Chase Lotito
ECE426 - VLSI Design
March 22, 2024

HW6

Question 1:

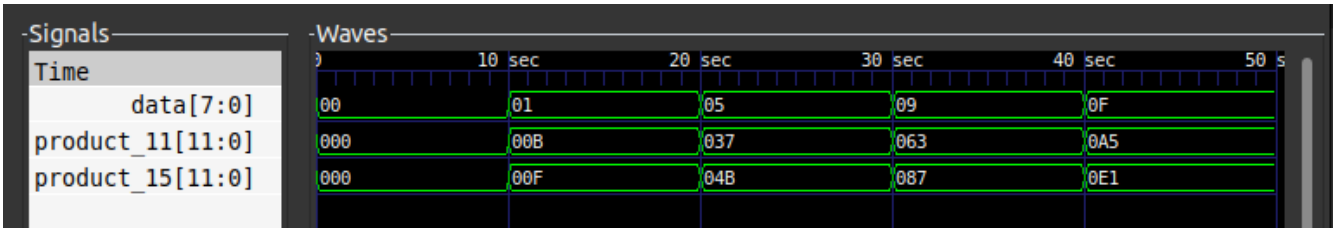
q1.v

```
// Chase Lotito - SIUC - SP2024
// ECE426 - HW6 - Q1

module multiply(data, product_11, product_15);

input [7:0] data;
output [11:0] product_11;
output [11:0] product_15;
wire [11:0] product_11;
wire [11:0] product_15;

assign product_11 = 8*data + 2*data + data;
assign product_15 = 8*data + 4*data + 2*data + data;

endmodule
```

tb1.v

```
// Chase Lotito - HW6 Q1 Testbench

`include "q1.v"

module tb();

initial begin : GTKWAVE
    $dumpfile("q1.vcd");
    $dumpvars(0, tb);
end

// I/O
reg [7:0] data;
wire [11:0] product_11;
wire [11:0] product_15;

initial begin : stimmy
    data = 0;
```

```
    #10 data = 1;
    #10 data = 5;
    #10 data = 9;
    #10 data = 15;
    #10 $finish;
end

// intialize multiplier
multiply U1 (
    .data(data),
    .product_11(product_11),
    .product_15(product_15)
);

endmodule
```



Question 2:

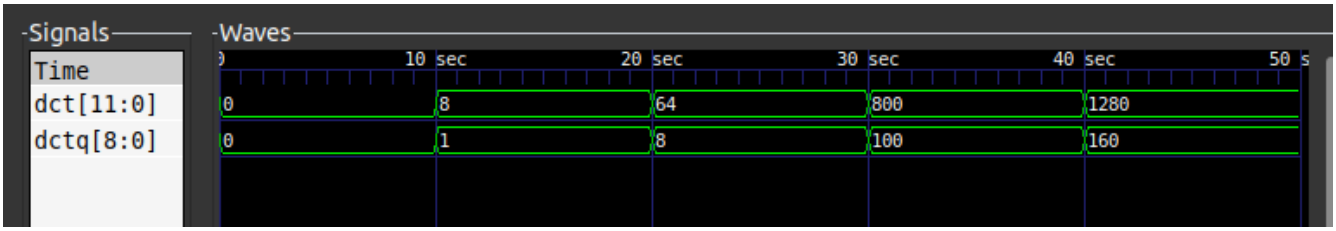
q2.v

```
// Chase Lotito - SIUC - SP2024
// ECE426 - HW6 - Q2

module a5_9(dct, dctq);

input [11:0] dct;
output [8:0] dctq;
wire [8:0] dctq;

assign dctq = dct / 8;      // quantize

endmodule
```

tb2.v

```
// Chase Lotito - HW6 Q2 Testbench

`include "q2.v"
```

```
module tb();

initial begin : GTKWAVE
    $dumpfile("q2.vcd");
    $dumpvars(0, tb);
end

// I/O
reg [11:0] dct;
wire [8:0] dctq;

initial begin : stimmy
    dct = 0;
    #10 dct = 8;
    #10 dct = 64;
    #10 dct = 800;
    #10 dct = 1280;
    #10 $finish;
end

// intialize
a5_9 U1 (
    .dct(dct),
    .dctq(dctq)
);

endmodule
```



Question 3:

q3.v

```
// Chase Lotito - SIUC - SP2024
// ECE426 - HW6 - Q3

module tff (
```

```
    clk, reset_n, sel, q
);

// I/O
input clk, reset_n, sel;
output q;
reg q;
wire d;

initial q = 1'b0;   // need to prevent unknown

assign d = sel ? !q : q;

always @ (posedge clk or negedge reset_n) begin
    if(reset_n == 1'b0)
        q = 1'b0;
    else
        q = d;
end

endmodule
```

tb3.v

```
// Chase Lotito - HW6 Q1 Testbench

`include "q3.v"

module tb();

initial begin : GTKWAVE
    $dumpfile("q3.vcd");
    $dumpvars(0, tb);
end

// I/O
reg clk, reset_n, sel;
wire q;

initial begin : stimmy
    clk = 1'b0;
    reset_n = 1'b1;
    sel = 1'b0;
```

```
    repeat(15)
        #15 sel = ~sel;

    reset_n = 1'b0;

    repeat(15)
        #15 sel = ~sel;

    #10 $finish;
end

always #10 clk = ~clk;

// intialize
tff U1 (
    .clk(clk),
    .reset_n(reset_n),
    .sel(sel),
    .q(q)
);

endmodule
```