CpE 186 Computer Hardware Design
Professor: Dr. Jing Pang

Project 2 Report
PCIe Virtual Channel Arbiter with Low-and High-Priority Implementations

Names: Alexis Ho, Andrew Peterson, Jayven Khoonsirivong, Jeb Chua, Mario Palacios
California State University, Sacramento

Table of Contents

Introduction *3*

Virtual Channel Arbiter *3*

Design Purpose **3**

Engineering Data **3**

Verilog Design **4**

Verilog Testbench **11**

Results Discussion **12**

Conclusion *12*

Introduction

The PCI arbiter provides arbitration for two to eight PCI master agents, and it follows a rotating (round robin) or a fixed (strict) priority scheme to provide fair access to the PCI bus. In a rotating priority scheme, the requestor that is most recently granted the bus receives the lowest priority, while the requestor position next to it receives the highest priority. In a fixed priority scheme, lower priority transactions are starved for bandwidth and have longer latencies, but the higher priorities receive very high bandwidth with minimal latencies. Using a virtual channel arbitration ensures that forwards progress is made for all transactions and prevents inadvertent split transaction time-outs.

Virtual Channel Arbiter

Design Purpose

The purpose of this project is to grow and develop an understanding of PCIe virtual channel arbiters, which are universally use in hardware design. The implementation would take in an 8bit segment where each are represented with a VC ID. The first 3bits are high-priority bits which follow a strict priority scheme. While the remaining 5bits are low-priority and follow a round robin priority scheme, all this can be seen in the figure below.

Engineering Data

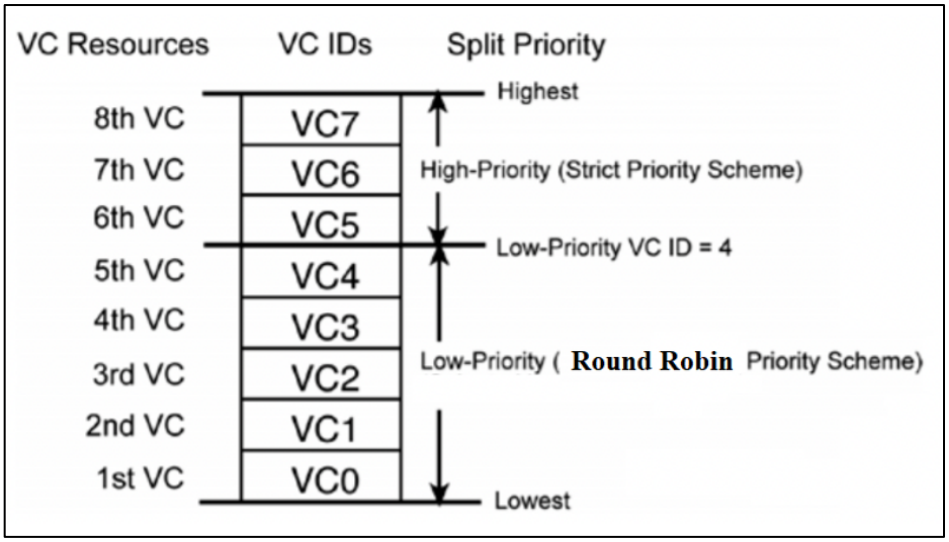

Figure 1. Virtual Channel Arbitration Diagram

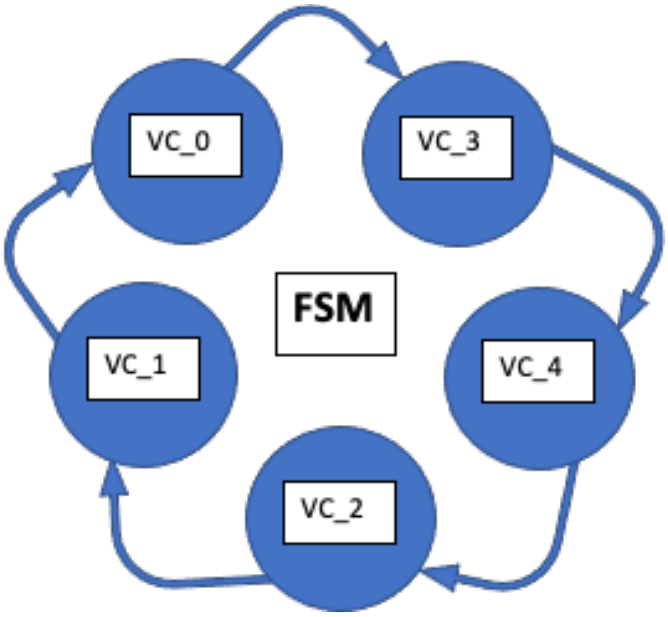

Figure 2. FSM Diagram for Round Robin Priority Scheme (Low – Priority)

