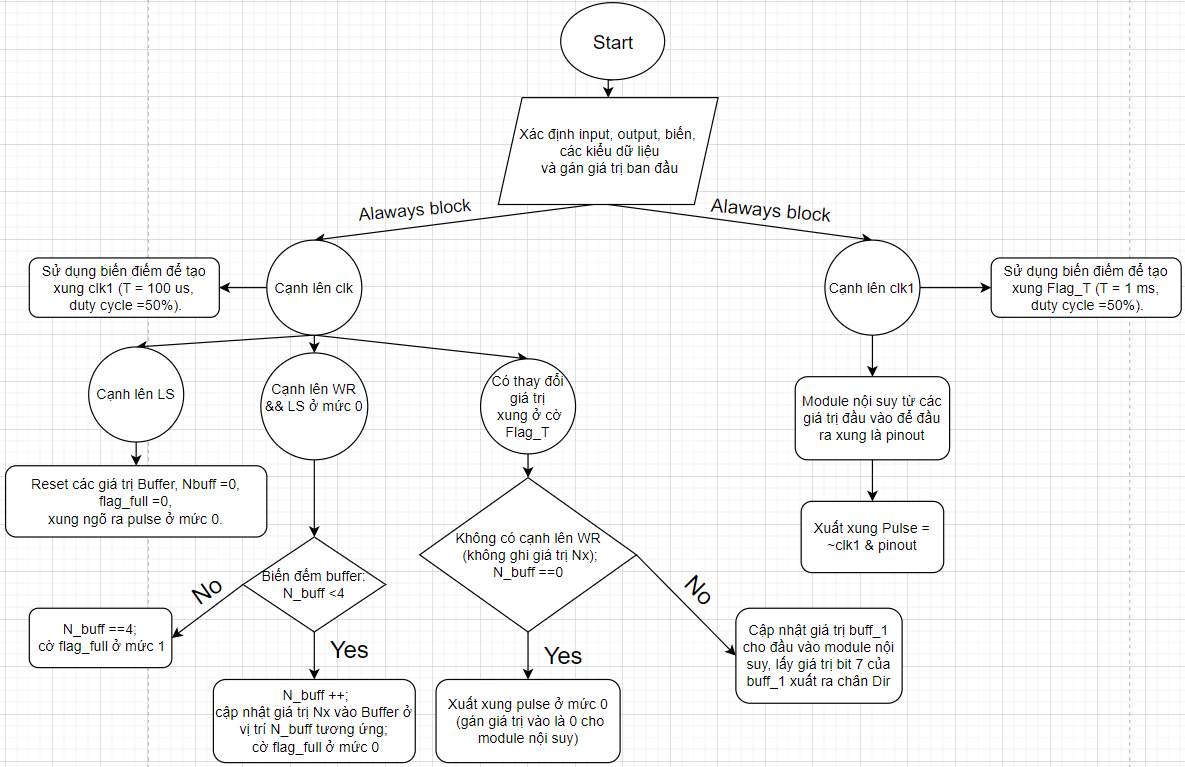
