ECE 571 – DMA Controller Test Plan

Draft 1

# STIMULUS

* **Sanity Test**:

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

* **Directed Tests**:

Driving request signals (DREQ) from IO to DMA (DUT).

Passing HLDA Signal to DMA.

CPU response time (Delay)

* **Random Tests**:

Driving random DREQ signal (4 different channels).

Randomizing CPU response time (delay).

# 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.

# Test Scenarios for Priority Encoder (Arbiter)

## 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.

## 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.

# Assertion Based Verification scenarios:

* In any cycle, there can be only one grant signal that can be asserted.
* Each requesting agent should get a grant signal in a maximum 4 cycle window.
* If one requesting agent gets a grant, it cannot receive another grant unless no other agent is requesting.
* If there are no requests, there cannot be any grants asserted