

Prova finale  
**Progetto di reti logiche**  
2018 - 19

Alessandro Nazzari – 10546223 (866419)  
Giuseppe Andrea Luzzi – 10565152 (868057)

# Indice

Indice .....	2
Presentazione del progetto e dell'algoritmo .....	3
Diagramma di flusso dell'algoritmo .....	4
Struttura della macchina a stati finiti .....	5
Segnali interni .....	6
Stati .....	7
Test generale .....	8
Test specifici .....	9
Test #1: centroidi equidistanti .....	9
Test #2: centroidi a distanza decrescente .....	10
Test #3: massima distanza .....	11
Test #4: minima distanza .....	11
Test #5: nessun centroide valido .....	12
Test #6: reset improvviso .....	12
Test #7: due esecuzioni con cambio di RAM .....	13
Conclusioni .....	14
Allegato 1: Report di sintesi .....	15

# Presentazione del progetto e dell'algoritmo

Il progetto consiste in un unico componente strutturato come una macchina a stati finiti con reset asincrono.

A partire da uno stato iniziale in cui si inizializzano tutti i segnali interni il componente procede leggendo dalla memoria la maschera d'ingresso e le coordinate del centro.

Ottenuti questi dati il componente verifica, tramite un AND bit a bit tra la maschera e un identificativo del centroide a 8 bit in codifica one-hot inizialmente pari a 00000001, se il centroide è da considerare.

In caso positivo, prosegue chiedendo alla RAM le coordinate del centroide attualmente considerato e calcolandone la distanza dal centro.

Si prosegue poi con un controllo tra la distanza ottenuta e la minima distanza salvata, inizialmente impostata al massimo valore possibile su 8 bit ovvero 511. Se la distanza calcolata risulta essere minore della più piccola distanza calcolata fino a questo momento, si procede con l'aggiornamento del valore minimo e della maschera di output; se invece la distanza risulta pari a quella minima precedentemente trovata si provvede al solo aggiornamento della maschera di output tramite un OR bit a bit con l'identificativo del centroide. Infine, nel caso in cui la distanza trovata sia maggiore di quella minima si lasciano invariati tutti i valori.

Il processo si ripete per tutti i centroidi, ovvero fino a quando l'identificativo del centroide ottenuto con uno shift a sinistra raggiungerà il valore 00000000.

Analizzati gli 8 centroidi, il componente prosegue in uno stato finale in cui provvede a scrivere in RAM la maschera di output calcolata e a impostare il segnale *o\_done* a 1, il segnale *o\_done* manterrà il valore 1 fino a quando *i\_start* non tornerà al valore 0.

La scelta implementativa è stata quella di creare due process distinti:

1. **Processo di clock**

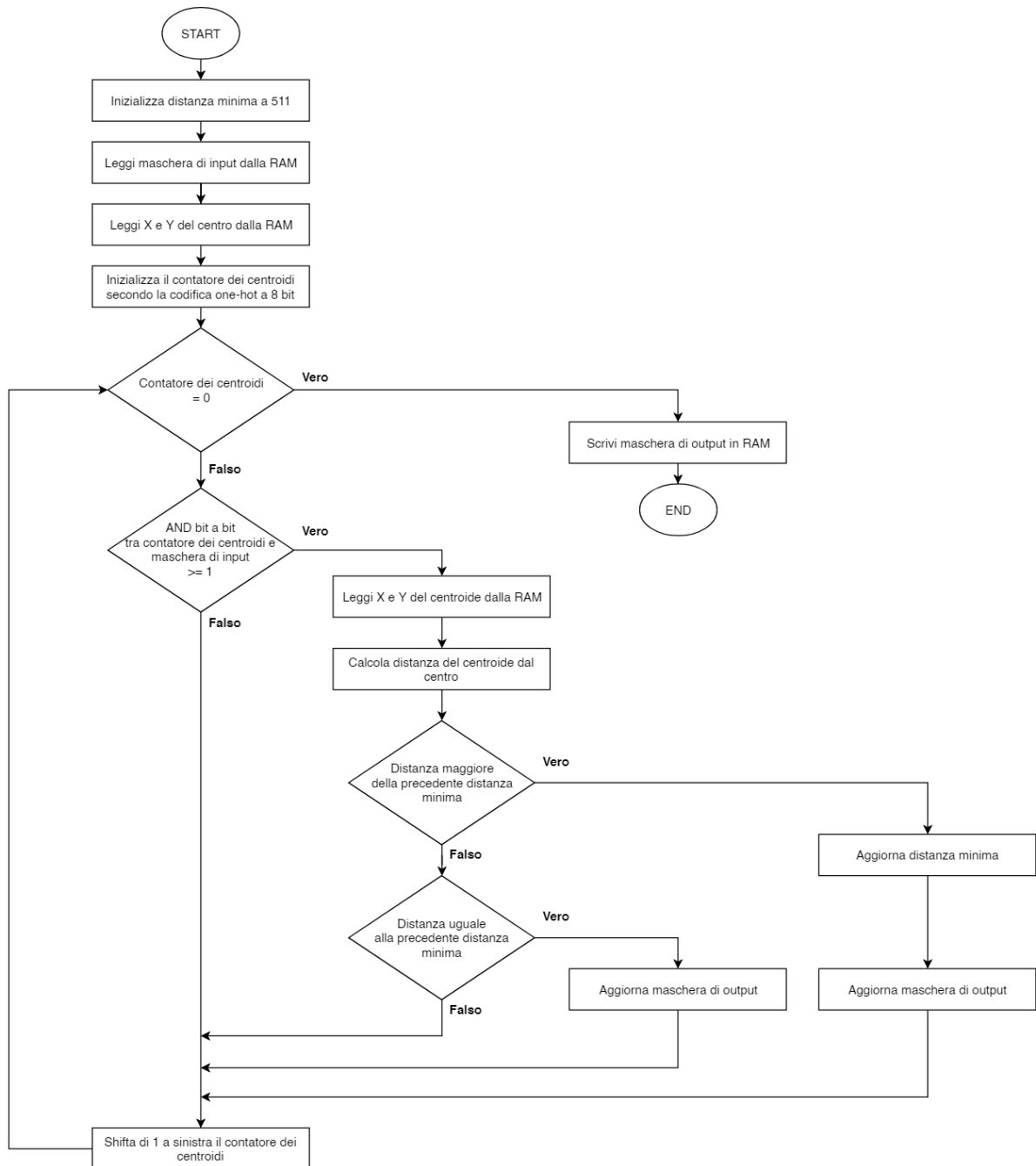
Si attiva ad ogni fronte di salita del clock ed è responsabile dell'aggiornamento di tutti i segnali rappresentanti lo stato attuale dell'automa con il loro valore futuro.