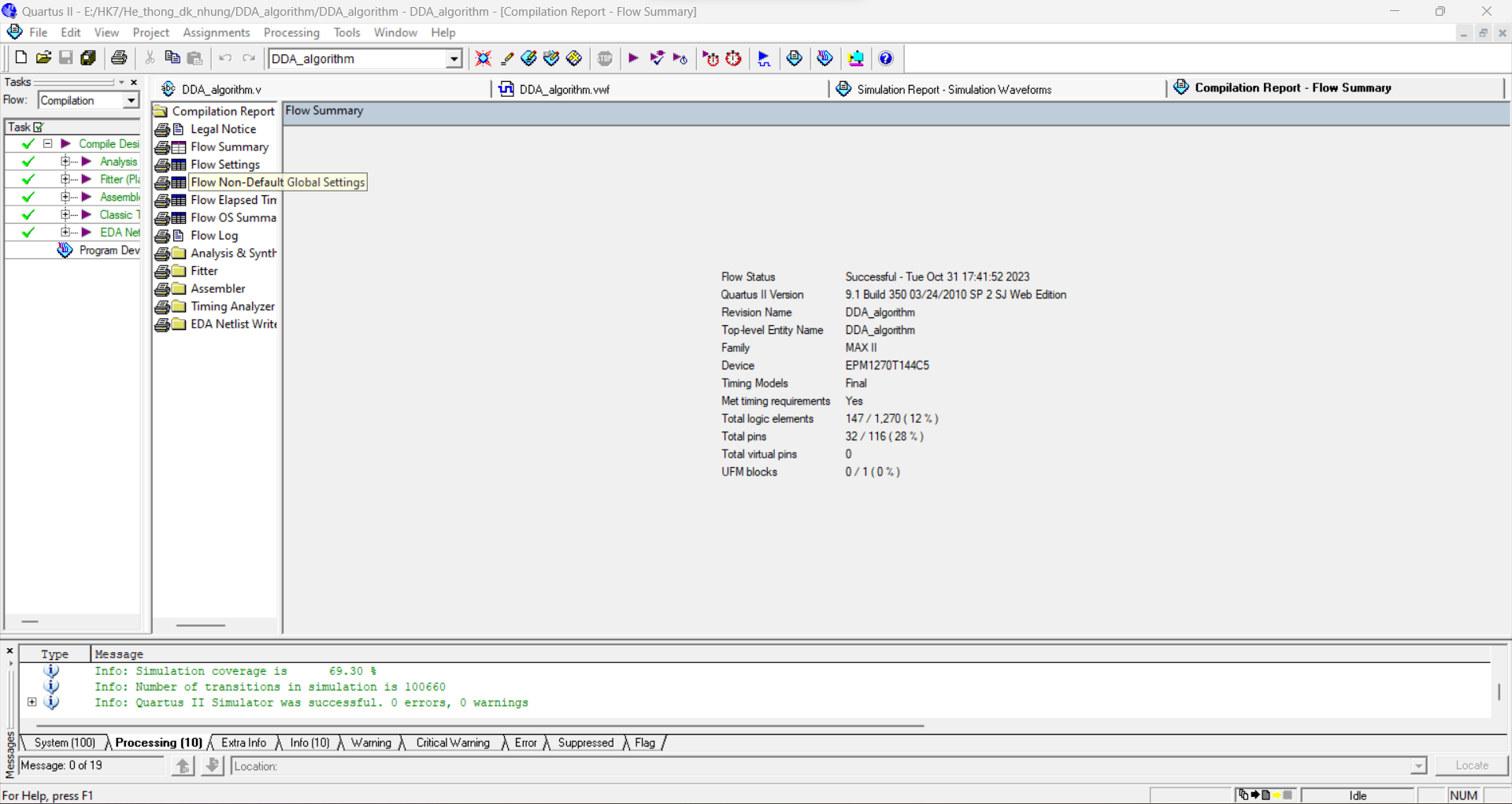
**Lưu đồ giải thuật:**



