

Università degli Studi di Padova – Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Elettronica

Relazione per la prova finale
***«Implementazione digitale di un filtro Butterworth
di 4° ordine su ARM:
confronto tra FMAC e CMSIS-DSP»***

Relatore: Prof. Buso Simone

Laureando: Fagherazzi Marco
Matricola n.: 2104305

Padova, 23/09/2026

Filtri digitali

I filtri elettronici sono molto diffusi nell'ambito dell'elaborazione dei segnali per eliminare o modificare determinate frequenze.

Vantaggi dei filtri digitali rispetto alla controparte analogica:

- Immunità a tolleranze, variazioni di temperatura e invecchiamento dei componenti elettronici;
- Maggiore flessibilità: si possono cambiare i parametri del filtro senza modificare l'hardware;
- Pendenza di taglio molto ripida;
- Riduzione costi e minore occupazione di spazi: non richiedono componenti fisici specifici per la loro implementazione.

Filtri FIR E IIR

I filtri digitali si dividono in:

- Filtri FIR (Finite Impulse Response):
 - Sempre BIBO stabili
 - Non sono retroazionati
 - Possono avere fase lineare
 - Non derivano da prototipi analogici

Hanno equazione generica:

$$y(n) = \sum_{k=0}^M b_k \cdot x(n - k)$$

- Filtri IIR (Infinite Impulse Response):
 - Potenziale instabilità
 - Presentano cicli di retroazione
 - Fase intrinseca non lineare
 - Derivano da prototipi analogici

Hanno equazione generica:

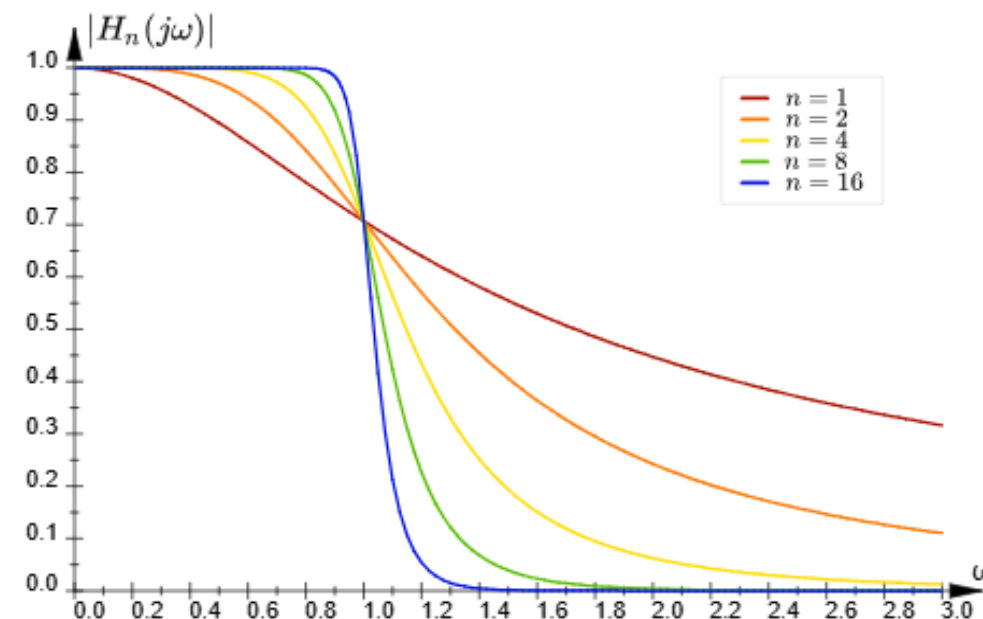
$$y[n] = \sum_{k=0}^N b_k \cdot x[n - k] - \sum_{j=1}^M a_j \cdot y[n - j]$$

Filtro Butterworth

Il filtro Butterworth è caratterizzato da:

- Risposta massimamente piatta in banda passante;
- Non presenta ripple né in banda passante né in banda attenuata;
- Pendenza di taglio proporzionale con l'ordine del filtro;
- Modulo della risposta in frequenza:

$$|G(\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_0}\right)^{2n}}}$$



Obiettivo: realizzare un filtro Butterworth del 4° ordine

Specifiche di progetto:

- **Frequenza di taglio:** $f_c = 20\text{kHz}$, tipica banda audio;
- **Frequenza di campionamento:** $f_s = 200\text{kHz}$, per rispettare teorema di Nyquist e ottenere una buona risoluzione temporale;
- **Guadagno in banda:** 0 dB con oscillazione minima, idealmente piatto.

L'implementazione dell'algoritmo è stata eseguita e analizzata seguendo due differenti modalità:

- Implementazione software: tramite la libreria ottimizzata **CMSIS-DSP** fornita da ARM;
- Implementazione hardware: utilizzo dell'**FMAC**, integrato nel microcontrollore, con l'obiettivo di ridurre drasticamente il carico computazionale sulla CPU.

MICROCONTROLLORE ARM: scheda NUCLEO-G474RE

Microcontrollore e periferiche utilizzate

La scheda di sviluppo NUCLEO-G474RE di STMicroelectronics ha montato a bordo il microcontrollore a 32 bit STM32G474RE, basato su ARM Cortex-M4 operante ad una frequenza di clock massima di 170 MHz.

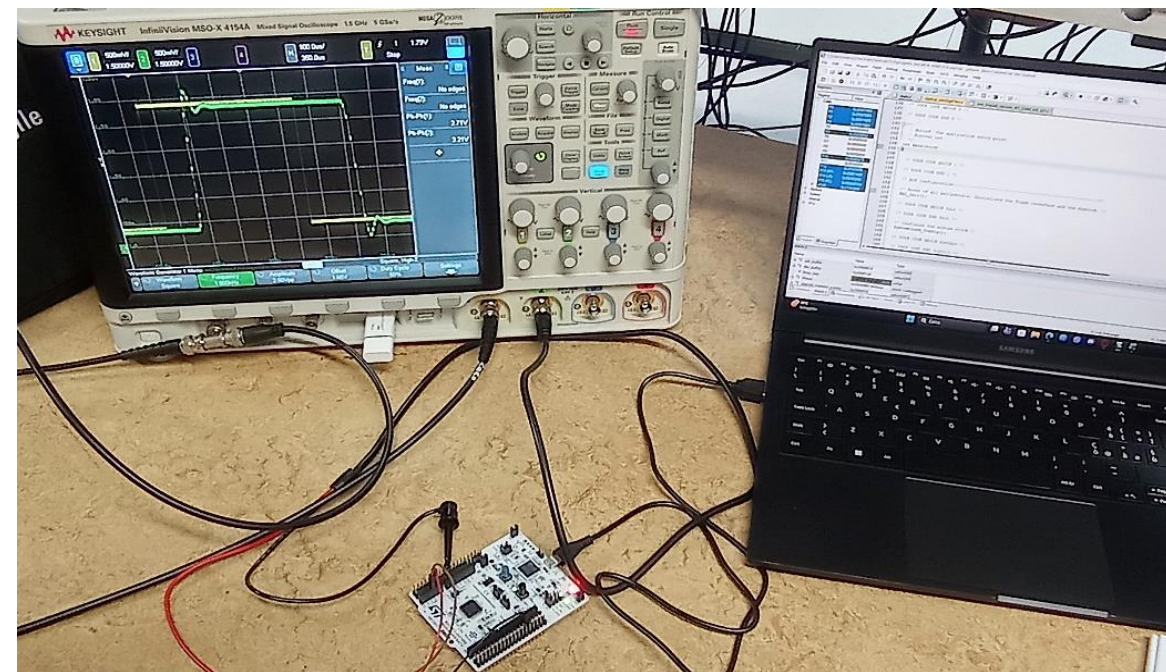
A livello pratico sono state utilizzate diverse periferiche hardware integrate nel chip:

- Convertitore analogico-digitale (ADC);
- Convertitore digitale-analogico (DAC);
- Timer ad alta risoluzione (HRTIM);
- Filter Math Accelerator (FMAC);
- Direct Memory Access (DMA).



Il banco di prova è stato allestito con i seguenti strumenti:

- **Oscilloscopio** a segnali misti (MSO): l'oscilloscopio usato è il Keysight InfiniiVision MSO-X 4154A, caratterizzato da una banda passante di 1,5 GHz, una frequenza di campionamento massima di 5 GSa/s e 2 generatori di forme d'onda interni;
- **Generatore di segnali:** per lo studio del sistema è stato impiegato il generatore interno allo strumento Keysight (Gen1);
- **Computer portatile:** utile per lo sviluppo del firmware tramite ambiente IDE e il debug della scheda NUCLEO-G474RE in tempo reale.



Cosa è la libreria CMSIS-DSP

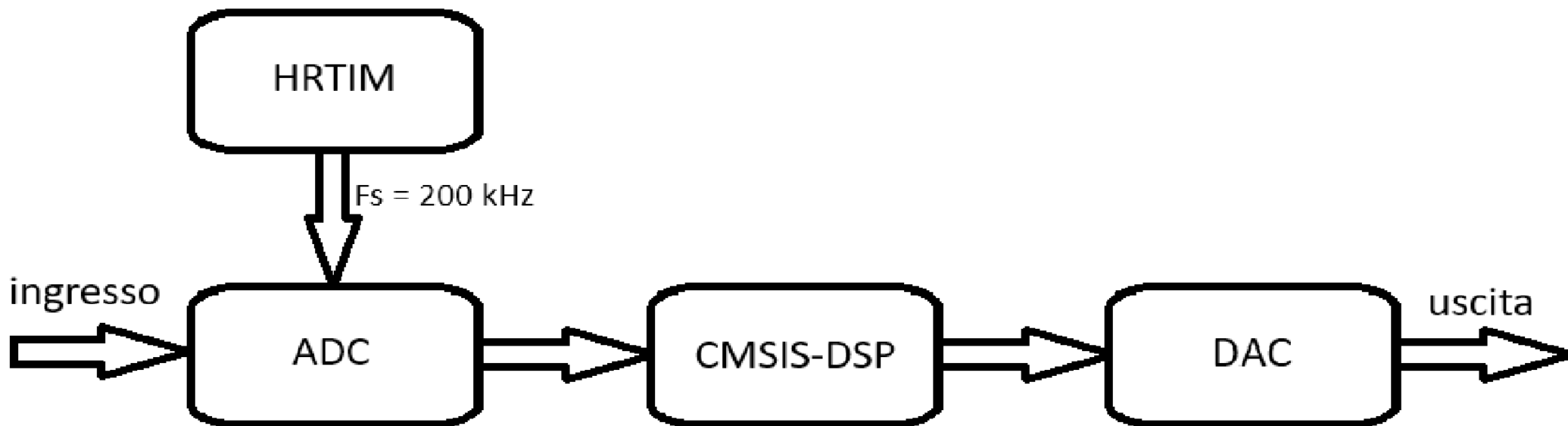
La libreria CMSIS-DSP (*Cortex Microcontroller Software Interface Standard - Digital Signal Processing*) porta vari vantaggi:

- Rappresenta uno standard industriale;
- Fornita da ARM ed è ottimizzata per l'architettura Cortex-M;
- Sfrutta al massimo la FPU (Floating Point Unit);
- Oltre 60 funzioni matematiche e per l'elaborazione di segnali:
 - Trasformate
 - Statistica
 - Filtri



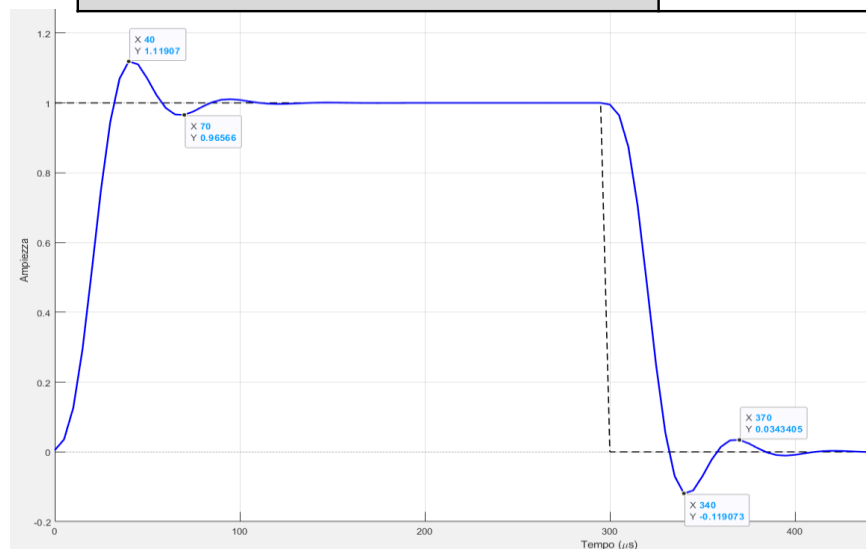
Il flusso dati si sviluppa:

1. **HRTIM** invia segnale di trigger all'ADC alla frequenza di campionamento (f_s);
2. **ADC** campiona il segnale;
3. La CPU tramite la libreria **CMSIS-DSP** elabora il dato;
4. **DAC** converte il segnale da digitale ad analogico e aggiorna l'uscita.



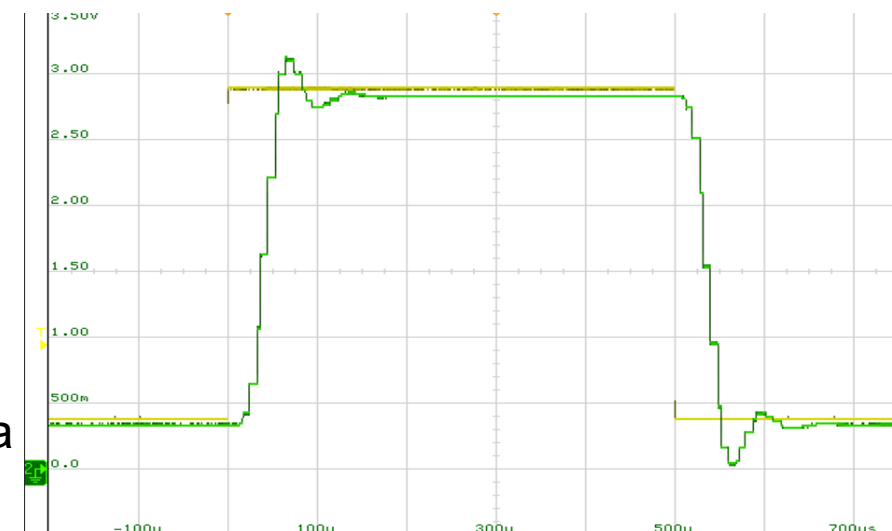
LIBRERIA CMSIS-DSP: Risposta al gradino

Parametro	Valore teorico	Valore sperimentale	Errore percentuale
Sovraelongazione (S%)	11,90%	12,05%	+1,26%
Sottoelongazione (U%)	11,90%	11,75%	-1,26%
Tempo di salita/discesa (tr/td)	20,2 μ s	21 μ s	+3,96%
Tempo di assestamento (ts)	76,5 μ s	81 μ s	+5,88%
Frequenza di ringing	16,67 kHz	16,1 kHz (salita) 14,7 kHz (discesa)	-3,42% (salita) -11,82% (discesa)
Latenza	0 μ s (ideale)	16 μ s	Pari a circa 3 periodi di campionamento ($T = 5\mu$ s a 200kHz)



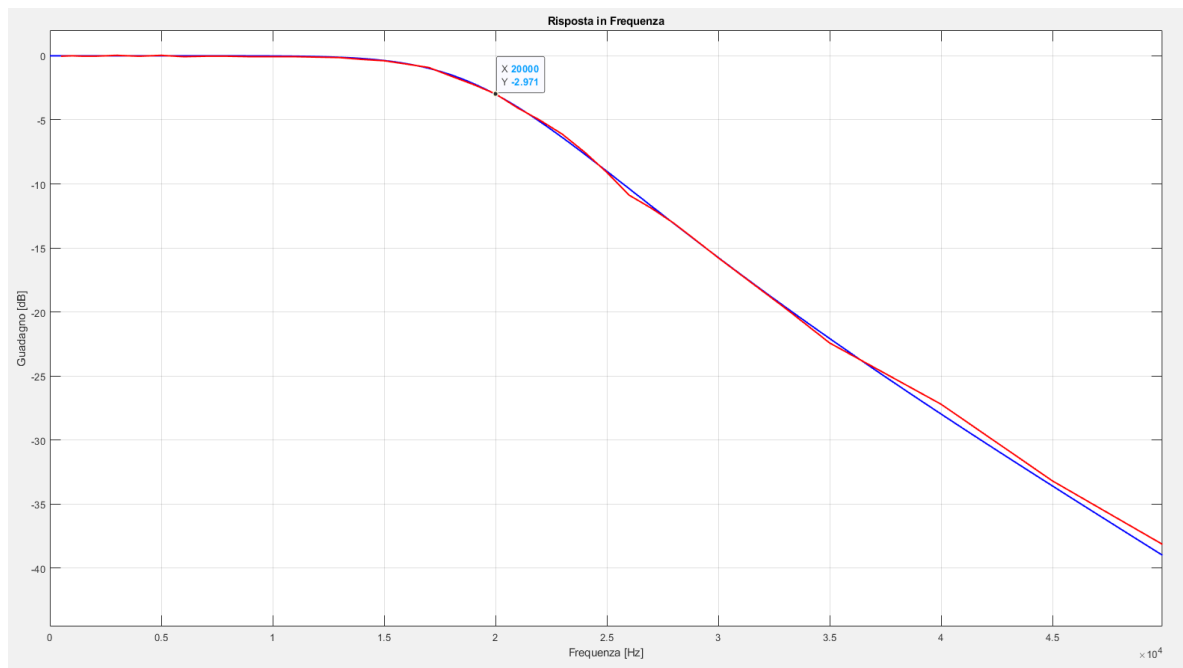
Curva teorica

Curva misurata



LIBRERIA CMSIS-DSP:

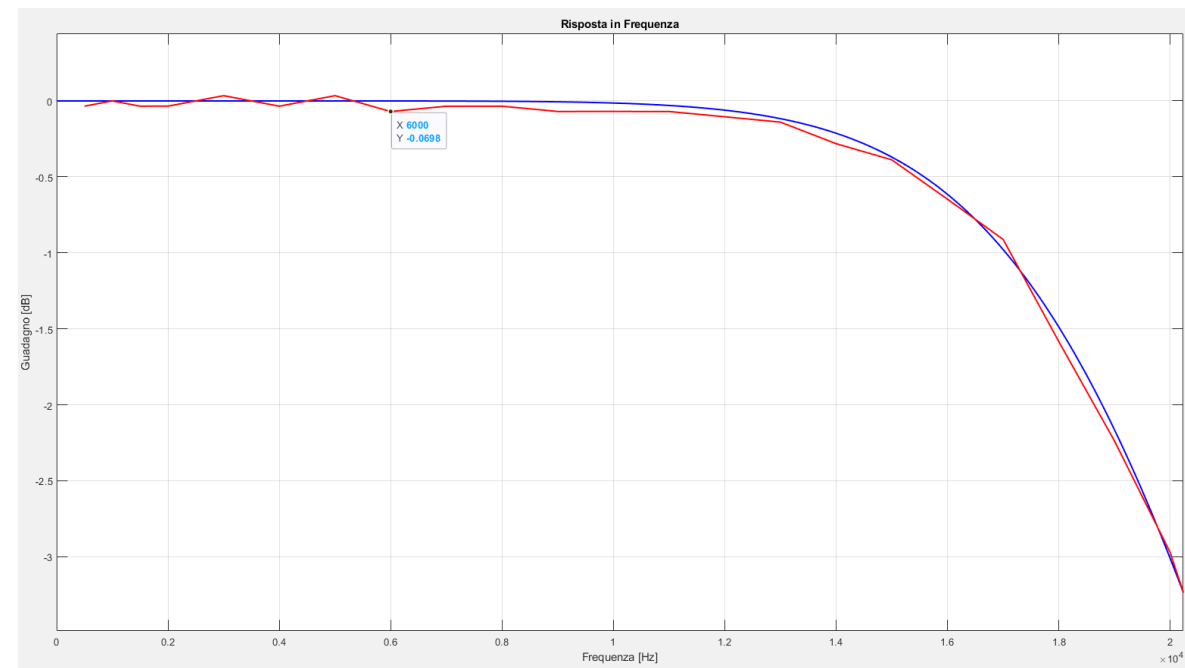
Risposta in frequenza



Modulo risposta in frequenza fino a 50 kHz

$$|G(20 \text{ kHz})| = -2,971 \text{ dB}$$

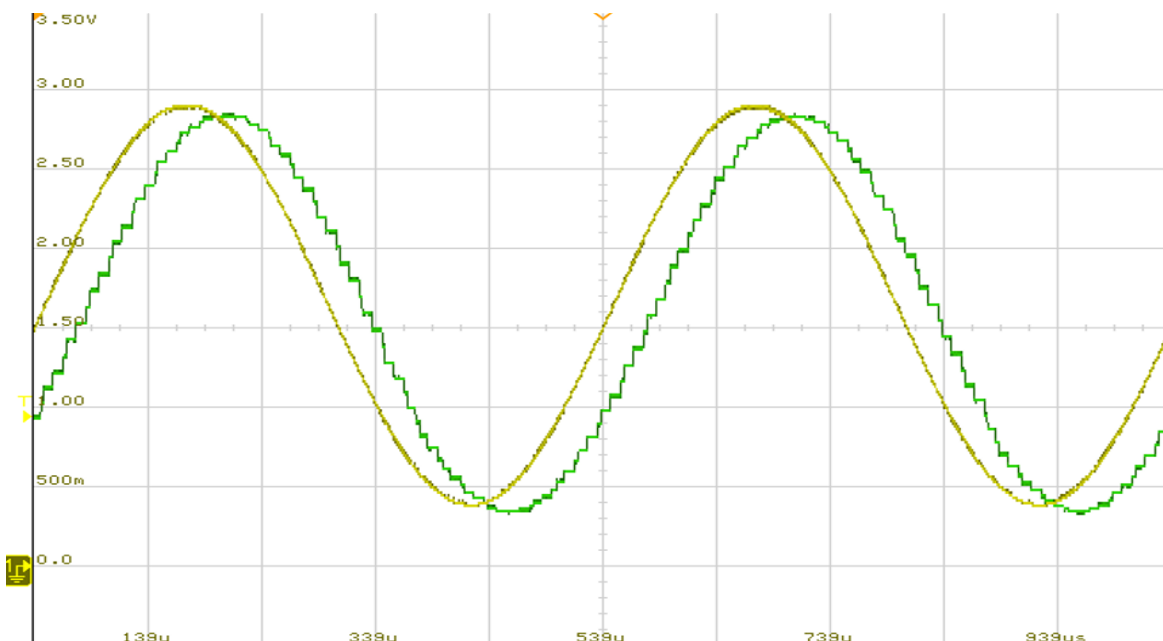
Pendenza di **-80 dB/dec**



Ingrandimento modulo risposta in frequenza fino alla frequenza di taglio di 20 kHz

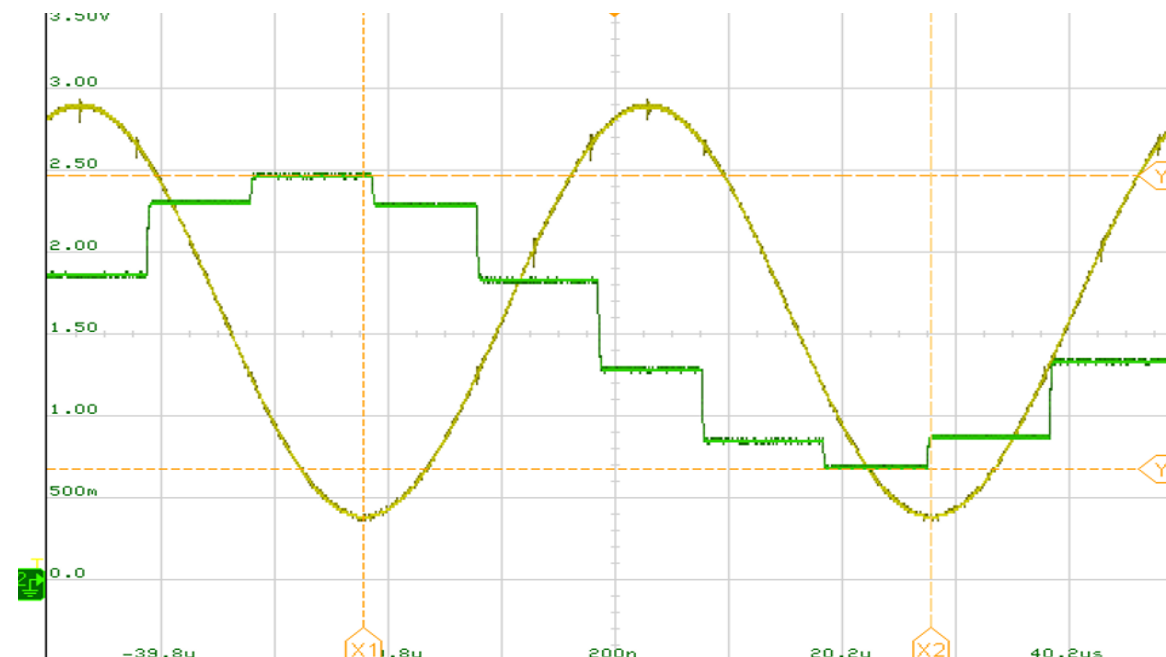
Modulo scostamento massimo pari a 0,0698 dB

LIBRERIA CMSIS-DSP: Frequenze di prova



Sinusoide a **2 kHz**

- $|G(2 \text{ kHz})| = V_{pp,out}/V_{pp,in} = 0,995 = -0,043 \text{ dB}$
- 100 campioni per periodo
- 14° di latenza intrinseca ($19,4 \mu\text{s}$ teorico)
- Ritardo totale di fase di $37 \mu\text{s}$



Sinusoide a **20 kHz**

- $|G(20 \text{ kHz})| = V_{pp,out}/V_{pp,in} = 0,71 = -2,971 \text{ dB}$
- 10 campioni per periodo
- 180° di latenza intrinseca ($25 \mu\text{s}$ teorico)
- Ritardo totale di fase di $43 \mu\text{s}$

Cosa sono FMAC e DMA

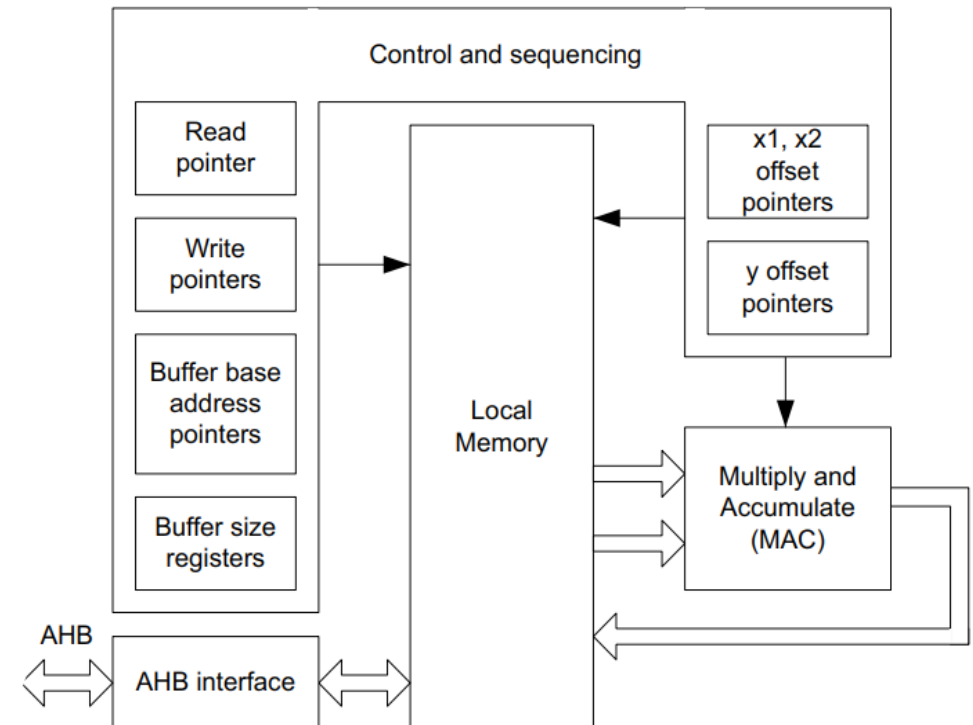
FMAC (Filter Math Accelerator):

- Unità hardware dedicata al calcolo di filtri IIR e FIR;
- Esecuzione operazioni MAC (Multiply-Accumulate) in un ciclo di clock;
- Riduce carico computazionale della CPU.

DMA (Direct Memory Access):

- Trasferisce dati da periferica a memoria e viceversa;
- Gestisce il flusso di dati in background.

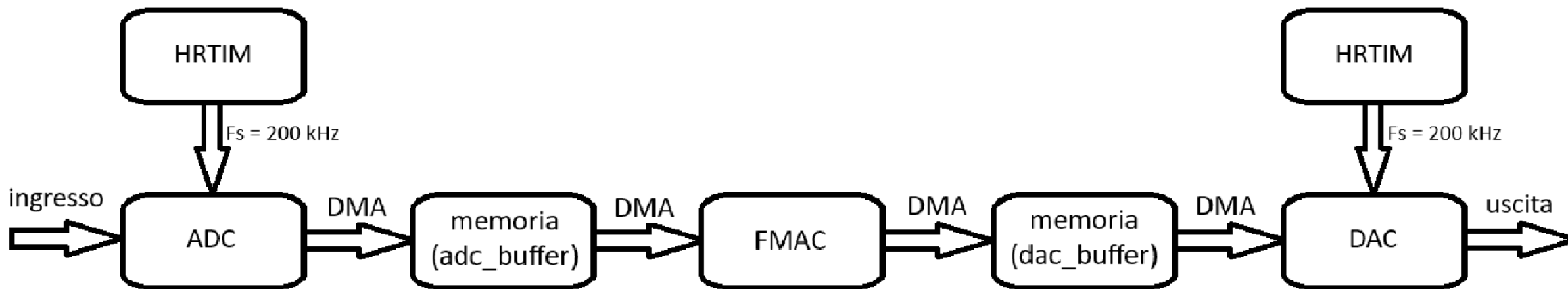
Architettura interna FMAC: memoria locale, blocco MAC e interfaccia AHB (Advanced High-performance Bus)



FMAC: Architettura e flusso dati

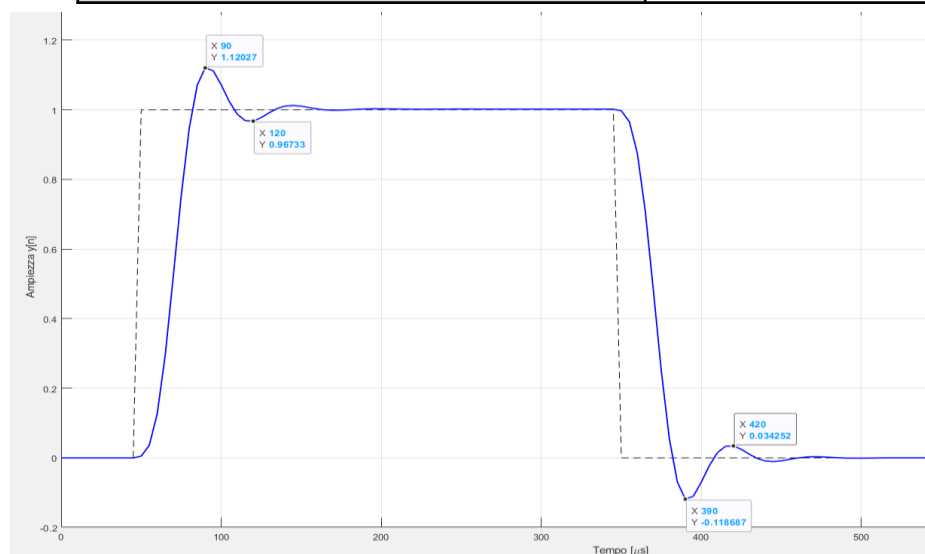
Il flusso dati si sviluppa:

1. **ADC** campiona l'ingresso quando riceve segnale di trigger dall'**HRTIM**;
2. **DMA** trasferisce i dati in un array in memoria (adc_buffer);
3. **DMA** trasferisce i dati dalla memoria all'FMAC (WDATA);
4. **FMAC** elabora il dato;
5. **DMA** trasferisce i dati dall'FMAC (RDATA) ad un array in memoria (dac_buffer);
6. **DMA** trasferisce i dati al DAC;
7. **DAC** converte il segnale da digitale ad analogico e aggiorna l'uscita quando riceve segnale di trigger dall'**HRTIM**.



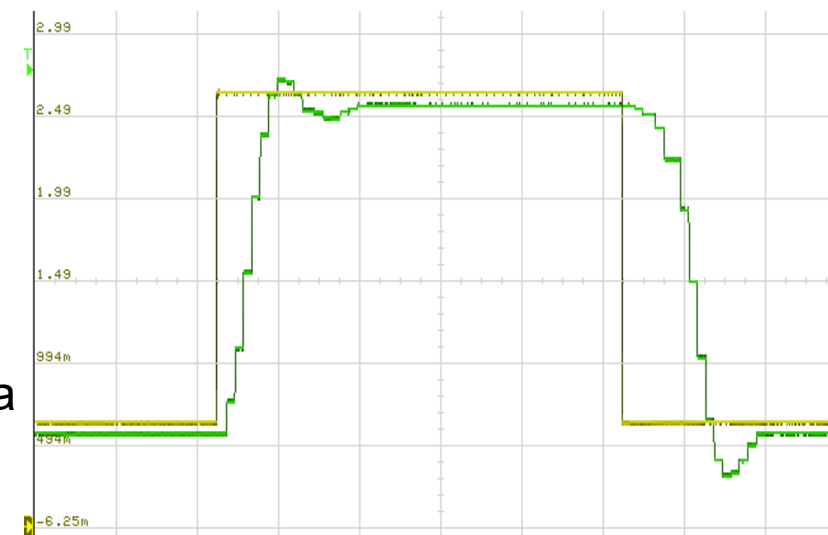
FMAC: Risposta al gradino

Parametro	Valore teorico	Valore sperimentale	Errore percentuale
Sovraelongazione (S%)	12,02%	11,50%	-4,32%
Sottoelongazione (U%)	11,87%	11,40%	-3,96%
Tempo di salita/discesa (tr/td)	20,3 μ s (salita) 20,3 μ s (discesa)	21,4 μ s (salita) 21,6 μ s (discesa)	+5,4% (salita) +6,4% (discesa)
Tempo di assestamento (ts)	81 μ s (salita) 76,5 μ s (discesa)	80,2 μ s (salita) 78 μ s (discesa)	-0,98% (salita) +1,96% (discesa)
Frequenza di ringing	16,67 kHz	15,6 kHz (salita) 15,1 kHz (discesa)	-6,4% (salita) -9,4% (discesa)

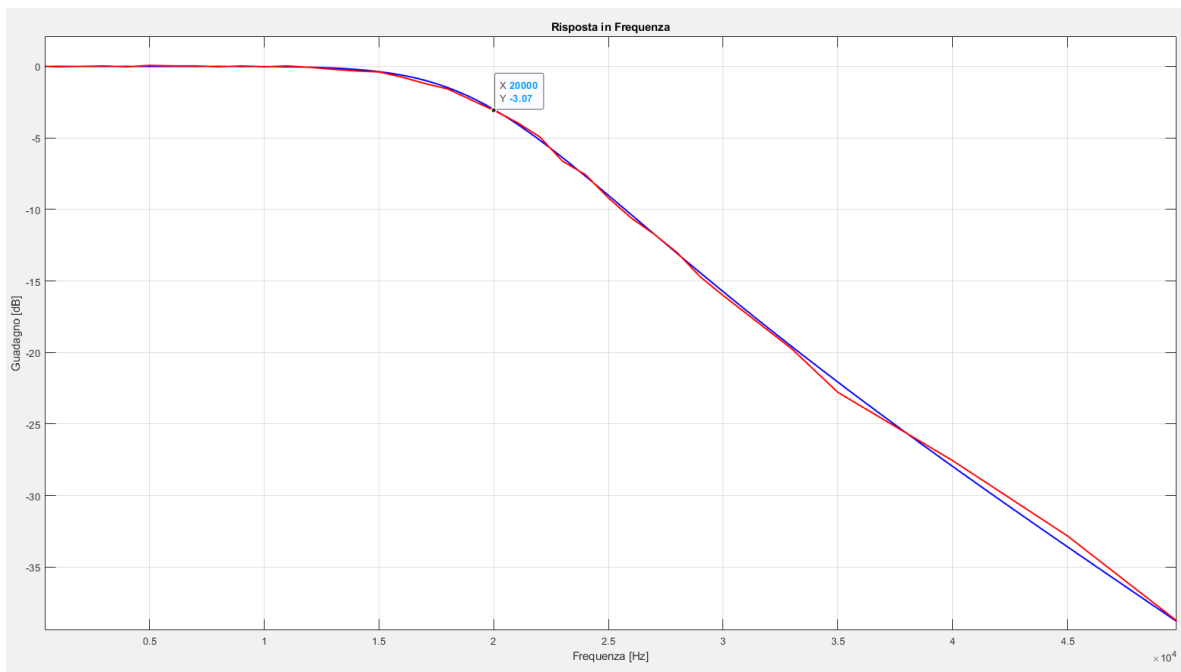


Curva teorica

Curva misurata



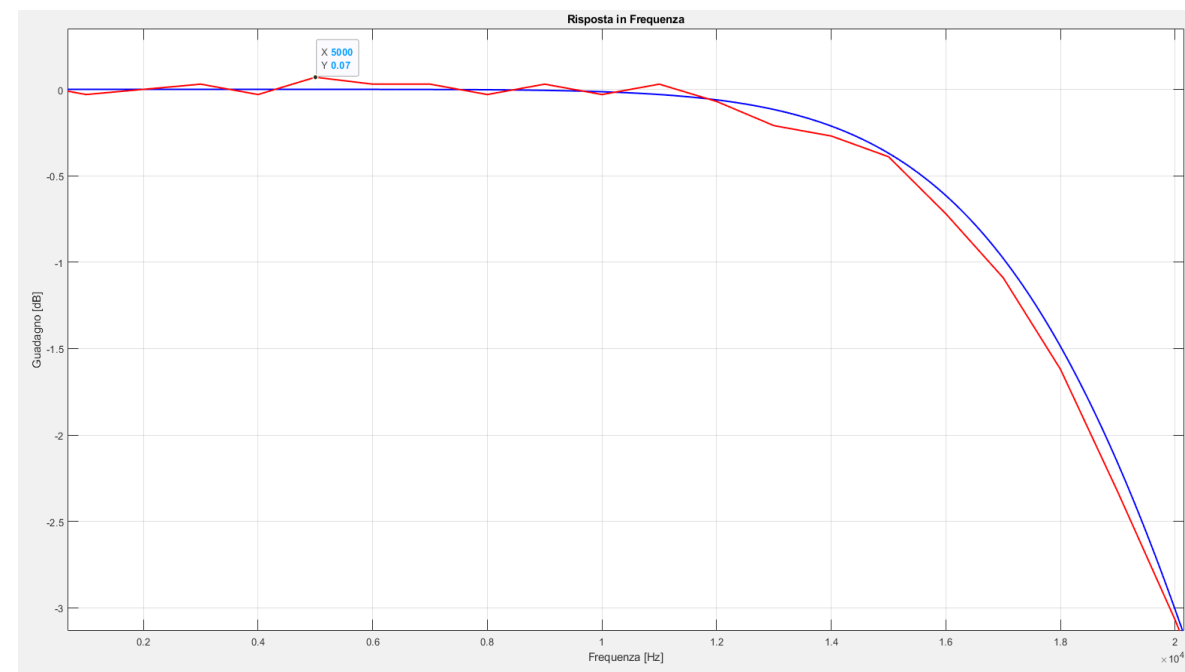
FMAC: Risposta in frequenza



Modulo risposta in frequenza fino a 50 kHz

$$|G(20 \text{ kHz})| = -3,07 \text{ dB}$$

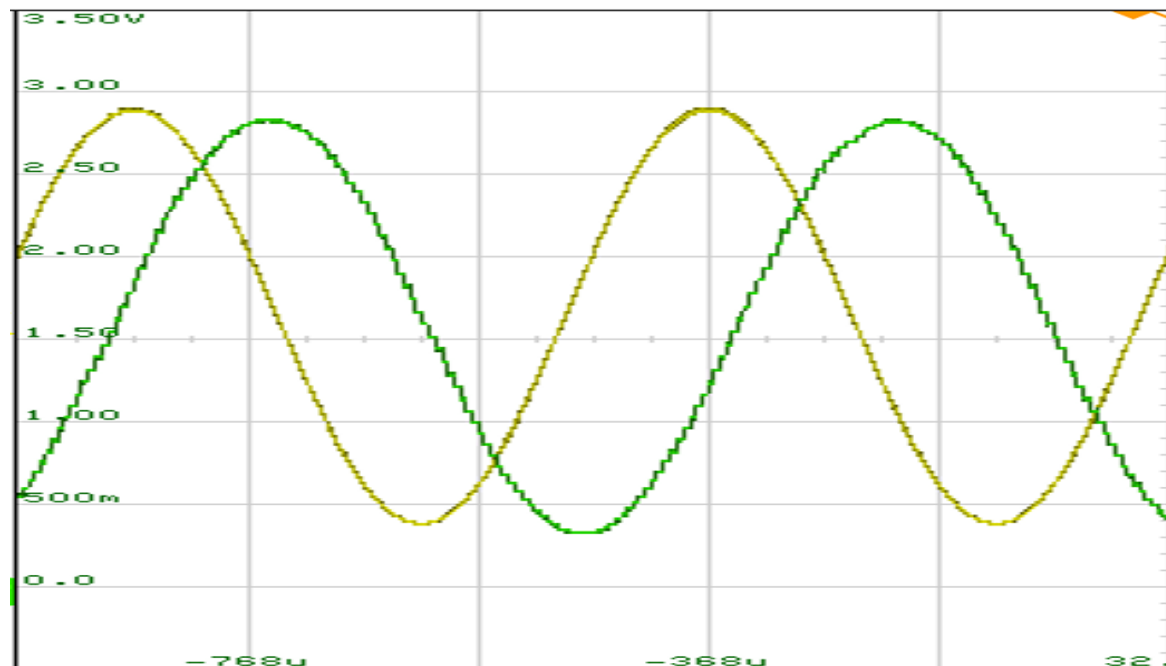
Pendenza di **-80 dB/dec**



Ingrandimento modulo risposta in frequenza fino alla
frequenza di taglio di 20 kHz

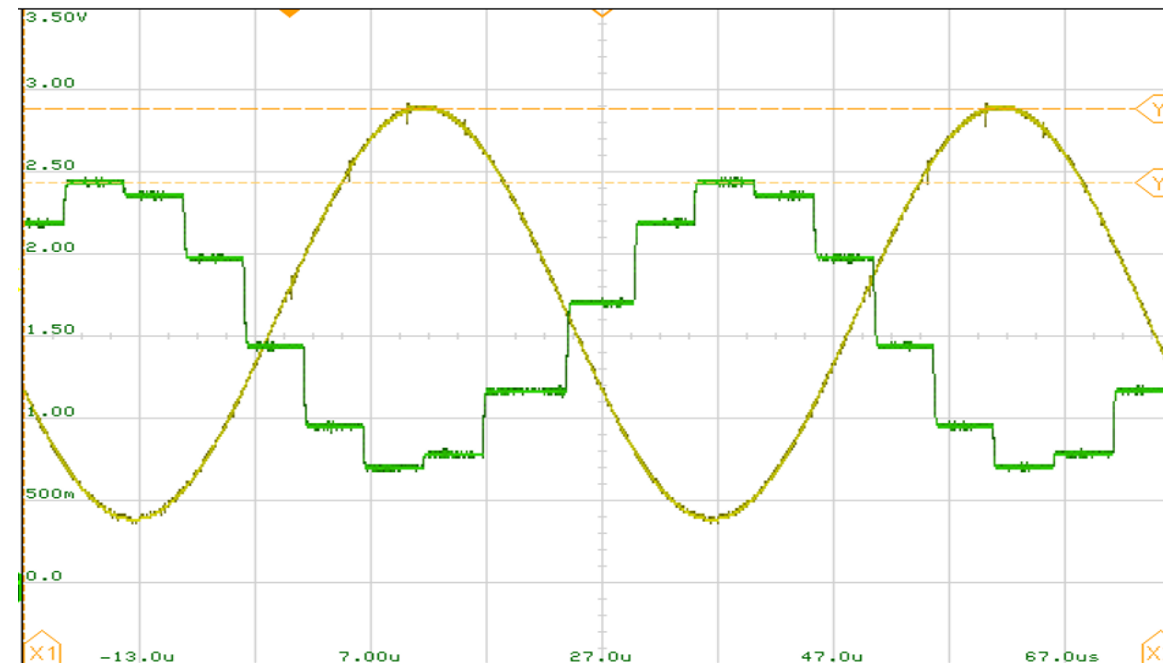
Modulo scostamento massimo pari a 0,07 dB

FMAC: Frequenze di prova



Sinusoide a 2 kHz

- $|G(2 \text{ kHz})| = V_{pp,out}/V_{pp,in} = 1,002 = \mathbf{0,018 \text{ dB}}$
- 100 campioni per periodo
- 14° di latenza intrinseca (19,4 μs teorico)



Sinusoide a 20 kHz

- $|G(20 \text{ kHz})| = V_{pp,out}/V_{pp,in} = 0,702 = \mathbf{-3,07 \text{ dB}}$
- 10 campioni per periodo
- 180° di latenza intrinseca (25 μs teorico)
- Ritardo totale di fase di 27 μs

CONCLUSIONI

Caratteristica	FMAC (Hardware)	CMSIS-DSP (Software)
Complessità di realizzazione	Elevata (Setup FMAC e 4 canali DMA)	Ridotta (funzioni standard)
Velocità di esecuzione / latenza	Pochissimi cicli di clock	Decine di cicli di clock CPU
Occupazione CPU	0% (Elaborazione in background)	Elevata (Proporzionale alla frequenza di campionamento)
Precisione numerica	Virgola fissa a 16 bit con segno (Q1.15)	Virgola mobile a 32 bit (Float32)
Determinismo temporale	Assoluto (Immune agli interrupt della CPU)	Algoritmico (Soggetto a jitter di sistema)
Memoria FLASH	Trascurabile (Solo codice di configurazione)	Modesta (300-500 byte per la funzione di filtraggio)
Memoria RAM	Uso SRAM interna dedicata dell'FMAC	Uso RAM di sistema
Portabilità del codice	Bassa (Periferica specifica di alcune famiglie ST)	Elevata (Disponibile per tutti i microcontrollori ARM)

Quale implementazione scegliere?

Dipende molto dalle specifiche di progetto:

- **FMAC** se sono richiesti:
 - Alta frequenza di campionamento
 - Elaborazione Real-Time del segnale
 - Minima latenza hardware
- **CMSIS-DSP** se:
 - Frequenza di campionamento garantisce un margine d'elaborazione sufficiente alla CPU
 - Portabilità del codice
 - Semplicità di sviluppo
 - Maggiore precisione numerica