2. **Processo combinatorio**

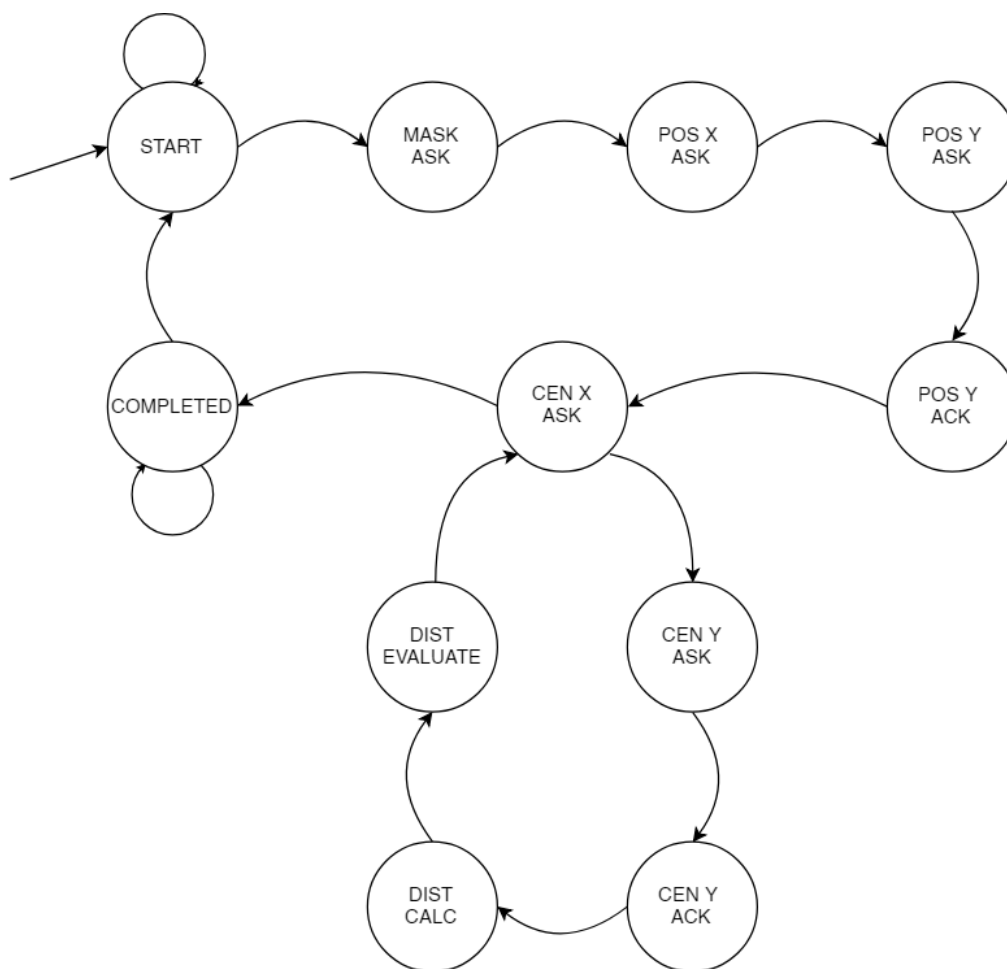
Si attiva a ogni cambio di stato dell'automa ed è responsabile del calcolo di tutti i valori futuri e più in generale dell'intera computazione.

Per ridurre al minimo gli accessi alla memoria e quindi i tempi di esecuzione, la maschera di output viene calcolata prima su un segnale interno e aggiornata in memoria solo alla fine della computazione. Inoltre, in questo modo la RAM rappresenta sempre uno stato coerente dell'informazione e non sensibile agli stati intermedi della computazione. Anche le informazioni riguardanti la maschera, la posizione iniziale e la posizione del centroide attualmente considerato sono salvate in dei segnali interni con lo scopo di velocizzare la computazione.

# Diagramma di flusso dell'algoritmo



## Struttura della macchina a stati finiti



# Segnali interni

Il componente fa uso di alcuni segnali interni, per ognuno di essi è previsto un segnale ausiliario caratterizzato dal suffisso “\_next” da utilizzare per assegnare il valore futuro al rispettivo segnale.

- **state\_curr - state\_next**  
Stato corrente e stato futuro della macchina a stati finiti.  
*In VHDL rappresentato con un tipo personalizzato “eg\_state\_type” per indicare ogni possibile stato.*
- **current\_x - current\_x\_next**  
Valore della coordinata X del centroide attualmente considerato e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(7 downto 0) inizializzato a 00000000.*
- **current\_y - current\_y\_next**  
Valore della coordinata Y del centroide attualmente considerato e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(7 downto 0) inizializzato a 00000000.*
- **pos\_x - pos\_x\_next**  
Valore della coordinata X del centro e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(7 downto 0) inizializzato a 00000000.*
- **pos\_y - pos\_y\_next**  
Valore della coordinata Y del centro e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(7 downto 0) inizializzato a 00000000.*
- **min\_distance - min\_distance\_next**  
Valore della minima distanza trovata e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(8 downto 0) inizializzato a 11111111.*
- **current\_distance - current\_distance\_next**  
Valore della distanza calcolata per il centroide attuale e valore futuro  
*In VHDL rappresentato con std\_logic\_vector(8 downto 0) inizializzato a 11111111.*
- **mask - mask\_next**  
Maschera di input e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(7 downto 0) inizializzato a 00000000.*
- **config\_cursor - config\_cursor\_next**  
Indirizzo della RAM per la coordinata necessaria del centroide attualmente considerato e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(7 downto 0) inizializzato a 00000000.*
- **current\_mask - current\_mask\_next**  
Identificativo del centroide attualmente considerato secondo una codifica one-hot e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(7 downto 0) inizializzato a 00000000.*
- **output\_mask - output\_mask\_next**  
Maschera di output e valore futuro.  
*In VHDL rappresentato con std\_logic\_vector(7 downto 0) inizializzato a 00000000.*

