

# **Sistema di sviluppo con DSP floating-point per il Rapid Prototyping**

**Versione 1.0**

*Graziano Bertini, Lucio Di Giovannantonio, Diego Gonzales,  
Lorenzo Grassi, Massimo Magrini, Massimiliano Dedola*

*Nota interna B4-*

**ISTI-CNR**



# Sistema di sviluppo con DSP floating-point per il Rapid Prototyping

Graziano Bertini, Lucio Di Giovannantonio\*, Diego Gonzales\*\*, Lorenzo Grassi\*,  
Massimo Magrini\*, Massimiliano Dedola\*\*\*

\* Collaboratore esterno, \*\*FSSG-CNR (VE), \*\*\* Stage di tesi Ing. Telecomunicazioni

## Indice

<b>1 Abstract.....</b>	<b>4</b>
<b>2 Sommario.....</b>	<b>4</b>
<b>3 Cenni sulla struttura dell' algoritmo.....</b>	<b>5</b>
<b>4 Implementazione real-time su HW.....</b>	<b>6</b>
4.1 Piattaforma DSP utilizzata.....	6
4.1.1 Il DSP TMS320C6711:Caratteristiche principali.....	7
4.1.2 La scheda DSK con TMS320C6711.....	9
<b>5 Implementazione SW.....</b>	<b>9</b>
5.1 Tool di sviluppo.....	10
5.2 Modalità di sviluppo.....	10
5.2.1 Inserimento del modello dei blocchi fondamentali del DSK.....	10
5.2.2 Configurazione del modello.....	11
5.3 Cenni sul profiling del software.....	11
<b>6 Realizzazione del prototipo.....</b>	<b>12</b>
6.1 Preamplificatore.....	13
<b>7 Test audio.....</b>	<b>13</b>
<b>8 Conclusioni, sviluppi futuri.....</b>	<b>14</b>
<b>9 Appendice.....</b>	<b>15</b>
9.1 Schema elettrico preamplificatore.....	15
9.2 Note sui componenti utilizzati.....	16
9.3 Misura del rapporto segnale rumore.....	16
9.3.1 Strumenti utilizzati.....	16
9.3.2 Esito della misura.....	17
9.4 Pannello frontale.....	18
9.5 Pannello posteriore.....	18
9.6 Interno.....	19
<b>10 Bibliografia.....</b>	<b>20</b>

# 1 Abstract

Rapid prototyping is becoming a new approach in advanced Digital Signal Processing development. It allows fast tuning of complex algorithms without spending many work-days coding in typical DSP oriented languages, typically assembler or C/C++ dialects.

With the advent of Mathworks MATLAB's Real Time Workshop (RTW) is now possible to compile, load and execute graphically designed Mathworks Simulink models into a real DSP platform. RTW supports many Developer Starter Kit DSP boards, including the pretty powerful Texas Instruments C6000 series.

As described in this document, we developed an HW system based on the TMS320C6711 DSK, enclosing it in a chassis together with additional analog circuitry. This system has been used for testing a Virtual Dubbing algorithm currently under development at Musical & Architectural Acoustic Lab. (FSSG-CNR, Venice Italy) proving the advantages of rapid aided prototyping approach

# 2 Sommario

Nella nota viene descritto un sistema HW/SW realizzato in forma prototipale, basato su PC che gestisce tramite parallela un processore per DSP esterno, per l'implementazione real time di algoritmi di elaborazione di segnali in banda audio.

La particolarità del sistema è quella di consentire l'impiego dei più recenti tools sviluppati da Texas Instruments insieme a Matworks per l'esecuzione di codice su target DSP, a partire da algoritmi scritti in linguaggio ad alto livello, tecnica riferita come "rapid prototyping". (s dire qualcosa di più qui?..).

Il lavoro è stato svolto su iniziativa del Laboratorio di Acustica Musicale e Architettonica di Venezia (FSSG-CNR) nell'ambito dei finanziamenti del progetto della CE (RACINE-S) del quale i ricercatori del suddetto laboratorio sono partners. L'obiettivo del progetto riguardava la ricerca di tecniche per il restauro audio-video di vecchie pellicole cinematografiche fortemente deteriorate.

Uno degli aspetti del progetto prevedeva la realizzazione presso l'ISTI-CNR di un sistema con il quale poter effettuare le prime sperimentazioni del processo di Virtual Dubbing a partire dalle tracce audio (parlato) di pellicole deteriorate.

Il cuore del sistema è costituito dalla scheda Texas Instruments basata sul TMS320C6711, un processore floating point con architettura DSP-RISC innovativa e istruzioni di tipo VLIW (1), più altri componenti e varie interfacce audio (Codec).

La messa punto del sistema, oltre alla realizzazione di un apparato HW completo con rack standard, pannelli ecc, ha comportato lo studio delle funzionalità della scheda, in particolare di quelle per la corretta gestione delle interfacce audio, e le necessarie verifiche delle specifiche di qualità dell'elaborazione del segnale. In figura una visione del sistema in una fase intermedia di sviluppo.





