Team ssh

Sukhbir Singh

Greg Schmit

Philip Horwitz

Table of vectors: The first 21 (green) are from the pdf table handout.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | f | expected\_y | expected\_z |
| 00000000 | 00000000 | 2 | 00000000 | 1 |
| 00000000 | ffffffff | 2 | ffffffff | 0 |
| 00000001 | ffffffff | 2 | 00000000 | 1 |
| 000000ff | 00000001 | 2 | 00000100 | 0 |
| 00000000 | 00000000 | 6 | 00000000 | 1 |
| 00000000 | ffffffff | 6 | 00000001 | 0 |
| 00000001 | 00000001 | 6 | 00000000 | 1 |
| 00000100 | 00000010 | 6 | 000000f0 | 0 |
| 00000000 | 00000000 | 7 | 00000000 | 1 |
| 00000000 | 00000001 | 7 | 00000001 | 0 |
| 00000000 | ffffffff | 7 | 00000000 | 1 |
| 00000001 | 00000000 | 7 | 00000000 | 1 |
| ffffffff | 00000000 | 7 | 00000001 | 0 |
| ffffffff | ffffffff | 0 | ffffffff | 0 |
| ffffffff | 12345678 | 0 | 12345678 | 0 |
| 12345678 | 87654321 | 0 | 02244220 | 0 |
| 00000000 | ffffffff | 0 | 00000000 | 1 |
| ffffffff | ffffffff | 1 | ffffffff | 0 |
| 12345678 | 87654321 | 1 | 97755779 | 0 |
| 00000000 | ffffffff | 1 | ffffffff | 0 |
| 00000000 | 00000000 | 1 | 00000000 | 1 |
| ffffffff | 00000000 | 4 | ffffffff | 0 |
| ffffffff | ffffffff | 4 | 00000000 | 1 |
| 00000000 | ffffffff | 4 | 00000000 | 1 |
| 01010101 | 00000000 | 4 | 01010101 | 0 |
| 00000000 | 00000000 | 5 | ffffffff | 0 |
| ffffffff | ffffffff | 5 | ffffffff | 0 |
| 00000000 | ffffffff | 5 | 00000000 | 1 |
| ffffffff | 00000000 | 5 | ffffffff | 0 |
| 01010101 | 00000000 | 5 | ffffffff | 0 |
| 33333333 | 33333333 | 5 | ffffffff | 0 |

**alu.sv**

module mux2to1\_32bit(input wire [31:0] i0, i1, input wire s, output wire [31:0] out);

assign out = s ? i1 : i0;

endmodule

module mux4to1\_32bit(input wire [31:0] i0, i1, i2, i3, input wire [1:0] s, output wire [31:0] out);

wire [31:0] m1\_out, m2\_out;

mux2to1\_32bit m1(i0, i1, s[0], m1\_out);

mux2to1\_32bit m2(i2, i3, s[0], m2\_out);

mux2to1\_32bit m3(m1\_out, m2\_out, s[1], out);

endmodule

module adder\_32bit(input wire [31:0] a, b, input wire op, output wire [31:0] y);

assign y = op ? a + b + 1 : a + b;

endmodule

module zero\_extender(input wire a, output wire [31:0] aex);

assign aex[0] = a;

assign aex[31:1] = 30'b0;

endmodule

module alu(input logic [31:0] a, b, input logic [2:0] f, output logic [31:0] y, output logic zero);

wire [31:0] bb;

wire [31:0] s;

wire cout;

wire [31:0] zeroex;

mux2to1\_32bit bbmux(b, ~b, f[2], bb);

adder\_32bit adder(a, bb, f[2], s);

zero\_extender ze(s[31], zeroex);

mux4to1\_32bit ymux(a & bb, a | bb, s, zeroex, f[1:0], y);

assign zero = ~(|y);

