

Thiết kế tổng hợp hệ thống số

Họ và tên: Đỗ Xuân Sơn

MSSV: 20133299

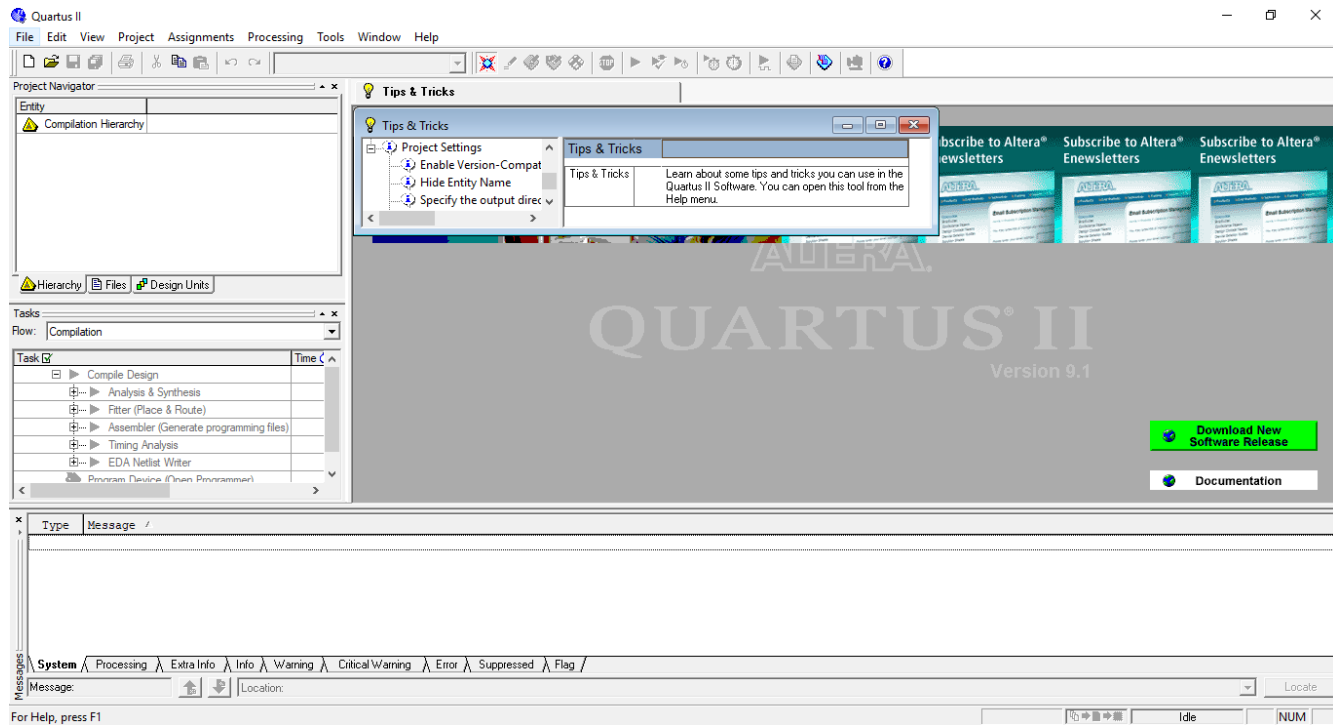
Lớp: ĐTTT-06 K58

Bài tập

1. Cài đặt quartus II (bản 9.0 cho nhẹ)
2. Làm các bài hôm trước trên máy tính (viết code)
3. Đọc phần slide 1.4 và 1.5
4. Tự tìm hiểu viết code cho bài chia tần số 50MHZ về 1HZ

=> Nộp phần 4 và chụp màn hình phần 1, 2.

