

Paula & il Marchingegno

di Maurizio Damiani Chersoni, Luca Galeani, Giuliano Peritore

Secondo appuntamento con l'hardware di Amiga. Dopo lo strepitoso successo raccolto per il numero scorso (bugia, come sapete quando scriviamo questa parte del numero X il numero X-1 non è ancora in edicola!) i nostri baldi giovani neo collaboratori di MC tornano alla carica con una nuova sfornata di ferraglie (hardware) relativa al nostro amato-odiato Amiga. Non vi nascondiamo che originariamente questo articolo era lungo il doppio e comprendeva anche una modifica hardware da effettuare, saldatore in mano, sul circuito del nostro malcapitato computer per migliorare una particolare feature. Il taglio, non dovuto a censura come i più maligni penseranno, si è reso necessario per i soliti stramaledettissimi motivi di spazio. Non dubitate però, non vi chiediamo di acquistare la parte mancante su disco (ohibò), ma semplicemente di pazientare, i più coraggiosi, fino al mese prossimo. Volete sapere di cosa si tratterà? Eh, no! MC non è solita fare anticipazioni. E ora morite pure di curiosità, nel frattempo «cuccatevi» queste propedeutiche righe. Buona lettura...

adp

Introduzione

Salve a tutti!

Siamo sempre noi, i tre hardwaristi pazzi che il mese scorso vi hanno presentato un pomposissimo Disk-Monitor... hardware. Ovviamente coloro che non hanno comprato il numero di ottobre saranno così rosi dal dolore che non perderanno un numero di MCmicrocomputer fino a quando camperanno.

Scherzi a parte, vediamo gli argomenti di cui ci occuperemo in questo mese.

Come progetto hardware costruiremo un circuito funzionalmente equivalente al ben noto «Marchingegno» della RAI. Se non avete molta memoria, o se siete lettori saltuari di MC (beh, è sempre meglio che non leggerlo mai) alzatevi e cercate nella vostra MC-teca il numero 73 (aprile '88), posto di averlo, andate a pagina sessantanove e leggetevi l'articolo di AdP.

Dalla costruzione della piccola scheda (il Marchingegno) passeremo, con un chiarissimo legame concettuale, alla trattazione della gestione delle porte POT (sennò il dispositivo come lo usate?), alla spiegazione del DMA Audio, alla effettuazione di una modifica hardware da apportare al vostro beneamato computer (l'Amiga).

Parlavamo di legami concettuali... tutte le cose di cui parleremo questo mese sono infatti state riunite in un unico articolo dato che tutte quante hanno a che fare con segnali audio. Scherzetto! Prevedendo le lamentele di un così banale legame, ho subito trovato un altro legame molto più profondo, si tratta di una entità che sovrintende e collega il tutto, un essere con un nome di donna, Agnus? Denise?, no ragazzi il mostro di questo mese è Paula.

Paula

Già a vedere le dimensioni di questa bella integratona ci spaventiamo. Si trat-

ta di ben 90-60-90... ops ... 48 piedini che tramite piste più o meno lunghe arrivano dappertutto: linee di interrupt, gestione del colore, gestione del drive, gestione della porta seriale, controllo dei canali DMA dell'audio e controllo della porta delle paddle.

Ed è proprio da tale porta che inizierà il nostro discorso.

Le porte delle paddle

Le più semplici periferiche che possiamo connettere alla porta delle paddle sono per l'appunto... le paddle. Tale porta viene però utilizzata anche da mouse, joystick proporzionali, tavolette grafiche (anche la KoalaPad può essere attaccata all'Amiga) e marchingegni vari. Tutti questi aggeggi hanno lo stesso modo di gestire la porta con cui sono collegati.

