TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH

Khoa Điện – Điện tử

Bộ môn Điện tử

_____***_____

BÁO CÁO CẤU TRÚC MÁY TÍNH

LAB 1: THIẾT KẾ BỘ CỘNG 4 BIT

GVHD: Trần Hoàng Linh

Sinh viên thực hiện: Đinh Thế Bảo

MSSV: 1510152

Tp. Hồ Chí Minh, ngày 20 tháng 11 năm 2019

I. Mục tiêu

Mục tiêu của bài thí nghiệm này là xây dựng bộ cộng 4 bit bằng 2 cách:

a. Sử dụng bộ cộng toàn phần để thực hiện (Structural model).
b. Sử dụng mô tả hành vi để thực hiện (Behavioral model).

II. Lý thuyết hoạt động của mạch

Với phương pháp sử dụng bộ cộng toàn phần để xây dựng bộ cộng 4 bit, ta cần thực hiện việc ghép 4 bộ cộng toàn phần 1 bit lại.

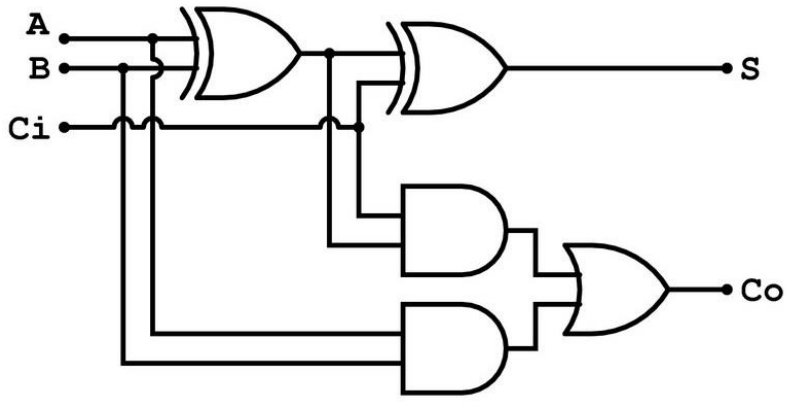
Một bộ cộng toàn phần 1 bit (full adder 1 bit) gồm có 3 ngõ vào là A, B, Ci (bit nhớ thường có giá trị là 0 đối với bộ cộng toàn phần 1 bit) và 2 ngõ ra là tổng S và bit nhớ Co.

Bảng chứa các giá trị ngõ ra tương ứng với giá trị ngõ vào của bộ cộng toàn phần:

A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

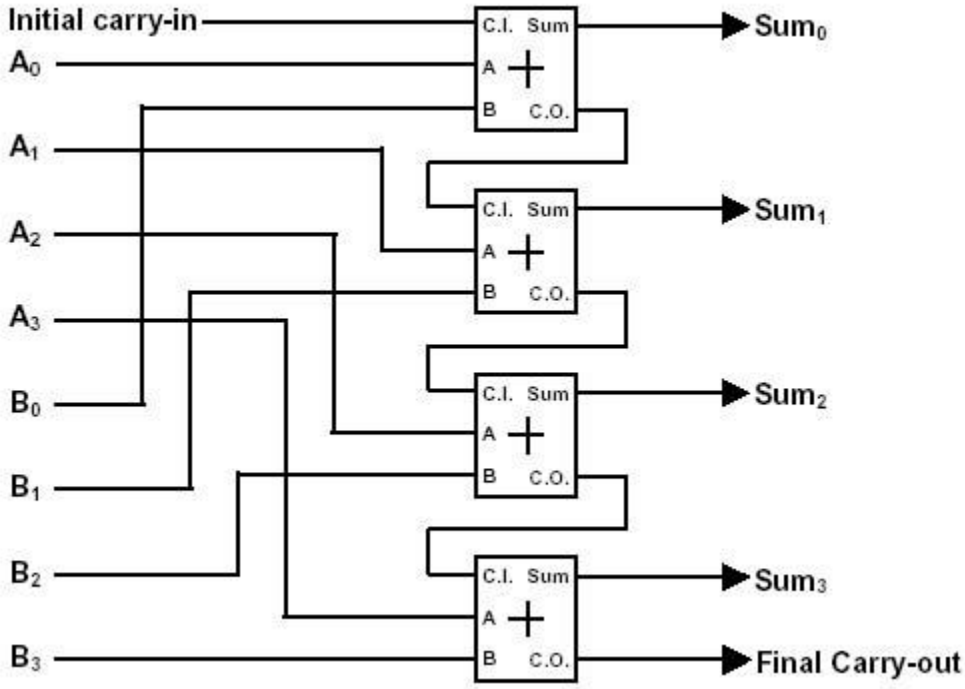
Bảng 2.1: Bảng sự thật của bộ cộng full adder 1 bit