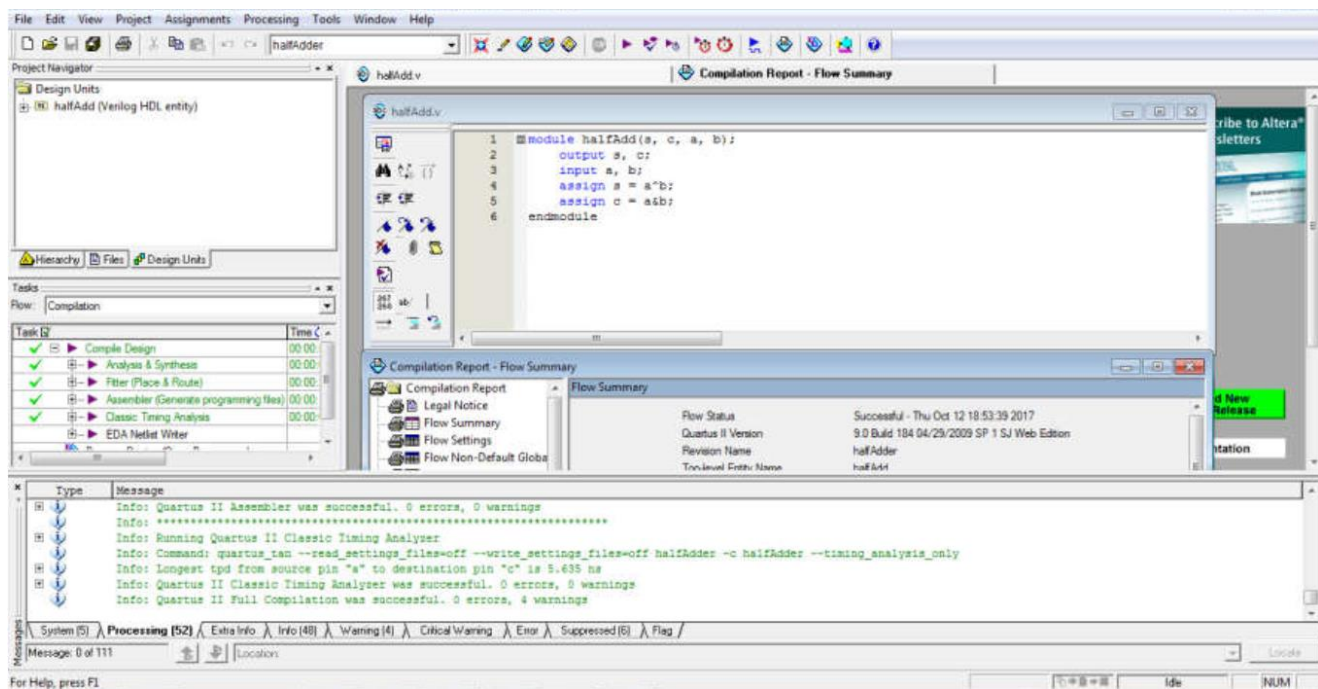
1.Cài đặt Quartus II



2. Viết code bài tập hôm trước trên máy tính

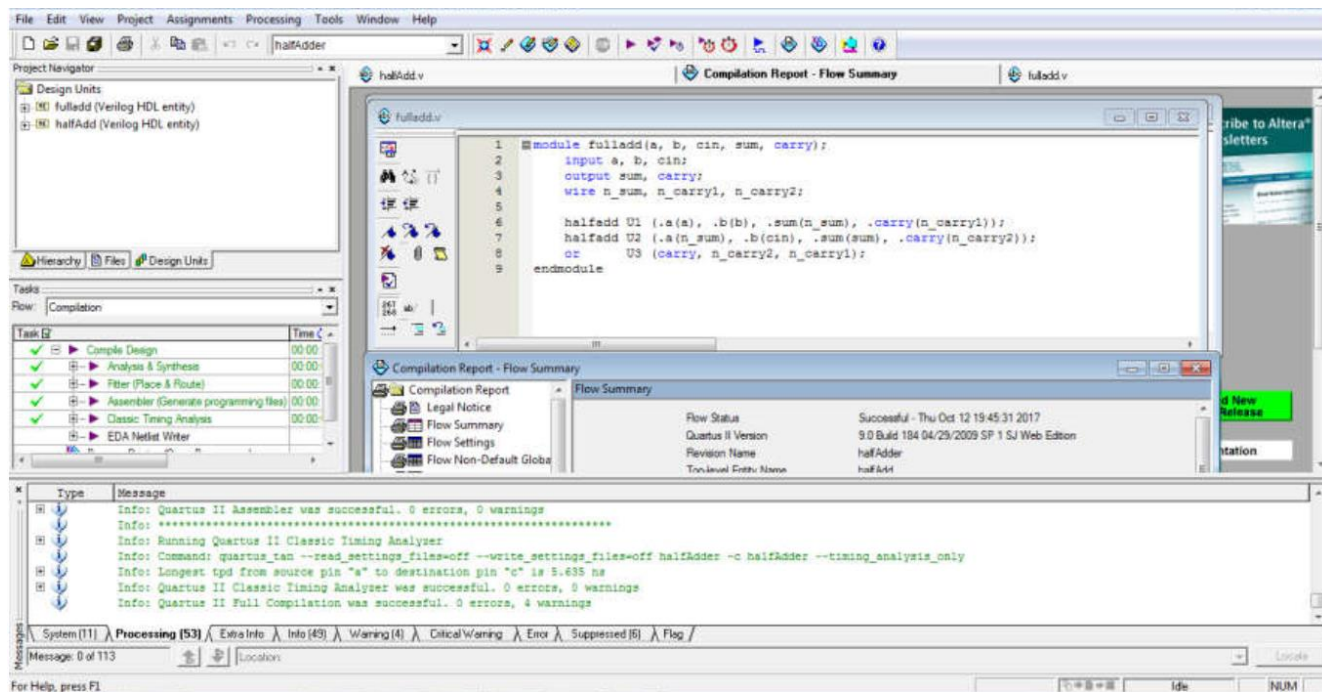
2.1. Half adder

```
module halfAdd(s, c, a, b);  
  
    output s, c;  
  
    input a, b;  
  
    assign s = a^b;  
  
    assign c = a&b;  
  
endmodule
```

