

PROGETTO di RETI LOGICHE



POLITECNICO
MILANO 1863

ZHANG YUEDONG 张跃东

PREFAZIONE

Studente : ZHANG YUEDONG

Matricola : *****

Codice Persona : *****

Email : *****

Professore : Palermo Gianluca

Anno Accademico : 2018/2019

Esito di Test : Risultati positivi per tutte le simulazioni (Behavioral Simulation, Post-Synthesis Functional Simulation, Post-Synthesis Timing Simulation, Post-Implementation Functional Simulation, Post-Implementation Timing Simulation)

INDICE

INTRODUZIONE	04
STRUTTURA del CODICE	05
FSM SCELTO	08
SCHEMATIC	10
RISULTATO di SIMULATION	11
TEST PERSONALI	14

INTRODUZIONE

Questo progetto è la prova finale del corso di Reti logiche (codice insegnamento 085877) A.A. 2018 - 2019. È stato utilizzato il linguaggio VHDL per la descrizione dell'hardware, la realizzazione dell'input e dell'output e il calcolo.

Lo scopo del progetto è la ricerca dei punti che si trovano più vicini (a seconda della distanza Manhattan) al "Punto da valutare". La rete logica ha 4 porte di input e 5 porte di output.

Per ulteriori dettagli sulle regole si faccia riferimento ai materiali "Specifica _PFRL.pdf" e "REGOLE_PFRL.pdf"

STRUTTURA del CODICE

Nel codice sono stati definiti 9 signals e uno state_type per realizzare il FSM.

```
type state_type is (IDLE,RST,S0,S1,S2,S3,S4,S5,S6,S7);
signal current_state : state_type := IDLE;
signal next_state : state_type := IDLE;
signal operand_valid : std_logic := '0';
signal todo_output : std_logic := '0';
signal index_masc : integer range 0 to 7;
signal masc_di_entrata : std_logic_vector(7 downto 0) := (others => '0');
signal punt_da_valutare_x : std_logic_vector(7 downto 0) := (others => '0');
signal punt_da_valutare_y : std_logic_vector(7 downto 0) := (others => '0');
signal distance_min : std_logic_vector(15 downto 0) := (others => '0');
```

Nel codice sono presenti in totale 3 processi:

— —state_reg process

```
--state reg
state_reg:process(i_clk,i_rst)
begin
    if i_rst='1' then
        current_state <= RST;
    elsif i_clk'event and i_clk = '0' then
        current_state <= next_state;
    end if;
end process state_reg;
```

Realizza il cambiamento di stato di FSM. Quando input signal i_rst viene portato a 1, FSM entra nello stato RST, dopo di che quando il clk'event avviene e **viene portato a 0**, avviene un cambio di stato del FSM.

--lambda process

Definisce il cambiamento di stato del FSM. Come lo state_reg process, utilizza il **falling edge** per la sincronizzazione. La definizione specifica della FSM si trova nelle sezioni successive.

```
lambda: process(current_state,i_rst,i_start,operand_valid,todo_output,i_clk)
begin
  if i_rst = '1' then
    next_state <= RST;
  elsif i_clk'event and i_clk = '0' then
    case current_state is
      when IDLE =>

        when RST =>
          if i_start = '1' then
            next_state <= S0;
          end if;

        when S0 =>
          next_state <= S1;

        when S1 =>
          next_state <= S2;

        when S2 =>
          next_state <= S3;

        when S3 =>
          if operand_valid = '1' then --11/10
            next_state <= S4;
          elsif todo_output = '0' then --00
            next_state <= S3;
          else --01
            next_state <= S6;
          end if;

        when S4 =>
          if operand_valid = '1' then --10/11
            next_state <= S5;
          elsif todo_output = '0' then --00
            next_state <= S3;
          else --01
            next_state <= S6;
          end if;

        when S5 =>
          if todo_output = '1' then --01/11
            next_state <= S6;
          else --00/10
            next_state <= S3;
          end if;

        when S6 =>
          next_state <= S7;

        when S7 =>
          next_state <= S7;

      end case;
    end if;
  end process lambda ;
```