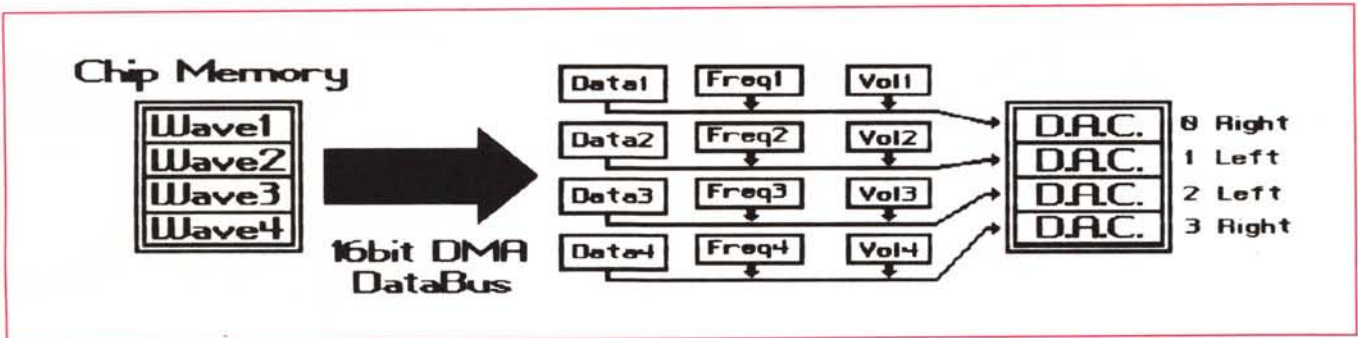
Se andiamo a vedere la piedinatura di una delle due Gameport, che sono pressoché (e non totalmente) identiche, troveremo, oltre a +5v, GND e le cinque linee del joystick, anche due piedini chiamati POT0X, POT0Y nella porta uno e POT1X, POT1Y nella porta due.

Le paddle non sono altro che semplici potenziometri lineari da 470k collegati tra il piedino 7 (5 volt) e il piedino 9 per la paddle sinistra (o Y), o 5 per la paddle destra (o X).

Mentre sul buon vecchio caro C64 per leggere la porta delle paddle bastavano due semplici peek (54297 e 54298) la faccenda sull'Amiga è leggermente più complicata.

Durante il vertical blanking l'utente dovrebbe settare a \$0001 il registro POTGO. Questa operazione provoca l'azzeramento dei valori POTX e POTY e l'avviamento dell'hardware di controllo che avviene dopo circa sette linee raster.

Il controllo avviene praticamente in questo modo: in serie alla paddle è collegato un condensatore che inizia a



Audio DMA.

caricarsi non appena viene settato il registro POTGO. Ad ogni linea raster un comparatore confronta la carica del condensatore con un valore prefissato. Se il valore è minore di quello di confronto il contatore della paddle viene incrementato. Se il valore è maggiore o uguale il valore viene congelato, badate bene, fino al prossimo video frame in cui il contatore continuerà ad essere incrementato.

L'operazione di lettura andrebbe quindi fatta in questo modo: subito dopo il vertical blanking (circa alla terza-sesta linea) dovrebbe essere settato a \$0001 il registro POTGO (\$DFF034), poi si dovrebbero svolgere le altre operazioni fino al successivo vertical blanking dove, nelle primissime linee di scan si dovrebbero leggere i valori dei registri POT0DAT (\$DFF012) e POT1DAT (\$DFF014), registri che contengono nei bit

15-8 il valore del POTY e nei bit 7-0 il valore dei POTX.

Guardandosi intorno...

In effetti il bit importante del registro POTGO è il bit 0. Vi sono tuttavia altri otto bit che generalmente non vengono mai usati ma che sarebbe un errore trascurare data la loro importanza. Ciò è dovuto al fatto che i quattro canali delle paddle possono essere programmati in uscita o in entrata come una mezza porta parallela con tempo minimo di cambio di stato di 300 microsecondi.

I bit 15, 13, 11, 9 di POTGO (\$DFF034) sono i cosiddetti Output Enable rispettivamente per i canali Right9, Right5, Left9, Left5, mentre i bit 14, 12, 10, 8 sono i veri e propri dati. Queste linee che non vengono utilizzate praticamente mai in uscita sono utilizzate co-

me entrate non analogiche molto spesso. Un esempio è il tasto destro del mouse che utilizza la linea POTY o piedino 9.