**Code Quatus:**

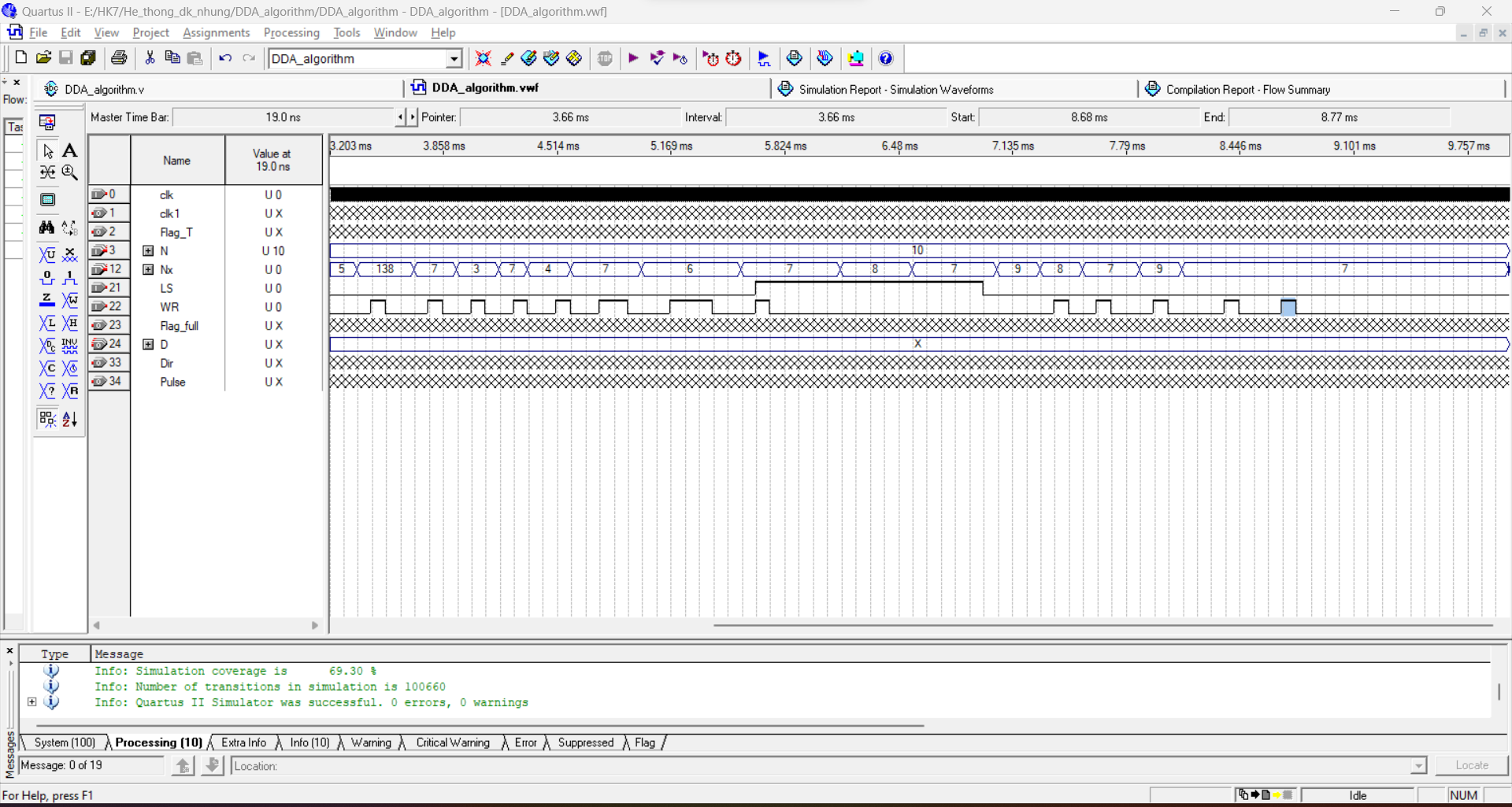
|  |
| --- |
| module DDA\_algorithm(clk, WR, LS, N, Nx, Pulse, Dir, Flag\_T,Flag\_full,D,clk1);  input clk, WR, LS;  input [7:0] N, Nx;  output Pulse, Dir;  reg Dir;  reg pinout;  output reg [7:0] D = 0;  output reg Flag\_T = 0;  output reg Flag\_full = 0;  output reg clk1 = 0;  reg [7:0] counter\_clk1 = 0;  reg [3:0] count\_T = 0;  reg [7:0] acc = 10;  reg [2:0] N\_buff = 0;  reg [7:0] buff\_1,buff\_2,buff\_3,buff\_4 = 0;  reg pre\_WR = 0;  reg pre\_LS = 0;  reg pre\_Flag\_T = 0;    always@(posedge clk) begin  pre\_WR<=WR;  pre\_LS <= LS;  //---------------Tạo tín hiệu xung clk1-----------  pre\_Flag\_T <= Flag\_T;  if (counter\_clk1 <= 49)  counter\_clk1 = counter\_clk1 +1;  else begin  clk1 = ~clk1;  counter\_clk1 = 0;  end    //---------------Xét khi có cạnh lên LS-----------  if ({pre\_LS, LS} == 2'b01) begin  N\_buff = 0;  buff\_1 = 0;  buff\_2 = 0;  buff\_3 = 0;  buff\_4 = 0;  D = 0;  end  //---------------Xét khi có