# Stati

Gli stati che compongono l'automa e le loro rispettive funzionalità sono:

1.     **START**  
Stato iniziale in cui tutti i segnali vengono inizializzati. La macchina rimarrà in questo stato fino a quando *i\_start* non diventerà 1. Svolge anche la funzione di stato di reset.
2.     **MASK\_\_ASK**  
Il componente richiede il valore della maschera di input alla RAM.
3.     **POS\_\_X\_\_ASK**  
Il componente riceve il valore della maschera di input e richiede il valore della coordinata X del centro alla RAM.
4.     **POS\_\_Y\_\_ASK**  
Il componente riceve il valore della coordinata X del centro e richiede il valore della coordinata Y del centro alla RAM.
5.     **POS\_\_Y\_\_ACK**  
Il componente riceve il valore della coordinata Y del centro alla RAM.
6.     **CEN\_\_X\_\_ASK**  
Il componente verifica se sono presenti altri centroidi da valutare. In caso positivo valuta se il centroide da analizzare rispetta la maschera e quindi chiede la sua coordinata X alla RAM.  
In caso negativo il componente si porta nello stato finale COMPLETED.
7.     **CEN\_\_Y\_\_ASK**  
Il componente riceve il valore della coordinata X del centroide attualmente considerato e richiede il valore della coordinata Y.
8.     **CEN\_\_Y\_\_ACK**  
Il componente riceve il valore della coordinata Y del centroide attualmente considerato.
9.     **DIST\_\_CALC**  
Il componente calcola la distanza tra il centro e il centroide attualmente considerato.
10.    **DIST\_\_EVALUATE**  
Il componente confronta la distanza calcolata con la minima distanza ottenuta precedentemente ed eventualmente provvede ad aggiornare distanza minima e la maschera di output.
11.    **COMPLETED**  
Il componente scrive in RAM la maschera di output calcolata e porta il segnale *o\_done* a 1, il quale verrà mantenuto a 1 fino a quando il segnale *i\_start* non avrà assunto il valore 0.