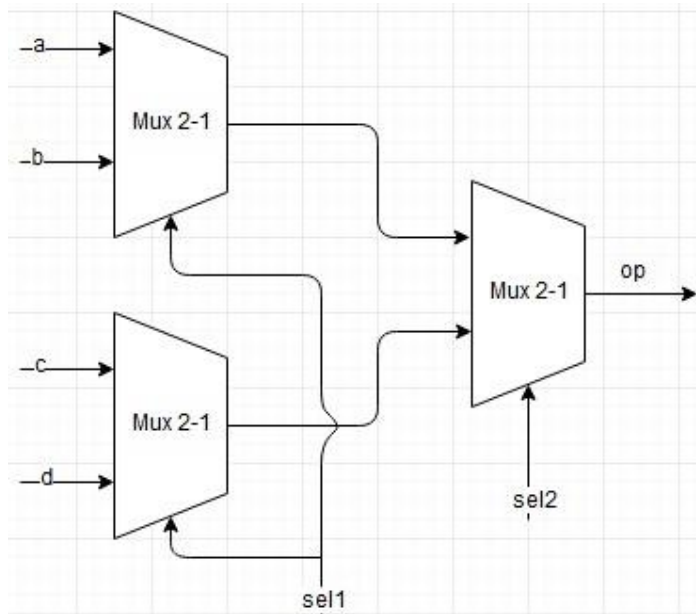


2.2.Full adder

```
module fulladd(a, b, cin, sum, carry);  
  
    input a, b, cin;  
  
    output sum, carry;  
  
    wire n_sum, n_carry1, n_carry2;  
  
    halfadd U1 (.a(a), .b(b), .sum(n_sum), .carry(n_carry1));  
  
    halfadd U2 (.a(n_sum), .b(cin), .sum(sum), .carry(n_carry2));  
  
    or  
        U3 (carry, n_carry2, n_carry1);  
  
endmodule
```



2.3.Code mô tả bộ mux 4-1 từ mux 2-1



```
module mux4x1_from3mux2x1(op, a, b, c, d, sel1, sel2);  
    input a, b, c, d, sel1, sel2;  
    output op;  
    reg op;  
    always@(a or b or c or d or sel1 or sel2) begin  
        case ({sel1, sel2})  
            2'b00 : op = a;  
            2'b01 : op = b;  
            2'b10 : op = c;  
            2'b11 : op = d;  
        endcase  
    end  
endmodule
```

File Edit View Project Assignments Processing Tools Window Help

halfAdder

Design Units

- fullAdd (Verilog HDL entity)
- halfAdd (Verilog HDL entity)
- mux4x1_from3mux2x1 (Verilog HDL entity)

Hierarchy Files Design Units

Tasks

Flow: Compilation

Task	Time
Complete Design	00:00
Analysis & Synthesis	00:00
Filter (Place & Route)	00:00
Assembler (Generate programming files)	00:00
Classic Timing Analysis	00:00
EDA Netlist Writer	00:00

Compilation Report - Flow Summary

Compilation Report

Legal Notice

```
1 module mux4x1_from3mux2x1(op, a, b, c, d, sel1, sel2);
2   input a, b, c, d, sel1, sel2;
3   output op;
4
5   reg op;
6
7   always@(a or b or c or d or sel1 or sel2) begin
8       case ((sel1, sel2))
9           2'b00 : op = a;
10          2'b01 : op = b;
11          2'b10 : op = c;
12          2'b11 : op = d;
13      endcase
14   end
15 endmodule
```

Compilation Report - Flow Summary

Flow Summary

Download New Release

Information

Type	Message
Info	Quartus II Assembler was successful. 0 errors, 0 warnings
Info	*****
Info	Running Quartus II Classic Timing Analyzer
Info	Command: quartus_tan --read_settings_files=off --write_settings_files=off halfAdder -c halfAdder --timing_analysis_only
Info	Longest tpd from source pin "a" to destination pin "c" is 5.635 ns
Info	Quartus II Classic Timing Analyzer was successful. 0 errors, 0 warnings
Info	Quartus II Full Compilation was successful. 0 errors, 4 warnings