endmodule

**Alu.tv**

00000000\_00000000\_2\_00000000\_1

00000000\_ffffffff\_2\_ffffffff\_0

00000001\_ffffffff\_2\_00000000\_1

000000ff\_00000001\_2\_00000100\_0

00000000\_00000000\_6\_00000000\_1

00000000\_ffffffff\_6\_00000001\_0

00000001\_00000001\_6\_00000000\_1

00000100\_00000010\_6\_000000f0\_0

00000000\_00000000\_7\_00000000\_1

00000000\_00000001\_7\_00000001\_0

00000000\_ffffffff\_7\_00000000\_1

00000001\_00000000\_7\_00000000\_1

ffffffff\_00000000\_7\_00000001\_0

ffffffff\_ffffffff\_0\_ffffffff\_0

ffffffff\_12345678\_0\_12345678\_0

12345678\_87654321\_0\_02244220\_0

00000000\_ffffffff\_0\_00000000\_1

ffffffff\_ffffffff\_1\_ffffffff\_0

12345678\_87654321\_1\_97755779\_0

00000000\_ffffffff\_1\_ffffffff\_0

00000000\_00000000\_1\_00000000\_1

ffffffff\_00000000\_4\_ffffffff\_0

ffffffff\_ffffffff\_4\_00000000\_1

00000000\_ffffffff\_4\_00000000\_1

01010101\_00000000\_4\_01010101\_0

00000000\_00000000\_5\_ffffffff\_0

ffffffff\_ffffffff\_5\_ffffffff\_0

00000000\_ffffffff\_5\_00000000\_1

ffffffff\_00000000\_5\_ffffffff\_0

01010101\_00000000\_5\_ffffffff\_0

33333333\_33333333\_5\_ffffffff\_0

**Testbench.sv**

`include "ALU.sv"

module testbench();

logic [31:0] a;

logic [31:0] b;

logic [3:0] f;

logic [31:0] y; //actual y value

logic [31:0] expected\_y; //expected y value

logic z;

logic [3:0]expected\_z;//expected zero value

//change first bracket numbers to reflect size of line

//change second bracket numbers to reflect the number of lines

logic [103:0] testvector[30:0];

integer num; // used for iterating through testvector

ALU dut(a,b,f[2:0],y,z); //please make sure I/Os are in correct order

initial

begin

$readmemh("C:/Modeltech\_pe\_edu\_10.4a/examples/testvector.tv",testvector);

for (num = 0; num <31;num = num + 1)

begin

{a,b,f,expected\_y,expected\_z} = testvector[num];#10;

//use num + 1 since file lines start at 1

$display("num=%d a=%h b=%h f=%h",num+1,a,b,f);

$display("exp\_y=%h y=%h exp\_z=%h z=%h",expected\_y,y,expected\_z,z);

if (expected\_y !=y | expected\_z != z)

$display("Incorrect y or z line:%d---------------",num+1);

end

end

endmodule

**Display:**

# num= 1 a=00000000 b=00000000 f=2

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 2 a=00000000 b=ffffffff f=2

# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

# num= 3 a=00000001 b=ffffffff f=2

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 4 a=000000ff b=00000001 f=2

# exp\_y=00000100 y=00000100 exp\_z=0 z=0

# num= 5 a=00000000 b=00000000 f=6

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 6 a=00000000 b=ffffffff f=6

# exp\_y=00000001 y=00000001 exp\_z=0 z=0

# num= 7 a=00000001 b=00000001 f=6

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 8 a=00000100 b=00000010 f=6

# exp\_y=000000f0 y=000000f0 exp\_z=0 z=0

# num= 9 a=00000000 b=00000000 f=7

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 10 a=00000000 b=00000001 f=7

# exp\_y=00000001 y=00000001 exp\_z=0 z=0

# num= 11 a=00000000 b=ffffffff f=7

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 12 a=00000001 b=00000000 f=7

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 13 a=ffffffff b=00000000 f=7