— —delta process

Definisce l'esecuzione dei vari stati. Come nei processi precedenti, utilizza il **falling edge** per la sincronizzazione. Il processo contiene 3 variabili:

- difference_value indica la differenza di coordinate tra il punto da valutare per ogni centroide;
- punt_centroide_x e punt_centroide_y indicano rispettivamente la coordinata del centroide sull'asse x e l'asse y.

```
---Define state
delta:process(current_state,i_clk,i_start,i_data,todo_output,index_masc,
              masc_di_entrata,operand_valid,punt_da_valutare_x, punt_da_valutare_y,distance_min)

variable difference_value :std_logic_vector(15 downto 0) := (others => '0');
variable punt_centroide_x : std_logic_vector(7 downto 0) := (others => '0');
variable punt_centroide_y : std_logic_vector(7 downto 0) := (others => '0');

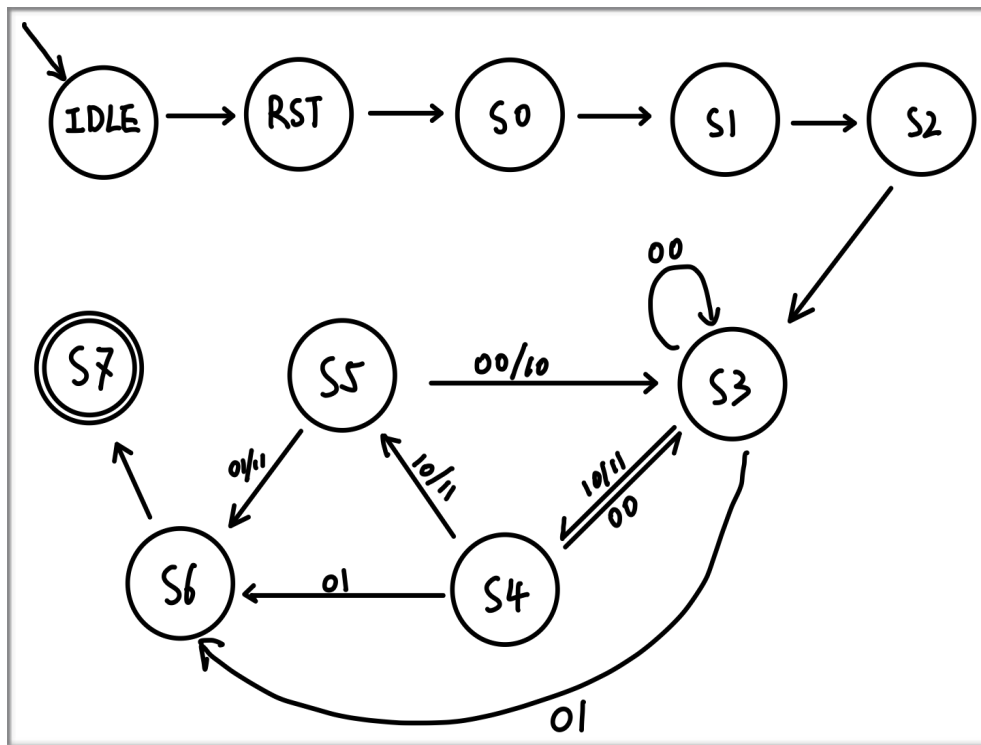
begin
  if i_clk'event and i_clk = '0' then
    case current_state is
```

FSM SCELTO

Signal:

Signal operand_valid: Indica se il valore di maschera è valido.

Signal todo_output: Indica se è possibile effettuare un output.



FSM Minimo :

	00	01	10	11
S3	S3	S6	S4	S4
S4	S3	S6	S5	S5
S5	S3	S6	S3	S6

Stato:

IDLE: Attende l'i_rst signal senza effettuare nessun lavoro.

RST: Inizializza tutti i signal e i variabili , quando l'i_start signal viene portato a 1, alza l'o_en signal a 1.

S0: Memorizza il valore di maschera.

S1: Memorizza il valore X del punto da valutare.

S2: Memorizza il valore Y del punto da valutare.

S3: Trova il bit valido della maschera.

