**Jk触发器**

library ieee;

use ieee.std\_logic\_1164.all;

entity jk\_trigger is

port(j, k, clr, cls: in std\_logic;

clk: in std\_logic;

q, qn: buffer std\_logic);

end jk\_trigger;

architecture one of jk\_trigger is

begin

process(clk, clr, cls)

variable jk: std\_logic\_vector(1 downto 0);

begin

jk := j & k;

if clr = '0' then q <= '0'; qn <= '1';

elsif cls = '0' then q <= '1'; qn <= '0';

elsif clk'event and clk = '0' then

case jk is

when "00" => q <= q; qn <= qn;

when "01" => q <= '0'; qn <= '1';

when "10" => q <= '1'; qn <= '0';

when "11" => q <= not q; qn <= not qn;

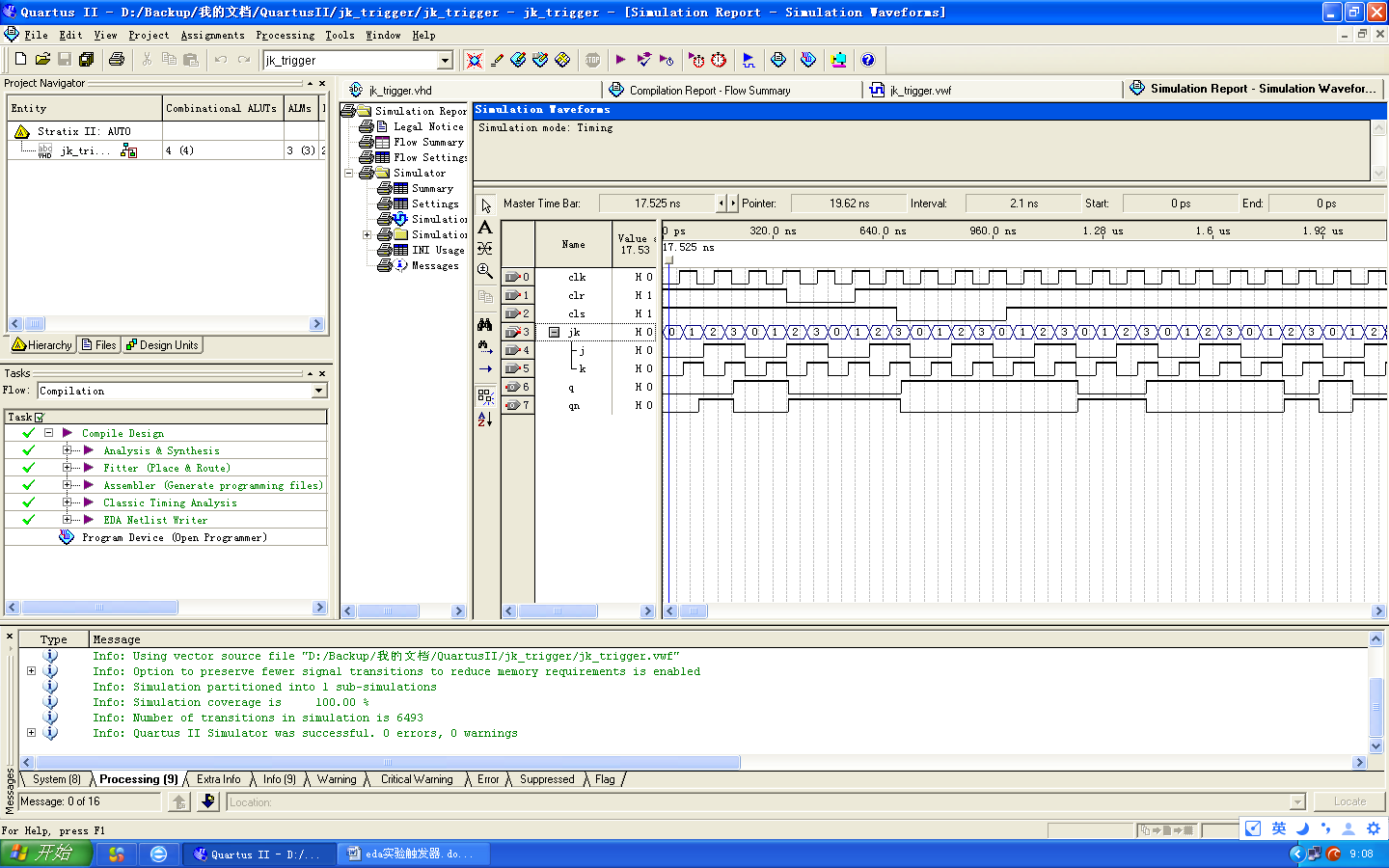
when others => null;

end case;

end if;

end process;

end one;