Từ bảng sự thật trên ta xác định được hàm ngõ ra phụ thuộc vào ngõ vào của bộ cộng toàn phần 1 bit: $S = (A \oplus B) \oplus Ci, Co = ((A \oplus B) \bullet Ci) + (A \bullet B)$. Từ đó ta có sơ đồ mạch của bộ cộng toàn phần 1 bit:



Hình 2.1: Sơ đồ mạch cộng toàn phần 1 bit.

Bộ cộng toàn phần 4 bit là kết quả của việc ghép 4 bộ cộng toàn phần 1 bit theo sơ đồ dưới đây:



Hình 2.2: Bộ cộng toàn phần 4 bit.

III. Chương trình thực hiện

3.1 Sructural model

3.1.1 Full adder 1 bit

```
module fullAdder (A, B, Ci, Co, S);
input A;
input B;
input Ci;
output Co;
output S;
assign S = (A ^ B) ^ Ci;
assign Co = A&B | (A ^ B) & Ci;
endmodule
```

3.1.2 Full adder 4 bit

```
module fullAdder4bit (A, B, Ci, S, Co);
    input [3:0] A;
    input [3:0] B;
    input Ci;
    output [3:0] S;
    output Co;


    wire carry0;
    wire carry1;
    wire carry2;


    fullAdder fullAdder4bit_0 (
        .A  (A[0]),
        .B  (B[0]),
        .Ci (Ci),
        .S  (S[0]),
        .Co (carry0)
```

```
        );

    fullAdder fullAdder4bit_1 (
        .A  (A[1]),
        .B  (B[1]),
        .Ci (carry0),
        .S  (S[1]),
        .Co (carry1)
        );

    fullAdder fullAdder4bit_2 (
        .A  (A[2]),
        .B  (B[2]),
        .Ci (carry1),
        .S  (S[2]),
        .Co (carry2)
        );

    fullAdder fullAdder4bit_3 (
        .A  (A[3]),
        .B  (B[3]),
        .Ci (carry2),
        .S  (S[3]),
        .Co (Co)
        );

endmodule
```

