

I comandi con valori compresi fra 32 e 255 non faranno altro che impartire il print del carattere equivalente al codice inviato.

Poc'anzi si accennava alle capacità del VDU Driver dedicato al controllo della stampante.

Secondo il modo di procedere che mi sono dovuto imporre, questo è da ricondurre all'argomento più generale del Video Device Unit.

Un argomento questo che oltre a

ricoprire un ruolo di massima importanza nella gestione sotto RISC-OS, è anche di estrema complessità. Non è quindi possibile riassumerne le caratteristiche in una overview genericamente dedicata al sistema per l'output dei caratteri.

In tal senso, il *microcosmo VDU*, sarà analizzato, in modo completo e (spero) esaustivo, in un prossimo articolo unicamente dedicato alla gestione video.

Una sorta di "inserto speciale" comunque compreso nel corso.

Character Input

Specularmente al Sistema per l'Output dei Caratteri, il Kernel dispone anche del Character Input System delegato dal RISC-OS a ricevere l'input dai device annessi al sistema. La tastiera anzitutto e poi la porta seriale e i file gestiti dal Filing System (che quindi

Piccola scheda tecnica: l'ARM Hardware

d Bruno Rosati

Contemporaneamente al primo articolo riguardante il RISC-OS, inauguriamo la serie delle "piccole schede tecniche" con le quali si cercherà d'integrare gli argomenti trattati nelle varie edizioni mensili.

La prima scheda tecnica è relativa alla configurazione hardware della macchina, con particolare riferimento ai quattro chip principali che compongono l'ARM-set

L'architettura degli Archimedes è basata su di un ARM-set composto di quattro chip dedicati.

L'ARM (acronimo di Acorn RISC Machine), quale processore principale; il VIDC (VIDeo Controller) dedicato al controllo delle modalità di output sia del video che del suono; lo IOC (Input Output Controller) predisposto alla gestione degli interrupt nella comunicazione con le periferiche connesse; e dal MEMC (MEMory Controller) il quale è destinato all'azione d'interfacciamento con gli altri chip per la gestione della memoria. Se osservate la figura 1 essa ci permette un primo, schematico sguardo sulla struttura hardware di un RISC-OS computer. Come semplice osservazione generale possiamo già rilevare il tipo di connessione con cui vengono organizzati i quattro chip. Con l'ARM collegato direttamente al MEMC, bidirezionalmente al data-bus e indirizzato verso quello degli indirizzi.

Poi il VIDC, con la sua rispettiva connessione al controller della memoria, al monitor e all'utilizzatore sonoro verso i quali, attraverso i VDU driver, convoglia le informazioni ricevute attraverso il data-bus. La dislocazione dello IOC è costruita invece in modo più articolato; essendo tale controller addetto alla gestione dell'Input-Output, è stato ovviamente reso disponibile un data-bus dedicato (chiamato I/O data-bus) al quale si annettono, con linee bidirezionali, tutti i vari device supportabili dal sistema (unità a dischi, collegamenti in rete, porte seriali e podule varie). Attraverso un latch diretto lo stesso IOC è quindi collegato al bus di sistema. Da notare infine il solito interfacciamento diretto con il MEMC. Anche la RAM e la ROM contenente il RISC-OS sono chiaramente connesse al

bus di sistema. L'unico chip a non avere una connessione diretta al bus di sistema è il MEMC.

Interfacciato con tutti i chip-set e il RISC-OS, risulta in connessione con la sola RAM della quale opera la conseguente gestione.

Dall'interpretazione dello schema ad una breve scheda dei singoli chip del set, passiamo ad introdurre l'ARM. Un micro come è ormai notorio, a ridotto set d'istruzioni, viaggiante intorno ai 5 MIPS (Milioni d'Informazione Per Secondo) che usa parole complete

da 32 bit, con un data-bus che completa un giro d'istruzione con un solo ciclo ed un bus d'indirizzamento a 26 bit che porta il totale della memoria ai fatidici 64 Mbyte.

I suoi modi operativi sono quattro. L'User Mode (quello usato nei programmi) il Supervisor Mode (SVC) usato con le chiamate a sistema SWI; l'Interrupt Mode (IRQ) dedito alle periferiche ed il Fast Interrupt Mode (FIQ) simile al precedente ma ovviamente "più fast". Escluso l'User gli altri tre modi operativi si definiscono convenzionalmente