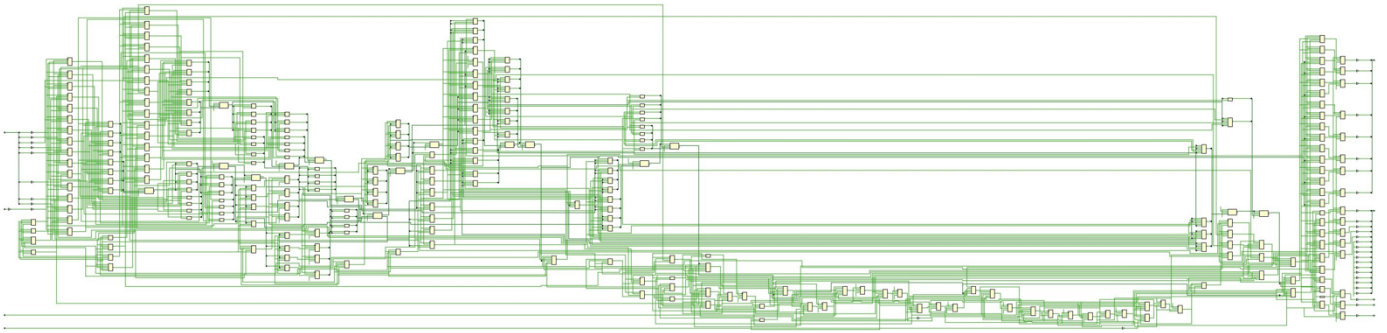
S4: Calcola la distanza sull'asse X tra il centroide e il punto da valutare, e lo compara con la distanza minima trovata in precedenza (512 come valore iniziale), se fosse più grande, non entrerà nello stato S5, assegna il valore 0 all'operand_valid signal e asseconda del todo_output signal il FSM entrerà nello stato S3 oppure S6.

S5: Calcola la distanza sull'asse Y, e lo confronta il difference_value con la distanza minima.

S6: Effettua l'output.

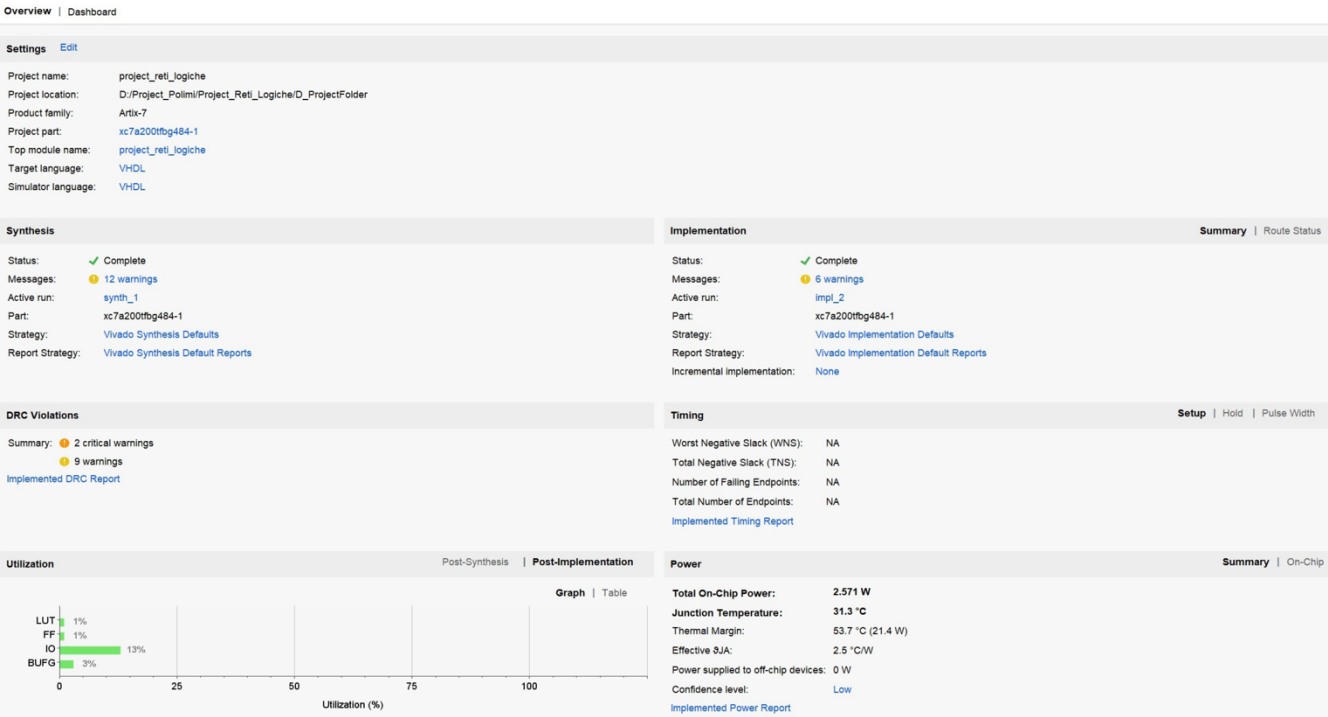
S7: Pone l'o_done signal a 0.