Verilog Design

```
`timescale 1ns / 1ps
module arbiter(clk,clr,PCIe,id);
	input 			clk, clr;
	input	[7:0] PCIe;
	output reg [7:0] id;

	reg [1:0] cs, ns;
	reg [3:0] vc, nvc;
	reg [3:0] state = 0;
	reg v4=0, v3=0, v2=0, v1=0, v0=0;

	parameter vc7=4'b0111, vc6=4'b0110, vc5=4'b0101, vc4=4'b0100,
			vc3=4'b0011, vc2=4'b0010, vc1=4'b0001, vc0=4'b0000,
			idle=4'b1000, start=4'b1111;

	always @ (posedge clk or negedge clr)
	if(clr)
	begin
		cs <= 2'b00;
		vc <= start;
	end
	else
	begin
		cs <= ns;
```

```
        vc <= nvc;
    end

    //FSM
    always @ (*)
    begin
        case(vc)
            vc7:
                begin
                    id = 8'b10000000;
                    if (PCIe[6])
                        nvc <= vc6;
                    else if (!PCIe[6] && PCIe[5])
                        nvc <= vc5;
                    else
                    begin
                        if (state == 0)
                        begin
                            if (PCIe[4]) nvc <= vc4;
                            else if (PCIe[3]) nvc <= vc3;
                            else if (PCIe[2]) nvc <= vc2;
                            else if (PCIe[1]) nvc <= vc1;
                            else if (PCIe[0]) nvc <= vc0;
                            else nvc <= idle;
                        end
                        if (state == 1)
                        begin
                            if (PCIe[3]) nvc <= vc3;
                            else if (PCIe[2]) nvc <= vc2;
                            else if (PCIe[1]) nvc <= vc1;
                            else if (PCIe[0]) nvc <= vc0;
                            else nvc <= idle;
                        end
                        if (state == 2)
                        begin
                            if (PCIe[2]) nvc <= vc2;
                            else if (PCIe[1]) nvc <= vc1;
                            else if (PCIe[0]) nvc <= vc0;
                            else nvc <= idle;
                        end
                        if (state == 3)
                        begin
                            if (PCIe[1]) nvc <= vc1;
                            else if (PCIe[0]) nvc <= vc0;
```

```
                                else nvc <= idle;
                        end
                        else
                        begin
                                if (PCIe[0]) nvc <= vc0;
                                else nvc <= idle;
                        end
                end
        end
vc6:
        begin
                id = 8'b01000000;
                if (PCIe[5])
                        nvc <= vc5;
                else
                begin
                        if (state == 0)
                        begin
                                if(PCIe[4]) nvc <= vc4;
                                else if (PCIe[3]) nvc <= vc3;
                                else if (PCIe[2]) nvc <= vc2;
                                else if (PCIe[1]) nvc <= vc1;
                                else if (PCIe[0]) nvc <= vc0;
                                else nvc <= idle;
                        end
                        if (state == 1)
                        begin
                                if(PCIe[3]) nvc <= vc3;
                                else if (PCIe[2]) nvc <= vc2;
                                else if (PCIe[1]) nvc <= vc1;
                                else if (PCIe[0]) nvc <= vc0;
                                else nvc <= idle;
                        end
                        if (state == 2)
                        begin
                                if(PCIe[2]) nvc <= vc2;
                                else if (PCIe[1]) nvc <= vc1;
                                else if (PCIe[0]) nvc <= vc0;
                                else nvc <= idle;
                        end
                        if (state == 3)
                        begin
                                if(PCIe[1]) nvc <= vc1;
                                else if (PCIe[0]) nvc <= vc0;
```

```
                                else nvc <= idle;
                        end
                        else
                        begin
                                if (PCIe[0]) nvc <= vc0;
                                else nvc <= idle;
                        end
                end
        end
vc5:
        begin
                id = 8'b00100000;
                if (state == 0)
                begin
                        if (PCIe[4]) nvc <= vc4;
                        else if (PCIe[3]) nvc <= vc3;
                        else if (PCIe[2]) nvc <= vc2;
                        else if (PCIe[1]) nvc <= vc1;
                        else if (PCIe[0]) nvc <= vc0;
                        else nvc <= idle;
                end
                if (state == 1)
                begin
                        if (PCIe[3]) nvc <= vc3;
                        else if (PCIe[2]) nvc <= vc2;
                        else if (PCIe[1]) nvc <= vc1;
                        else if (PCIe[0]) nvc <= vc0;
                        else nvc <= idle;
                end
                if (state == 2)
                begin
                        if (PCIe[2]) nvc <= vc2;
                        else if (PCIe[1]) nvc <= vc1;
                        else if (PCIe[0]) nvc <= vc0;
                        else nvc <= idle;
                end
                if (state == 3)
                begin
                        if (PCIe[1]) nvc <= vc1;
                        else if (PCIe[0]) nvc <= vc0;
                        else nvc <= idle;
                end
                else
                begin
```

```
                        if(PCIe[0]) nvc <= vc0;
                        else nvc <= idle;
                end
