO and IO to Memory).

   b. Unit Level:

      Check the functional blocks provided by the design team.

      - Timing and Control Block
      - Priority Block
      - Register/ DataPath

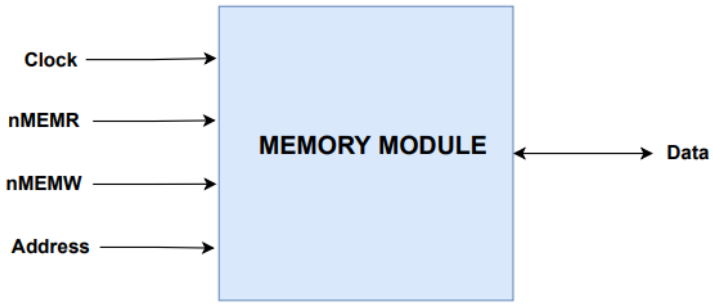
   c. System Level:

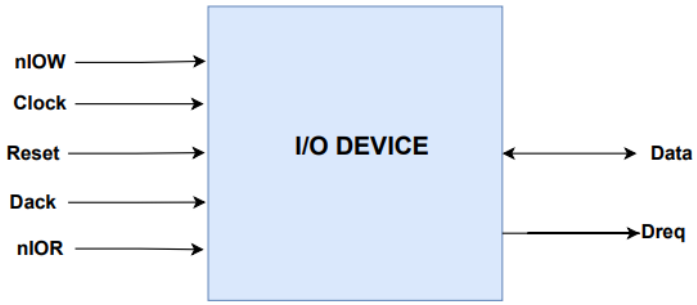
      We plan to test the multiple block interaction in the system level check. The plan to make assertions and check the timing block diagrams to verify the FSM logic for timing and Control Block and check the register contents.

   d. Testbench Components and Interactions:

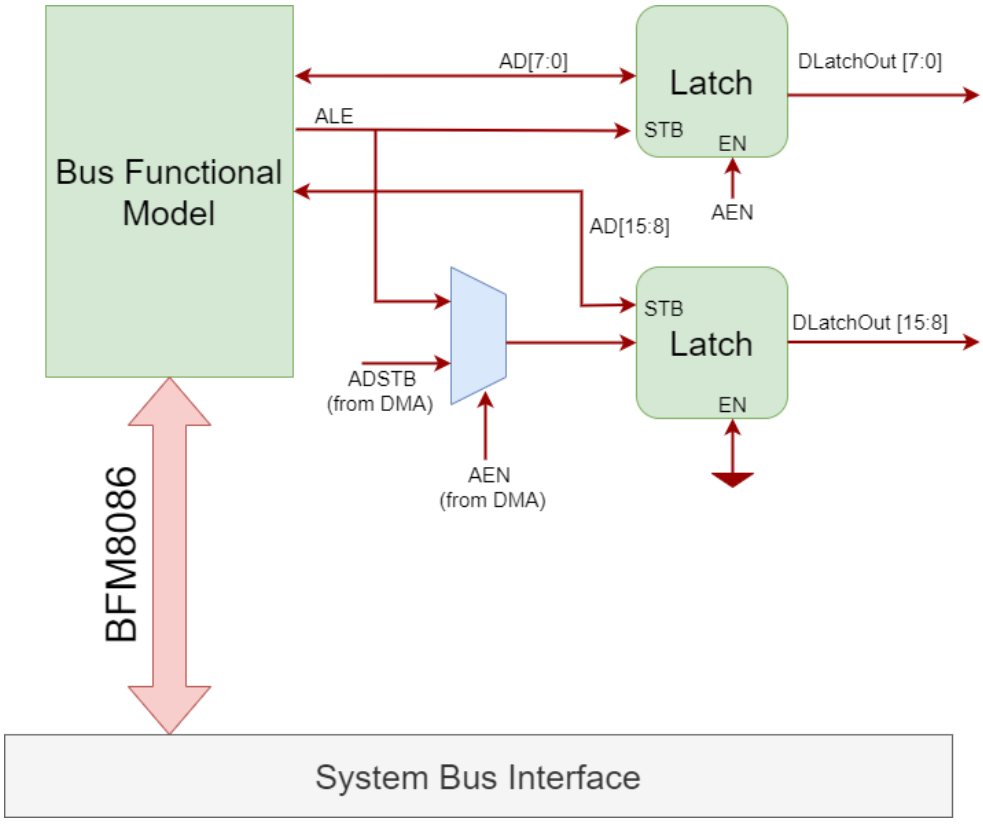
      We plan to include the testbench environment with instantiation of timing and control block, Bus functional model, Priority Block, Datapath Control Module and a top module. The environments plans to use, transaction class, generator, driver, mailbox, master interface, master unit, master-slave interface, checkers, assertion module. Errors are injected such as giving an out of bounds address and the output is checked for error.