# Test generale

- **Risultato previsto**

Maschera di output: 00010001

- **Risultato effettivo**

Maschera di output: 00010001

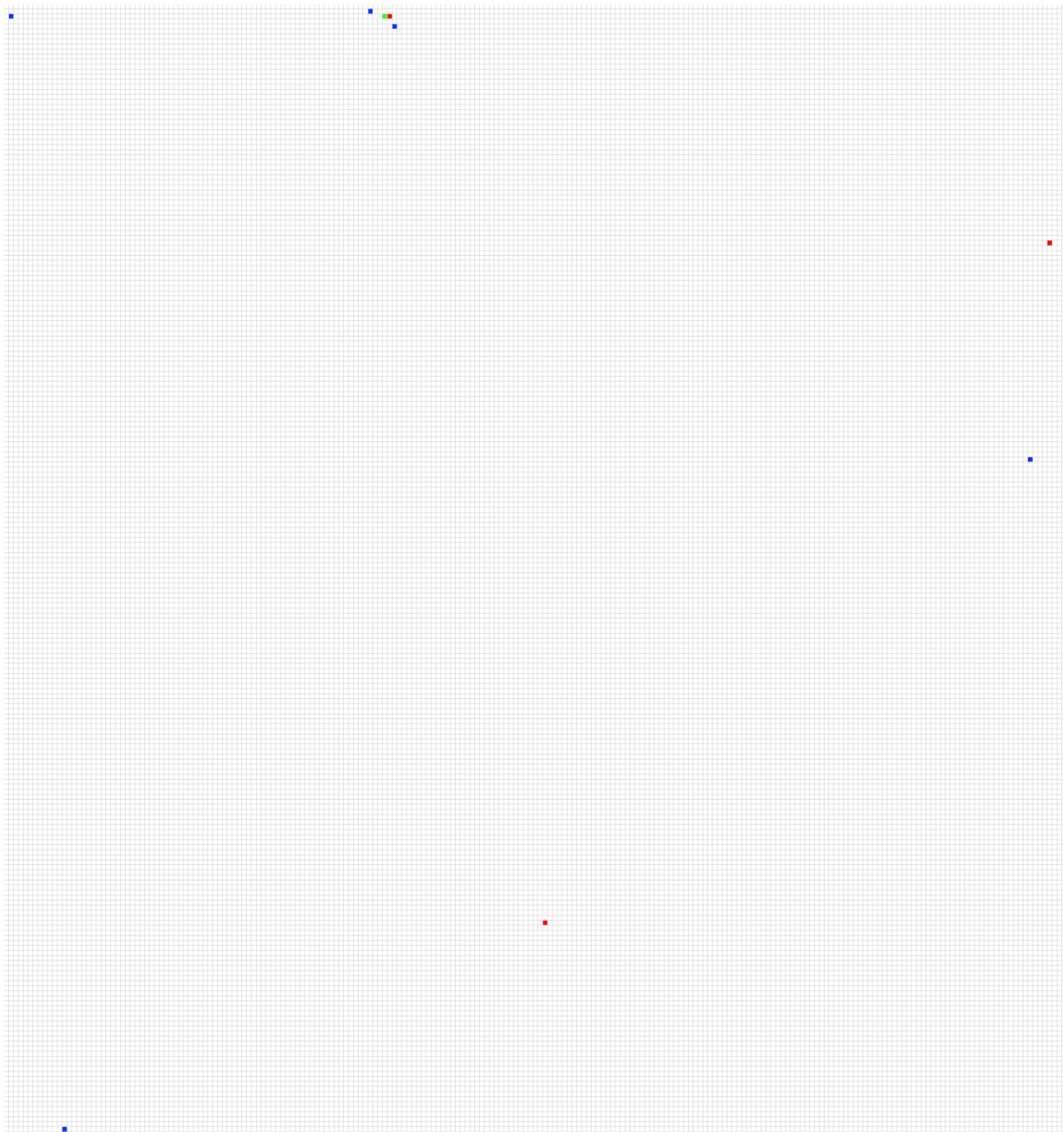
- **Benchmark**

Behavioral: 5250ns

Post-Synthesis Functional: 5250100ps

Post-Synthesis Timing: 5257654ps

- **Rappresentazione**



Legenda: • centroide valido • centroide non valido • centro



# Test specifici

Di seguito viene presentato un set di test realizzato con lo scopo di testare il comportamento del componente in alcune situazioni particolari. Per ogni test è fornita una breve spiegazione con il suo obiettivo, le informazioni sui tempi impiegati dal componente per completarlo e quando necessario una rappresentazione della griglia secondo la stessa legenda del test generale.

## Test #1: centroidi equidistanti

- **Obiettivo**

Valutare il comportamento del componente con 8 centroidi equidistanti e validi, in questo modo la maschera andrà sempre in OR con la maschera precedente (segnale *current\_mask*) e mai sovrascritta.

- **Risultato previsto**

Distanza: 20

Maschera di output: 11111111

- **Risultato effettivo**

Distanza: 20

Maschera di output: 11111111

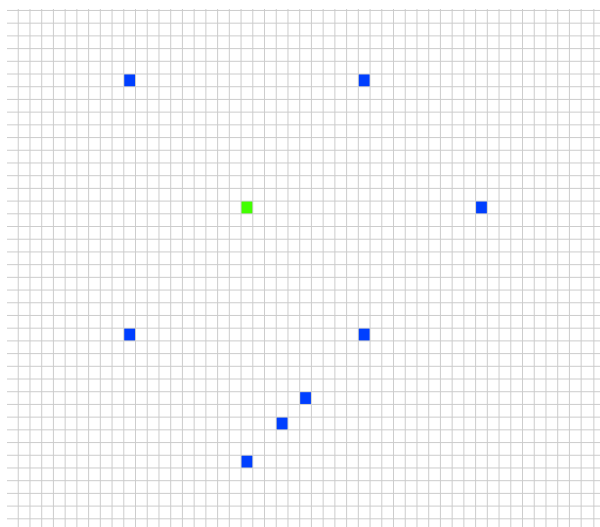
- **Benchmark**

Behavioral: 5150ns

Post-Synthesis Functional: 5150100ps

Post-Synthesis Timing: 5157589ps

- **Rappresentazione**



## Test #2: centroidi a distanza decrescente

- **Obiettivo**

Valutare il comportamento del componente con 8 centroidi validi posti a distanza decrescente, in questo modo la maschera andrà sovrascritta interamente e mai in OR con la maschera precedente (segnale *current\_mask*).

