VLSI Desing Project 2
Matrix ALU

Apostolos Kontarinis axk220238
Athanasia Karanika axk230133

Dallas 2023

Contents

Code Changes

Matrix ALU Scale Code Changes

The changes to the RTL code were minimal since the code was fully scallable. Thus there were changes only in the top module of the design *matrix_alu*. Regarding the scale of the module in order for it to be at least 3000 cells but not too much more than that, the only parameter changed was the *word_size* to a value of 8 (marked with red in the code below). Apart from the scale of the module, there was an issue with its output. The module was designed to set its unused outputs at high impedance "z" something that could not be synthesized using GF60 because of the lack of tri-state buffers. For that reason there was a change (depicted in red in the code below) for the unused outputs to be turned to 0 instead from "z". There were no other changes to the other modules' RTL codes.

```
module matrix_alu #(parameter word_size = 8,
                    parameter Amatrixrownum = 2,
                    parameter Amatrixcolnum = 2,
                    parameter Bmatrixrownum = 2,
                    parameter Bmatrixcolnum = 2)(
input   wire        clk,
input   wire        resetn,
input   wire        [(Amatrixcolnum * Amatrixrownum) * word_size -
    1 : 0]A,
input   wire        [(Bmatrixcolnum * Bmatrixrownum) * word_size -
    1 : 0]B,
input   wire        [1:0] op,
output  wire        [(Amatrixcolnum * Amatrixrownum) * (
   Bmatrixcolnum * Bmatrixrownum) * word_size - 1 : 0] C
);

wire    [(Amatrixrownum * Bmatrixcolnum) * word_size - 1 : 0]
   m_add, m_sub, m_mul;
wire    [(Amatrixcolnum * Amatrixrownum) * (Bmatrixcolnum *
   Bmatrixrownum) * word_size - 1 : 0] m_kro;
reg     [(Amatrixcolnum * Amatrixrownum) * (Bmatrixcolnum *
   Bmatrixrownum) * word_size - 1 : 0] C_reg;
always@(posedge clk or negedge resetn)begin
    if(!resetn)begin
        C_reg = 'd0;
    end else begin
        if(op==2'b00) begin
            C_reg = {{(((Amatrixcolnum * Amatrixrownum) * (
               Bmatrixcolnum * Bmatrixrownum) - 1)* word_size - 1)
               {1'b0}},m_add};
        end else if (op==2'b01) begin
            C_reg = {{(((Amatrixcolnum * Amatrixrownum) * (
               Bmatrixcolnum * Bmatrixrownum) - 1)* word_size - 1)
               {1'b0}},m_sub};
        end else if (op==2'b10) begin
            C_reg = {{(((Amatrixcolnum * Amatrixrownum) * (
               Bmatrixcolnum * Bmatrixrownum) - 1)* word_size - 1)
               {1'b0}},m_mul};
        end else if (op==2'b11) begin
            C_reg = m_kro;
        end
    end
end
assign C = C_reg;

matrix_add_sub #(.word_size(word_size),
                 .Amatrixrownum(Amatrixrownum),
                 .Amatrixcolnum(Amatrixcolnum),
                 .Bmatrixrownum(Bmatrixrownum),
                 .Bmatrixcolnum(Bmatrixcolnum))mat_add(
```

```
.op(1'b0),
.A(A),
.B(B),
.ASP(m_add)
);
matrix_add_sub #(.word_size(word_size),
                 .Amatrixrownum(Amatrixrownum),
                 .Amatrixcolnum(Amatrixcolnum),
                 .Bmatrixrownum(Bmatrixrownum),
                 .Bmatrixcolnum(Bmatrixcolnum))mat_sub(
.op(1'b1),
.A(A),
.B(B),
.ASP(m_sub)
);
matrix_mul #(.word_size(word_size),
                 .Amatrixrownum(Amatrixrownum),
                 .Amatrixcolnum(Amatrixcolnum),
                 .Bmatrixrownum(Bmatrixrownum),
                 .Bmatrixcolnum(Bmatrixcolnum))mat_mul(
.A(A),
.B(B),
.MP(m_mul)
);
matrix_kronecker #(.word_size(word_size),
                 .Amatrixrownum(Amatrixrownum),
                 .Amatrixcolnum(Amatrixcolnum),
                 .Bmatrixrownum(Bmatrixrownum),
                 .Bmatrixcolnum(Bmatrixcolnum))mat_kron(
.A(A),
.B(B),
.TP(m_kro)
);
endmodule
```

