**ASD ASSIGNMENT – 2**

**Rohan Muthyala**

**1602-22-735-093**

Question : 1.Packet Generation:

Modify the code to generate 10 AXI packets instead of 5. Ensure the addresses and data values are still randomized.

2.Sorting Packets by ID:

Write a simulation to sort the axi\_packets dynamic array by packet IDs in ascending order and display the sorted packets.

3.Status Check:

Add functionality to check if a packet with ID 3 exists in the packet\_status associative array. If it exists, print its status; otherwise, mark it as "Not Found".

4. Packet Filtering:

Write a simulation to filter out packets from the axi\_packets array where the burst\_type is 0 (FIXED). Store the filtered packets in a new dynamic array and display them.

**Solution:**

Code:

module axi\_packet\_generator;

typedef struct packed {

logic [31:0] addr; // Address

logic [31:0] data; // Data

logic [3:0] id; // Packet ID

logic [1:0] burst\_type; // Burst Type (e.g., INCR, FIXED)

} axi\_packet\_t;

// Dynamic array for generated AXI packets

axi\_packet\_t axi\_packets[];

axi\_packet\_t filtered\_packets[];

// Associative array to track packet statuses (Key: Packet ID, Value: Status)

string packet\_status[int];

// Queue to simulate a pipeline of packets for processing

axi\_packet\_t processing\_queue[$];

int id = 0;

int filtered\_array\_count = 0;

initial begin

// Step 1: Generate 10 AXI packets

axi\_packets = new[10]; // Allocate space for 10 packets

for (int i = 0; i < axi\_packets.size(); i++) begin

axi\_packets[i].addr = $urandom\_range(0, 2\*\*32-1); // Random address

axi\_packets[i].data = $urandom\_range(0, 2\*\*32-1); // Random data

axi\_packets[i].id = i; // Packet ID

axi\_packets[i].burst\_type = $urandom\_range(0, 2); // Random burst type (0, 1, or 2)

$display("Generated Packet %0d - Addr: %h, Data: %h, ID: %0d, Burst: %0d",

i, axi\_packets[i].addr, axi\_packets[i].data, axi\_packets[i].id, axi\_packets[i].burst\_type);

end

// Step 2: Store packet statuses in an associative array

foreach (axi\_packets[i]) begin

packet\_status[axi\_packets[i].id] = "Pending"; // Initially mark all packets as "Pending"

end

// Display the packet statuses

$display("\nPacket Statuses:");

if (packet\_status.first(id) != 0) begin

do

begin

$display("Packet ID: %0d, Status: %s", id, packet\_status[id]);

end while(packet\_status.next(id));

end

// Step 3: Process packets using a queue

foreach (axi\_packets[i]) begin

processing\_queue.push\_back(axi\_packets[i]); // Add packets to the processing queue

end

// Process packets one by one (FIFO order)

$display("\nProcessing Packets:");

while (processing\_queue.size() != 0) begin

axi\_packet\_t packet;

packet = processing\_queue.pop\_front(); // Remove the first packet

packet\_status[packet.id] = "Processed"; // Update its status

$display("Processed Packet ID: %0d, Addr: %h, Data: %h", packet.id, packet.addr, packet.data);

end

// Step 4: Display final statuses of all packets

$display("\nFinal Packet Statuses:");

if (packet\_status.first(id) != 0) begin

do

begin

$display("Packet ID: %0d, Status: %s", id, packet\_status[id]);

end while(packet\_status.next(id));

end

// Check if a packet with ID 3 exists in the packet\_status associative array

if ( packet\_status . exists (3) ) begin

$display ( "\n\nPacket ID 3 exists in the associative array, details are as follows : " ) ;

$display ( "ID: %0d, Addr: %h, Data: %h, Burst: %0d", axi\_packets[3].id, axi\_packets[3].addr, axi\_packets[3].data, axi\_packets[3].burst\_type ) ;

end else begin

$display ( "Packet ID 3 not found in the array " ) ;

end

//check for packets with burst size zero

$display("\n\nPackets with burst size of zero : ");

for(int i = 0; i < axi\_packets.size(); i++) begin

if (axi\_packets[i].burst\_type == 0) begin

filtered\_packets = new[filtered\_array\_count + 1];

filtered\_packets[filtered\_array\_count] = axi\_packets[i];