3.1.3 Test bench

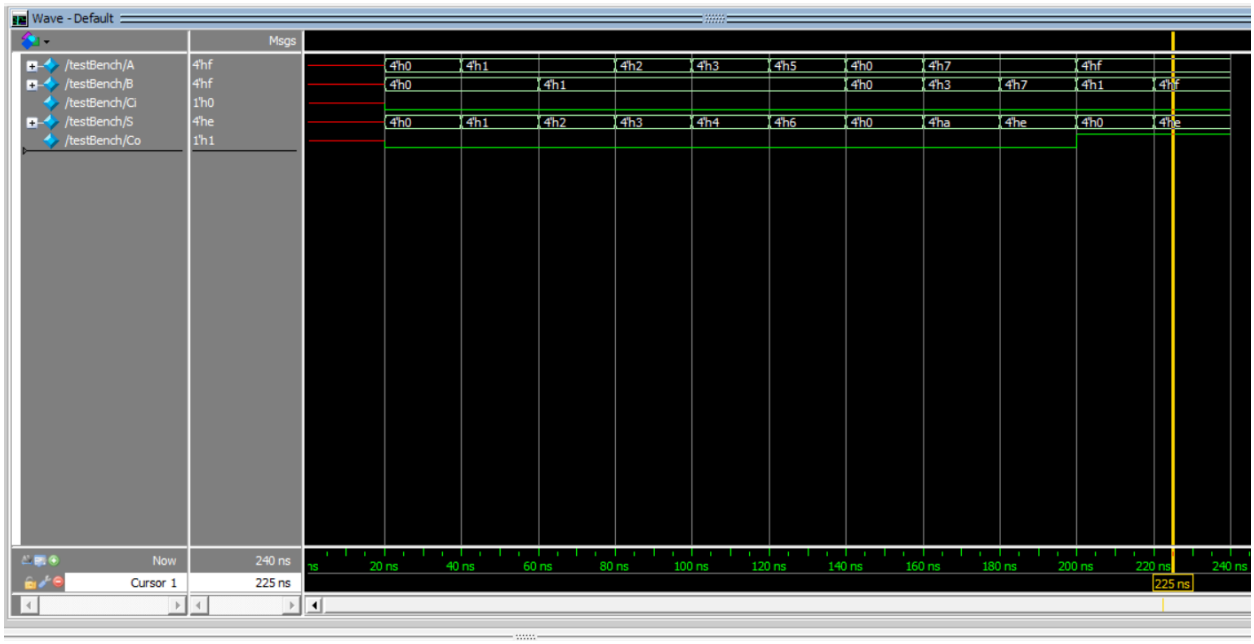
```
module testBench;
    reg [3:0] A, B;
    reg Ci;
    wire [3:0] S;
    wire Co;

    fullAdder4bit testFullAdder4bit (   .A  (A),
                                        .B  (B),
                                        .Ci (Ci),
                                        .S  (S),
                                        .Co (Co)
                                        );

    initial begin
        #20
        A = 4'b0000;
        B = 4'b0000;
        Ci = 1'b0;
        #20
        A = 4'b0001;
        B = 4'b0000;
        Ci = 1'b0;
        #20
        A = 4'b0001;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b0010;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b0011;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b0101;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b0000;
        B = 4'b0000;
        Ci = 1'b0;
        #20
```

```
        A = 4'b0111;
        B = 4'b0011;
        Ci = 1'b0;
        #20
        A = 4'b0111;
        B = 4'b0111;
        Ci = 1'b0;
        #20
        A = 4'b1111;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b1111;
        B = 4'b1111;
        Ci = 1'b0;
        #20
        $finish;
    end
endmodule //
```

3.1.4 Kết quả mô phỏng



Hình 3.1: Kết quả mô phỏng một số trường hợp.

3.2 Behavioral model

3.2.1 Chương trình

```
module behavioral (A, B, Ci, S, Co);
```

```
    input [3:0] A, B;
    input Ci;
    output wire [3:0] S;
    output wire Co;
    wire [4:0] temp;
    assign temp = A + B + Ci;
    assign S = temp[3:0];
    assign Co = temp[4];

endmodule
```