Testbench Code Changes

The following is the updated testbench for both the behavioral *matrix_alu* module and its *matrix_alu_mapped* netlist which was produced by the *DC Compiler* tool.

```
// testbench (matrix_alu_tb) for testing a matrix arithmetic logic
    unit (Matrix ALU) //
// Initialize parameters, clock and reset signals, input matrices
   A and B, and the   //
// operation code (op), and then connects these signals to the
   Matrix ALU module.    //
// It toggles the clock signal, simulates a reset sequence, and
   then performs a      //
// sequence of operations on matrices A and B. The output result
   is captured in the  //
// signal C

   //

module matrix_alu_tb();

// Parameter Definitions //
parameter word_size = 8; // Word size for matrix elements //
parameter Amatrixrownum = 2;
parameter Amatrixcolnum = 2;
parameter Bmatrixrownum = 2;
parameter Bmatrixcolnum = 2;

// Signal Declarations //
```

```
reg     clk, resetn;
reg     [(Amatrixcolnum * Amatrixrownum) * word_size - 1 : 0] A, B
   ;
reg     [1:0] op;
wire    [(Amatrixcolnum * Amatrixrownum) * (Bmatrixcolnum *
   Bmatrixrownum) * word_size - 1 : 0] C;
wire    [(Amatrixcolnum * Amatrixrownum) * (Bmatrixcolnum *
   Bmatrixrownum) * word_size - 1 : 0] C_net;

// Clock Generation //
always #10 clk = ~clk;
initial begin
clk = 1'b0;
resetn = 1'b1;
#20
resetn = 1'b0;
#20
resetn = 1'b1;
end

//Initial Block for Matrix and Operation Initialization //
initial begin
A={8'h01,8'h02,8'h03,8'h00};
B={8'h05,8'h06,8'h07,8'h08};

op = 2'b00;
#100
op = 2'b01;
#100
op = 2'b10;
#100
op = 2'b11;
end

// Module Instantiation //
matrix_alu #(.word_size(word_size),
             .Amatrixrownum(Amatrixrownum),
             .Amatrixcolnum(Amatrixcolnum),
             .Bmatrixrownum(Bmatrixrownum),
             .Bmatrixcolnum(Bmatrixcolnum)) alu (
.clk(clk),
.resetn(resetn),
.A(A),
.B(B),
.op(op),
.C(C)
);
matrix_alu_mapped alu_netlist (
.clk(clk),
.resetn(resetn),
.A(A),
.B(B),
.op(op),
.C(C_net)
);
endmodule
```

Waveform Comparison

Below circled with purple is the output of the behavioral code (*martix_alu*) and circled with orange is the output of the netlist (*martix_alu_mapped*). Circled with red are the different tests (the different operations the ALU performs).

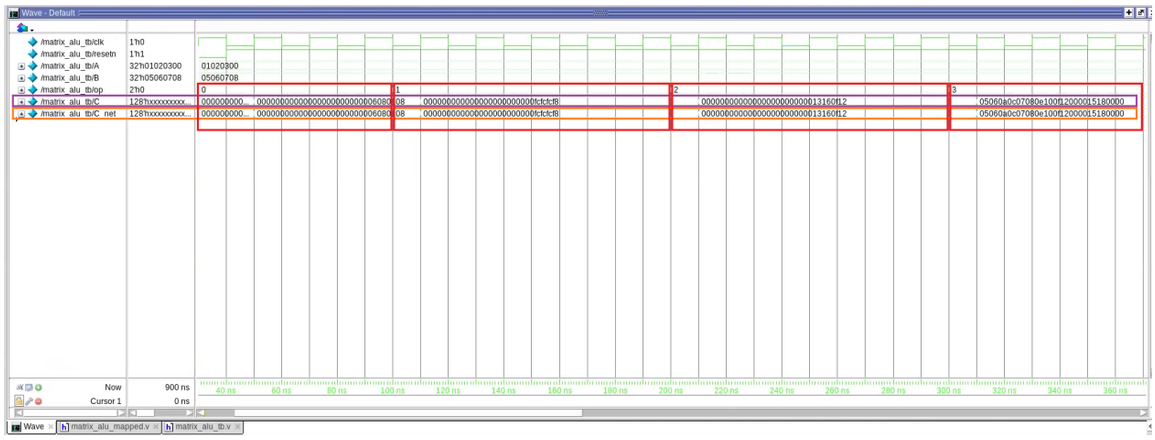

Figure 1: *matrix_alu* and *matrix_alu_netlist* Output Comparison

Cell Report

As it can be seen at the end of the report the total number of cells is 5577.

```
****************************************
Report : cell
Design : matrix_alu
Version: O-2018.06-SP1
Date   : Fri Sep 22 10:33:25 2023
****************************************

Attributes:
    b - black box (unknown)
    d - dont_touch
    h - hierarchical
    n - noncombinational
    r - removable
    u - contains unmapped logic

Cell                        Reference       Library              Area
     Attributes
--------------------------------------------------------------------------------

C402                        nand2           library          1.000000
C405                        nand2           library          1.000000
C408                        nand2           library          1.000000
C410                        nand2           library          1.000000
C_reg_reg[0]                dff             library          7.000000
     n
C_reg_reg[1]                dff             library          7.000000
     n
C_reg_reg[2]                dff             library          7.000000
     n
C_reg_reg[3]                dff             library          7.000000
     n
C_reg_reg[4]                dff             library          7.000000
     n
C_reg_reg[5]                dff             library          7.000000
     n
C_reg_reg[6]                dff             library          7.000000
     n
C_reg_reg[7]                dff             library          7.000000
     n
C_reg_reg[8]                dff             library          7.000000
     n
```

