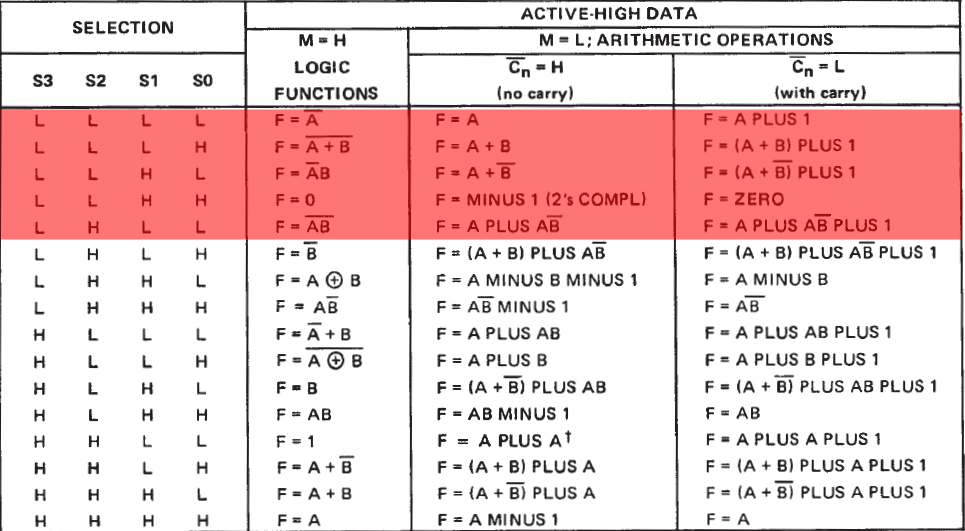
 Universidade Federal de Itajubá – Mestrado em Engenharia Elétrica

EM604 – Síntese Lógica

Aluno: Leonardo Ribeiro Gonçalves

Comportamento do CI 74181 em Verilog

O objetivo desse trabalho é desenvolver um programa em verilog simulando o comportamento do CI 74181. O sistema será capaz de realizar 5 operações lógicas e 5 aritméticas destacadas em vermelho na tabela abaixo.

Tabela 1 - Operações lógicas e aritméticas

Foi realizada uma simulação do sistema para verificar o funcionamento das funções lógicas e aritméticas. A figura 1 mostra o comportamento do sistema.

As entradas A e B assumem nesses caso valores fixos (0000 e 1111 respectivamente), a entrada S e M recebem os valores para selecionar as operações a serem realizadas assim como descritas na tabela 1.

Figura 1 - Simulação do projeto

O código para a execução dessas operações segue abaixo nesse documento:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* UNIVERSIDADE FEDERAL DE ITAJUBA 1S/2017 - EM604 \*

\* \*

\* Function: 74181 Behavior System \*

\* \*

\* Written by: Leonardo Gonçalves \*

\* \*

\* Last modified: April 24, 2017 \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

module projeto(S, A, B, M, CIN\_N, F, COUT\_N);

input [3:0] A, B;

input [2:0] S;

input CIN\_N, M;

output [3:0] F;

output COUT\_N;

wire [3:0] F;

wire cout, COUT\_N;

projeto2b P1(S, A[1:0], B[1:0], M, CIN\_N, F[1:0], cout);

projeto2b P2(S, A[3:2], B[3:2], M, cout, F[3:2], COUT\_N);

endmodule

module projeto2b(S, A, B, M, CIN\_N, F, COUT\_N);

input [1:0] A, B;

input [2:0] S;

input CIN\_N, M;

output [1:0] F;

output reg COUT\_N;

reg [1:0] F;

always @(S or A or B)

begin

if(M)

begin

case (S)

4'b000:

begin

F <= ~A;

end

4'b001:

begin

F <= ~(A|B);

end

4'b010:

begin

F <= ~A & B;

end

4'b011:

begin

F <= 2'b00;

end

4'b100:

begin

F <= ~(A&B);

end

default:

begin

end

endcase

end

else

begin

if(CIN\_N)

begin

case (S)

4'b000:

begin

F <= A;

end

4'b001:

begin

F <= A | B;

end

4'b010:

begin

F <= A | ~B;

end

4'b011:

begin

F <= 2'b00;

end

4'b100:

begin

F <= A + (A&~B);

end

default:

begin

end

endcase

end

else

begin

case (S)

4'b000:

begin

F <= A + 2'b01;

end

4'b001:

begin

F <= (A | B) + 2'b01;

end

4'b010:

begin

F <= (A | ~B) + 2'b01;

end

4'b011:

begin

F <= 2'b00;

end

4'b100:

begin

F <= A + A&~B + 2'b01;

end

default:

begin

end

endcase

end

end

end

endmodule