Audio DMA

Avevamo detto che Paula si occupa anche della gestione della parte audio. Infatti è proprio in questo chip che si trovano i quattro convertitori DAC (Digital to Analog Converter) che l'Amiga utilizza per produrre segnali audio. Il principio di funzionamento è praticamente identico, se non consideriamo la precisione, a quello utilizzato nei compact disc. Dato che un computer non potrebbe registrare i valori analogici che compongono un brano musicale, attraverso opportune interfacce, i digitalizzatori audio, il suono viene codificato in memoria in forma digitale, praticamente

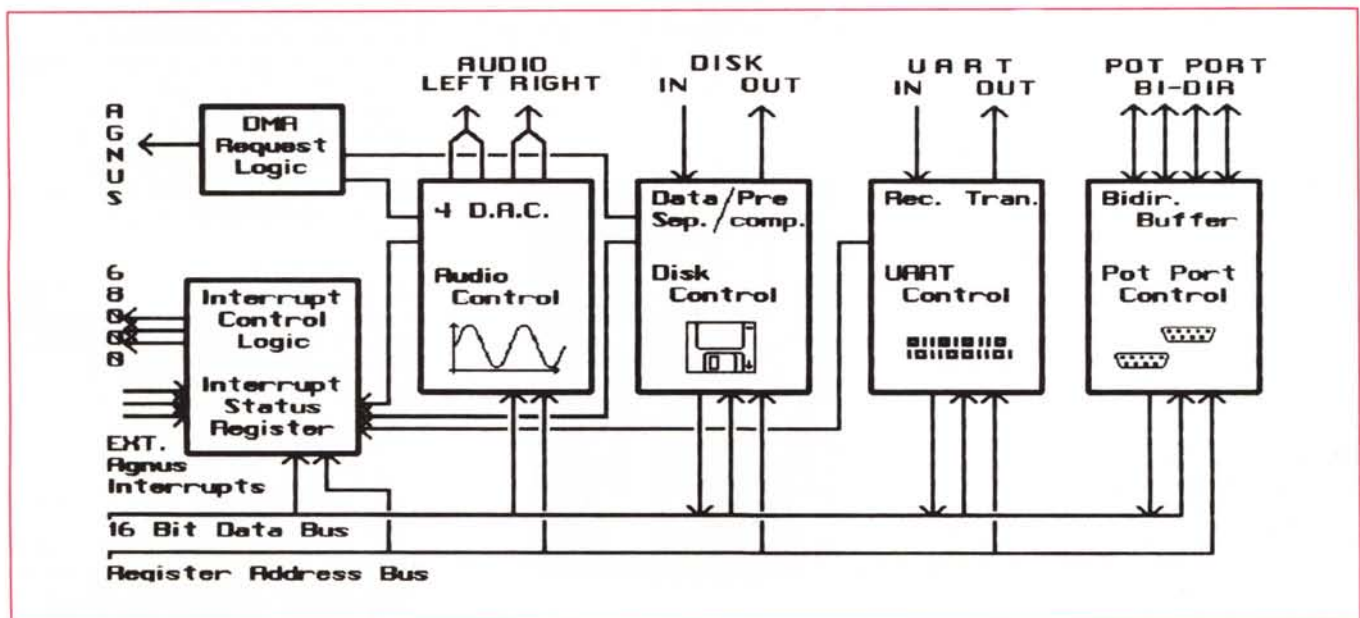
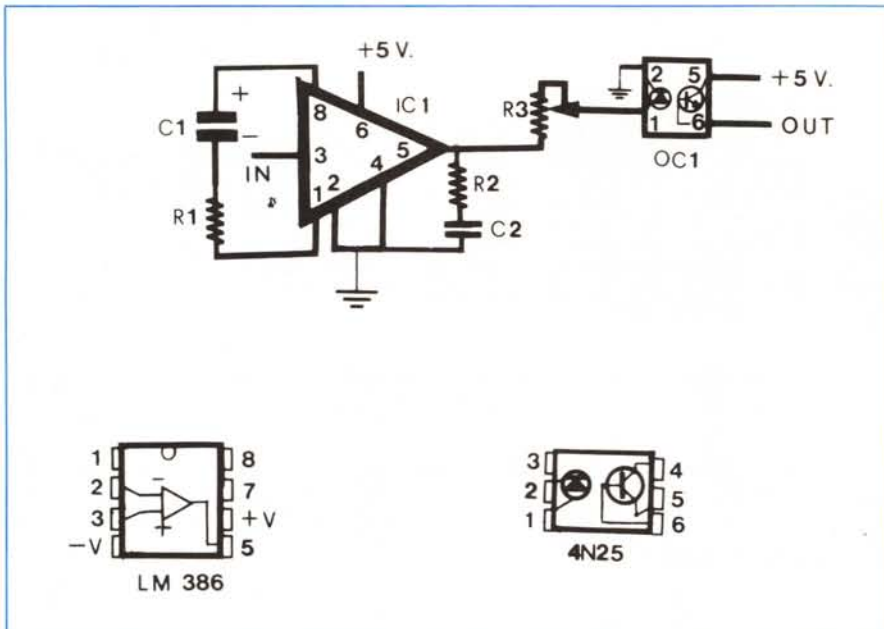