- **Risultato previsto**

Distanza: 2

Maschera di output: 10000000

- **Risultato effettivo**

Distanza: 2

Maschera di output: 10000000

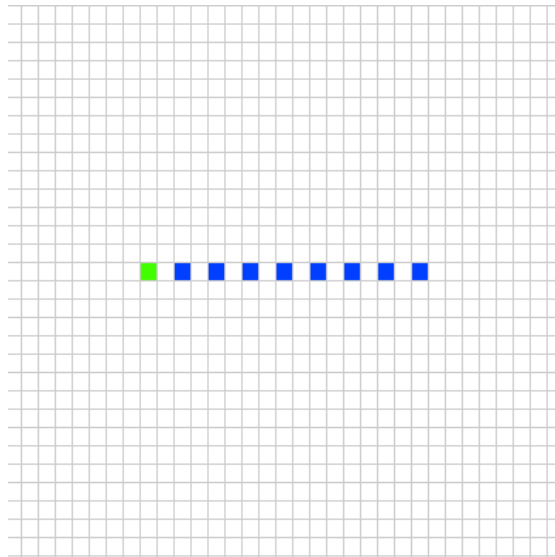
- **Benchmark**

Behavioral: 5150ns

Post-Synthesis Functional: 5150100ps

Post-Synthesis Timing: 5157589ps

- **Rappresentazione**



## Test #3: massima distanza

- **Obiettivo**

Solo il primo centroide è da analizzare e si trova nell'angolo in basso a destra della matrice. Il centro è invece posizionato nell'angolo opposto, in alto a sinistra, andando così a valutare la capacità del componente di salvare correttamente la massima distanza ottenibile in una matrice 256x256 cioè 510.

- **Risultato previsto**

Distanza: 510

Maschera di output: 10000000

- **Risultato effettivo**

Distanza: 510

Maschera di output: 10000000

- **Benchmark**

Behavioral: 2350ns

Post-Synthesis Functional: 2350100ps

Post-Synthesis Timing: 2357654ps

## Test #4: minima distanza

- **Obiettivo**

Solo il primo centroide è da analizzare e si trova sovrapposto al centro rendendo così la distanza tra i due pari a 0.

- **Risultato previsto**

Distanza: 0

Maschera di output: 10000000

- **Risultato effettivo**

Distanza: 0

Maschera di output: 10000000

- **Benchmark**

Behavioral: 5150ns

Post-Synthesis Functional: 5150100ps

Post-Synthesis Timing: 5157654ps

## Test #5: nessun centroide valido

- **Obiettivo**

La maschera d'ingresso vale 00000000 e quindi non si deve considerare nessun centroide.

È da notare come il tempo impiegato per completare questo test sia sensibilmente più basso di quello richiesto dagli altri test, questo è conforme alle aspettative in quanto il componente verifica la validità del centroide prima di qualsiasi altra operazione rendendo in questo caso inutile il passaggio in tutti gli stati successivi a CEN\_X\_ASK.

- **Risultato previsto**

Maschera di output: 00000000

- **Risultato effettivo**

Maschera di output: 00000000

- **Benchmark**

Behavioral: 1950ns

Post-Synthesis Functional: 1950100ps

Post-Synthesis Timing: 1957654ps

## Test #6: reset improvviso

- **Obiettivo**

Dopo l'inizio dell'esecuzione la testbench attende 10 cicli di clock prima di resettare il componente e far ripartire dall'inizio l'esecuzione.

Lo scopo è quello di verificare la funzionalità di reset del componente e che non si verifichino comportamenti imprevisti e interferenze tra le due esecuzioni.

- **Risultato previsto**

Maschera di output: 00010001

- **Risultato effettivo**

Maschera di output: 00010001

- **Benchmark**

Behavioral: 5250ns

Post-Synthesis Functional: 5250100ps

Post-Synthesis Timing: 5257654ps

## Test #7: due esecuzioni con cambio di RAM

- **Obiettivo**

Dopo una prima esecuzione andata a buon fine, la testbench resetta il componente e sostituisce tutte le informazioni presenti in RAM prima di far ripartire l'esecuzione. Lo scopo è quello di verificare che non si verifichino interferenze tra le due esecuzioni.

