



UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base
Corso di Laurea Magistrale in Ingegneria Informatica

Progetto di Architettura di Sistemi Digitali

Anno Accademico 2021/2022

Gruppo 23
Conte Alfonso matr. M63001378

Indice

1	Sistema Handshaking	1
1.1	Traccia	1
1.2	Descrizione soluzione teorica	1
1.2.1	Schematici Componenti ed implementazione in VHDL	2
1.2.2	Testbenches	7

Progetto 1

Sistema Handshaking

1.1 Traccia

Progettare, implementare in VHDL e simulare il seguente sistema. Un'unità A e un'unità B trasmettono N byte ciascuna (in parallelo) ad un'unità C mediante handshaking. L'unità C, ricevuti gli N byte sia da A che da B, li memorizza e conta quanti byte in posizioni omologhe sono uguali. La comparazione fra il byte $X[i]$ ricevuto da A e il byte $Y[i]$ ricevuto da B può essere fatta appena dopo la ricezione di $X[i]$ e $Y[i]$, oppure al termine della ricezione complessiva. Lo studente può fare qualsiasi assunzione sulle modalità di invio da parte di A e B, e procedere all'implementazione di conseguenza.

1.2 Descrizione soluzione teorica

L'esercizio è stato risolto strutturalmente progettando separatamente 3 nodi: rispettivamente A, B, C. In particolare i nodi A e B presentano esattamente la stessa struttura in termini di unità operativa e di componenti. In particolare il protocollo di handshaking è stato realizzato prendendo in considerazione entrambi i segnali di "ready" dei nodi A,B procedendo alla memorizzazione delle stringhe

ricevute solo quando entrambi i nodi manifestano la volontà di trasmissione, ovvero quando sia `r_a` che `r_b` sono alti; ricevuti tali segnali il protocollo procede in maniera classica, associando ai due ack per i 2 nodi un unico segnale. Per quanto riguarda il Comparatore, esso è stato ancora una volta realizzato strutturalmente a partire da comparatori di un sol bit per ricavare il numero di bit di pari valore delle due stringhe in posizione omologa.

1.2.1 Schematici Componenti ed implementazione in VHDL

SistemaHandshaking

Il top module è rappresentato dal `SistemadiHandshaking` che risulta costituito dai tre nodi A,B e C; pertanto è stato chiaramente progettato con approccio strutturale. Di seguito se ne riporta l'interfaccia:

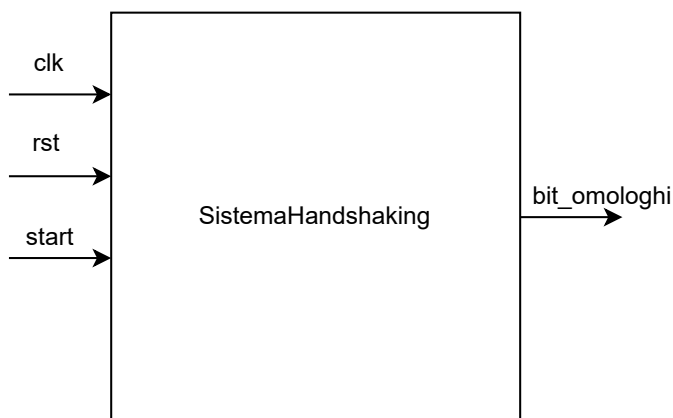


Figura 1.1: Sistema Handshaking

Esso presenta in ingresso il segnale di clock, il reset e un segnale di start che dà inizio alla comunicazione. In uscita ritroviamo invece il numero di bit omologhi delle stringhe trasmesse.

Per quanto riguarda il nodo A (e conseguentemente quello B) si è proceduto alla progettazione mediante scomposizione in unità operativa e unità di controllo. L'unità operativa in particolare risulta costituita da tre componenti:

- Una Memoria ROM contenente le N (pari a 4) stringhe da trasmettere.
- Un Contatore per indirizzare la memoria ROM e consentire la terminazione della procedura di trasmissione.
- Un registro tampone per memorizzare la stringa da trasmettere.

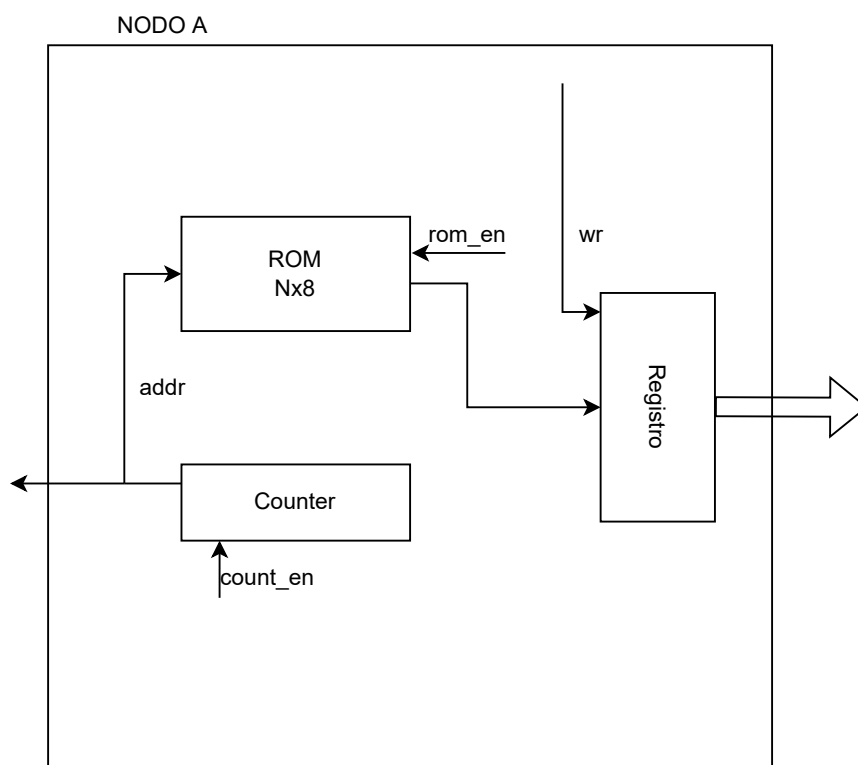


Figura 1.2: Datapath Nodo A e Nodo B

L'unità di controllo corrispondente è stata invece modellata come un automa a stati finiti. La transizione tra stati procede come stabilito dal protocollo di handshaking classico.

Control Unit Nodo A e Nodo B

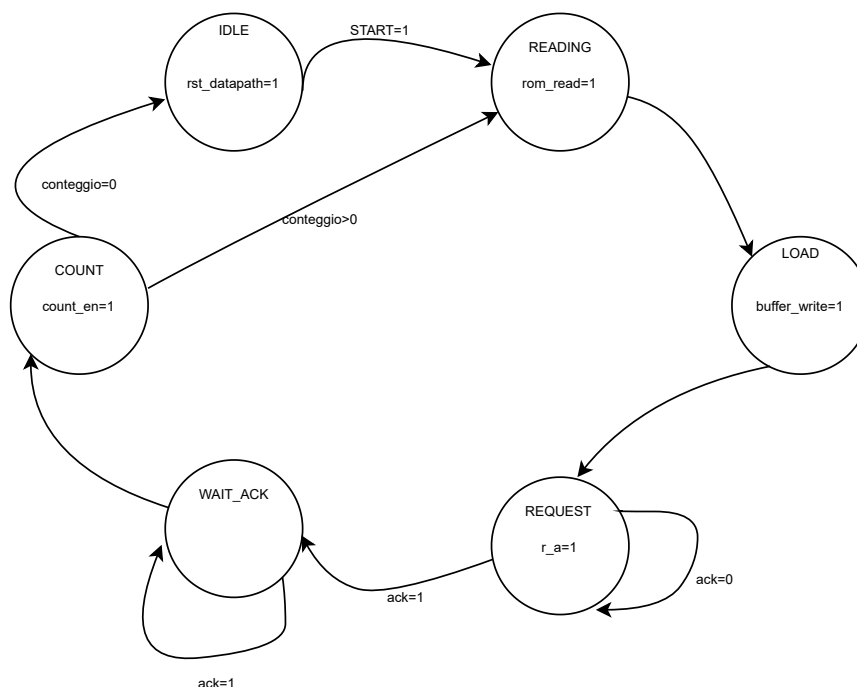


Figura 1.3: Unità di Controllo Nodo A e Nodo B

Attendiamo dunque il segnale di start in corrispondenza del quale ci si sposta nello stato di READING in cui si abilita la lettura dalla ROM, al successivo colpo di clock si scrive il contenuto prelevato dalla ROM all'interno del registro buffer nello statp di LOAD, in seguito come il protocollo di handshaking classico prevede viene alzato il segnale di ready nello stato di REQUEST dove si permane fintanto che non viene asserito il segnale di ack, passando poi nello stato di WAIT_ACK dove si attende che il segnale di ack venga abbassato, infine sulla base del valore di conteggio del contatore si ritorna nello stato di IDLE o si procede prelevando la successiva stringa di bit da trasmettere.

Per quanto riguarda il nodo C invece, ancora una volta si è scelta la scomposizione in unità operativa e unità di controllo. In particolare l'unità operativa risulta costituita da:

- 2 Registri che memorizzano le stringhe ricevute da i rispettivi nodi A e B.

- Un Nibble Comparator che accetta in ingresso le due stringe di 8 bit e restituisce in uscita un vettore contenente tanti bit pari ad 1 quanti sono i bit uguali in posizione omologa
- Una memoria per salvare le stringhe ricevute dai due nodi.
- Un contatore per indirizzare la memoria.
- Un Contatore di bit che accetta in ingresso la stringa di bit prodotta dal NibbleComparator per contare il numero di bit pari ad '1'.
- Un Multiplexer per selezionare opportunamente la stringa da memorizzare all'interno della memoria.

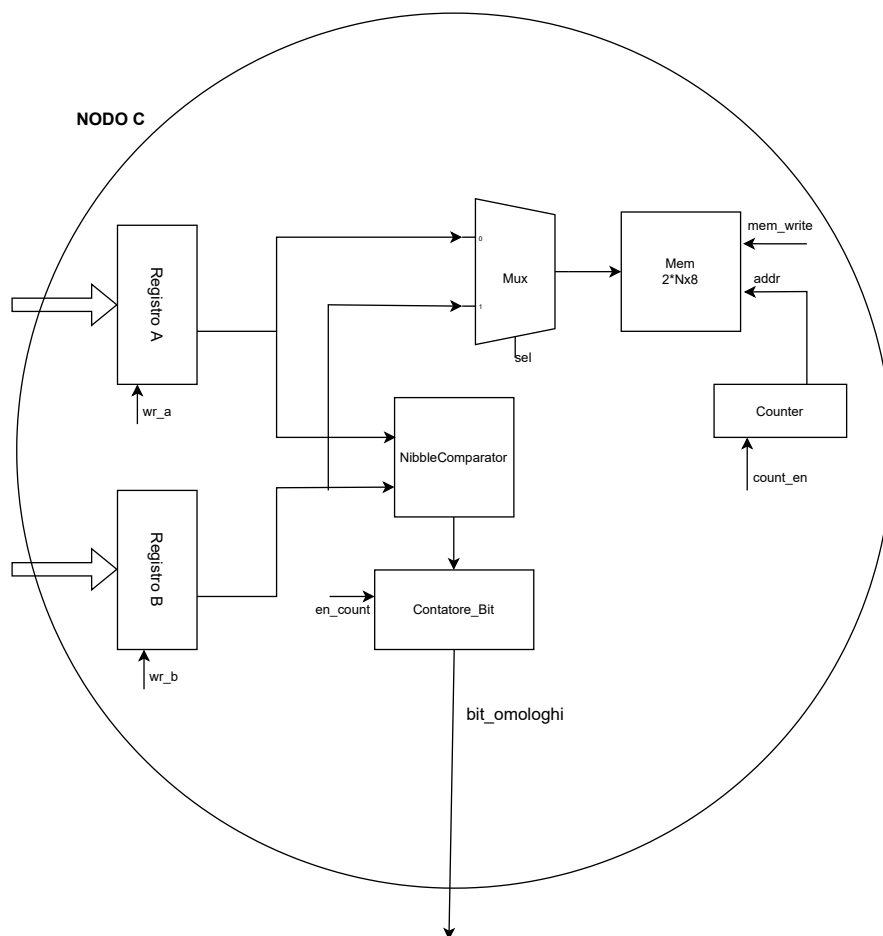


Figura 1.4: Datapath Nodo C

Occorre osservare, inoltre, che il NibbleComparator è stato realizzato (sfruttando un for..generate) con approccio strutturale come composizione di comparatori a singolo bit che presentano la seguente interfaccia in VHDL:

```
entity Comparator is port (
    a : in std_logic;
    b : in std_logic;
    gt : in std_logic;
    eq : in std_logic;
    lt : in std_logic;

    a_gt_b : out std_logic;
    a_eq_b : out std_logic;
    a_lt_b : out std_logic
);
end Comparator;
```

Esso ci è tornato estremamente utile per determinare i bit uguali in posizione omologa delle due stringhe, assegnando a ciascuno di essi uno dei bit delle stesse e ponendo in un opportuno segnale i risultati delle uscite a_eq_b.

Per quanto riguarda l'unità di controllo essa procede nel seguente modo: Attende in IDLE che entrambi i segnali di ready dei nodi siano alti per transitare nello stato di PRELIEVO dove abilita la scrittura nei due registri, al successivo colpo di clock essa si porta nello stato di CALC dove viene abilitato il contatore di bit omologhi. Al successivo colpo di clock invece si procede a memorizzare la stringa ricevuta dal nodo A all'interno della memoria nello stato di SAVE_A, successivamente nello stato di INC viene incrementato il conteggio del contatore per memorizzare la stringa ricevuta dal nodo B nello stato di SAVE_B, viene nuovamente incrementato il contatore e tornando nello stato di INC si valuta il segnale interno 'sel' che se posto ad 1 porta la fsm nello stato di ACK_OUT dove

vengono alzati i segnali di ack per i 2 nodi. Infine quando entrambi i segnali di ready tornano a zero si ritorna nello stato di IDLE.

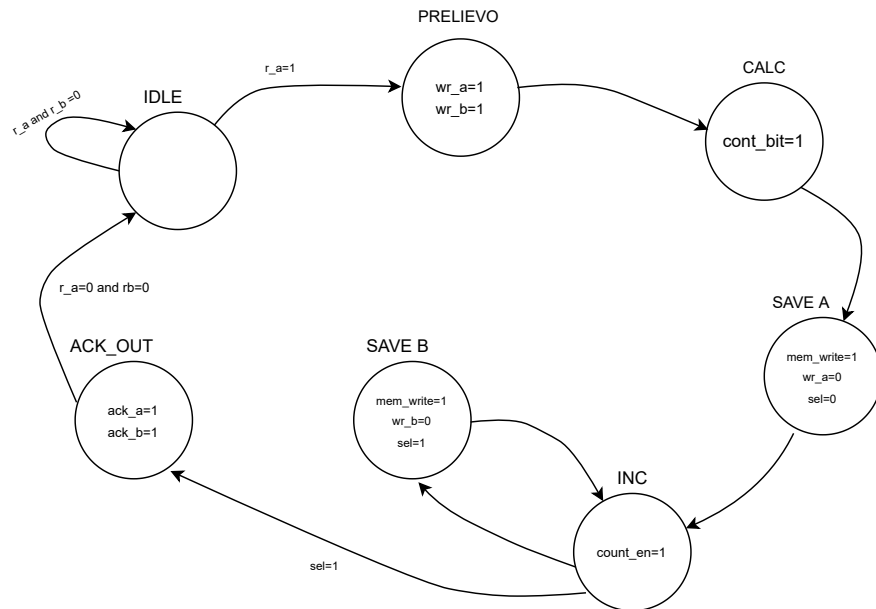


Figura 1.5: Unità Controllo Nodo C

1.2.2 Testbenches

Per quanto riguarda i Testbenches si è proceduto a testare dapprima il nodo A fornendo gli opportuni segnali.

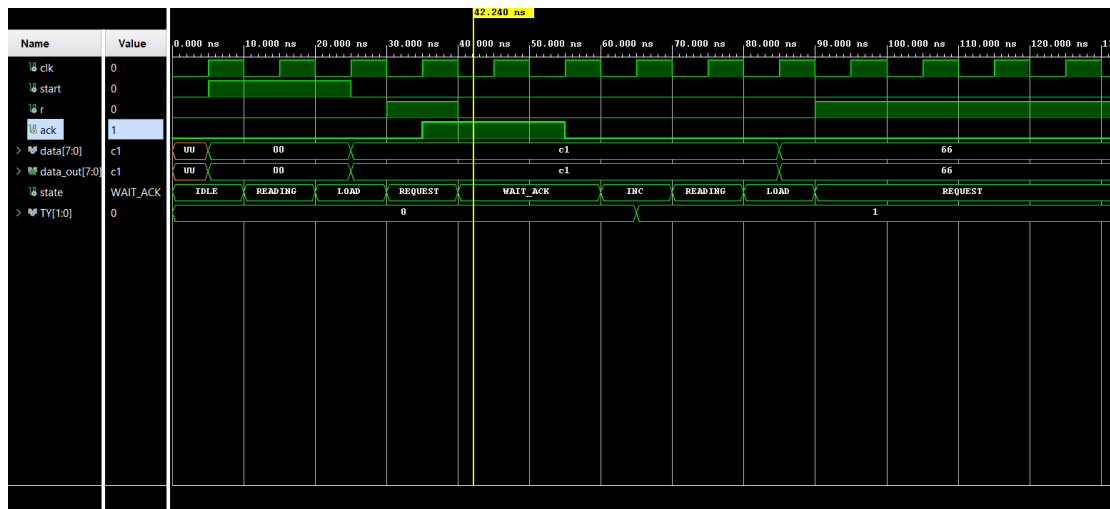


Figura 1.6: Nodo_A TB



Da notare inoltre come essendo tempificate dallo stesso segnale di clock entrambi i sistemi procedono in parallelo alla trasmissione delle stringhe, un'alternativa maggiormente realistica è quella di utilizzare segnali di tempificazione differenti per i due nodi A e B in comunicazione con C (di seguito se ne riporta una simulazione).

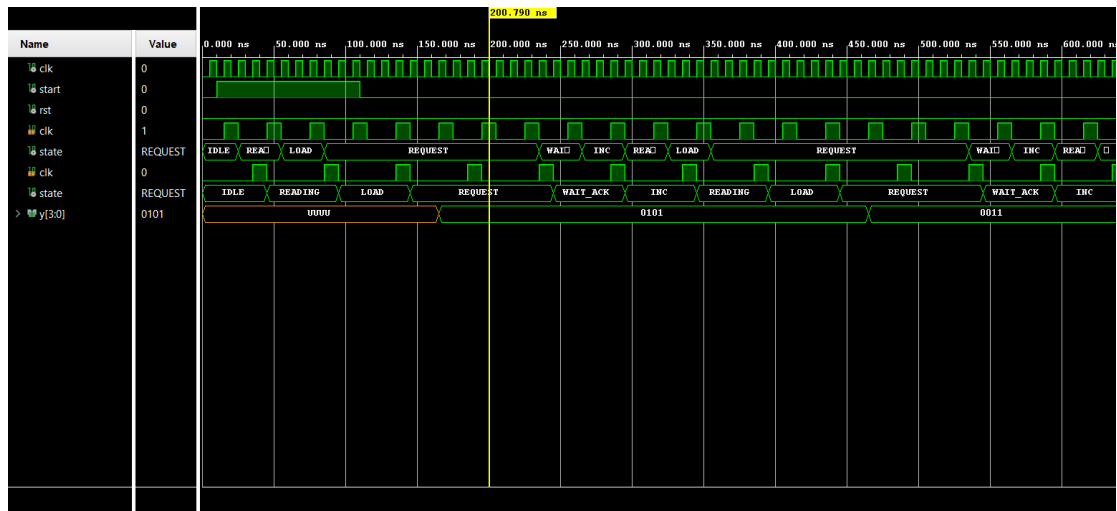


Figura 1.9