Diagramma a blocchi di PAULA.



Schema elettrico del circuito e piedinatura integrati.

misurando a intervalli regolari l'ampiezza della forma d'onda e convertendo tale valore in un numero la cui precisione varia a seconda del sistema usato. Nell'Amiga è generalmente di otto bit. I convertitori DAC non fanno altro che rileggere tali valori della memoria e riconvertirli in valori analogici da inviare alle uscite audio.

Il fatto importante è però che i DAC leggono la memoria tramite il DMA (Direct Memory Access) e quindi praticamente, quando sono in funzione, non rallentano il 68000.

Questo avviene perché il 68000 accede alla memoria durante i cicli pari, mentre effettua operazioni interne durante quelli dispari, lasciando così libero l'accesso alla memoria a tutte le altre funzioni che usano il DMA, quali Copper, DAC, Blitter, Disco, BitPlane e Sprite.

Abbiamo detto che vi sono quattro convertitori. Essi sono abbinati a due a due e, precisamente, lo zero ed il terzo sul canale destro ed il primo e il secondo sul canale sinistro. Questo artificio consente la produzione di brani stereofonici o di simpatici effetti audio.

La gestione dei convertitori è semplicissima. Una volta creata in memoria una zona contenente i dati (ne parleremo nel prossimo paragrafo) basta far partire il convertitore che continuerà indisturbato a ripetere quella forma d'onda fino a quando, terminate le nostre operazioni, ci decideremo a fermarlo.

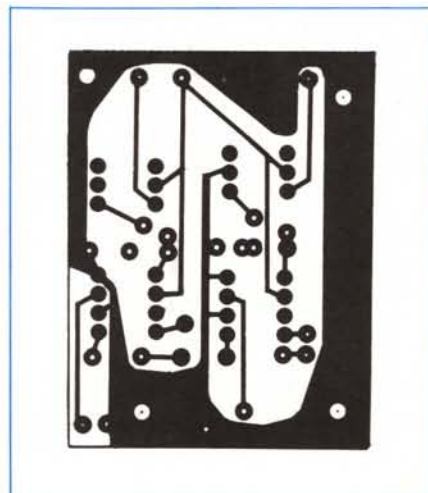
Tratteremo la gestione del convertito-

re zero, ma comunque sappiate che l'unica differenza fra le gestioni dei quattro DAC consiste nel fatto che per passare da un DAC al DAC successivo basta sommare agli indirizzi del convertitore zero \$10 per il primo DAC, \$20 per il secondo e \$30 per il terzo.