$display("ID: %0d, Addr: %h, Data: %h, Burst: %0d", filtered\_packets[filtered\_array\_count].id, filtered\_packets[filtered\_array\_count].addr, filtered\_packets[filtered\_array\_count].data, filtered\_packets[filtered\_array\_count].burst\_type);

filtered\_array\_count++;

end

end

end

endmodule

Output:

# Generated Packet 0 - Addr: ce46aa23, Data: a01b6e32, ID: 0, Burst: 2  
# Generated Packet 1 - Addr: 8b4e9f1d, Data: a80385b8, ID: 1, Burst: 0  
# Generated Packet 2 - Addr: 13c11ac0, Data: e0a228ac, ID: 2, Burst: 2  
# Generated Packet 3 - Addr: 17e9241c, Data: a3ea30be, ID: 3, Burst: 0  
# Generated Packet 4 - Addr: 3fb8d2d8, Data: 88daf049, ID: 4, Burst: 2  
# Generated Packet 5 - Addr: 186e66d3, Data: 61968fd2, ID: 5, Burst: 1  
# Generated Packet 6 - Addr: 76d40342, Data: f731fc33, ID: 6, Burst: 2  
# Generated Packet 7 - Addr: caf5baf4, Data: caa56f5e, ID: 7, Burst: 1  
# Generated Packet 8 - Addr: 3e0f89da, Data: f3cd9f14, ID: 8, Burst: 1  
# Generated Packet 9 - Addr: 60370af1, Data: 158f2193, ID: 9, Burst: 0  
#   
# Packet Statuses:  
# Packet ID: 0, Status: Pending  
# Packet ID: 1, Status: Pending  
# Packet ID: 2, Status: Pending  
# Packet ID: 3, Status: Pending  
# Packet ID: 4, Status: Pending  
# Packet ID: 5, Status: Pending  
# Packet ID: 6, Status: Pending  
# Packet ID: 7, Status: Pending  
# Packet ID: 8, Status: Pending  
# Packet ID: 9, Status: Pending  
#   
# Processing Packets:  
# Processed Packet ID: 0, Addr: ce46aa23, Data: a01b6e32  
# Processed Packet ID: 1, Addr: 8b4e9f1d, Data: a80385b8  
# Processed Packet ID: 2, Addr: 13c11ac0, Data: e0a228ac  
# Processed Packet ID: 3, Addr: 17e9241c, Data: a3ea30be  
# Processed Packet ID: 4, Addr: 3fb8d2d8, Data: 88daf049  
# Processed Packet ID: 5, Addr: 186e66d3, Data: 61968fd2  
# Processed Packet ID: 6, Addr: 76d40342, Data: f731fc33  
# Processed Packet ID: 7, Addr: caf5baf4, Data: caa56f5e  
# Processed Packet ID: 8, Addr: 3e0f89da, Data: f3cd9f14  
# Processed Packet ID: 9, Addr: 60370af1, Data: 158f2193  
#   
# Final Packet Statuses:  
# Packet ID: 0, Status: Processed  
# Packet ID: 1, Status: Processed  
# Packet ID: 2, Status: Processed  
# Packet ID: 3, Status: Processed  
# Packet ID: 4, Status: Processed  
# Packet ID: 5, Status: Processed  
# Packet ID: 6, Status: Processed  
# Packet ID: 7, Status: Processed  
# Packet ID: 8, Status: Processed  
# Packet ID: 9, Status: Processed  
#   
#   
# Packet ID 3 exists in the associative array, details are as follows :   
# ID: 3, Addr: 17e9241c, Data: a3ea30be, Burst: 0  
#   
#   
# Packets with burst size of zero :   
# ID: 1, Addr: 8b4e9f1d, Data: a80385b8, Burst: 0  
# ID: 3, Addr: 17e9241c, Data: a3ea30be, Burst: 0  
# ID: 9, Addr: 60370af1, Data: 158f2193, Burst: 0  
# exit  
# End time: 13:23:33 on Mar 19,2025, Elapsed time: 0:00:01  
# Errors: 0, Warnings: 0  
End time: 13:23:33 on Mar 19,2025, Elapsed time: 0:00:02