- **Risultato previsto**

Prima maschera di output: 10100110

Seconda maschera di output: 00000010

- **Risultato effettivo**

Prima maschera di output: 10100110

Seconda maschera di output: 00000010

- **Benchmark**

Behavioral: 7250ns

Post-Synthesis Functional: 7250100ps

Post-Synthesis Timing: 7257654ps

# Conclusioni

L'automa realizzato sarebbe ulteriormente minimizzabile sacrificando la significatività dei singoli stati, tuttavia si è scelto di non condensarlo ulteriormente in modo da mantenere divise le singole funzionalità degli stati anche in vista di possibili modifiche future.

Infatti, grazie alla divisione netta degli stati, nel caso di una complessità maggiore del problema sarebbe possibile dividere l'automa in più moduli da sviluppare singolarmente in modo da ridurre le scelte affidate al sintetizzatore di Vivado.

In questo caso, visto il numero ridotto di centroidi da analizzare e le dimensioni ridotte della griglia la scelta adottata è risultata ottimizzata.

# Allegato 1: Report di sintesi

[...]

Finished RTL Elaboration : Time (s): cpu = 00:00:05 ; elapsed = 00:00:09 . Memory (MB): peak = 398.258 ; gain = 149.688

[...]

Report Check Netlist:

	Item	Errors	Warnings	Status	Description
1	multi_driven_nets	0	0	Passed	Multi driven nets

[...]

INFO: [Synth 8-802] inferred FSM for state register 'state\_cur\_reg' in module 'project\_reti\_logiche'  
INFO: [Synth 8-5546] ROM "o\_address" won't be mapped to RAM because it is too sparse  
INFO: [Synth 8-5546] ROM "o\_en" won't be mapped to RAM because it is too sparse  
INFO: [Synth 8-5546] ROM "config\_cursor\_next" won't be mapped to RAM because it is too sparse  
INFO: [Synth 8-5546] ROM "state\_next" won't be mapped to RAM because it is too sparse  
INFO: [Synth 8-5544] ROM "state\_next" won't be mapped to Block RAM because address size (1) smaller than threshold (5)

State	New Encoding	Previous Encoding
start	00000000001	0000
mask_ask	00000000010	0001
pos_x_ask	00000000100	0010
pos_y_ask	00000001000	0011
pos_y_ack	00000010000	0100
cen_x_ask	00000100000	0101
completed	00001000000	1010
cen_y_ask	00010000000	0110
cen_y_ack	00100000000	0111
dist_calc	01000000000	1000
dist_evaluate	10000000000	1001

INFO: [Synth 8-3354] encoded FSM with state register 'state\_cur\_reg' using encoding 'one-hot' in module 'project\_reti\_logiche'

Finished RTL Optimization Phase 2 : Time (s): cpu = 00:00:20 ; elapsed = 00:00:29 . Memory (MB): peak = 785.500 ; gain = 536.930

[...]

Start RTL Component Statistics

Detailed RTL Component Info :

---Adders :  
3 Input 9 Bit Adders := 2  
2 Input 9 Bit Adders := 3  
2 Input 8 Bit Adders := 2  
---Registers :  
9 Bit Registers := 2  
8 Bit Registers := 8  
---Muxes :  
11 Input 16 Bit Muxes := 1  
11 Input 11 Bit Muxes := 1  
2 Input 11 Bit Muxes := 4  
2 Input 9 Bit Muxes := 2  
11 Input 9 Bit Muxes := 2  
2 Input 8 Bit Muxes := 4  
11 Input 8 Bit Muxes := 8  
2 Input 5 Bit Muxes := 1  
2 Input 1 Bit Muxes := 7  
11 Input 1 Bit Muxes := 13

Finished RTL Component Statistics

Start RTL Hierarchical Component Statistics

Hierarchical RTL Component report

Module project\_reti\_logiche

Detailed RTL Component Info :