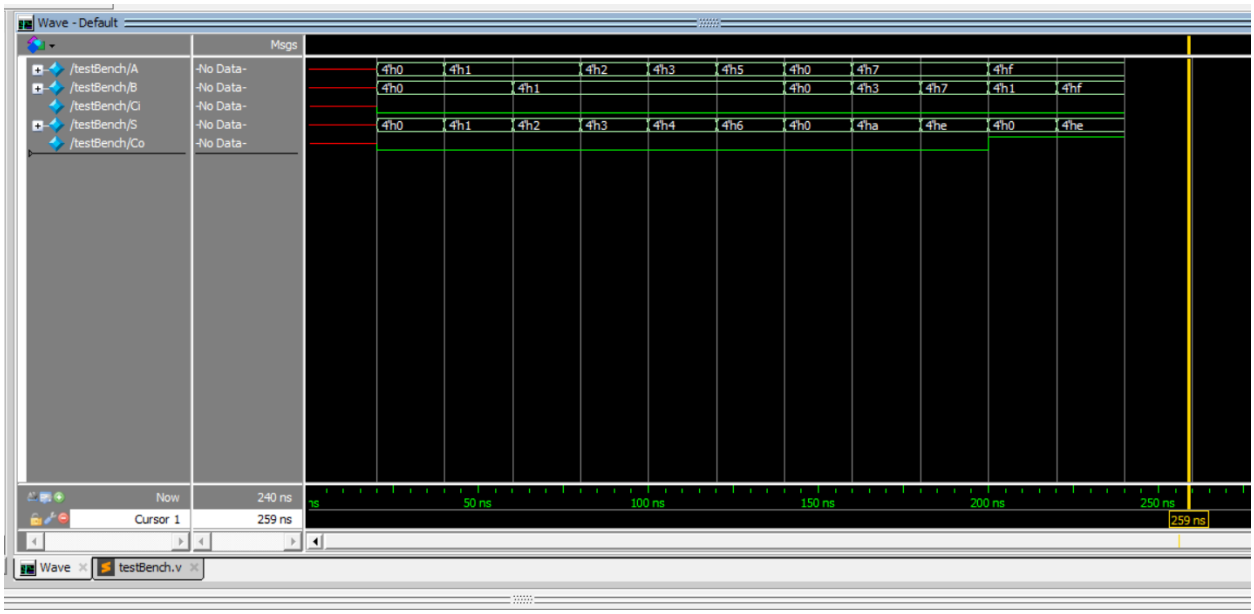
3.2.2 Test Bench

```
module testBench;
    reg [3:0] A, B;
    reg Ci;
    wire [3:0] S;
    wire Co;
 /*
    fullAdder4bit testFullAdder4bit (   .A  (A),
                                        .B  (B),
                                        .Ci (Ci),
                                        .S  (S),
                                        .Co (Co)
                                        );
*/
    behavioral behavioral ( .A  (A),
                            .B  (B),
                            .Ci (Ci),
                            .S  (S),
                            .Co (Co)
                            );

    initial begin
        #20
        A = 4'b0000;
        B = 4'b0000;
        Ci = 1'b0;
        #20
        A = 4'b0001;
        B = 4'b0000;
        Ci = 1'b0;
        #20
        A = 4'b0001;
        B = 4'b0001;
```

```
        Ci = 1'b0;
        #20
        A = 4'b0010;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b0011;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b0101;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b0000;
        B = 4'b0000;
        Ci = 1'b0;
        #20
        A = 4'b0111;
        B = 4'b0011;
        Ci = 1'b0;
        #20
        A = 4'b0111;
        B = 4'b0111;
        Ci = 1'b0;
        #20
        A = 4'b1111;
        B = 4'b0001;
        Ci = 1'b0;
        #20
        A = 4'b1111;
        B = 4'b1111;
        Ci = 1'b0;
        #20
        $finish;
    end
endmodule //
```

3.2.3 Kết quả mô phỏng



3.3 Kiểm tra kết quả và nhận xét

Bảng sự thật ứng với các trường hợp mô phỏng

A	B	Ci	S
0000	0000	0	0
0001	0000	0	0001
0001	0001	0	0010
0010	0001	0	0011
0011	0001	0	0100
0101	0001	0	0110
0000	0000	0	0000
0111	0011	0	1010
0111	0111	0	1110
1111	0001	0	10000
1111	1111	0	11110

Bảng 3.1: Bảng sự thật của các trường hợp mô phỏng

Nhận xét: Kết quả mô phỏng đúng với kết quả tính toán các theo lý thuyết.