C_reg_reg[9]	dff	library	7.000000
n			
C_reg_reg[10]	dff	library	7.000000
n			
C_reg_reg[11]	dff	library	7.000000
n			
C_reg_reg[12]	dff	library	7.000000
n			
C_reg_reg[13]	dff	library	7.000000
n			
C_reg_reg[14]	dff	library	7.000000
n			
C_reg_reg[15]	dff	library	7.000000
n			
C_reg_reg[16]	dff	library	7.000000
n			
C_reg_reg[17]	dff	library	7.000000
n			
C_reg_reg[18]	dff	library	7.000000
n			
C_reg_reg[19]	dff	library	7.000000
n			
C_reg_reg[20]	dff	library	7.000000
n			
C_reg_reg[21]	dff	library	7.000000
n			
C_reg_reg[22]	dff	library	7.000000
n			
C_reg_reg[23]	dff	library	7.000000
n			
C_reg_reg[24]	dff	library	7.000000
n			
C_reg_reg[25]	dff	library	7.000000
n			
C_reg_reg[26]	dff	library	7.000000
n			
C_reg_reg[27]	dff	library	7.000000
n			
C_reg_reg[28]	dff	library	7.000000
n			
C_reg_reg[29]	dff	library	7.000000
n			
C_reg_reg[30]	dff	library	7.000000
n			
C_reg_reg[31]	dff	library	7.000000
n			
C_reg_reg[32]	dff	library	7.000000
n			
C_reg_reg[33]	dff	library	7.000000
n			
C_reg_reg[34]	dff	library	7.000000
n			
C_reg_reg[35]	dff	library	7.000000
n			
C_reg_reg[36]	dff	library	7.000000
n			
C_reg_reg[37]	dff	library	7.000000
n			
C_reg_reg[38]	dff	library	7.000000
n			
C_reg_reg[39]	dff	library	7.000000
n			
C_reg_reg[40]	dff	library	7.000000
n			

C_reg_reg[41]	dff	library	7.000000
n			
C_reg_reg[42]	dff	library	7.000000
n			
C_reg_reg[43]	dff	library	7.000000
n			
C_reg_reg[44]	dff	library	7.000000
n			
C_reg_reg[45]	dff	library	7.000000
n			
C_reg_reg[46]	dff	library	7.000000
n			
C_reg_reg[47]	dff	library	7.000000
n			
C_reg_reg[48]	dff	library	7.000000
n			
C_reg_reg[49]	dff	library	7.000000
n			
C_reg_reg[50]	dff	library	7.000000
n			
C_reg_reg[51]	dff	library	7.000000
n			
C_reg_reg[52]	dff	library	7.000000
n			
C_reg_reg[53]	dff	library	7.000000
n			
C_reg_reg[54]	dff	library	7.000000
n			
C_reg_reg[55]	dff	library	7.000000
n			
C_reg_reg[56]	dff	library	7.000000
n			
C_reg_reg[57]	dff	library	7.000000
n			
C_reg_reg[58]	dff	library	7.000000
n			
C_reg_reg[59]	dff	library	7.000000
n			
C_reg_reg[60]	dff	library	7.000000
n			
C_reg_reg[61]	dff	library	7.000000
n			
C_reg_reg[62]	dff	library	7.000000
n			
C_reg_reg[63]	dff	library	7.000000
n			
C_reg_reg[64]	dff	library	7.000000
n			
C_reg_reg[65]	dff	library	7.000000
n			
C_reg_reg[66]	dff	library	7.000000
n			
C_reg_reg[67]	dff	library	7.000000
n			
C_reg_reg[68]	dff	library	7.000000
n			
C_reg_reg[69]	dff	library	7.000000
n			
C_reg_reg[70]	dff	library	7.000000
n			
C_reg_reg[71]	dff	library	7.000000
n			
C_reg_reg[72]	dff	library	7.000000
n			