**Verilog HDL**

module jk\_trigger\_verilog(j, k, clk, clr, cls, q, qn);

input clk, clr, cls, j, k;

output reg q, qn;

always @(negedge clk or negedge clr or negedge cls)

begin

if(clr == 0) begin q = 0; qn = 1; end

else if(cls == 0) begin q = 1; qn = 0; end

else

begin

case ({j, k})

'b00: begin q = q; qn = qn; end

'b01: begin q = 0; qn = 1; end

'b10: begin q = 1; qn = 0; end

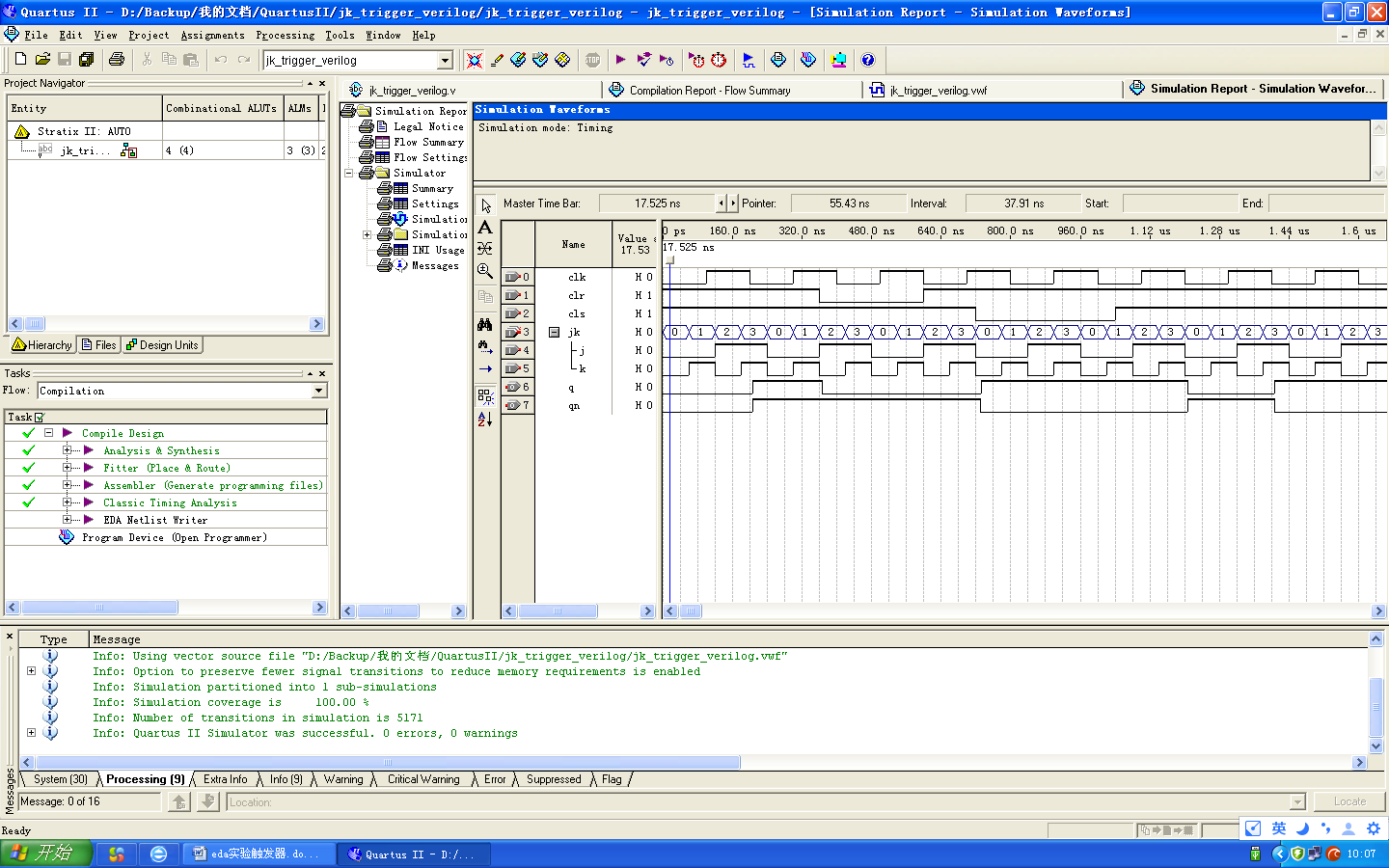
'b11: begin q = ~q; qn = ~qn; end

endcase;

end

end

endmodule



算术逻辑运算单元

module alu\_circuit(s, a, b, d);

input [2:0]s;

input [7:0]a;

input [7:0]b;

output d;

reg [7:0] d;

always @(posedge s, posedge a, posedge b)

begin

case (s)

'b000: begin d = a + b; end

'b001: begin d = a - b;end

'b010: d = a + 1;

'b011: d = a - 1;

'b100: d = ~a;

'b101: d = a & b;

'b110: d = a | b;

'b111: d = a ^ b;

endcase

end

endmodule