Per controllare il tutto ci bastano cinque istruzioni in LM. Con la prima comunichiamo al convertitore dove si trovano i dati da leggere e lo facciamo così:

```
move.l address, $DFF0A0
```

Con la seconda dobbiamo definire la lunghezza dei dati. Badate bene che la



Circuito stampato lato rame.

lunghezza va espressa in word e non in byte (lunghezza in word = lunghezza in byte/2):

```
move.w length, $DFF0A4
```

La terza istruzione ci serve per definire la velocità di lettura espressa in numero di cicli tra due campionamenti successivi. Dato che il valore minimo è 124 avremo una frequenza massima riproducibile di 28.867 kHz. L'istruzione è la seguente:

```
move.w period, $DFF0A6
```

A questo punto non ci rimane che settare il volume:

```
move.w # val, $DFF0A8
```

dove val è in un range tra zero e sessantaquattro, e far partire il DMA:

```
move.w # $8201, $DFF096
```

il valore \$8201 è per il convertitore zero. I valori per gli altri convertitori sono \$8202, \$8204, \$8208.

Ah... dimenticavo, se non volete che i vostri timpani siano martellati in eterno potete sempre fermare i convertitori con le seguenti istruzioni:

```
move.w # $0001, $DFF096 ;conv 0
```

```
move.w # $0002, $DFF096 ;conv 1
```

```
move.w # $0004, $DFF096 ;conv 2
```

```
move.w # $0008, $DFF096 ;conv 3
```

Dati audio e IFF

I dati che manderete ai convertitori possono essere di vari tipi: suoni calcolati da programmi da voi scritti, suoni prodotti da programmi in commercio (Synthia ecc.), suoni digitalizzati, e dati processati da programmi vari tipo AudioMaster, ecc.

Nel caso di suoni da voi calcolati non ci sono problemi in quanto ne sapete tutte le caratteristiche salienti, quali lunghezza, frequenza, volume, ecc.

Se però provate a caricare in memoria un suono creato per esempio dall'AudioMaster, anche se effettuerete tutto alla perfezione, con tanto di allocazione di memoria, ecc. sarà difficile che indovinerete al primo tentativo il periodo e che non troviate del rumore all'inizio del suono.

La causa di tutto ciò è che l'AudioMaster registra i file in forma IFF. Purtroppo data la mancanza di spazio non è questa la sede adatta per una trattazione completa dei file IFF e quindi possiamo solo accennare brevemente alla struttura di tali file.

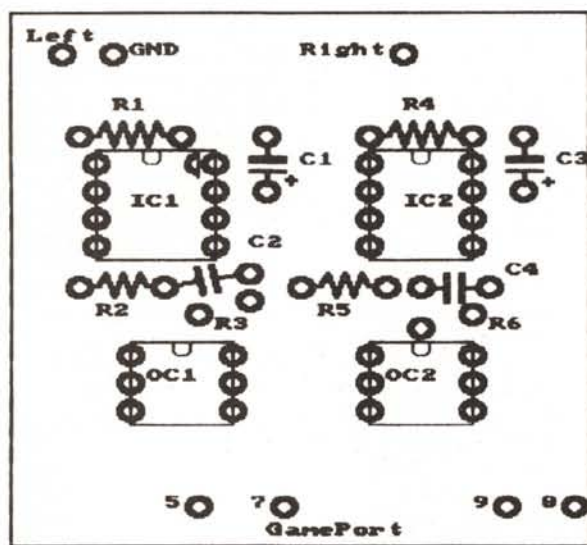
Supponiamo di aver creato un file con l'AudioMaster. Se andiamo a leggerlo troviamo diverse zone di dati chiamate Chunk.

Per prima cosa troviamo la parola «FORM» seguita dalla lunghezza del file e dalla parola «8SVX» che esprime il tipo di file IFF. Continuando a leggere troviamo «VHDR» seguito dalla lun-

Lista componenti:

R1, R4: 560 Ohm
 R2, R5: 10 Ohm 1 Watt
 R3, R6: 10 KOhm potenziometro
 C1, C3: 22 μ F 16 V
 C2, C4: 470.000 pF poliest.
 IC1, IC2: LM386
 OC1, OC2: 4N25 (fotoaccopp.)