C_reg_reg[73]	dff	library	7.000000
n			
C_reg_reg[74]	dff	library	7.000000
n			
C_reg_reg[75]	dff	library	7.000000
n			
C_reg_reg[76]	dff	library	7.000000
n			
C_reg_reg[77]	dff	library	7.000000
n			
C_reg_reg[78]	dff	library	7.000000
n			
C_reg_reg[79]	dff	library	7.000000
n			
C_reg_reg[80]	dff	library	7.000000
n			
C_reg_reg[81]	dff	library	7.000000
n			
C_reg_reg[82]	dff	library	7.000000
n			
C_reg_reg[83]	dff	library	7.000000
n			
C_reg_reg[84]	dff	library	7.000000
n			
C_reg_reg[85]	dff	library	7.000000
n			
C_reg_reg[86]	dff	library	7.000000
n			
C_reg_reg[87]	dff	library	7.000000
n			
C_reg_reg[88]	dff	library	7.000000
n			
C_reg_reg[89]	dff	library	7.000000
n			
C_reg_reg[90]	dff	library	7.000000
n			
C_reg_reg[91]	dff	library	7.000000
n			
C_reg_reg[92]	dff	library	7.000000
n			
C_reg_reg[93]	dff	library	7.000000
n			
C_reg_reg[94]	dff	library	7.000000
n			
C_reg_reg[95]	dff	library	7.000000
n			
C_reg_reg[96]	dff	library	7.000000
n			
C_reg_reg[97]	dff	library	7.000000
n			
C_reg_reg[98]	dff	library	7.000000
n			
C_reg_reg[99]	dff	library	7.000000
n			
C_reg_reg[100]	dff	library	7.000000
n			
C_reg_reg[101]	dff	library	7.000000
n			
C_reg_reg[102]	dff	library	7.000000
n			
C_reg_reg[103]	dff	library	7.000000
n			
C_reg_reg[104]	dff	library	7.000000
n			

C_reg_reg[105]	dff	library	7.000000
n			
C_reg_reg[106]	dff	library	7.000000
n			
C_reg_reg[107]	dff	library	7.000000
n			
C_reg_reg[108]	dff	library	7.000000
n			
C_reg_reg[109]	dff	library	7.000000
n			
C_reg_reg[110]	dff	library	7.000000
n			
C_reg_reg[111]	dff	library	7.000000
n			
C_reg_reg[112]	dff	library	7.000000
n			
C_reg_reg[113]	dff	library	7.000000
n			
C_reg_reg[114]	dff	library	7.000000
n			
C_reg_reg[115]	dff	library	7.000000
n			
C_reg_reg[116]	dff	library	7.000000
n			
C_reg_reg[117]	dff	library	7.000000
n			
C_reg_reg[118]	dff	library	7.000000
n			
C_reg_reg[119]	dff	library	7.000000
n			
C_reg_reg[120]	dff	library	7.000000
n			
C_reg_reg[121]	dff	library	7.000000
n			
C_reg_reg[122]	dff	library	7.000000
n			
C_reg_reg[123]	dff	library	7.000000
n			
C_reg_reg[124]	dff	library	7.000000
n			
C_reg_reg[125]	dff	library	7.000000
n			
C_reg_reg[126]	dff	library	7.000000
n			
C_reg_reg[127]	dff	library	7.000000
n			
I_0	inv	library	1.000000
I_1	inv	library	1.000000
I_2	inv	library	1.000000
I_4	inv	library	1.000000
...			
...			
...			
mat_sub/U17	inv	library	1.000000
mat_sub/U19	inv	library	1.000000
mat_sub/U21	inv	library	1.000000
mat_sub/U23	inv	library	1.000000
mat_sub/U25	inv	library	1.000000
mat_sub/U27	inv	library	1.000000
mat_sub/U29	inv	library	1.000000
mat_sub/U31	inv	library	1.000000
mat_sub/U33	inv	library	1.000000
mat_sub/U35	inv	library	1.000000
mat_sub/U37	inv	library	1.000000

```
mat_sub/U39                inv              library          1.000000
mat_sub/U41                inv              library          1.000000
mat_sub/U43                inv              library          1.000000
mat_sub/U45                inv              library          1.000000
mat_sub/U47                inv              library          1.000000
mat_sub/U49                inv              library          1.000000
mat_sub/U51                inv              library          1.000000
mat_sub/U53                inv              library          1.000000
mat_sub/U55                inv              library          1.000000
mat_sub/U57                inv              library          1.000000
mat_sub/U59                inv              library          1.000000
mat_sub/U61                inv              library          1.000000
mat_sub/U63                inv              library          1.000000
mat_sub/U65                inv              library          1.000000
mat_sub/U67                inv              library          1.000000
mat_sub/U69                inv              library          1.000000
mat_sub/U71                inv              library          1.000000
mat_sub/U73                inv              library          1.000000
mat_sub/U75                inv              library          1.000000
mat_sub/U77                inv              library          1.000000
mat_sub/U79                inv              library          1.000000
--------------------------------------------------------------------------------

Total 5577 cells
   9485.000000
1
```