**Lavoro di gruppo di:**

**848834 - Ali El Wahsh  
847218 – Mathyas Giudici**

DOCUMENTAZIONE ALLEGATA AL PROGETTO DI RETI LOGICHE

# PREMESSA

Al fine di poter implementare una rete logica in grado di calcolare l’area del rettangolo minimo che circoscrive la figura in memoria (dato un valore di soglia) si decide di sviluppare il progetto verso un’ottimizzazione in frequenza.

Se da un lato si ha un vantaggio nella velocità della risposta del sistema si perderà dal punto di vista di dimensionale; il circuito sintetizzato a partire dal codice VHDL occuperà una considerevole quantità di silicio.

L’entità project\_reti\_logiche da noi implementata si sviluppa nelle seguenti macro-componenti:

* Memoria interna;
* Comparatore di soglia;
* Convertitore Coordinate;
* Logica visita immagine;
* Moltiplicatore e salvataggio in memoria.

# memoria INTERNa

Per poter implementare i singoli elementi di memoria si utilizza l’entità **registro**:

entity registro is

Port( clk : in std\_logic;

i\_set : in std\_logic;

i\_reset : in std\_logic;

i\_mem : in std\_logic\_vector (7 downto 0);

o\_mem : out std\_logic\_vector (7 downto 0));

end registro;

architecture Behavioral of registro is

begin

process(clk)

begin

if(clk'event and clk='0') then

if(i\_set='1' and i\_reset ='0') then

o\_mem <= i\_mem;

end if;

if(i\_set='0' and i\_reset ='1') then

o\_mem <= "XXXXXXXX";

end if;

end if;

end process;

end Behavioral;

Il registro è un componente regolato da un segnale di clock. Sul fronte di discesa del segnale di clock in caso vi sia un segnale *i\_set* in ingresso con valore logico ‘1’ il vettore in ingresso *i\_mem* viene salvato in uscita (*o\_mem*). Nel caso invece vi sia in ingresso un segnale *i\_reset* con valore logico ‘1’; l’uscita viene resettata e posta al valore unknown. Non sono ammesse in ingreso al registro configurazioni *i\_set=’0’ i\_set=’0’*, oppure *i\_set=’1’ i\_set=’1*.

Al fine di poter ottimizzare al meglio la reperibilità dei dati si è pensato di implementare all’interno della nostro entità project\_reti\_logiche una piccola memoria con 7 registri di memoria.

entity memoria\_interna is

Port( i\_clk : in std\_logic;

i\_reset : in std\_logic;

i\_set : in std\_logic;

i\_addr: in std\_logic\_vector(2 downto 0);

i\_mem : in std\_logic\_vector (7 downto 0);

o\_mem\_soglia : out std\_logic\_vector (7 downto 0);

o\_mem\_colonne : out std\_logic\_vector (7 downto 0);

o\_mem\_righe : out std\_logic\_vector (7 downto 0);

o\_mem\_nord : out std\_logic\_vector (7 downto 0);

o\_mem\_sud : out std\_logic\_vector (7 downto 0);

o\_mem\_west : out std\_logic\_vector (7 downto 0);

o\_mem\_est: out std\_logic\_vector (7 downto 0));

end memoria\_interna;

La memoria interna è progettata in modo che attraverso un unico segnale di reset (*i\_reset*) tutti i registri al suo interno vengano resettati e portati al valore unknown. Attraverso un segnale *i\_set* ed un segnale per l’indirizzo *i\_addr* può essere deciso su quale registro salvare il byte in ingresso *i\_mem*,

Gli indirizzi sono associati al loro valore in questo ordine:

1. Valore della soglia sfondo/immagine;
2. Numero di colonne;
3. Numero di righe;
4. Numero riga nord;
5. Numero riga sud;
6. Numero colonna est;
7. Numero colonna ovest.

Si è ritenuto opportuno avere in uscita dall’entità memoria\_interna tutti i valori salvati in memoria perché sono dati utilizzati in molti confronti e questa soluzione ne facilita la facile reperibilità all’interno dell’entità project\_reti\_logiche.