cạnh lên WR và LS ở mức 0-----------  if ({pre\_WR,WR} == 2'b01 && LS ==0) begin  if (N\_buff < 4) begin  N\_buff = N\_buff +1;  case(N\_buff)  1: buff\_1 = Nx;  2: buff\_2 = Nx;  3: buff\_3 = Nx;  4: buff\_4 = Nx;  endcase  end  else begin  N\_buff = 4;  end  end    if (N\_buff < 4) Flag\_full = 0;  else Flag\_full = 1;  //---------------Xét khi có thay đổi ở cờ Flag\_T-----------  if (pre\_Flag\_T != Flag\_T) begin  if (N\_buff !=0) begin  N\_buff = N\_buff - 1;  D = buff\_1;  buff\_1 = buff\_2;  buff\_2 = buff\_3;  buff\_3 = buff\_4;  buff\_4 = 0;  Dir = D[7]; end  else D=0;  end  end    always@(posedge clk1)  begin  //---------------Tạo tín hiệu xung cờ Flag\_T-----------  count\_T = count\_T +1;  if (count\_T == 10) begin  count\_T = 0;  Flag\_T = ~Flag\_T;  end  //---------------Module nội suy-----------  acc = acc + D[6:0];  if (acc > N) begin  acc = acc - N;  pinout = 1;  end  else begin  pinout = 0;  end  end  assign Pulse = ~clk1 & pinout;  endmodule end  end  end  endmodule |

