VLSI Signal Processing Class Assignment

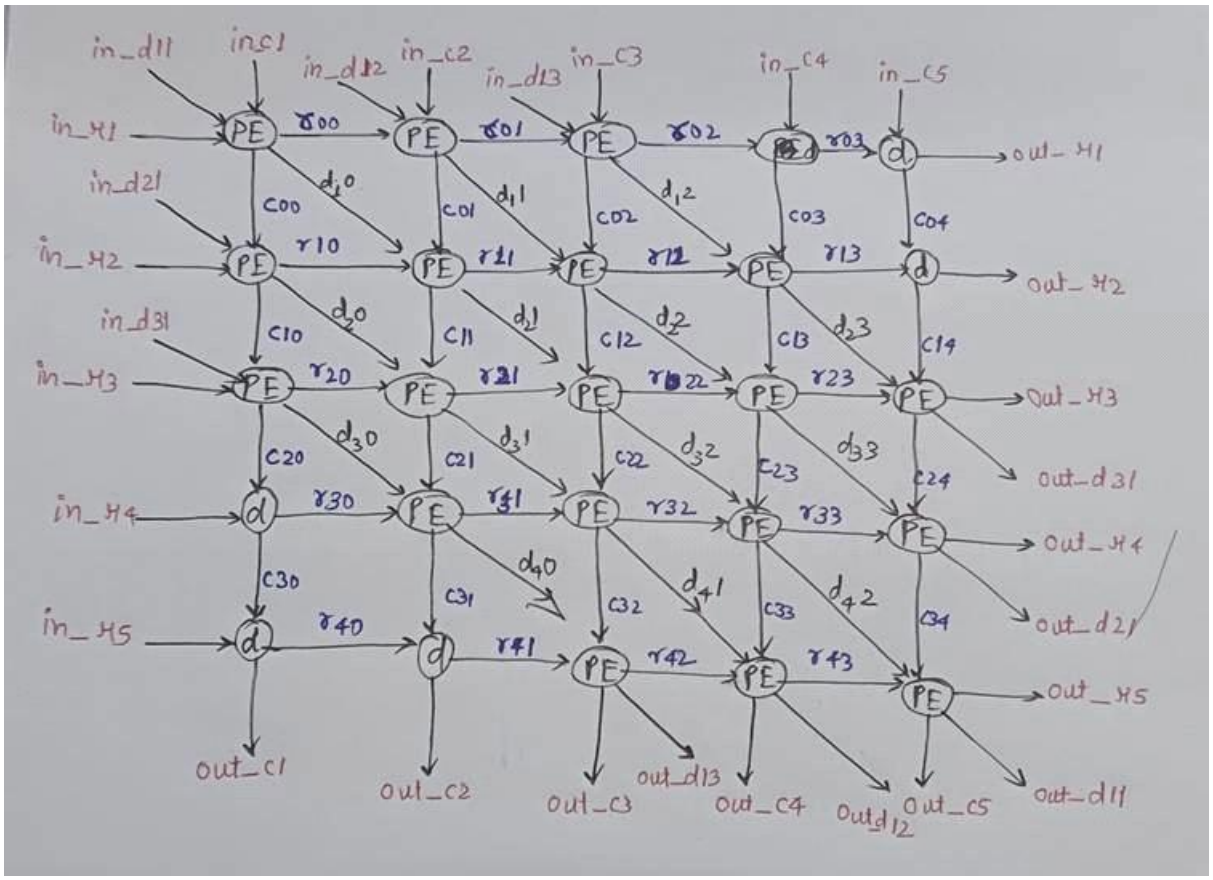
Systolic Multiplication

Submitted by:

Saurabh Kumar

SC22B146

Systolic array:



Verilog source code:

```
`timescale 1ns / 1ps


// delay element
module d(clk, in1, in2, out1, out2);
   input clk;
```

```
    input [7:0] in1, in2;
    output reg [7:0] out1, out2;

    always @(posedge clk) begin
        out1 <= in1;
        out2 <= in2;
    end

endmodule

// processing element
module pe(clk, in1, in2, in3, out1, out2, out3);
    input clk;
    input [7:0] in1, in2, in3;
    output reg [7:0] out1, out2, out3;

    always @(posedge clk) begin
        out1 <= in1;
        out2 <= in2;
        out3 <= (in1 * in2) + in3;
    end
endmodule

// systolic array
module array(clk, in_c1, in_c2, in_c3, in_c4, in_c5,
                in_r1, in_r2, in_r3, in_r4, in_r5,
                in_d11, in_d12, in_d13, in_d21, in_d31,
                out_c1, out_c2, out_c3, out_c4, out_c5,
                out_r1, out_r2, out_r3, out_r4, out_r5,
                out_d11, out_d12, out_d13, out_d21, out_d31);
    input clk;
    input [7:0] in_c1, in_c2, in_c3, in_c4, in_c5,
                in_r1, in_r2, in_r3, in_r4, in_r5,
                in_d11, in_d12, in_d13, in_d21, in_d31;
    output [7:0] out_c1, out_c2, out_c3, out_c4, out_c5,
                out_r1, out_r2, out_r3, out_r4, out_r5,
                out_d11, out_d12, out_d13, out_d21, out_d31;
```

```
wire [7:0] wire_c[0:3][0:4];
wire [7:0] wire_r[0:4][0:3];
wire [7:0] wire_d1[0:2];
wire [7:0] wire_d2[0:3];
wire [7:0] wire_d3[0:3];
wire [7:0] wire_d4[0:2];

// row1
pe pe11(clk, in_c1, in_r1, in_d11, wire_c[0][0], wire_r[0][0], wire_d1[0]);
pe pe12(clk, in_c2, wire_r[0][0], in_d12, wire_c[0][1], wire_r[0][1], wire_d1[1]);
pe pe13(clk, in_c3, wire_r[0][1], in_d13, wire_c[0][2], wire_r[0][2], wire_d1[2]);
d d14(clk, in_c4, wire_r[0][2], wire_c[0][3], wire_r[0][3]);
d d15(clk, in_c5, wire_r[0][3], wire_c[0][4], out_r1);

// row2
pe pe21(clk, wire_c[0][0], in_r2, in_d21, wire_c[1][0], wire_r[1][0], wire_d2[0]);
pe pe22(clk, wire_c[0][1], wire_r[1][0], wire_d1[0], wire_c[1][1], wire_r[1][1], wire_d2[1]);
pe pe23(clk, wire_c[0][2], wire_r[1][1], wire_d1[1], wire_c[1][2], wire_r[1][2], wire_d2[2]);
pe pe24(clk, wire_c[0][3], wire_r[1][2], wire_d1[2], wire_c[1][3], wire_r[1][3], wire_d2[3]);
d d25(clk, wire_c[0][4], wire_r[1][3], wire_c[1][4], out_r2);

// row3
pe pe31(clk, wire_c[1][0], in_r3, in_d11, wire_c[2][0], wire_r[2][0], wire_d3[0]);
pe pe32(clk, wire_c[1][1], wire_r[2][0], wire_d2[0], wire_c[2][1], wire_r[2][1], wire_d3[1]);
pe pe33(clk, wire_c[1][2], wire_r[2][1], wire_d2[1], wire_c[2][2], wire_r[2][2], wire_d3[2]);
pe pe34(clk, wire_c[1][3], wire_r[2][2], wire_d2[2], wire_c[2][3], wire_r[2][3], wire_d3[3]);
pe pe35(clk, wire_c[1][4], wire_r[2][3], wire_d2[3], wire_c[2][4], out_r3, out_d31 );

// row4
d d41(clk, wire_c[2][0], in_r4, wire_c[3][0], wire_r[3][0]);
pe pe42(clk, wire_c[2][1], wire_r[3][0], wire_d3[0], wire_c[3][1], wire_r[3][1], wire_d4[0]);
pe pe43(clk, wire_c[2][2], wire_r[3][1], wire_d3[1], wire_c[3][2], wire_r[3][2], wire_d4[1]);
pe pe44(clk, wire_c[2][3], wire_r[3][2], wire_d3[2], wire_c[3][3], wire_r[3][3], wire_d4[2]);
pe pe45(clk, wire_c[2][4], wire_r[3][3], wire_d3[3], wire_c[3][4], out_r4, out_d21);

// row5
```

```
    d d51(clk, wire_c[3][0], in_r5, out_c1, wire_r[4][0]);
    d d52(clk, wire_c[3][1], wire_r[4][0], out_c2, wire_r[4][1]);
    pe pe53(clk, wire_c[3][2], wire_r[4][1], wire_d4[0], out_c3, wire_r[4][2], out_d13);
    pe pe54(clk, wire_c[3][3], wire_r[4][2], wire_d4[1], out_c4, wire_r[4][3], out_d12);
    pe pe55(clk, wire_c[3][4], wire_r[4][3], wire_d4[2], out_c5, out_r5, out_d11);
endmodule
```

Verilog TestBench:

```
`timescale 1ns / 1ps

module dut;
    reg clk;
    reg [7:0] in_c1, in_c2, in_c3, in_c4, in_c5,
              in_r1, in_r2, in_r3, in_r4, in_r5,
              in_d11, in_d12, in_d13, in_d21, in_d31;
    wire [7:0] out_c1, out_c2, out_c3, out_c4, out_c5,
               out_r1, out_r2, out_r3, out_r4, out_r5,
               out_d11, out_d12, out_d13, out_d21, out_d31;

    array array1(clk, in_c1, in_c2, in_c3, in_c4, in_c5,
                 in_r1, in_r2, in_r3, in_r4, in_r5,
                 in_d11, in_d12, in_d13, in_d21, in_d31,
                 out_c1, out_c2, out_c3, out_c4, out_c5,
                 out_r1, out_r2, out_r3, out_r4, out_r5,
                 out_d11, out_d12, out_d13, out_d21, out_d31);

    initial begin
        clk <= 0;

        in_c1 <= 8'h2;
        in_c2 <= 8'h3;
        in_c3 <= 8'h4;
        in_c4 <= 8'h0;
        in_c5 <= 8'h0;

        in_r1 <= 8'h1;
```

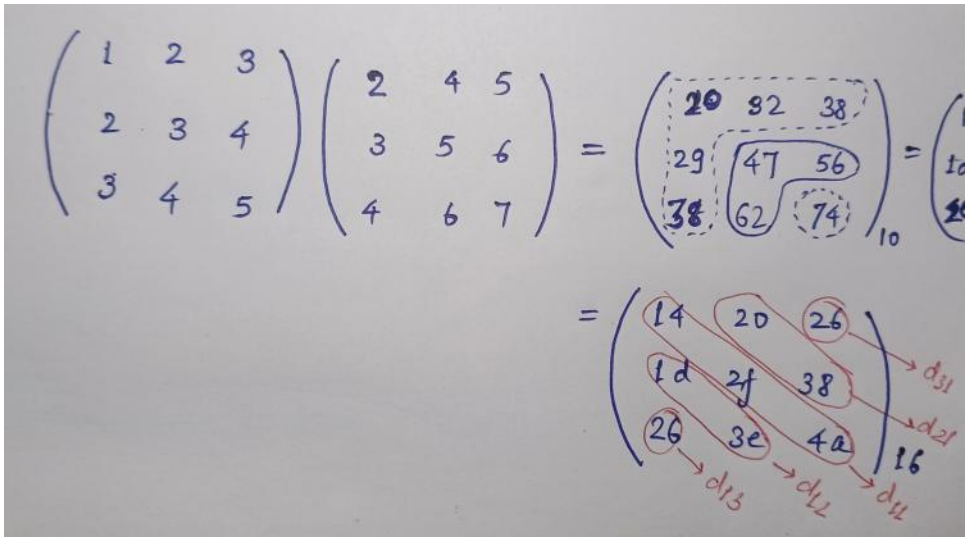
```
    in_r2 <= 8'h2;
    in_r3 <= 8'h3;
    in_r4 <= 8'h0;
    in_r5 <= 8'h0;

    in_d11 <= 8'h0;
    in_d12 <= 8'h0;
    in_d13 <= 8'h0;
    in_d21 <= 8'h0;
    in_d31 <= 8'h0;
end

always begin
    #5 clk <= ~clk;
end

always @(posedge clk) begin
    #5      in_c1 <= 8'h0;          #20   in_c1 <= 8'h0;          #35   in_c1 <= 8'h0;
            in_c2 <= 8'h4;                in_c2 <= 8'h0;                in_c2 <= 8'h0;
            in_c3 <= 8'h5;                in_c3 <= 8'h0;                in_c3 <= 8'h0;
            in_c4 <= 8'h6;                in_c4 <= 8'h0;                in_c4 <= 8'h0;
            in_c5 <= 8'h0;                in_c5 <= 8'h0;                in_c5 <= 8'h0;

            in_r1 <= 8'h0;                in_r1 <= 8'h0;                in_r1 <= 8'h0;
            in_r2 <= 8'h2;                in_r2 <= 8'h0;                in_r2 <= 8'h0;
            in_r3 <= 8'h3;                in_r3 <= 8'h0;                in_r3 <= 8'h0;
            in_r4 <= 8'h4;                in_r4 <= 8'h0;                in_r4 <= 8'h0;
            in_r5 <= 8'h0;                in_r5 <= 8'h0;                in_r5 <= 8'h0;

            in_d11 <= 8'h0;               in_d11 <= 8'h0;               in_d11 <= 8'h0;
            in_d12 <= 8'h0;               in_d12 <= 8'h0;               in_d12 <= 8'h0;
            in_d13 <= 8'h0;               in_d13 <= 8'h0;               in_d13 <= 8'h0;
            in_d21 <= 8'h0;               in_d21 <= 8'h0;               in_d21 <= 8'h0;
            in_d31 <= 8'h0;               in_d31 <= 8'h0;               in_d31 <= 8'h0;

    #10   in_c1 <= 8'h0;          #25   in_c1 <= 8'h0;          #40   in_c1 <= 8'h0;
            in_c2 <= 8'h0;                in_c2 <= 8'h0;                in_c2 <= 8'h0;
```

```
        in_c3 <= 8'h5;
        in_c4 <= 8'h6;
        in_c5 <= 8'h7;

        in_r1 <= 8'h0;
        in_r2 <= 8'h0;
        in_r3 <= 8'h3;
        in_r4 <= 8'h4;
        in_r5 <= 8'h5;

        in_d11 <= 8'h0;
        in_d12 <= 8'h0;
        in_d13 <= 8'h0;
        in_d21 <= 8'h0;
        in_d31 <= 8'h0;

#15     in_c1 <= 8'h0;
        in_c2 <= 8'h0;
        in_c3 <= 8'h0;
        in_c4 <= 8'h0;
        in_c5 <= 8'h0;

        in_r1 <= 8'h0;
        in_r2 <= 8'h0;
        in_r3 <= 8'h0;
        in_r4 <= 8'h0;
        in_r5 <= 8'h0;

        in_d11 <= 8'h0;
        in_d12 <= 8'h0;
        in_d13 <= 8'h0;
        in_d21 <= 8'h0;
        in_d31 <= 8'h0;

    end
```

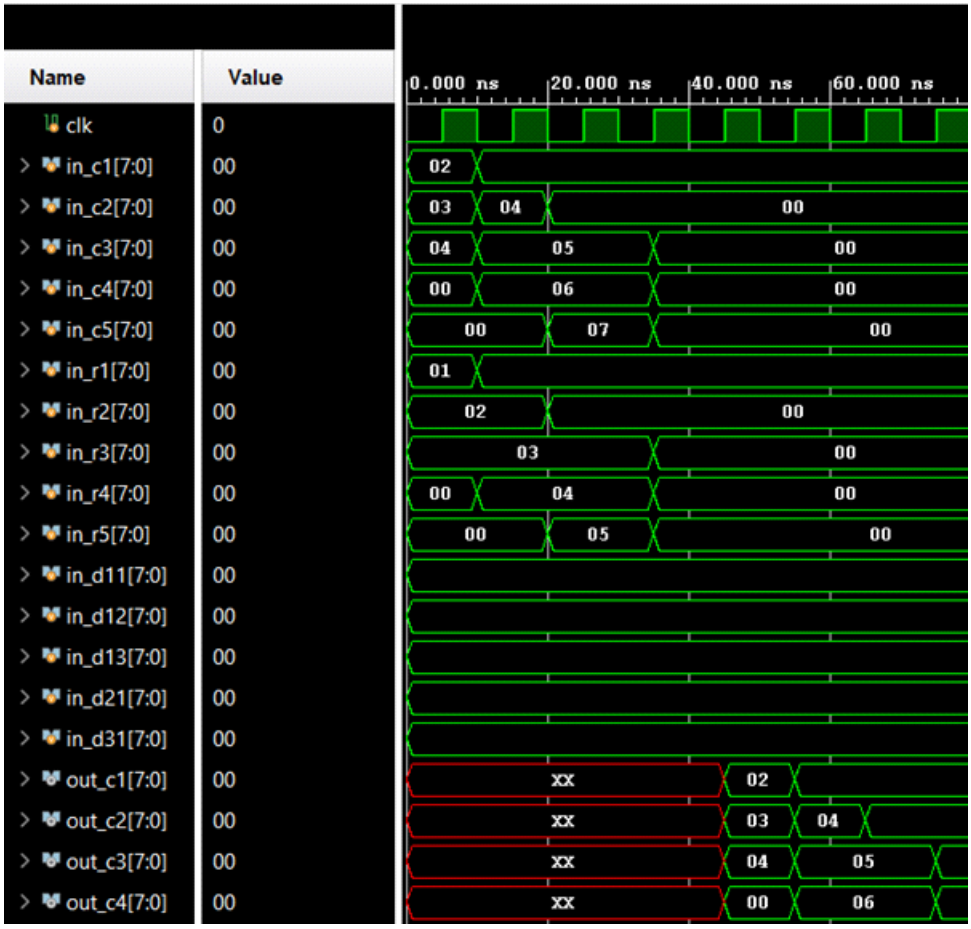
```
        in_c3 <= 8'h0;
        in_c4 <= 8'h0;
        in_c5 <= 8'h0;

        in_r1 <= 8'h0;
        in_r2 <= 8'h0;
        in_r3 <= 8'h0;
        in_r4 <= 8'h0;
        in_r5 <= 8'h0;

        in_d11 <= 8'h0;
        in_d12 <= 8'h0;
        in_d13 <= 8'h0;
        in_d21 <= 8'h0;
        in_d31 <= 8'h0;

#30     in_c1 <= 8'h0;
        in_c2 <= 8'h0;
        in_c3 <= 8'h0;
        in_c4 <= 8'h0;
        in_c5 <= 8'h0;

        in_r1 <= 8'h0;
        in_r2 <= 8'h0;
        in_r3 <= 8'h0;
        in_r4 <= 8'h0;
        in_r5 <= 8'h0;

        in_d11 <= 8'h0;
        in_d12 <= 8'h0;
        in_d13 <= 8'h0;
        in_d21 <= 8'h0;
        in_d31 <= 8'h0;
```

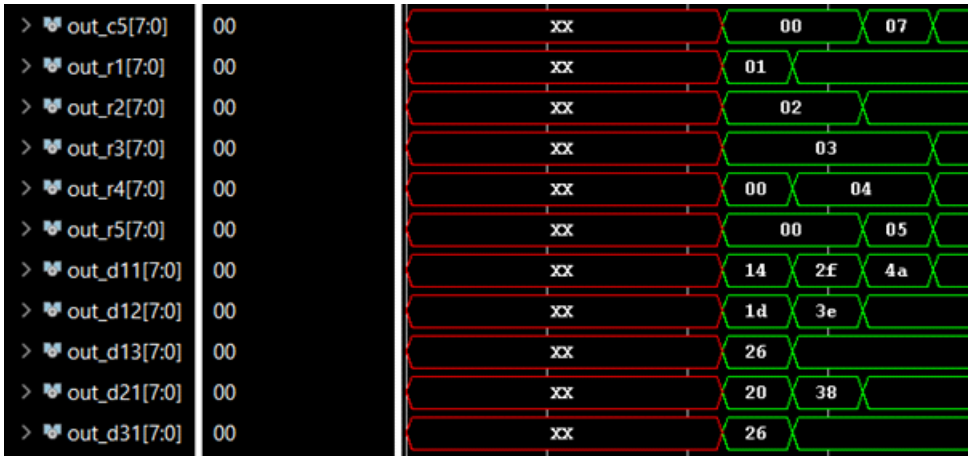
```
        in_c3 <= 8'h0;
        in_c4 <= 8'h0;
        in_c5 <= 8'h0;

        in_r1 <= 8'h0;
        in_r2 <= 8'h0;
        in_r3 <= 8'h0;
        in_r4 <= 8'h0;
        in_r5 <= 8'h0;

        in_d11 <= 8'h0;
        in_d12 <= 8'h0;
        in_d13 <= 8'h0;
        in_d21 <= 8'h0;
        in_d31 <= 8'h0;

#45     in_c1 <= 8'h0;
        in_c2 <= 8'h0;
        in_c3 <= 8'h0;
        in_c4 <= 8'h0;
        in_c5 <= 8'h0;

        in_r1 <= 8'h0;
        in_r2 <= 8'h0;
        in_r3 <= 8'h0;
        in_r4 <= 8'h0;
        in_r5 <= 8'h0;

        in_d11 <= 8'h0;
        in_d12 <= 8'h0;
        in_d13 <= 8'h0;
        in_d21 <= 8'h0;
        in_d31 <= 8'h0;
```

```
endmodule
```



Simulation:




out_c5[7:0]
out_r1[7:0]
out_r2[7:0]
out_r3[7:0]
out_r4[7:0]
out_r5[7:0]
out_d11[7:0]
out_d12[7:0]
out_d13[7:0]
out_d21[7:0]
out_d31[7:0]
00
xx
00
07
01
02
03
04
05
14
2f
4a
1d
3e
26
20
38