SCHEMATIC

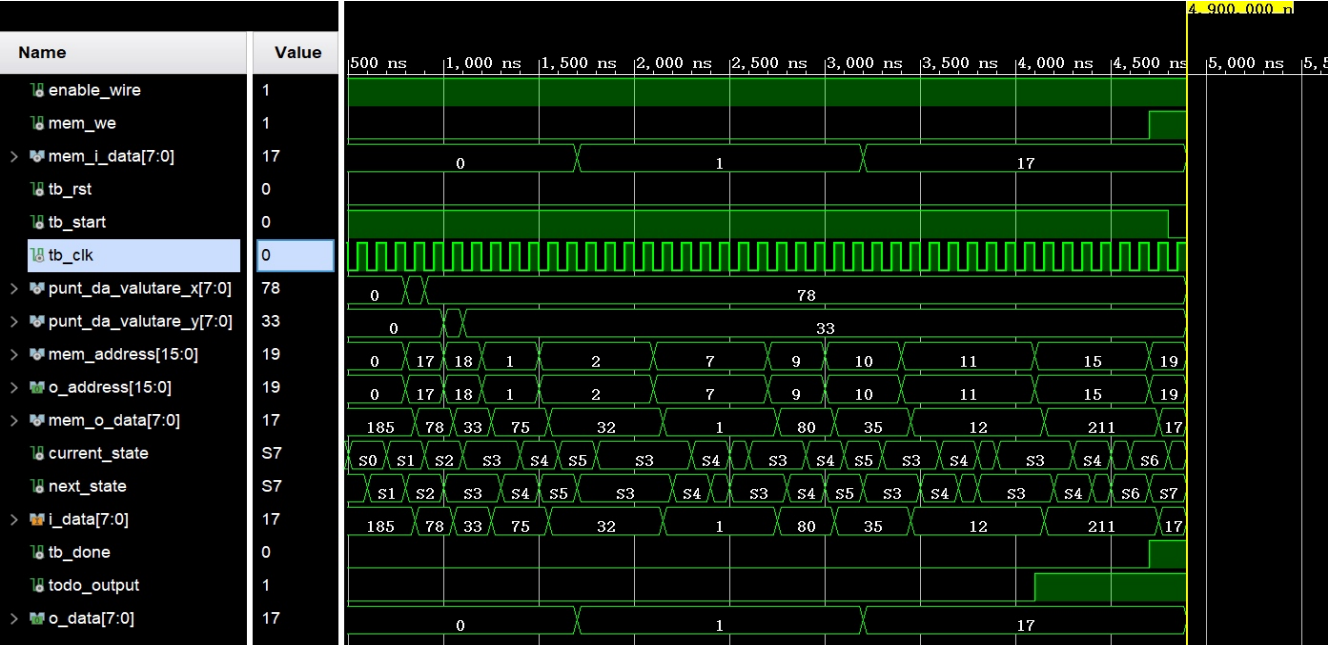


RISULTATO di SIMULATION

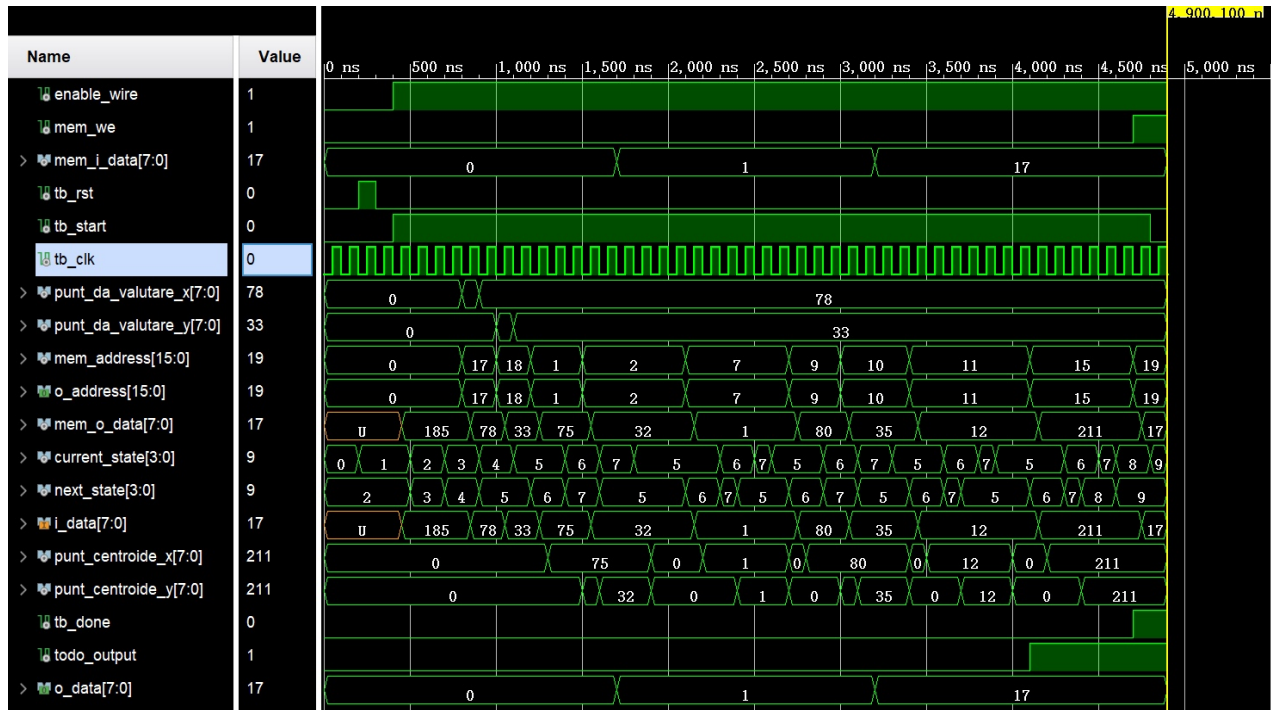
Project Summary



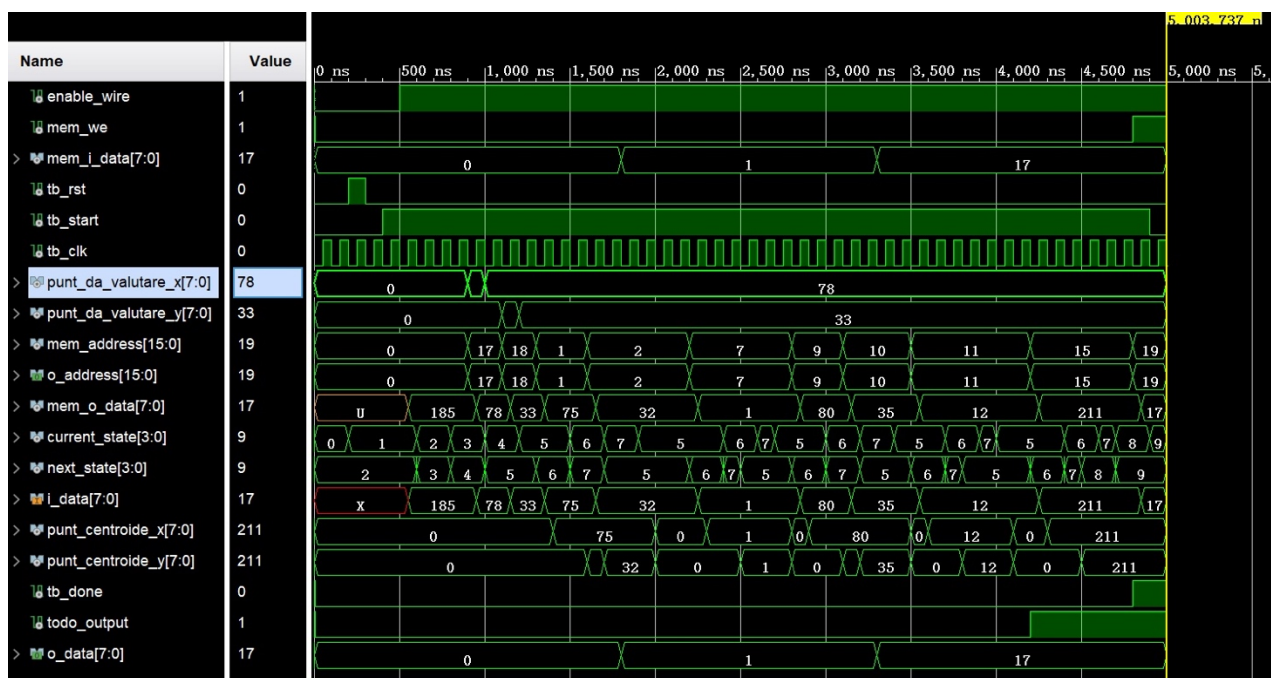