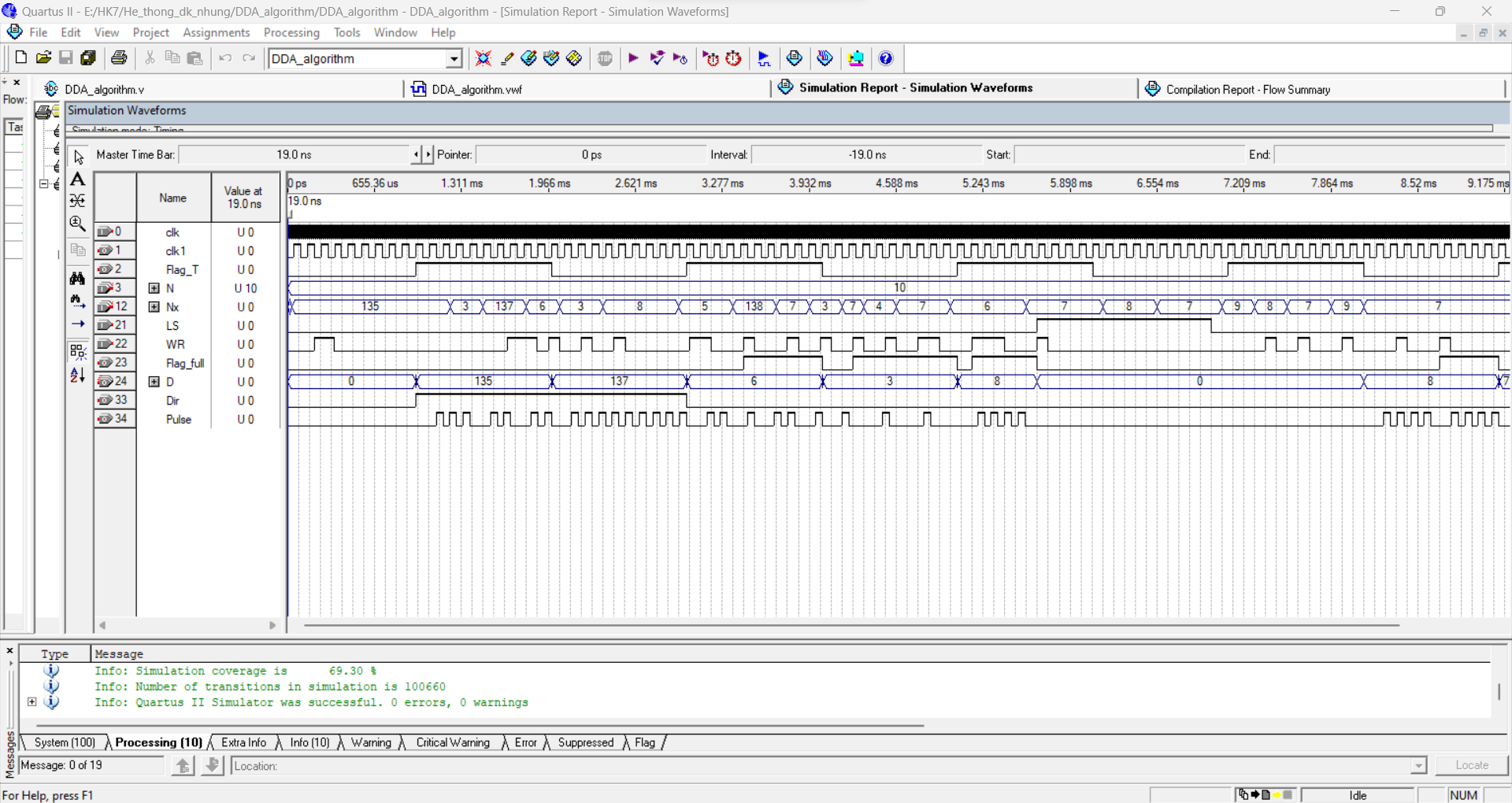
**Compile thành công:**



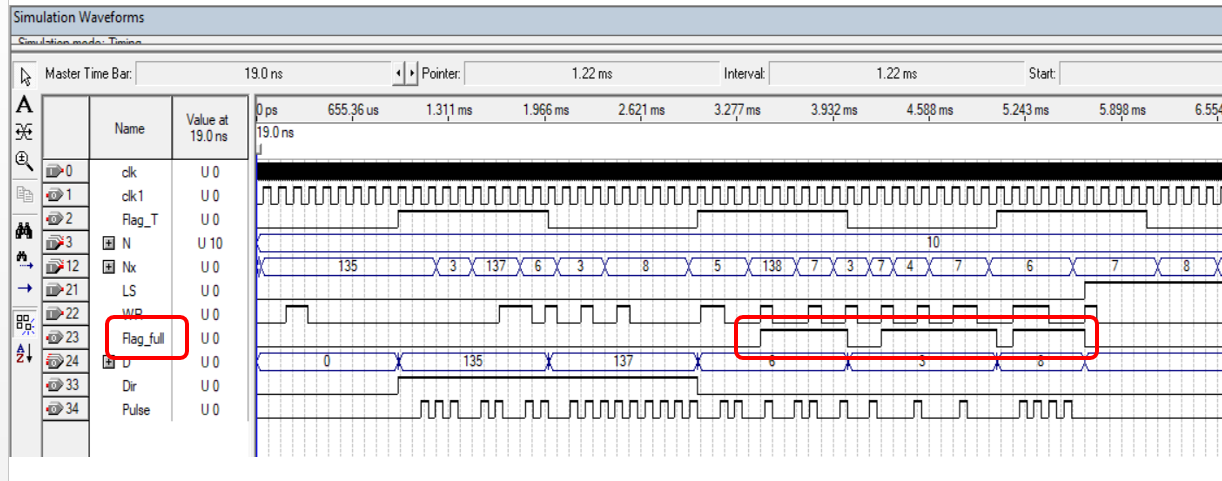
**Waveform**

Kết quả tổng quát:

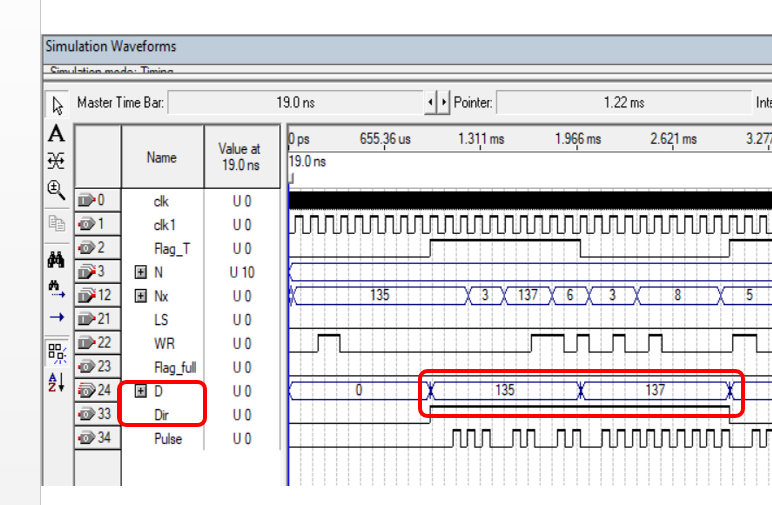




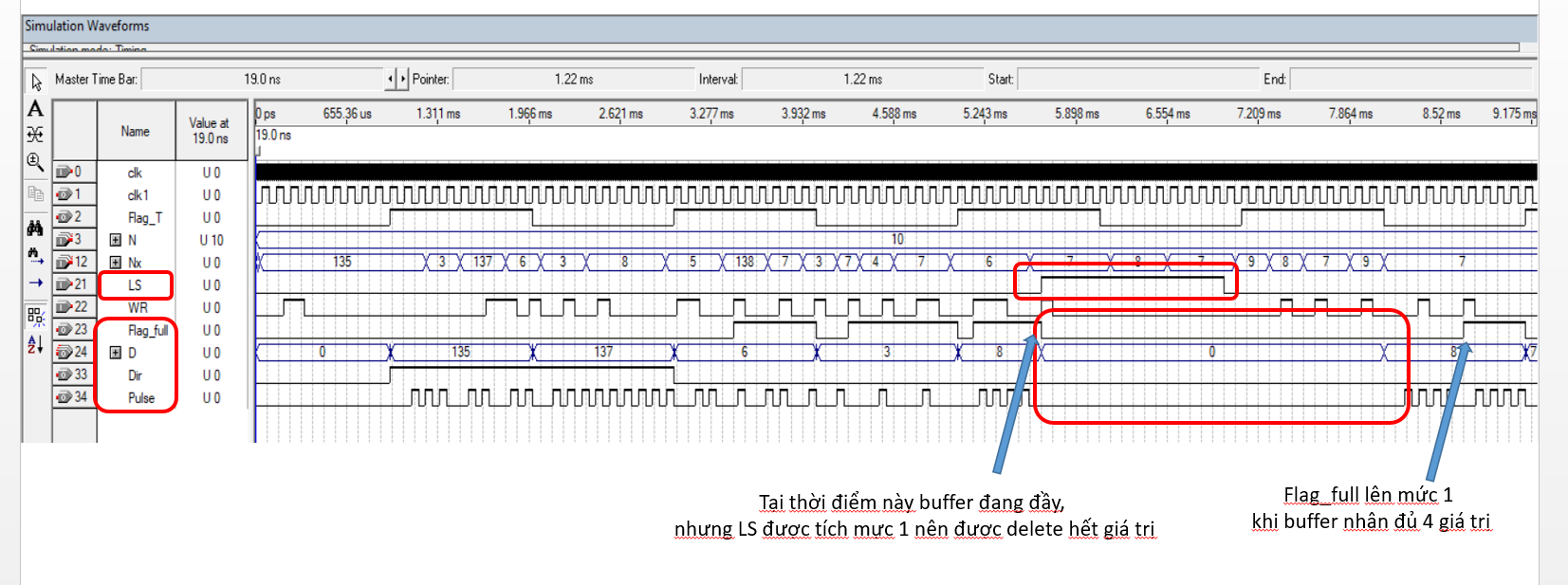
Phân tích kết quả có được:



*Nhận xét:* Tại thời điểm khi các buffer đã nhận đầy 4 giá trị thì cờ Flag\_full tích cực lên mức 1 báo hiệu là buffer đã đầy, ở chu kì tiếp theo khi buffer giải phóng 1 giá trị thì Flag\_full tích cực về mức 0.



*Nhận xét:* Chân Dir xuất ra được bit dấu từ giá trị bit thứ 7 cập nhật được từ Nx.



*Nhận xét:* Khi LS =1, ngõ ra pulse = 0, tất cả các giá trị trong buffer được xóa. Khi LS =0 thì mới thực hiện lại quá trình xuất xung theo nội suy, nhận giá trị Nx vào buffer khi có cạnh lên WR, khi buffer nhận đầy 4 giá trị thì cờ Flag\_full =1.

* Dạng module:

