Verilog Code Snippets

module mux16(input wire [0:15] i, input wire [3:0] sel, output wire o);

mux8 upper(i[0:7], sel[0], sel[1], sel[2], o1);

mux8 lower(i[8:15], sel[0], sel[1], sel[2], o2);

mux2 final(o1, o2, sel[3], o);

endmodule

module sll(input wire [15:0] i, input wire [3:0] shamt, output wire [15:0] o);

mux16 m\_15( { i[15], i[14], i[13], i[12], i[11], i[10], i[9], i[8], i[7], i[6], i[5], i[4], i[3], i[2], i[1], i[0] }, shamt, o[15]);

mux16 m\_14( { i[14], i[13], i[12], i[11], i[10], i[9], i[8], i[7], i[6], i[5], i[4], i[3], i[2], i[1], i[0], 1'b0 }, shamt, o[14]);

. . .

mux16 m\_1( { i[1], i[0], 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0 }, shamt, o[1]);

mux16 m\_0( { i[0], 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0 }, shamt, o[0]);

endmodule

module srl(input wire [15:0] i, input wire [3:0] shamt, output wire [15:0] o);

mux16 m\_0( { i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9], i[10], i[11], i[12], i[13], i[14], i[15] }, shamt, o[0]);

mux16 m\_1( { i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9], i[10], i[11], i[12], i[13], i[14], i[15], 1'b0 }, shamt, o[1]);

. . .

mux16 m\_14( { i[14], i[15], 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0 }, shamt, o[14]);

mux16 m\_15( { i[15], 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0, 1'b0 }, shamt, o[15]);

endmodule

module sra(input wire [15:0] i, input wire [3:0] shamt, output wire [15:0] o);

mux16 m\_0( { i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9], i[10], i[11], i[12], i[13], i[14], i[15] }, shamt, o[0]);

mux16 m\_1( { i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[8], i[9], i[10], i[11], i[12], i[13], i[14], i[15], i[15] }, shamt, o[1]);

. . .

mux16 m\_14( { i[14], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15] }, shamt, o[14]);

mux16 m\_15( { i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15], i[15] }, shamt, o[15]);

endmodule

Modified ALU with SLT

module alu (input wire [2:0] op, input wire [15:0] i0, i1, output wire [15:0] o, output wire carry);

    wire [15:0] add\_op, sub\_op, and\_op, or\_op, sll\_op, srl\_op, sra\_op;

wire add\_carry, sub\_carry;

STBitAddSub add\_unit( i0, i1, 1'b0, add\_op, add\_carry );

STBitAddSub sub\_unit( i0, i1, 1'b1, sub\_op, sub\_carry );

and2x and\_unit( i0, i1, and\_op );

or2x or\_unit( i0, i1, or\_op );

sll sll\_unit( i0, i1[3:0], sll\_op );

srl srl\_unit( i0, i1[3:0], srl\_op );

sra sra\_unit( i0, i1[3:0], sra\_op );

STBit8to1Mux outp( add\_op, sub\_op, and\_op, or\_op, sll\_op, srl\_op, sra\_op, {15'h0, sub\_op[15]}, op, o );

mux2 outc( add\_carry, sub\_carry, op[0], carry );

endmodule

module mux8\_16(input [15:0] i0, i1, i2, i3, i4, i5, i6, i7, input [2:0] s, output [15:0] o);

mux8 op\_0( { i0[0], i1[0], i2[0], i3[0], i4[0], i5[0], i6[0], i7[0] }, s[0], s[1], s[2], o[0] );

mux8 op\_1( { i0[1], i1[1], i2[1], i3[1], i4[1], i5[1], i6[1], i7[1] }, s[0], s[1], s[2], o[1] );

...

mux8 op\_15( { i0[15], i1[15], i2[15], i3[15], i4[15], i5[15], i6[15], i7[15] }, s[0], s[1], s[2], o[15] );

endmodule

