

Acquisizione e analisi spettrale segnali audio

Il nostro elaborato si suddivide in due file: la prima applicazione (fig. 1 front panel) implementa un sistema in grado di acquisire un segnale audio da microfono e, attraverso la scheda di acquisizione, campionarlo e convertirlo in digitale, prima di farne un'analisi completa a livello spettrale; mentre la seconda parte (fig. 2 front panel) realizza un sistema in grado di elaborare un segnale acquisito da file.

La nostra analisi del segnale si compone principalmente di tre parti: al primo passo visualizziamo il segnale campionato nel tempo, poi ne rappresentiamo lo spettro di potenza ed infine effettuiamo dei filtri a sei diverse frequenze mostrandone gli spettri.

Descrizione elaborato:

Parte I: acquisizione da microfono.

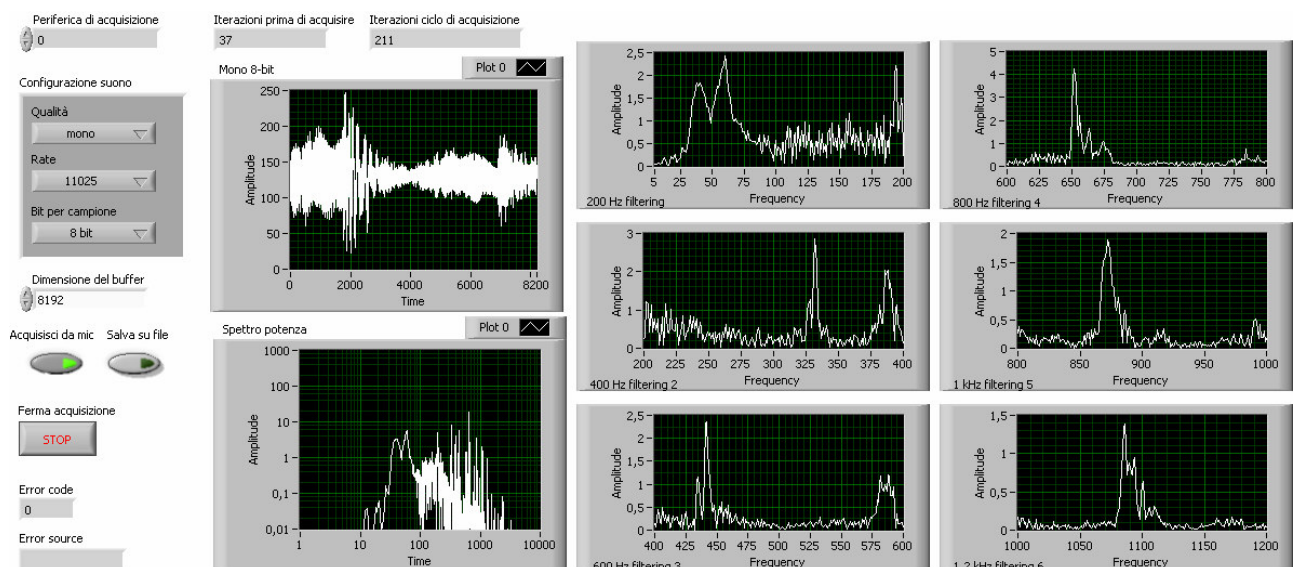


Figura 1 – Front Panel.

Adesso realizziamo una descrizione più approfondita del nostro elaborato, andando ad analizzare ogni singolo blocco componente l'applicazione.

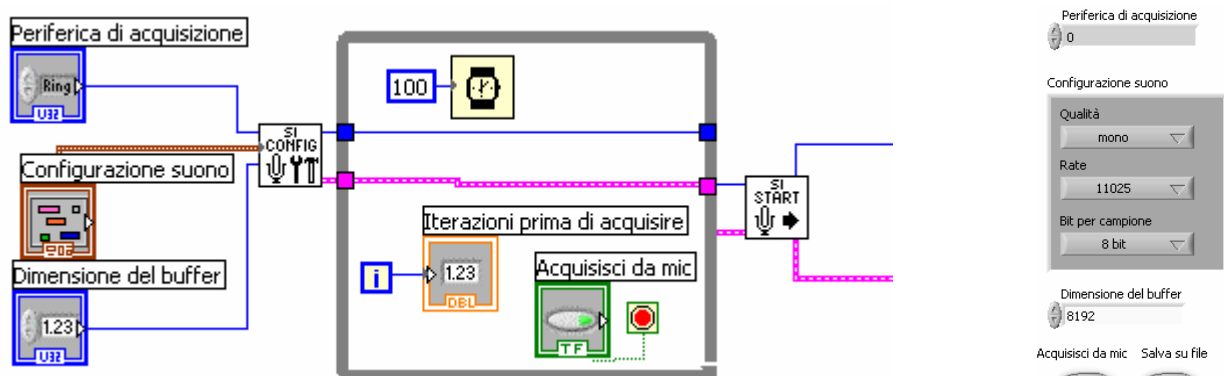


Figura 2 – Configurazione input.

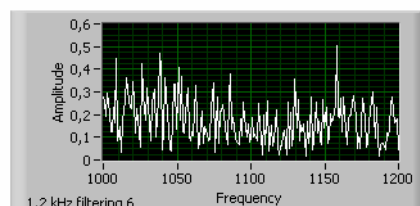
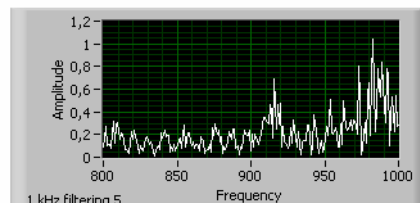
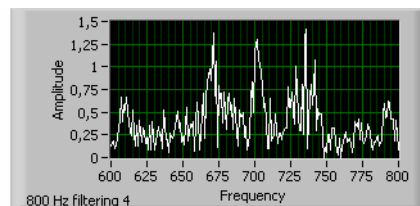
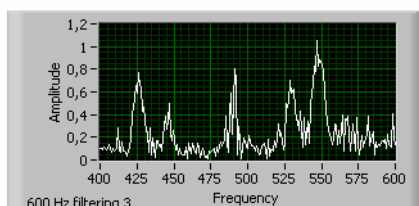
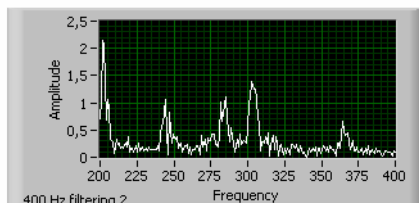
Ferma acquisizione

STOP

Error code

0

Error source



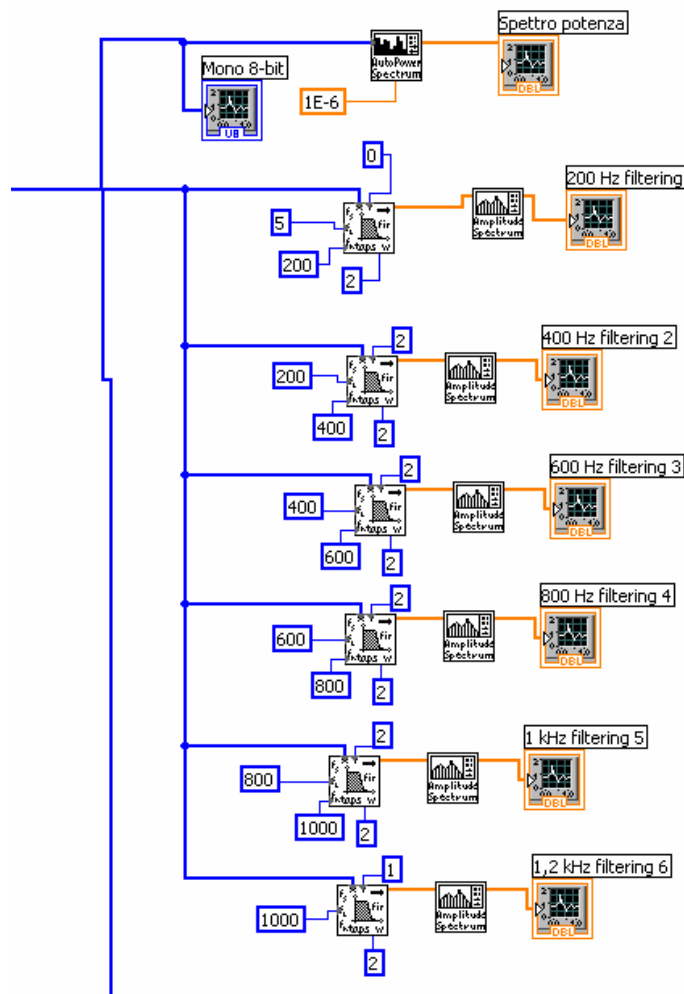


Figura 6 – Strutture di filtraggio e visualizzazione.

Durante l'acquisizione del segnale, il programma visualizza in tempo reale, con una memoria di 8192 campioni, il segnale campionato, lo spettro della potenza, e vari filtraggi: il primo è un filtro passa-basso con frequenza di taglio superiore a 200Hz, il secondo, terzo, quarto e quinto sono filtri passa-banda con banda passante di 200 Hz fino a 1 kHz e l'ultimo è un filtro passa-alto con frequenza di taglio inferiore di 1 kHz. I blocchi che implementano i filtraggi sono realizzati con filtri FIR, utilizzando una finestra di tipo Hamming.

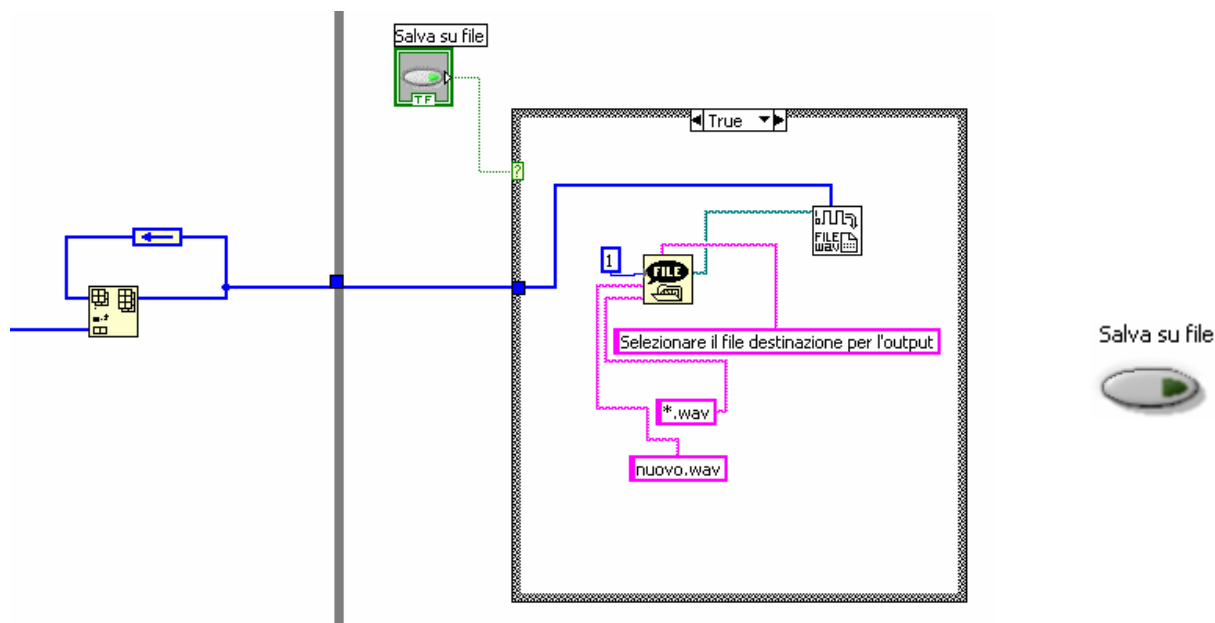


Figura 7 – Struttura per il salvataggio su file.

In seguito abbiamo aggiunto nel programma la possibilità di salvare su file il segnale acquisito, utilizzando sempre il blocco “Case structure”, al cui interno è contenuta un’opportuna struttura che apre la finestra di dialogo per il salvataggio dei file wave.

Parte II: acquisizione da file.

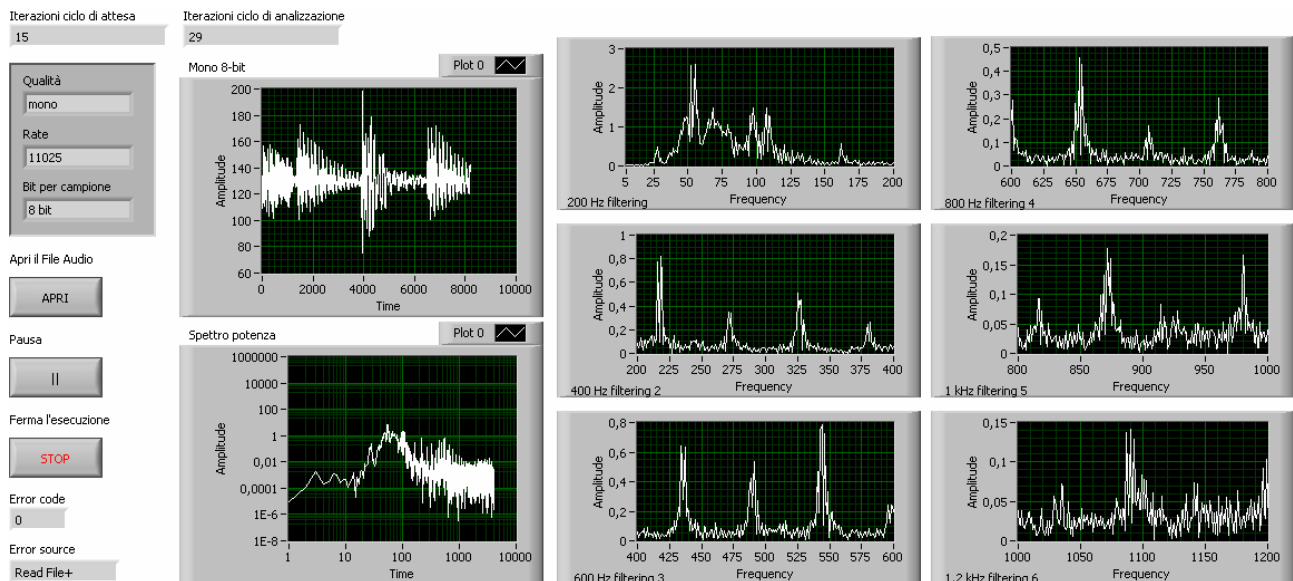


Figura 8 – Front Panel.

Come possiamo notare dal front panel, la seconda parte del nostro elaborato differisce dalla prima solamente per il modo in cui il segnale viene acquisito, cioè da file wave. Dunque illustreremo solo la parte dell’elaborato che differisce dalla precedente.

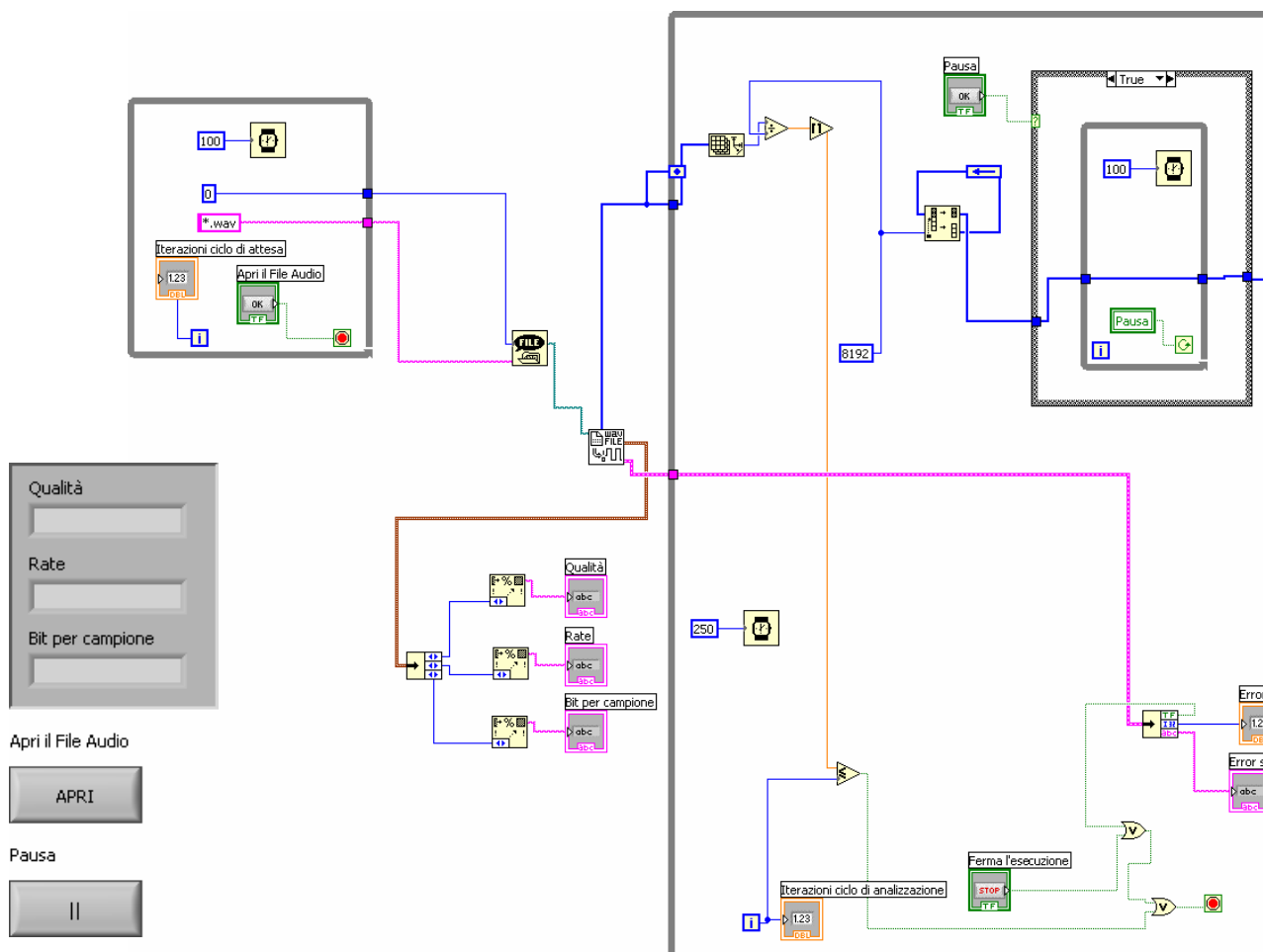


Figura 9 – Block Diagram.

L'acquisizione del file avviene premendo l'apposito tasto "APRI" che visualizza la relativa finestra di dialogo; in seguito il file viene memorizzato in un vettore utilizzato per inizializzare il ciclo di feedback (vedi figura).

Il suddetto ciclo viene passato in ingresso al blocco "Split 1D Array", che divide il vettore in due sottovettori; il primo di dimensione 8192 (8kB), il secondo contenente i restanti elementi del vettore. Questo accorgimento si è reso necessario per realizzare un buffer usato per simulare l'acquisizione in real-time del file; altrimenti avremmo visualizzato e processato l'intero file senza avere la sensazione che il tutto avvenisse in tempo reale.

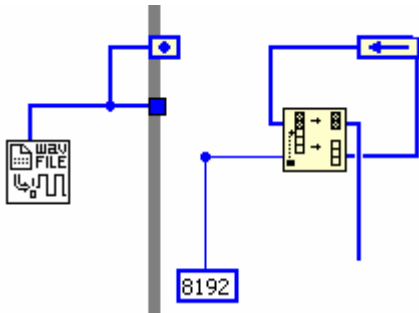


Figura 10 – Lettura file e inizializzazione ciclo acquisizione.

Il primo subarray viene quindi passato al blocco "Case structure"; premendo il pulsante "Pausa" l'esecuzione entra in un ciclo infinito dal quale può uscire solo dopo il rilascio del medesimo pulsante. Nell'altro caso l'esecuzione sarebbe passata dal "Case structure" senza interruzioni.

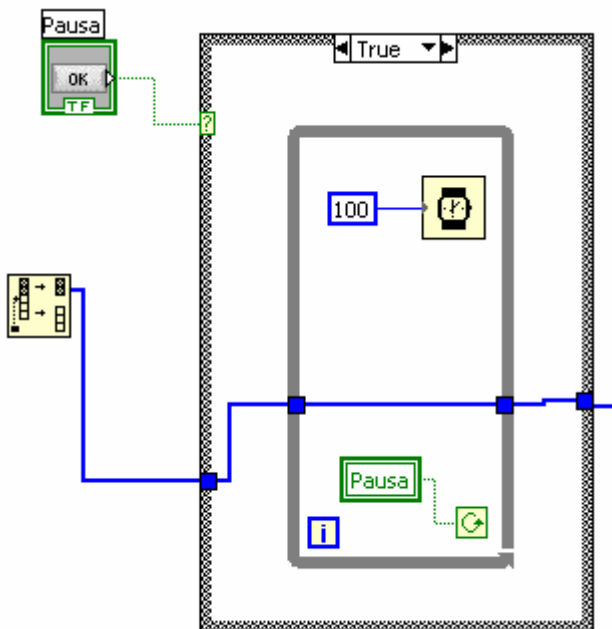


Figura 11 – In Pausa.

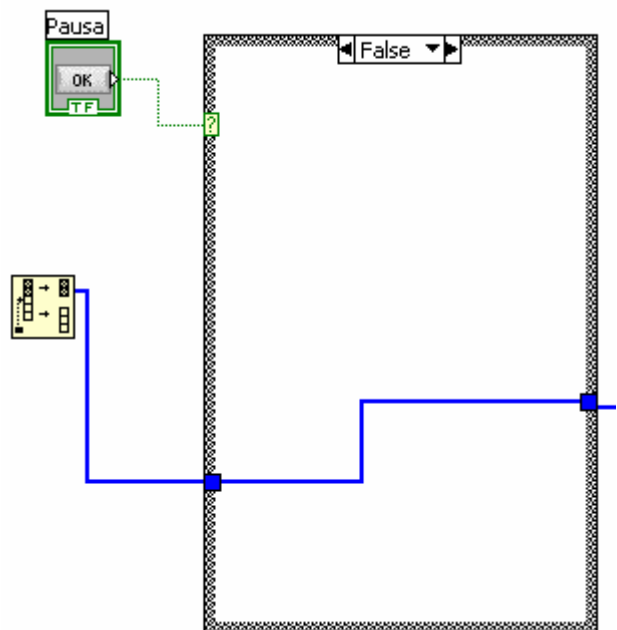


Figura 12- In esecuzione.

Dopo quest'ultimo passo l'esecuzione è la medesima della prima parte.

Infine un'ultima nota sull'esecuzione dell'analisi. Per farsi che quest'ultima non continuasse all'infinito, abbiamo aggiunto un controllo che, avendo come parametri la lunghezza del vettore e il numero di campioni utilizzati per il buffer, blocchi automaticamente l'esecuzione una volta raggiunta la fine del file. Come si può vedere dalla figura 13, il tutto avviene tramite un confronto tra il numero delle iterazioni necessarie al completamento dell'analisi e l'iterazione corrente.

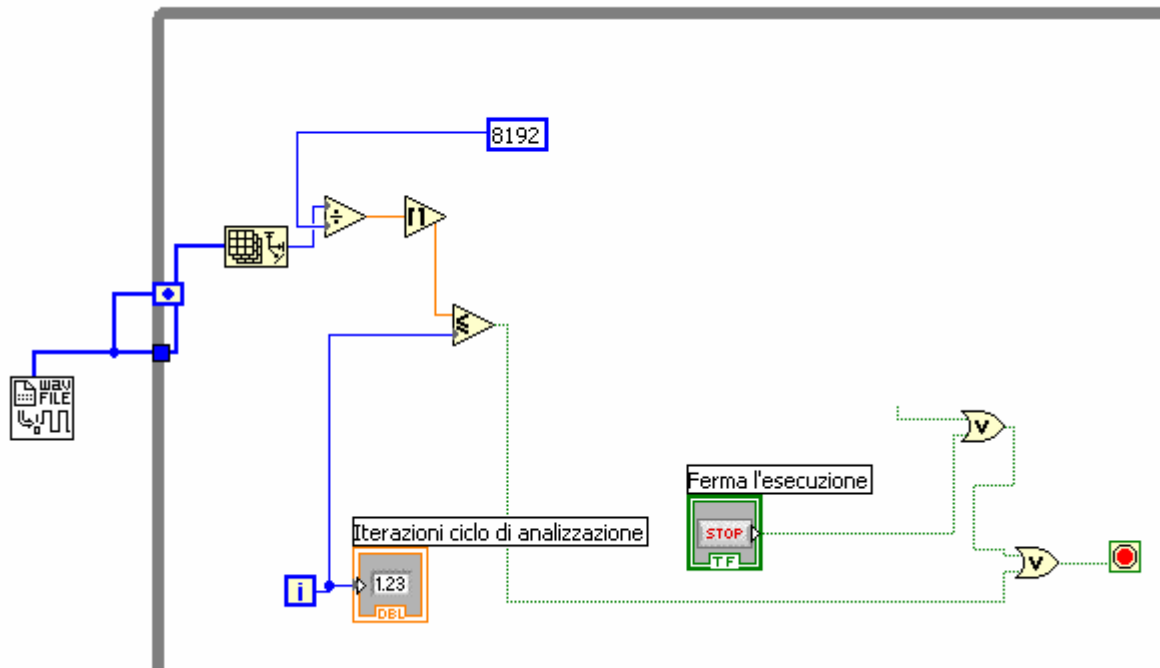


Figura 13 – Stop automatico dell'esecuzione.