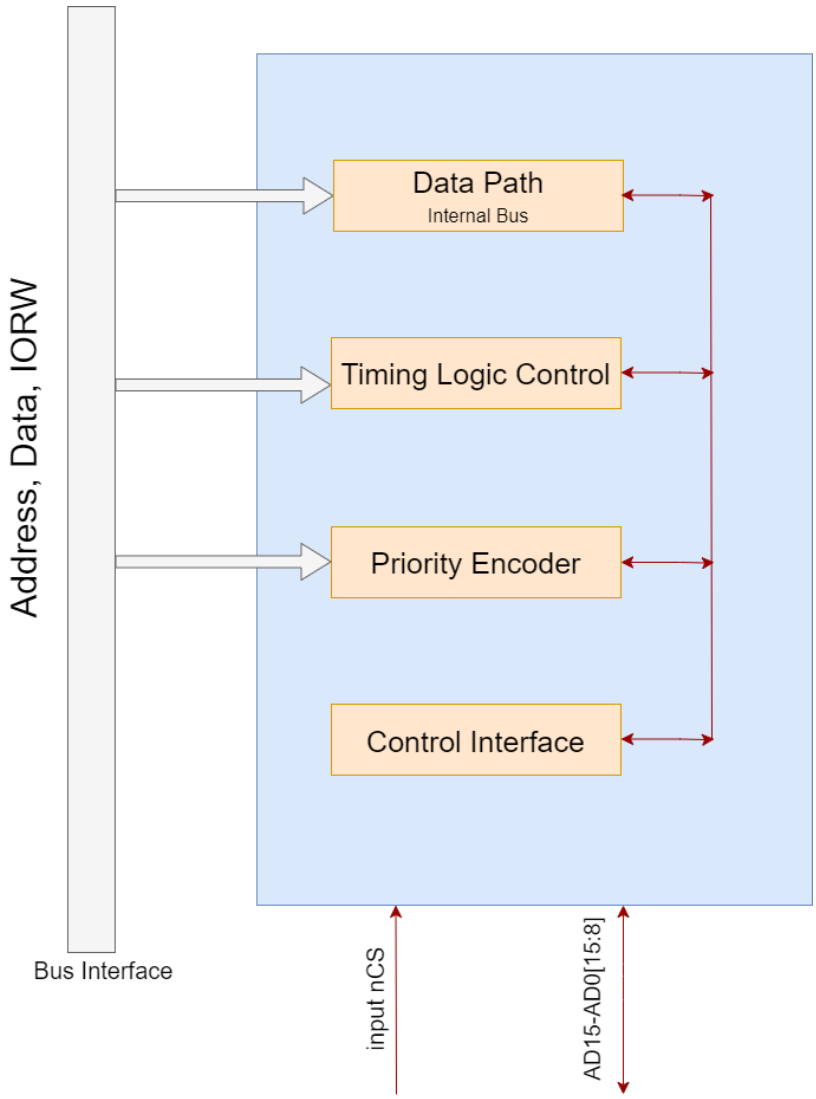
      The memory and I/o module diagram :




The Interface and Components:






2. Functions to be Verified:
    a. Critical Functions:
    - **Directed Tests**:
      Driving request signals (DREQ) from IO to DMA (DUT).
      Passing HLDA Signal to DMA.
    - **Random Tests**:
      Driving random DREQ signal (4 different channels).
      Randomizing CPU response time (delay).

    b. Secondary Functions:
      Checkers:

- Checking the channel priority and rotating priority (to verify proper order of DREQ).
- Checker for valid address.
- Independent polarity control check on DACK and DREQ.
- Enable and Disable Channels.
- End of Process check.

Assertion Based Checkers: -

- Verifying output signals from DUT (DMA) (AEN, ADSTB, DACK, IOR, IOW, MEMR. MEMW)
- Checking basic Handshake between peripheral devices and DMA.
    - DREQ and DACK
- Verifying transitions are not made during setup/hold time of READY signal.

## Priority Block:-

Test Scenarios for Priority Encoder (Arbiter)

1. Fixed Priority:

    - Programming 4 different Dreq request.
    - Verifying Command register bit [4] is set to 0.
    - Sending 2 dreq (say 2 and 3) and verify channel 2 gets the DACK.
    - Randomly sending **combination** of Dreq signals (16 combinations) and verifying highest priority channel gets the DACK and fixed priority is maintained for remaining requests in the sequence.
    - Randomly sending the **sequence** of three or four Dreq on each channel and verifying below scenarios:

    1. Highest priority channel gets the DACK and once the request is served the next priority dreq gets the DACK.

2. Checking for sequences like dreq (12310121) as its fixed priority all the requests on channel 0 should be served followed by all requests on channel 1 and then 2 and then 3.