# exp\_y=00000001 y=00000001 exp\_z=0 z=0

# num= 14 a=ffffffff b=ffffffff f=0

# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

# num= 15 a=ffffffff b=12345678 f=0

# exp\_y=12345678 y=12345678 exp\_z=0 z=0

# num= 16 a=12345678 b=87654321 f=0

# exp\_y=02244220 y=02244220 exp\_z=0 z=0

# num= 17 a=00000000 b=ffffffff f=0

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 18 a=ffffffff b=ffffffff f=1

# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

# num= 19 a=12345678 b=87654321 f=1

# exp\_y=97755779 y=97755779 exp\_z=0 z=0

# num= 20 a=00000000 b=ffffffff f=1

# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

# num= 21 a=00000000 b=00000000 f=1

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 22 a=ffffffff b=00000000 f=4

# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

# num= 23 a=ffffffff b=ffffffff f=4

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 24 a=00000000 b=ffffffff f=4

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 25 a=01010101 b=00000000 f=4

# exp\_y=01010101 y=01010101 exp\_z=0 z=0

# num= 26 a=00000000 b=00000000 f=5

# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

# num= 27 a=ffffffff b=ffffffff f=5

# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

# num= 28 a=00000000 b=ffffffff f=5

# exp\_y=00000000 y=00000000 exp\_z=1 z=1

# num= 29 a=ffffffff b=00000000 f=5

# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

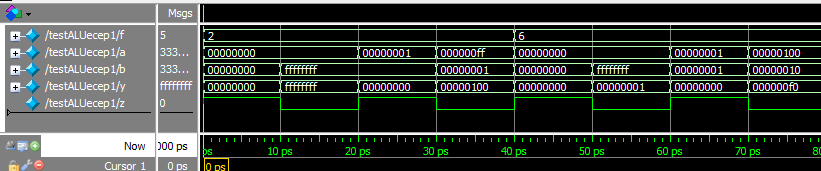
# num= 30 a=01010101 b=00000000 f=5

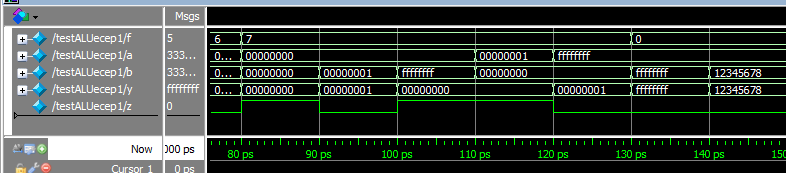
# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

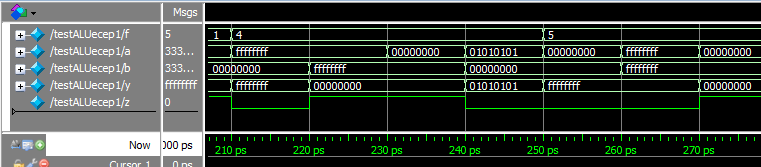
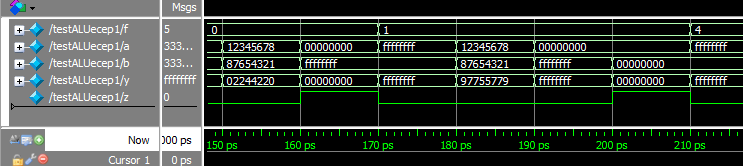
# num= 31 a=33333333 b=33333333 f=5

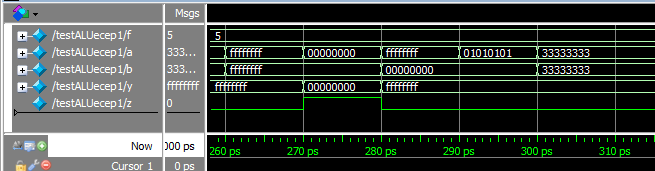
# exp\_y=ffffffff y=ffffffff exp\_z=0 z=0

Wave Forms:









Workload:

The Project took between 12-16 hrs. The most time consuming part has been troubleshooting the code because of syntax errors.