System (13) Processing (54) Extra Info (50) Warning (4) Critical Warning Error Suppressed (6) Flag

Message: 0 of 115

Location

For Help, press F1

Ln 8, Col 28

ide

NUM

3. Tìm hiểu code cho bài chia tần số 50MHz về 1Hz

3.1. Dao động và tần số

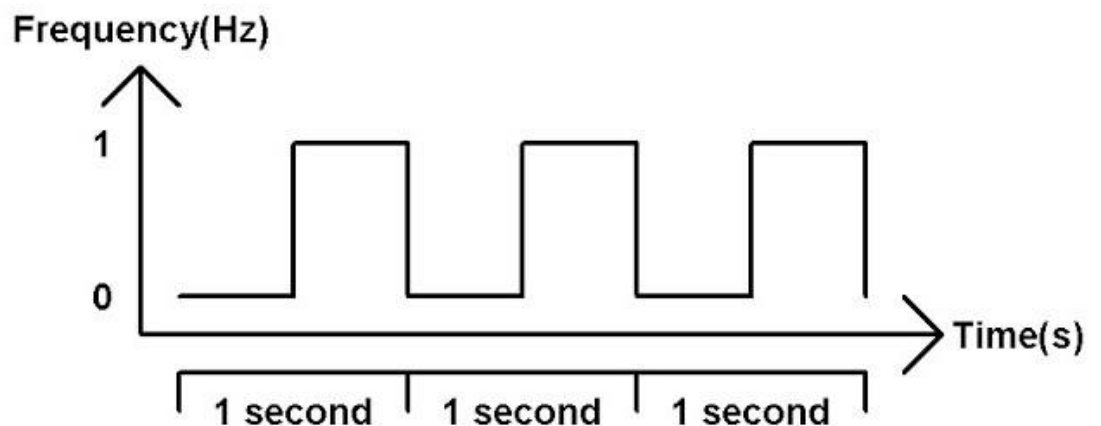
- Một thành phần rất quan trọng của bất kỳ board FPGA là bộ dao động crystal. Các dao động phát ra một dãy 0, 1 tại một thời gian nhất định trên tần số của nó. Oscillators sử dụng nhiều trong các thiết bị điện tử như TV, radio, điện thoại di động, máy vi tính....

- Ví dụ: một bộ vi xử lý có ghi tần số là 2 GHz, nó có nghĩa là các dao động có tần số 2GHz. Hay nói cách khác các dao động phát ra 2 tỉ chuỗi của các tín hiệu 0 và 1.

- Theo vật lý thì ta có thể tìm thời gian từ tần số và công thức để chuyển đổi từ tần số trong thời gian là:

- $F = 1/T$
- T là chu kỳ (giây)
- F là tần số (Hz)

Vậy từ công thức ta có thể suy ra được 1Hz là 1s. Một tín hiệu 1Hz được biểu diễn như sau:



3.2. Chia tần số

- Một công đoạn rất quan trọng trong bất cứ mạch nào liên quan đến sự đồng bộ và thời gian thực. Hoặc trong một mạch chúng ta cần nhiều tần số thì việc chia tần số chính của board FPGA (thường là 50MHz) ra nhiều tần số khác nhau.

$$1\text{ms} = 0.001\text{s}$$

$$1\mu\text{s} = 0.000\,001\text{s}$$

$$1\text{Hz} = 1\text{s}$$

$$10\text{Hz} = 0.1\text{s}$$

$$100\text{Hz} = 0.01\text{s}$$

$$1\text{KHz} = 1\text{ms}$$

$$1\text{MHz} = 1\mu\text{s}$$

- Trong bài toán của chúng ta cho tần số 50MHz chia về 1Hz.

- Tần số 50MHz trong 1s có 50 000 000 chuỗi bit 0, 1
- Trong một nửa chu kỳ (0.5s) thì có $50\text{triệu}/2 = 25\text{triệu}$ (ở trạng thái high hoặc low mỗi 0.5s)
- Với 25 triệu = 101111010111100001000000 có 25 bit => sử dụng bộ đếm từ 0 đến 24
- Code:

```
module clk_div (C_50Mhz,C_1Hz);

    reg[24:0] counter = 0; // Dem 25 000 000 trang thai

    input C_50Mhz;          // Ngo dau vao 50MHz

    output reg C_1Hz = 1; // Tin hieu dau ra


    always @(posedge C_50Mhz)

        begin

            counter = counter + 1;

            if(counter == 25_000_000)

                begin

                    counter = 0;

                    C_1Hz = ~C_1Hz;

                end

            end

        end

endmodule
```