- Verifying Dreq request is held high until Dack is received.
- When one of the channels gets the DACK verifying that the remaining channels are masked in Mask register.

2. Rotating Priority:

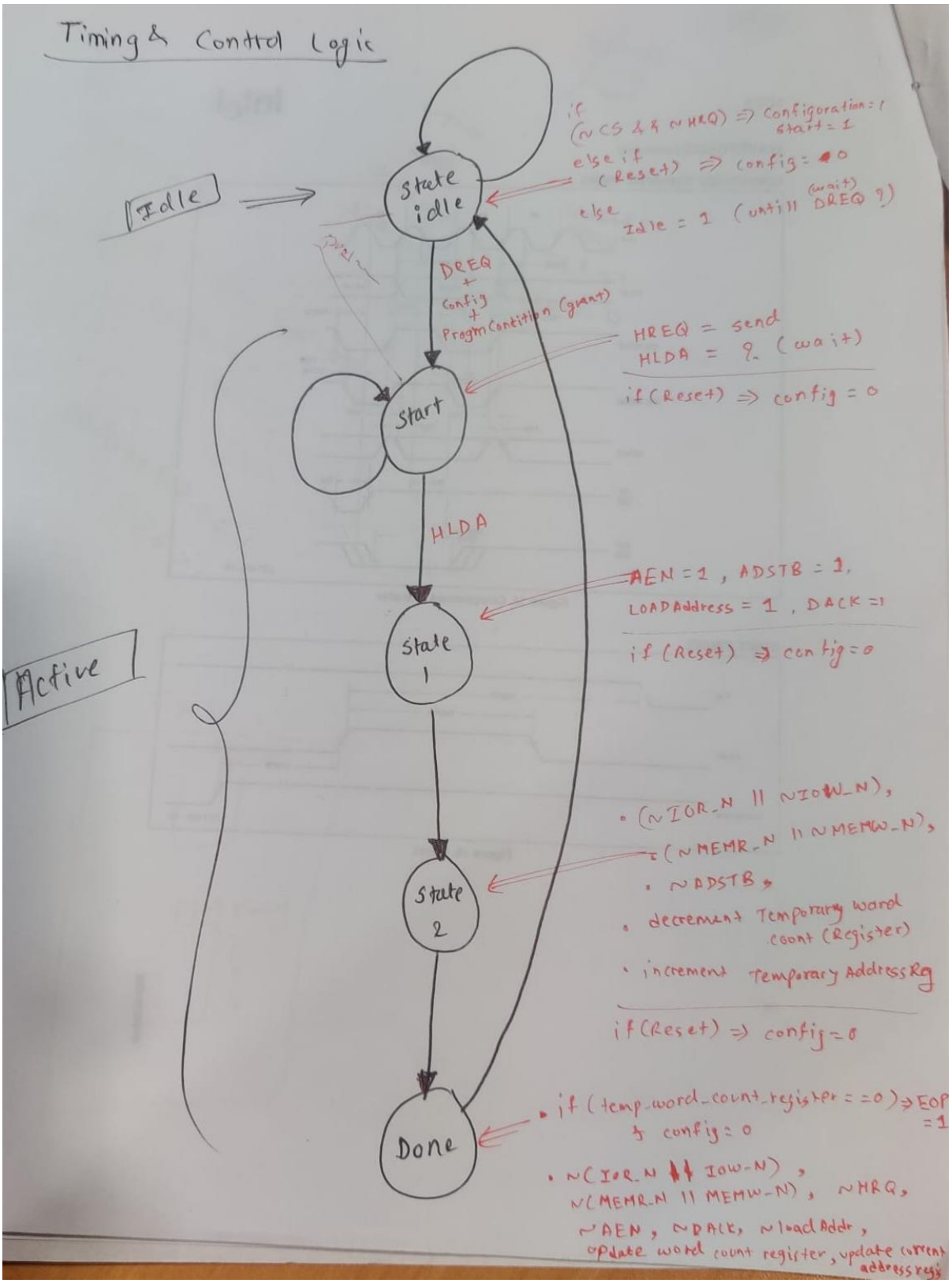
- Command Register bit [4] is set to 1.
- Sending 2 dreq (say 2 and 3) and verify channel 2 gets the DACK.
- Randomly sending **combination** of Dreq signals (16 combinations) and verifying highest priority channel gets the DACK and **Rotating priority** is maintained for remaining requests in the sequence.
- Randomly sending the **sequence** of three Dreq on each channel and verifying below scenarios:

1. Dreq0, dreq1,dreq2 asserted, DACK0 is served then DACK1 and DACK2.
2. Checking for sequences like dreq (12310121) DACK expected should be DACK1, DACK2, DACK3, DACK1, DACK0, DACK1, DACK2, DACK1.