Montaggio
 componenti.



ghezza dei dati che fanno parte di questo Chunk e che sono:

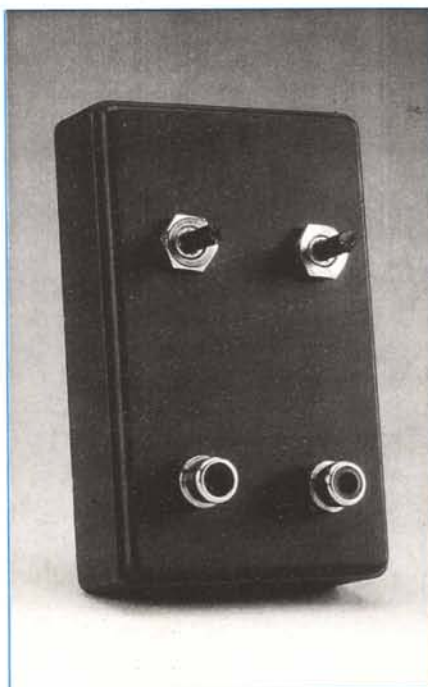
LONG	OneShotPerHiSamples
	RepeatHiSamples
	SamplesPerHiCycle
WORD	Sampling Rate
UBYTE	Octaves
	Compression
	Volume

Continuando a leggere il file troviamo il Chunk «NAME» seguito dalla lunghezza del nome e dal nome del suono, il Chunk «ANNO» in cui sono contenute annotazioni varie, sempre precedute dalla lunghezza, ed infine il Chunk «BODY», seguito dalla lunghezza dei dati da leggere e quindi dai dati del suono campionato.

Il Marchingegno

Questo mese vi proponiamo niente di meno che un circuitino funzionalmente equivalente al Marchingegno della RAI.

Ah, se avessimo saputo che alla RAI interessava una macchinetta del genere vi avremmo scritto dal nostro jet priva-



to: infatti la RAI pagava un fior di affitto per una macchinetta realizzabile con poche migliaia di lire che qualunque ragazzo «volenteroso» avrebbe costruito in poche ore.

Allora dove eravamo? Ah, sì il nostro marchingegno si basa essenzialmente sulla conversione dell'intensità della voce (sarebbe meglio dire ululati) provenienti da un microfono in un impulso Amiga-compatibile.

Nel nostro caso ci serviremo di un operazionale che, con specifico guadagno, amplifichi il segnale in ingresso in una tensione compresa tra 0 e 5 volt. Questo valore viene poi applicato ad un isolatore ottico che lo riporterà in uscita disaccoppiato. Inutile dirvi che le uscite del nostro circuito saranno lette dagli ingressi Paddle dell'Amiga.

Se il vostro cervello ha raggiunto la temperatura del saldatore fatevi pure due passi mentre io dico qualcosa ai geni che hanno capito tutto.

L'operazionale usato è un LM 386 che contiene nel suo interno un amplificatore a guadagno variabile che ci permette di amplificare l'onda in ingresso un numero ben preciso di volte: tale valore può essere determinato dai valori del condensatore e della resistenza posti tra i piedini 1 e 8 dell'integrato; si tenga presente che se si lasciano liberi i piedini si avrà un guadagno di 20 volte, con il solo condensatore (un 22 mF elettrolitico) si avrà il max guadagno pari a 200 volte.

Nel nostro caso abbiamo limitato il guadagno con una resistenza da 560 ohm.

Proseguendo nella descrizione, sull'uscita di tale operazionale abbiamo posto una resistenza e un condensatore per prevenire eventuali instabilità del circuito ed infine abbiamo pilotato l'accoppiatore ottico dopo aver opportunamente limitato il segnale con il potenziometro R3.

La funzione dell'accoppiatore è quella di separare fisicamente la parte del circuito ora descritta con l'Amiga; infatti nel suo interno troviamo un diodo led che pilota un fototransistor sulle cui uscite troviamo il valore da applicare sugli ingressi potenziometrici dell'Amiga.

Spero di essere stato abbastanza chiaro e comunque potete aiutarvi con lo schema elettrico allegato.

Un ultimo appunto riguardante il circuito stampato: questo è in versione stereo ovvero potremo applicare due ingressi separati ed avere due opportuni valori in uscita; a voi la realizzazione di un giochino controllato con un joystick vocale!!!

MC