## 4 Implementazione real-time su HW

### 4.1 Piattaforma DSP utilizzata

I Digital Signal Processor (DSP) come è noto sono stati introdotti negli anni '80 per risolvere in modo ottimale determinate classi di problemi di elaborazione digitale delle informazioni (in particolare segnali audio-video digitalizzati) da realizzare con stretti vincoli temporali. Negli ultimi anni si sono imposti come soluzione ideale a problematiche applicative di media ed elevata complessità anche in molti altri settori tecnologici. L'analisi e la sintesi dei segnali vocali e musicali, l'elaborazione delle immagini, l'audio digitale, l'automazione, la strumentazione biomedica e di laboratorio, e in generale l'elettronica destinata alle applicazioni consumer, sono solo alcune delle applicazioni dei DSP. Un processore DSP ha la funzione di realizzare manipolazioni matematiche di segnali provenienti dal "mondo reale" opportunamente convertiti dall'ambito analogico a quello digitale. Un DSP può essere considerato specifico per una determinata applicazione (task), oppure essere di supporto ad altri elaboratori svolgendo la funzione di "co-processore". In questo caso i DSP vengono utilizzati per elaborazioni ripetitive su segnali in tempo reale (ad es. operazioni di compressione video o acceleratori di grafica) mentre ai processori "general purpose" rimangono affidate tutte le altre operazioni dei computer, la manipolazione di grandi quantità di dati, il calcolo scientifico, la gestione del sistema operativo, delle unità di input/output, ecc.

Nel nostro caso si deve considerare il carattere prototipale dell'applicazione e quindi, più che l'ottimizzazione rapporto prezzo/prestazioni (dato fondamentale per un'applicazione industriale) si è reso necessario scegliere un dispositivo che presentasse dotazione di strumenti di sviluppo evoluti, in modo da facilitare la messa a punto ed il testing dell'algoritmo. Visto che la progettazione dell'algoritmo è stata condotta principalmente con MATLAB/Simulink la scelta è ricaduta su un dispositivo il cui ambiente di sviluppo fosse facilmente integrabile con esso.

In funzione dei seguenti fattori:

1. potenza di calcolo del processore
2. disponibilità di una scheda DSK a basso costo
3. ricchezza ambiente di sviluppo
4. integrazione con MATLAB

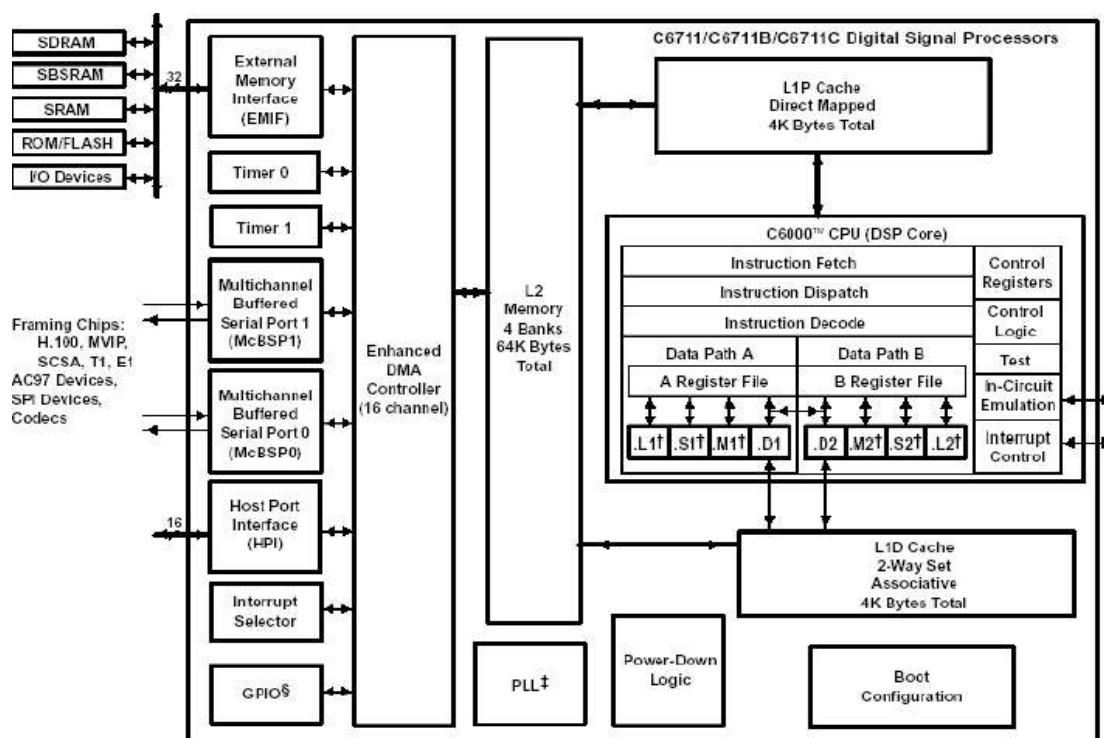
è stato deciso di utilizzare il DSP TMS320C6711 della Texas Instruments.

#### 4.1.1 Il DSP TMS320C6711: Caratteristiche principali

Come è noto alcune case produttrici di DSP, tra cui la Texas Instrument, stanno iniziando a immettere sul mercato dei DSP dotati di una piccola cache per i dati. A questo scopo la Quadrics Supercomputers World mette a disposizione il processore DSP TMS320C6711 che è la versione con cache del DSP TMS320C6701 nella linea

di produzione C6000 della Texas Instruments. Di seguito viene mostrata la struttura interna del DSP TMS320C6711.