## Timing Control FSM:-

The timing and control block has two modules for verification, one being the FSM for DMA Control signal generation and the assertions are grouped with the Memory Block to verify the Functionality

# Timing & Control Logic



c. Non-Verified Functions:
    - CPU response time (Delay)
    - FSM Timing and Control Block ( Not provided by design team)
    - The Individual Memory Cell Value Check.

3. Specific Tests and Methods:

←—**Bit Number**

| 7(MSB) | 6 | 5 | 4 | 3 | 2 | 1 | 0(LSB) |
|---|---|---|---|---|---|---|---|

The value of ‘i’ in below assertions can be 0,1,2,3. Used ‘i’ to indicate the assertion is valid for 4 channels of DMA.

- If DACK[i] and HLDA signals are asserted,the current address register of the channel ‘i’ should Increment if bit number 5 of mode register is 0, it should decrement if bit number 5 of mode register is 1.

  Assertion Failed:Current address register is not Incremented and Decremented during DMA transfer.

- If DACK[i], HLDA signals are asserted and the value in the word count register of the channel ‘i’ goes from 0 to FFFFH ,the terminal count should be generated and the bit number ‘i‘ of the status register should be set to 1.

  The Current word count is not Decrementing and Could not check if Status register is updated when word count goes from 0 to FFFFH

- When reset or master clear instruction is asserted , the command register and status register should be 0.

Assertion Passed.

- If controller disable(bit number 2 of Command Register) is 1, the control signals HRQ,HLDA,DACK[i] should not be asserted.

  Assertion Passed

- If DACK[i] is asserted and if its write transfer(bit number 3 and 2 of mode register is 0,1 respectively) , MEMW and IOR signals should be asserted in the next cycle.
  Assertion Passed

- If DACK[i] is asserted and if its read transfer(bit number 3 and 2 of mode register is 1,0 respectively) , MEMR and IOW signals should be asserted in the next cycle.
  Assertion Passed

- If DACK[i] is asserted ,word count is greater than 0 and 8 lower address bits(A7-A0) for channel ‘i’ is FFH , the ADSTB signal should be asserted.
  Assertion Passed

- If DACK[i] is asserted, the current word count register for channel ‘i’ should be decremented in each clock cycle until the value in the word count register goes from 0 to FFFFH.
  Assertion Failed: The current word count register is not decrementing.

- If DACK[i] is asserted and the value in the word count register of the channel ‘i’ goes from 0 to FFFFH ,the control signals HRQ,DACK and HLDA should be deasserted in the next cycle.

  Assertion Passed

4. Test Scenarios:

    a. System Level Test:

| Tests | Descriptions |
|---|---|
| Reset Test | When sif.Reset is asserted then all signals should reset |
| Write and Read from first Memory Location (write after read) | Single write and read to first memory location and check if the data is matched |
| Write and read from last memory location (write after read) | Single write and read to last memory location and check if the data is matched |
| Invalid address | Trying to access invalid memory location (not present in memory) |
| Reset after performing a certain operation on the memory | To check if memory retains its value on reset. |

## Verification Requirements:

1. Required Tools:

   The software simulation engine QuestaSim, which may be accessed via Portland State University's remote access Linux systems, is required for this project. System Verilog is used to write the design and verification code.

2. Risks and Dependencies:

   This project comprises RTL Design and Verification of Intel's 8237A DMA Controller, hence there is considerable risk of not finishing the task on time. The Verification team has had the disadvantage of terms of time and clarity as to what exactly is the design team strategy and planning for all the

blocks implemented, as no proper documentation was made or shared. The unit level testing is at high risk as the functionality hasn't been documented.
We do not have personal access to licensed software since it is quite expensive, thus the entire project is implemented on distant Linux servers.

## Resource Requirements:

For the RTL Design and Verification of the Intel's 8237A DMA Controller the requirements are, Project Guide- Prof. Mark Faust, Design Team- Gayatri, Satish, Aniruddh, Kirk and Verification Team- Saket, Kaavya, Manoj and Simulation software (QuestaSim) on the Linux system of Portland State University.

## Schedule Details:

| Date | Plan | Status |
|---|---|---|
| 04/28/2022 | Verification Plan | Done |
| 05/16/2022 | Basic Sanity Test | Done (Errors) |
| 05/24/2022 | Assertions | Worked on the it. |
| 06/08/2022 | Final Code and Documentation | Done |