---Adders :  
3 Input 9 Bit Adders := 2  
2 Input 9 Bit Adders := 3  
2 Input 8 Bit Adders := 2  
---Registers :  
9 Bit Registers := 2  
8 Bit Registers := 8  
---Muxes :  
11 Input 16 Bit Muxes := 1  
11 Input 11 Bit Muxes := 1  
2 Input 11 Bit Muxes := 4  
2 Input 9 Bit Muxes := 2  
11 Input 9 Bit Muxes := 2  
2 Input 8 Bit Muxes := 4  
11 Input 8 Bit Muxes := 8  
2 Input 5 Bit Muxes := 1  
2 Input 1 Bit Muxes := 7  
11 Input 1 Bit Muxes := 13

Finished RTL Hierarchical Component Statistics

Start Part Resource Summary

-----  
Part Resources:

DSPs: 740 (col length:100)  
BRAMs: 730 (col length: RAMB18 100 RAMB36 50)  
-----

Finished Part Resource Summary  
-----

Start Cross Boundary and Area Optimization  
-----

Warning: Parallel synthesis criteria is not met

INFO: [Synth 8-5546] ROM "state\_next" won't be mapped to RAM because it is too sparse  
INFO: [Synth 8-5546] ROM "config\_cursor\_next" won't be mapped to RAM because it is too sparse  
INFO: [Synth 8-5546] ROM "o\_en" won't be mapped to RAM because it is too sparse  
INFO: [Synth 8-5546] ROM "o\_address" won't be mapped to RAM because it is too sparse  
WARNING: [Synth 8-3917] design project\_reti\_logiche has port o\_address[15] driven by constant 0  
WARNING: [Synth 8-3917] design project\_reti\_logiche has port o\_address[14] driven by constant 0  
WARNING: [Synth 8-3917] design project\_reti\_logiche has port o\_address[13] driven by constant 0  
WARNING: [Synth 8-3917] design project\_reti\_logiche has port o\_address[12] driven by constant 0  
WARNING: [Synth 8-3917] design project\_reti\_logiche has port o\_address[11] driven by constant 0  
WARNING: [Synth 8-3917] design project\_reti\_logiche has port o\_address[10] driven by constant 0  
WARNING: [Synth 8-3917] design project\_reti\_logiche has port o\_address[9] driven by constant 0  
WARNING: [Synth 8-3917] design project\_reti\_logiche has port o\_address[8] driven by constant 0  
-----

Finished Cross Boundary and Area Optimization : Time (s): cpu = 00:00:22 ; elapsed = 00:00:31 . Memory (MB): peak = 785.500 ; gain = 536.930  
-----

[...]

Report Cell Usage:

	Cell	Count
1	IBUFG	1
2	ICARRY4	12
3	LUT2	40
4	LUT3	9
5	LUT4	39
6	LUT5	14
7	LUT6	38
8	FDCE	10
9	FDPE	1
10	FDRE	73
11	FDSSE	9
12	IBUF	11
13	OBUF	27

Report Instance Areas:

	Instance	Module	Cells
1	top		284

-----  
Finished Writing Synthesis Report : Time (s): cpu = 00:00:35 ; elapsed = 00:00:46 . Memory (MB): peak = 799.418 ; gain = 550.848  
-----

Synthesis finished with 0 errors, 0 critical warnings and 8 warnings.

Synthesis Optimization Runtime : Time (s): cpu = 00:00:20 ; elapsed = 00:00:32 . Memory (MB): peak = 799.418 ; gain = 163.605

Synthesis Optimization Complete : Time (s): cpu = 00:00:35 ; elapsed = 00:00:46 . Memory (MB): peak = 799.418 ; gain = 550.848

INFO: [Project 1-571] Translating synthesized netlist

INFO: [Netlist 29-17] Analyzing 12 Unisim elements for replacement

INFO: [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds

INFO: [Project 1-570] Preparing netlist for logic optimization

INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).

Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.001 . Memory (MB): peak = 799.715 ; gain = 0.000

INFO: [Project 1-111] Unisim Transformation Summary:

No Unisim elements were transformed.

INFO: [Common 17-83] Releasing license: Synthesis

24 Infos, 8 Warnings, 0 Critical Warnings and 0 Errors encountered.

synth\_design completed successfully

synth\_design: Time (s): cpu = 00:00:38 ; elapsed = 00:00:50 . Memory (MB): peak = 799.738 ; gain = 551.168

[...]