Le caratteristiche principali di questo DSP sono veramente notevoli con unità funzionali indipendenti, 32 registri a 32 bit, un range di indirizzamento di 4Gbyte (32 bit), supporto a formato dei numeri sia big endian che little endian e, in ultimo, la possibilità di eseguire 900 milioni di istruzioni in floating point al secondo.

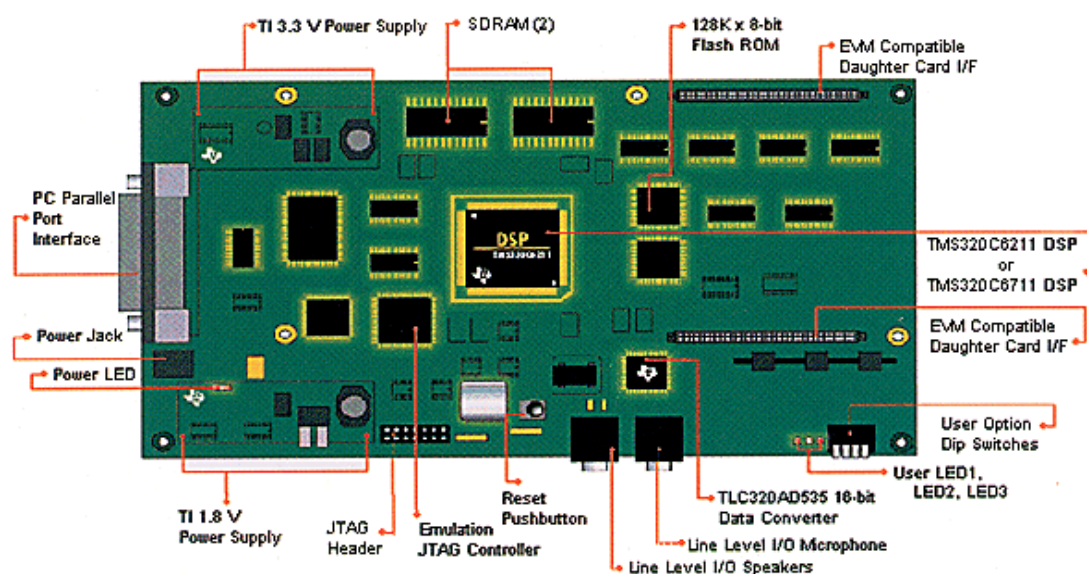


Questo DSP ha anche diverse periferiche integrate:

- *Interfaccia verso memoria esterna a 32 bit (external memory interface EMIF):*
- *Interfaccia diretta verso memorie sincrone (SDRAM e SBSRAM)*
- *Interfaccia diretta verso memorie asincrone (SDRAM e EPROM)*
- *Controller DMA (enhanced direct-memory-access EDMA)*
- *Interfaccia verso un host a 16-bit (host-port interface HPI)*
- *Due seriali multistandard (multi-channel buffered serial ports McBSPs)*
- *Due timer a 32-bit*
- *Blocco di generazione del Clock con un PLL integrato*
- *Supporto alla IEEE-1149.1 (JTAG) per l'emulazione e il test*

Il C6711 usa un'architettura basata su una cache a due livelli. La cache programma di primo livello (L1P) è da 32 Kbit, la cache dati di primo livello (L1D) è da 32 Kbit. La cache di secondo livello (L2) consiste di uno spazio di memoria di 512 Kbit che è diviso tra spazio di programma e di dati. La memoria L2 può essere configurata come una memoria mappata, una cache o una combinazione delle due.

### 4.1.2 La scheda DSK con TMS320C6711



Per il TMS320C6711 è disponibile una scheda Developer Started Kit che permette di valutare facilmente le capacità del processore. La scheda DSK è dotata di External memory Interface (EMIF) a 32 bit che supporta le interfacce SRAM e EPROM per memorie Asincrone e le interfacce SDRAM e SBSRAM per memorie Sincrone; lo spazio di memoria esterna indirizzabile è di 512 MB ed è presente il controller Enhanced Direct Memory Access (EDMA) da 16 canali indipendenti.

Installati on board sono presenti 16 MB di SDRAM e 128 KB di External Flash. La scheda è munita inoltre di un codec TLC320AD535 a 16-bit. Il JTAG Controller fornisce poi una connessione utile per un debugging approfondito. Infine la scheda è munita di un connettore di espansione per schede aggiuntive (daughter boards), come ad esempio Codec di migliori qualità etc.

## 5 Implementazione SW

Assieme alla scheda viene fornito l'ambiente di sviluppo integrato Code Composer Studio della Texas Instruments, che permette di sviluppo ed il debugging direttamente in linguaggio C. La connessione della scheda al PC viene effettuata tramite la porta parallela.

Il vantaggio fondamentale nell'uso di questa scheda e quest' ambiente di sviluppo sta nell'integrazione con l'ambiente MATLAB, con il quale è stata condotta la sperimentazione dell' algoritmo.

## 5.1 Tool di sviluppo

Nelle release più recenti dell'ambiente Matlab sono infatti disponibili una serie di funzioni e librerie Simulink per programmare direttamente schede target con specifici DSP a bordo, inclusa la scheda TI C6711 DSK già scelta, a partire da codice ad alto livello.

Una volta definito il modello che risolve un specifico algoritmo o parte di esso, e dopo aver effettuato le verifiche tramite le simulazioni, possiamo far sì che Matlab carichi il codice del modello sulla scheda DSP, crei il progetto per Code Composer Studio (CCS), lo compili, carichi in memoria il codice compilato e lanci l'esecuzione, senza scrivere direttamente nemmeno una riga di codice col CCS.

Per quanto riguarda la costruzione del modello Simulink, possiamo costruire il nostro modello sfruttando i blocchetti messi a disposizione dai vari toolbox, in particolar modo quelli messi a disposizione dal toolbox "Embedded Target for TI C6000 DSP", che consentono la gestione di alcuni dispositivi presenti sulla scheda: convertitori ADC e DAC, LED, Switch, Reset.

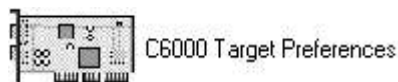
Ad esempio inserendo un ingresso il blocchetto "C6711 DSK ADC" indichiamo al modello che vogliamo usare come input l'ingresso della scheda DSP. Una volta creato il modello occorre impostare i corretti parametri di simulazione relativi al Real Time Workshop, occorre specificare quale system target vogliamo usare (nel nostro caso occorre selezionare la libreria relativa al `ti_c6000.tlc`), in questo modo verrà caricata la gestione relativa a queste famiglie di DSP, dobbiamo poi impostare la scheda DSP utilizzata (nel nostro caso C6711 DSK), a questo punto essendo completi gli elementi essenziali per poter far girare il modello sul DSP, premendo i tasti Ctrl+B si inizia la compilazione del modello e relativa esecuzione sul DSP.

E' inoltre possibile scrivere delle funzioni Matlab per lavorare col DSP senza ricorrere a Simulink: esiste infatti una libreria di funzioni che permette di gestire la scheda dalla riga di comando (o da uno script) di Matlab.

## 5.2 Modalità di sviluppo

La costruzione di un modello MATLAB Simulink adatto a ad essere pre-compilato in codice per Code Composer Studio, e quindi compilato in codice binario adatto a girare sulla scheda DSK deve essere condotta seguendo alcuni passi fondamentali:

### 5.2.1 Inserimento del modello dei blocchi fondamentali del DSK



Facendo doppio clic sul blocco è possibile verificare che la scheda venga riconosciuta correttamente.

Dal sottomenu di supporto per la scheda DSK è necessario, nel caso si voglia usare il Codec integrato, inserire i moduli Line in e Line out.



Questi blocchi hanno alcuni parametri di configurazione: sebbene i valori proposti per default siano adatti alla maggior parte delle applicazioni è necessario fare attenzione al numero di campioni per frame nel blocco di ingresso (1 nel nostro modello). Si noti che il sample rate del sistema, usando il codec integrato, è fissato a 8 kHz.

Se si vuole far uso della scheda figlia PCM3003 bisogna invece inserirne i blocchi AD/DA relativi (sempre reperibile nel menu dell'embedded target per C6000). Si noti che la questa scheda figlia è chiamata con il nome di catalogo Texas Instruments: TMDS326040.

### 5.2.2 Configurazione del modello

Prima di compilare il modello è necessario configurarlo correttamente per essere tradotto in codice C6000. A questo scopo è necessario aprire il pannello *Configuration parameters* del menu Simulation di Simulink per settare una serie di parametri. Di seguito vengono elencati i fondamentali:

- *Menu Solver*: Solver type: fixed step, Solver: discrete.
- *Stop time*: inf
- *Hardware Implementation*: Device type: C600
- *Real Time Workshop*: Target: ti\_c6000\_ert.tlc  
Questo settaggio è possibile solo se è installato l'embedded coder (ERT) per il C6000, coder che permette di compilare un codice opportunamente ottimizzato per il processore. In alternativa è possibile utilizzare solo la versione GRT.
- *Real Time Workshop/Target selection*: Target type: C6711DSK
- *Real Time Workshop/TIC6000 Code Generation*: Incorporate DSP/BIOS
- *Real Time Workshop/TIC6000 RunTime*: Overrun act: Notify and Continue.
- *Real Time Workshop/TIC6000 RunTime*: Overrun act: Turn\_on\_LEDs

Una volta impostate questi parametri, digitando Ctrl+b Matlab dovrebbe trasformare il modello in un progetto Code Composer Studio, lanciare quest'ultimo e quindi avviare la compilazione vera e propria. Una volta generato il file binario è possibile lanciarne l'esecuzione sulla scheda.

## 5.3 Cenni sul profiling del software.

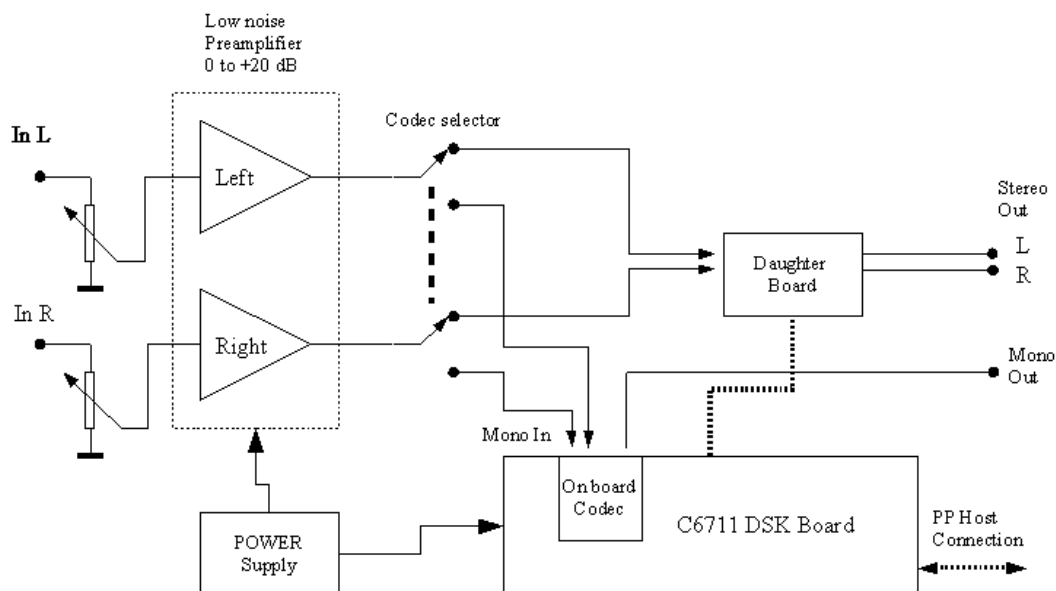
Usando l'embedded target per il C6000, con l'opzione *Profile performance at atomic subsystem* selezionata, è possibile avere un rapporto dettagliato sui tempi di

esecuzione di ciascun blocco: Il Real-Time Workshop inserirà infatti delle opportune chiamate all'inizio ed alla fine di ogni blocco di cui si desidera fare il profiling. from Texas Instruments.)

Lanciando e facendo girare il codice così generato per alcuni istanti sarà quindi possibile invocare la funzione profile dalla linea di comando di matlab, ottenendo così un rapporto dettagliato. Per i dettagli su questo tipo di operazione si rimanda alla documentazione MATLAB/Simulink (3).

Se si desidera invece un semplice misura della percentuale di utilizzo del tempo di CPU è possibile selezionare, con il programma in esecuzione, *CPU Load Graph* su Code Composer Studio. In questo modo verrà disegnato in tempo reale un grafico con l'andamento del tempo di utilizzo del processore. Per ulteriori dettagli su questo tipo di operazione si rimanda alla documentazione di Code Composer Studio (4).

## 6 Realizzazione del prototipo



E' stato assemblato uno chassis di dimensione Rack 2U standard, con alloggiata la scheda DSK con il processore TMS320C6711, l'alimentatore ed una piccola board, appositamente realizzata, contenente un preamplificatore stereo a basso rumore con guadagno in tensione tra 0 – 10, modificabile (altri particolari in appendice). Sul pannello frontale dello chassis sono stati montati gli interruttori e indicatori di presenza dell'alimentazione della scheda, i potenziometri per la regolazione del livello di ingresso del segnale analogico, vari spinotti jack stereo da 3,5 mm e da 6 mm di diametro per gli ingressi e le uscite.

Il controllo di superamento della dinamica dei convertitori è stato implementato in sw sul DSP: per la segnalazione sono utilizzati gli USERS LED disponibili come risorsa

“funzione utente” sulla scheda. Per comodità di uso sono quindi stati duplicati gli USERS LED, gli USER SWITCH ed il RESET sul pannello frontale del sistema, tenendo conto del fan-out dei circuiti logici della scheda.

## 6.1 Preamplificatore

E' stato progettato e realizzato un preamplificatore stereo a basso rumore per poter sfruttare appieno la dinamica d'ingresso dei codec, amplificando opportunamente i segnali in ingresso. Per sfruttare al meglio le funzionalità degli op-amp si è utilizzata una tensione duale +12,-12V ottenuta da un adatto convertitore DC-DC. Le caratteristiche del preamplificatore sono le seguenti:

- gain =  $V_{out}/V_{in}$  0 - 10
- impedenza d'ingresso 47K $\Omega$
- Vcc modulo alimentatore +5v
- Vcc preamplificatore +/-12v
- Banda passante (3db) 0 – 33khz

## 7 Test audio

Sono state condotte delle verifiche qualitative di laboratorio che sostanzialmente confermano le caratteristiche di risposta in frequenza dichiarate nei datasheet dei codec presenti nelle scheda DSK e nella scheda figlia PCM3003.

Per quanto riguarda il rapporto segnale rumore, intorno ai 60 dB, si nota un certo peggioramento rispetto alle caratteristiche dichiarate dovuto al posizionamento dei Codec adiacente ai circuiti digitali. La qualità complessiva è comunque da ritenersi sufficiente per soddisfare all'applicazione richiesta.

## 8 Conclusioni, sviluppi futuri

In questa prima serie di sperimentazioni l'algoritmo è stato testato, per semplicità, unicamente utilizzando il codec TLC320AD535 presente sulla scheda madre.

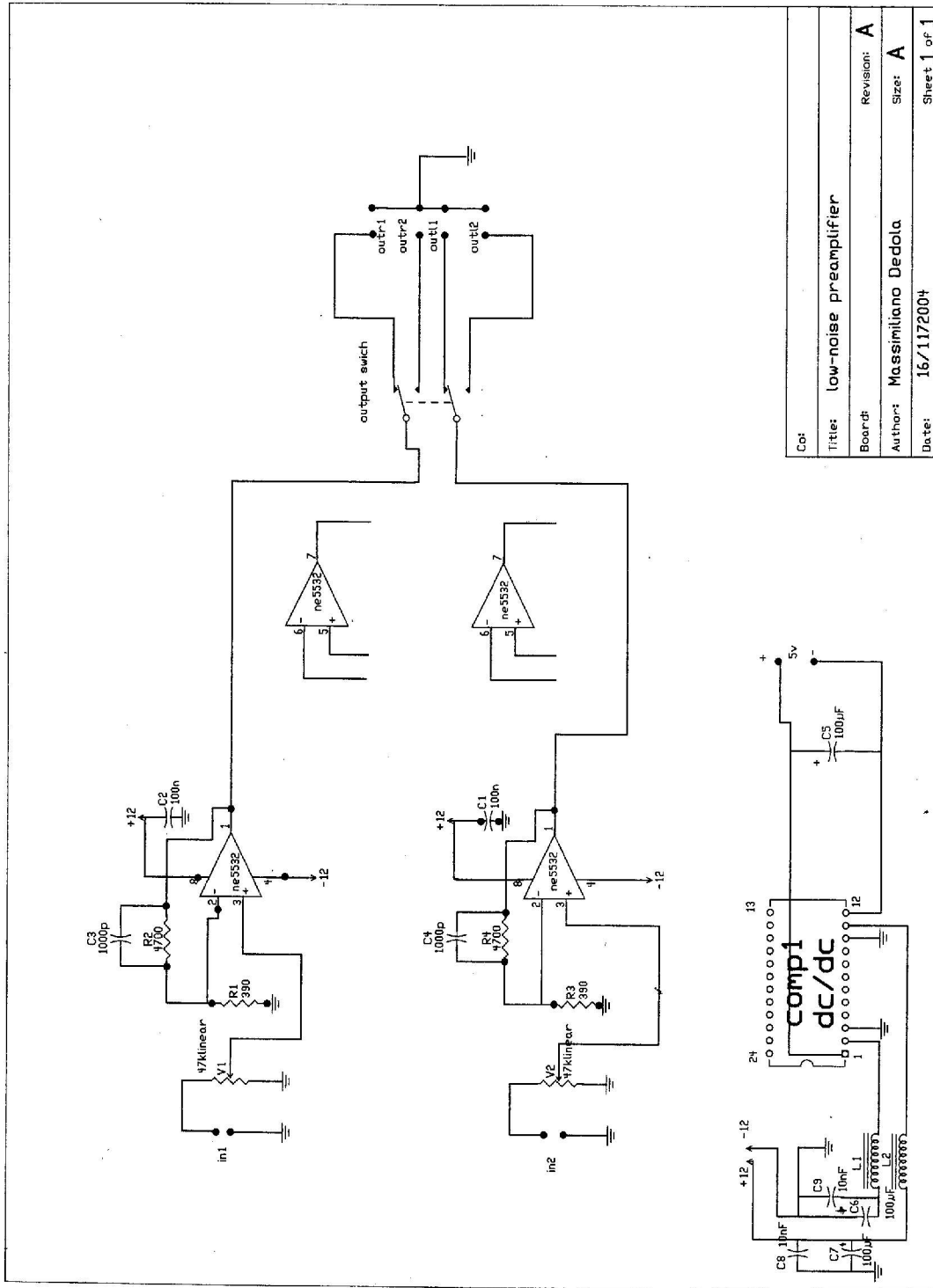
Pur con gli ovvi limiti della banda passante (4 kHz) del codec si sono comunque affrontate le problematiche relative all'approccio del Rapid Prototyping, che verranno discusse più approfonditamente in successivi documenti.

In particolare si sono evidenziate difficoltà nel caricamento dei programmi da PC alla scheda DSK, sia come velocità che come affidabilità, probabilmente dovuti all'utilizzo della connessione tramite porta parallela. A questo scopo è stato verificato che un approccio alternativo basato sull'utilizzo di un emulatore che sfrutta una connessione più diretta con il processore come quella JTAG, eliminerebbe questi inconvenienti.

Le risorse della scheda figlia contenente un codec di qualità migliore (16 bit, 48 kHz stereo) verranno sfruttate successivamente per ulteriori test dell'algoritmo.

# 9 Appendice

## 9.1 Schema elettrico preamplificatore



## 9.2 Note sui componenti utilizzati

Caratteristiche principali *Ne5532*

- Compensato internamente in frequenza
- Power supply +/- 22v max
- Slew rate 8V/ $\mu$ s
- Input noise voltage 8nv/ $\sqrt{\text{Hz}}$  at f = 3 Mhz
- Banda passante 140 Khz
- Prodotto guadagno banda 10 Mhz
- Impedenza d'ingresso 300 K $\Omega$
- Cmrr 100 db

Caratteristiche principali *TEM2-0521*

- Vin = 5 V +/- 10%
- Vout = +/-12 V
- Iout = 80 ma
- Efficienza 53%
- Ripple and Noise (20 Mhz ) 50 mv pk-pk max
- Switching frequency 80 Khz

Lista componenti:

R1=R3	390 $\Omega$ 1/4 watt
R2=R4	4700 $\Omega$ 1/4 watt
C1=C2	100nf
C3=C4	1nf
C5=C6=C7	100 $\mu$ f 25v
C8=C9	10nf
V1=V2	47K $\Omega$ lineare
comp1	DC/DC Converter Traco Power TEM 2-0521
L1=L2	impedenza tipo VK200
U1=U2	NE5532

## 9.3 Misura del rapporto segnale rumore

### 9.3.1 Strumenti utilizzati

- Generatore di segnali ESCORT EGC3233
- Voltmetro RMS/dB meter Hewlett Packard 3400A
- Oscilloscopio Tektronix 2445

Il generatore di segnali è stato impostato a 1000 Hz, connesso all'ingresso analogico del sistema ed alla traccia 1 dell'oscilloscopio. L'uscita della scheda è stata connessa al voltmetro RMS ed alla traccia 2 dell'oscilloscopio. Il guadagno di ingresso del preamplificatore della scheda è stato impostato a 1 (unitario).

Tramite l'uso della traccia dell'oscilloscopio connessa all'uscita della scheda si è regolata l'ampiezza del generatore di segnali in modo da avere in uscita il massimo di segnale indistorto. La misura in volt picco-picco del livello di ingresso d'altra parte è coerente con la dinamica consentita dei codec (circa 3 Vpp). A questo punto si è rilevato il livello misurato dal Voltmetro RMS, che rappresenta il massimo livello di segnale utile.

Per la misura del livello di rumore generato dal sistema si è proceduto come segue:

Si è scollegato il generatore di segnali e si è cortocircuitato l'ingresso alla scheda DSP. Si è misurato il livello di uscita dal sistema con il Voltmetro in posizione dB meter. Il rapporto S/N viene quindi calcolato come la differenza fra i due livelli misurati. Una riprova della misura tramite voltmetro è stata condotta misurando i livelli picco-picco del segnale e del rumore tramite l'oscilloscopio e calcolando il rapporto segnale-rumore come

$20 \cdot \log(\text{livello segnale/livello rumore})$ . Le misure condotte con le due modalità sono congruenti.

La prova viene ripetuta utilizzando il Codec integrato nella scheda madre ( un TLC320AD535C/1 16 bit mono, 8 kHz sampling rate) ed il codec della scheda figlia (un PCM3003 16 bit stereo, 48 kHz sampling rate).

### *9.3.2 Esito della misura*

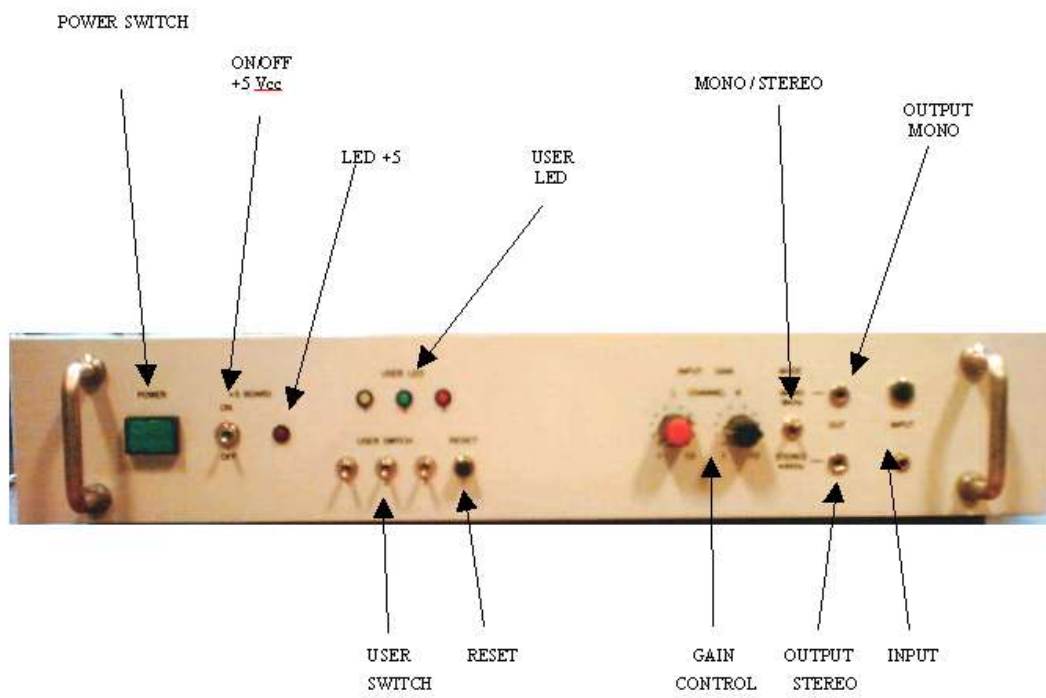
L'esito della misura, utilizzando i due diversi codec è il seguente:

- a) Codec scheda madre: 53 dB*
- b) Codec scheda figlia: 58 dB*

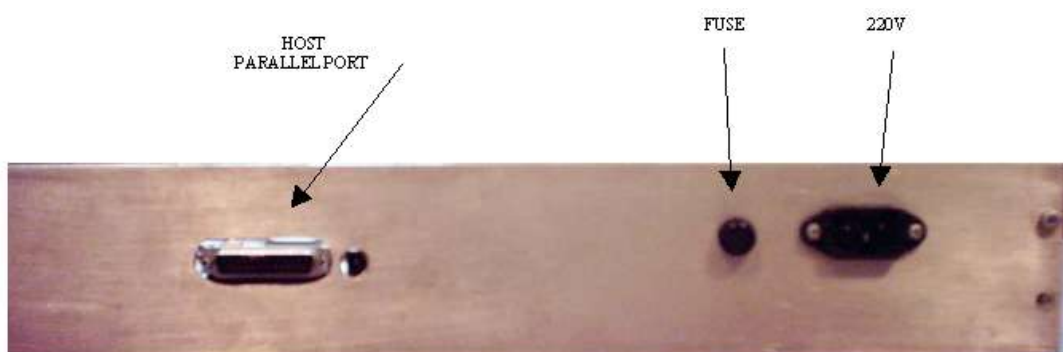
Includendo nella misura una pesatura con la curva A, si può stimare un miglioramento di circa 3 dB in entrambe le misure, per cui nel caso a) si ha *56 dB* e nel caso b) poco più di *60 dB*.

E' stato anche verificato che l'inserzione del preamplificatore è ininfluenza sui valori ottenuti. Esaminando le specifiche dei codec utilizzati, gli scopi e le applicazioni per cui è stata configurata la scheda Texas, pensiamo che le caratteristiche misurate con strumenti di maggiore precisione non si discostino molto dai livelli rilevati.

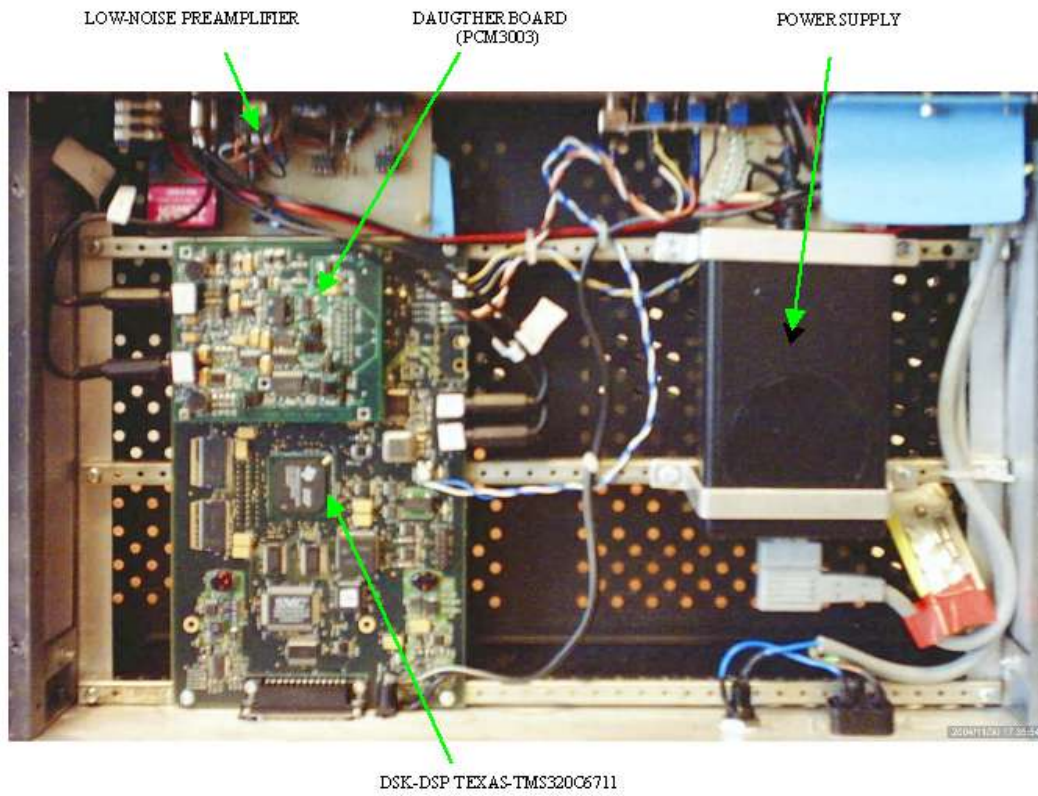
## 9.4 Pannello frontale



## 9.5 Pannello posteriore



## 9.6 Interno



## 10 Bibliografia

1. G.Bertini et. aavv. *MICROPROCESSORI PER DSP. Modelli recenti e tools di sviluppo: alcune applicazioni nel settore Telecom*. Nota interna ISTI-CNR B4-10, Maggio2003
2. D.L.Gonzales, O.A.Rosso  
“*Qualitative Modeling of Complex Biological Systems: a nonlinear dynamical system approach*” “2° Italian/Latinamerican conference on applied industrial mathematics, Roma 1996
3. Mathworks  
*MATLAB, Signal Processing Toolbox, Simulink user guide*
4. Texas Instruments  
“*TMS320C6700 One day workshop with Starter Kit c6711DSK*”,
5. Texas Instruments  
“*Code Composer Studio – Getting Starter guide*”