        end
vc4:
        begin
                id = 8'b00010000;
                v4 = 1'b1;
                if (state == 1) nvc <= idle;
                else
                begin
                        if (PCIe[3]) nvc <= vc3;
                        else if (PCIe[2])
                        begin
                                if (v2 == 1'b0)
                                begin
                                        nvc <= vc2;
                                end
                                else nvc <= idle;
                        end
                        else if (PCIe[1])
                        begin
                                if (v1 <= 1'b0)
                                begin
                                        nvc <= vc1;
                                end
                                else nvc <= idle;
                        end
                        else nvc <= idle;
                end
        end
vc3:
        begin
                id = 8'b00001000;
                v3 = 1'b1;
                if (state == 2) nvc <= idle;
                else
                begin
                        if (PCIe[2]) nvc <= vc2;
                        else if (PCIe[1]) nvc <= vc1;
                        else if (PCIe[0]) nvc <= vc0;
                        else nvc <= idle;
                end
        end
```

```
vc2:
        begin
                id = 8'b00000100;
                v2 = 1'b1;
                if (state == 3) nvc <= idle;
                else
                begin
                        if (PCIe[1]) nvc <= vc1;
                        else if (PCIe[0]) nvc <= vc0;
                        else nvc <= idle;
                end
        end
vc1:
        begin
                id = 8'b00000010;
                v1 = 1'b1;
                if (state == 4) nvc <= idle;
                else
                begin
                        if (PCIe[0]) nvc <= vc0;
                        else
                        begin
                                if (PCIe[4])
                                begin
                                        if (v4 == 1'b0)
                                        begin
                                                nvc <= vc4;
                                        end
                                        else nvc <= idle;
                                end
                                else nvc <= idle;
                        end
                end
        end
vc0:
        begin
                id = 8'b00000001;
                v0 = 1'b1;
                if (state == 0) nvc <= idle;
                else
                begin
                        if (PCIe[4]) nvc <= vc4;
                        else nvc <= idle;
                end
```

```
            end
idle:
            begin
                        id = 8'b00000000;
                        v4=1'b0; v3=1'b0; v2=1'b0; v1=1'b0; v0=1'b0;
                        if (state == 5) state = 0;
                        else
                                    state = state + 1;
                                    nvc <= start;
            end
start:
            begin
                        id = 8'b00000000;
                        if (PCIe[7]) nvc <= vc7;
                        else if (PCIe[6]) nvc <= vc6;
                        else if (PCIe[5]) nvc <= vc5;
                        else
                        begin
                                    if (state == 0)
                                    begin
                                                if (PCIe[4]) nvc <= vc4;
                                                else if (PCIe[3]) nvc <= vc3;
                                                else if (PCIe[2]) nvc <= vc2;
                                                else if (PCIe[1]) nvc <= vc1;
                                                else if (PCIe[0]) nvc <= vc0;
                                                else nvc <= start;
                                    end
                                    else if (state == 1)
                                    begin
                                                if (PCIe[3]) nvc <= vc3;
                                                else if (PCIe[2]) nvc <= vc2;
                                                else if (PCIe[1]) nvc <= vc1;
                                                else if (PCIe[0]) nvc <= vc0;
                                                else nvc <= start;
                                    end
                                    else if (state == 2)
                                    begin
                                                if (PCIe[2]) nvc <= vc2;
                                                else if (PCIe[1]) nvc <= vc1;
                                                else if (PCIe[0]) nvc <= vc0;
                                                else nvc <= start;
                                    end
                                    else if (state == 3)
                                    begin
```

```
                                        if (PCIe[1]) nvc <= vc1;
                                        else if (PCIe[0]) nvc <= vc0;
                                        else nvc <= start;
                                    end
                                    else
                                    begin
                                        if (PCIe[0]) nvc <= vc0;
                                        else nvc <= start;
                                    end
                                end
                            end
                        default:
                            id = 8'b00000000;
                endcase
        end
endmodule
```