Behavioral Simulation



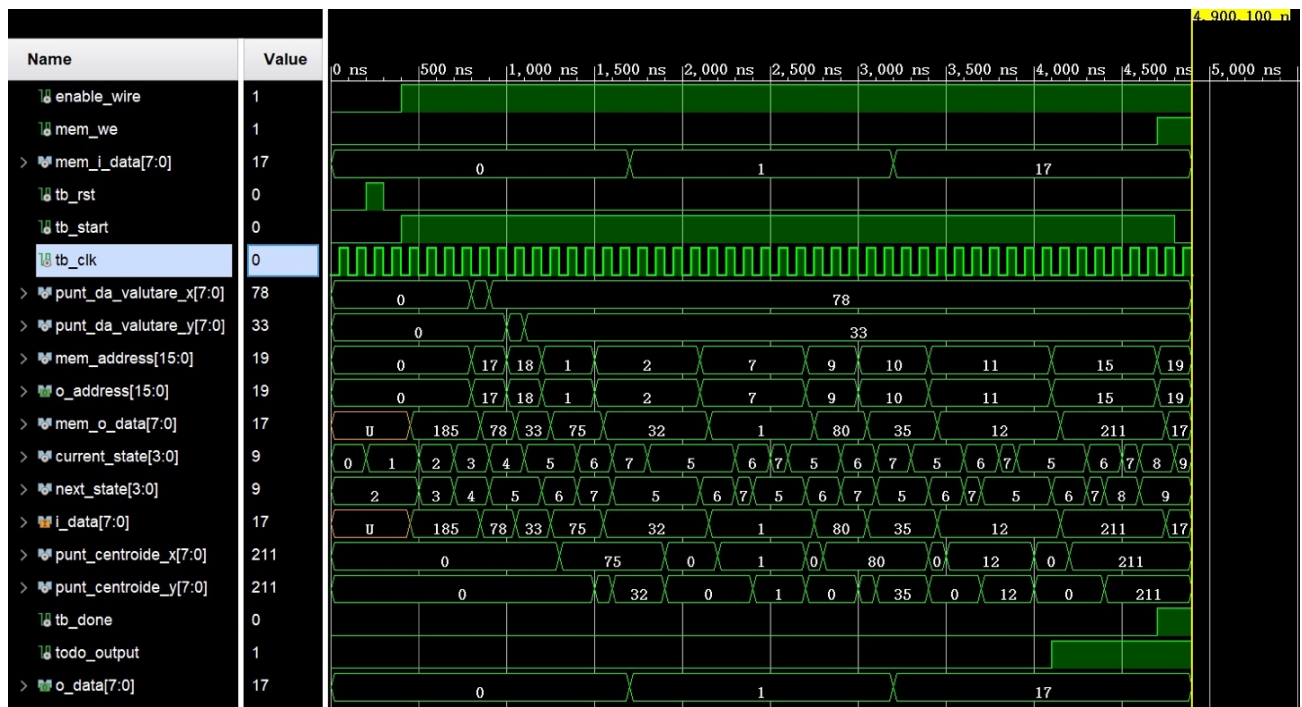
Post-Synthesis Functional Simulation



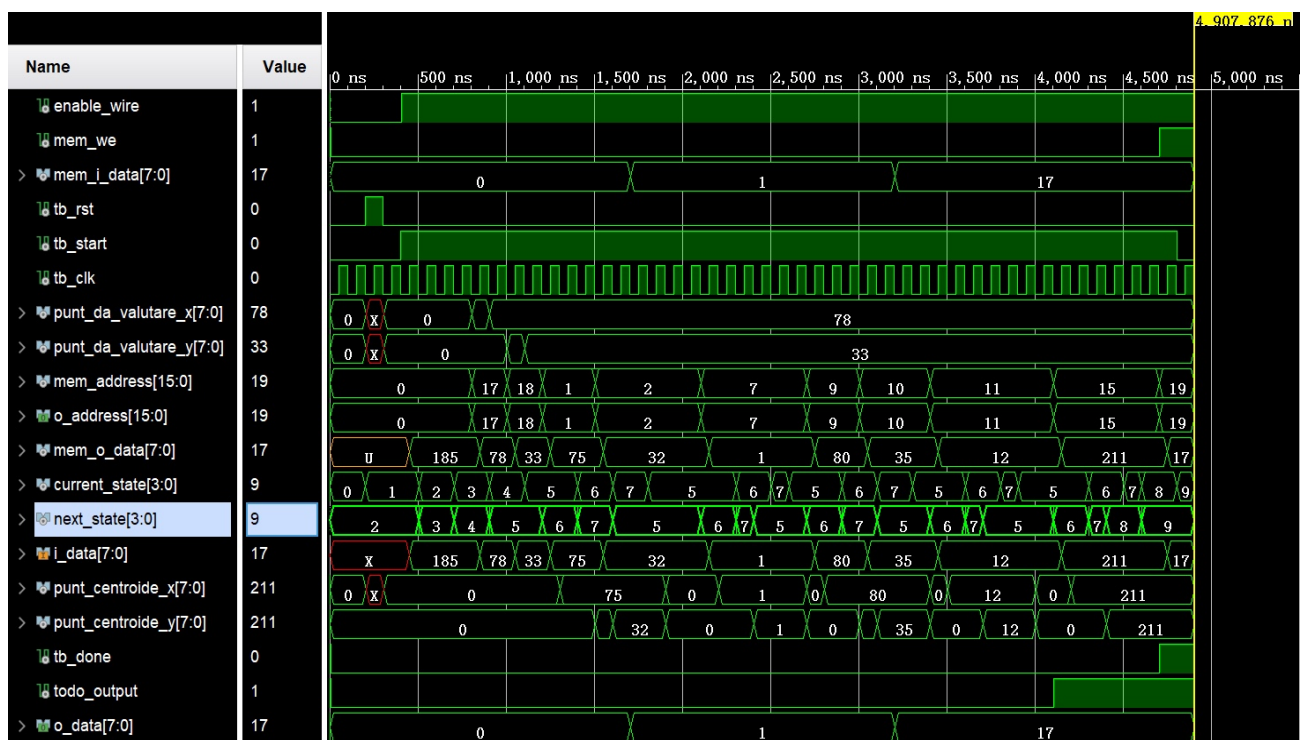
Post-Synthesis Timing Simulation



Post-Implementation Functional Simulation



Post-Implementation Timing Simulation



TEST PERSONALI

- All 0 test : Ponendo la maschera a "0000 0000"
- All 1 test : Ponendo la maschera a "1111 1111"
- Edge test : Ponendo il punto da valutare a (0 , 0) e il centroide a (255 , 255)
- Altri test assegnando valori casuali.