Verilog Testbench

```
`include "arbiter.v"
module arbiter_tb;
        reg clk, clr;
        reg [7:0] PCIe;
        wire [7:0] id;

initial
        $monitor ($time, " Reset = %b, PCIe = %b, VC ID = %b", clr, PCIe, id);

        arbiter arbiter_instant (.clk(clk), .clr(clr), .PCIe(PCIe), .id(id));

        initial begin
           clk = 0;
           forever #5 clk = ~clk;
        end

        initial begin
                clr=1; PCIe=8'b10100110;
           #10;   clr=0;
           #60;   PCIe=8'b00000110;
           #40;   PCIe=8'b00001110;
           #20;   PCIe=8'b01001110;
           #20;   PCIe=8'b00001110;
           #50    $finish;
        end
endmodule
```

Results Discussion

```
  0 Reset = 1, PCIe = 10100110, VC ID = 00000000
 10 Reset = 0, PCIe = 10100110, VC ID = 10000000
 15 Reset = 0, PCIe = 10100110, VC ID = 00100000
 25 Reset = 0, PCIe = 10100110, VC ID = 00000000
 45 Reset = 0, PCIe = 10100110, VC ID = 10000000
 55 Reset = 0, PCIe = 10100110, VC ID = 00100000
 65 Reset = 0, PCIe = 10100110, VC ID = 00000000
 70 Reset = 0, PCIe = 00000110, VC ID = 00000000
 85 Reset = 0, PCIe = 00000110, VC ID = 00000010
 95 Reset = 0, PCIe = 00000110, VC ID = 00000000
110 Reset = 0, PCIe = 00001110, VC ID = 00000000
130 Reset = 0, PCIe = 01001110, VC ID = 00000000
135 Reset = 0, PCIe = 01001110, VC ID = 01000000
145 Reset = 0, PCIe = 01001110, VC ID = 00000000
150 Reset = 0, PCIe = 00001110, VC ID = 00000000
165 Reset = 0, PCIe = 00001110, VC ID = 00001000
175 Reset = 0, PCIe = 00001110, VC ID = 00000100
185 Reset = 0, PCIe = 00001110, VC ID = 00000010
195 Reset = 0, PCIe = 00001110, VC ID = 00000000
```

Figure 3. Results from Arbiter Testbench

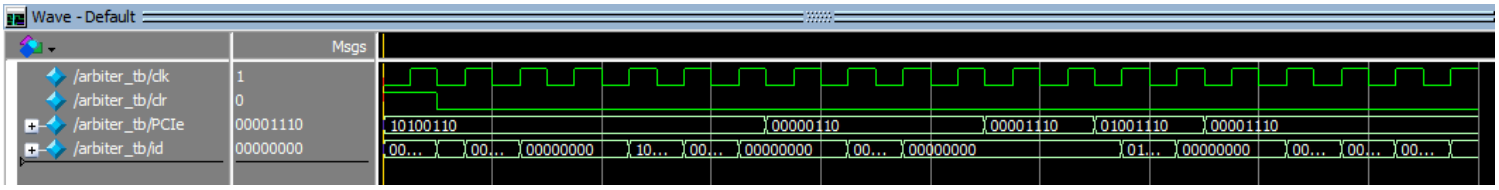

Figure 4. Waveform from Arbiter Testbench

We had planned out which numbers we wanted to use and then what the output should be, then we ran the file on Modelsim because Synopsys was giving us trouble. As you can see on figure 3, at clock signal 70, the bits [7:5] were zeroed since those are prioritized and the arbiter will not process [4:0] until [7:5] are take care of. Then at clock signal 130, bit 6 was added again to show that the arbiter then focuses on that bit until it is removed again.

Conclusion

In conclusion, we can see how the PCI arbiter performs arbitration using the round-robin method. It was essential for us to create the FSM so the correct VC ID would be displayed. At first this project was difficult because there was no code to help us understand what was going on, but the knowledge gained form the book and Dr. Pang, we were able to understand exactly how it should work. Overall, this project was very fun and challenging enough for us to get learn outside of class and use our resources. In the end we believe we were able to complete the project to our best understanding.