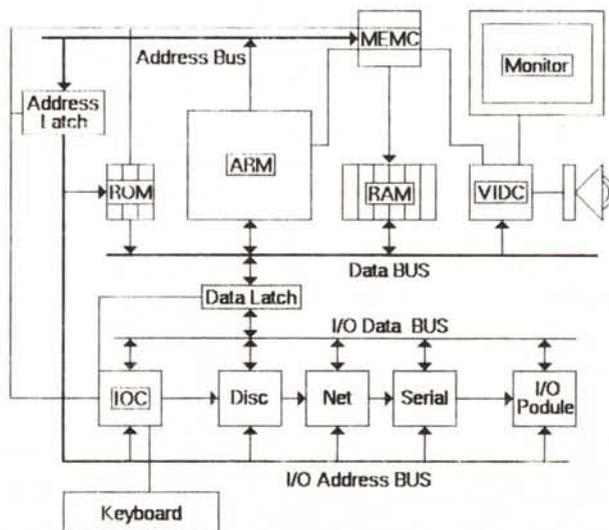


Figura 1 - Schematizzazione generale della struttura hardware di una macchina RISC-OS. Si notino in particolare le connessioni fra i quattro componenti del chip-set con particolare riferimento al MEMC interfacciato all'ARM, il VIDC e lo IOC per i quali gestisce la memoria disponibile.

operano la doppia funzione di I/O) sono i tre device controllati (permettetemi l'abbreviazione) dal C.I.S.

La stessa caratteristica principale dell'Output — ovvero la gestione tramite i sistemi di stream — è valida anche per l'Input. Uno stream per la tastiera, uno per i file, uno per la porta seriale. E se il centro vitale del sistema streamer per l'output è da assegnare alla chiamata OS_WriteC, analogamente per l'Input, sarà quella del tipo OS_ReadC. Lo stes-

come "modi privilegiati" e risultano — se li citiamo è proprio per questo — usati abbondantemente nella realizzazione del RISC-OS in modo che quest'ultimo risulti il più possibile prossimo al massimo sfruttamento delle caratteristiche del chip principale.

L'ARM contiene tra l'altro 27 registri a 32 bit pieni dei quali solo 16 sono a disposizione dell'user che può accedervi attraverso la selezione di uno dei quattro modi sopra elencati.

VIDC, ovvero il chip dedicato al controllo del video e del suono in uscita. Partendo dall'informazione letta dalla RAM con relativo trasferimento nel buffer del microprocessore — il VIDC dispone di tre differenti buffer: uno dedicato alle informazioni video, uno a quelle del suono ed un terzo a quelle dei movimenti mouse/puntatore — il dato verrà prima processato e quindi convertito, da digitale ad analogico, per rendere un valore compatibile per pilotare il cinescopio oppure il sistema di amplificazione del suono.

Il VIDC ha dalla sua l'elasticità operativa di essere programmato, attraverso le funzioni proprie del RISC-OS, producendo diversi formati di schermo. Per l'esattezza, sotto RISC-OS, sono possibili ben 27 screen-mode.

Lo **IOC** a sua volta svolge la propria attività, perlappunto di controllo, su interrupt e periferiche, attraverso il relativo I/O databus operando attraverso un set di funzioni predisposte estremamente elastiche e potenti.

In particolare è in grado di attivare quattro diversi tipi di contatori a 16 bit. Due dedicati al controllo della tastiera e della porta seriale, un terzo utilizzato per temporizzare gli eventi ed infine un quarto a completa disposizione del programmatore.

La programmabilità dei quattro contatori è il segno dell'estrema flessibilità della gestione generale dell'I/O delle macchine sotto RISC-OS.

Per quanto riguarda il **MEMC**, come già accennato, tale chip è connesso con tutti gli altri componenti il set, il banco della RAM e la ROM del RISC-OS.

In pratica la funzione del MEMory Controller è quella di gestire la distribuzione della memoria che l'ARM, il VIDC e lo IOC richiedono per svolgere la loro attività esecutiva.

Una vera e propria "interfaccia intelligente" posta fra le risorse del sistema e quelle

so dicasi per il procedimento di bufferizzazione.

Come si diceva la tastiera è la base principale per l'invio degli input e il relativo **Keyboard Driver** riveste la massima importanza.

Il funzionamento di una operazione di input da tastiera, tutto sommato è abbastanza facile da comprendere. Partendo dal fatto che ogni tasto premuto genera, o meglio: causa l'inserzione di uno stato d'interrupt al sistema, il codice del

PHIYSICAL RAM Size	PAGE Size	Number LOGICAL Pages
0.5 Mbyte	8 Kbyte	4K
1 Mbyte	8 Kbyte	4K
2 Mbyte	16 Kbyte	2K
4 Mbyte	32 Kbyte	1K

Figura 2 - La tabella di riferimento della memoria indirizzabile dal MEMC, suddivisa nelle pagine logiche che la versione attuale del chip è in grado di controllare.

che sono le rispettive attività dei componenti il chip-set.

Come valido esempio della funzionalità del MEMC possiamo riferirci ai tre buffer del VIDC che in realtà vengono predisposti ed organizzati proprio dal "controllore della memoria" sotto i canali di accesso diretti della memoria. Ovvero i DMA: Direct Memory Access.

Nel dettaglio si tratta di un circuito per il refresh del video, un buffer lineare per lo "sprite" del cursore e di un doppio buffer per le informazioni sonore.

Il MEMC, oltre a tale lavoro da "interfaccia", svolge l'importante compito del mapping della memoria fisica, ridistribuendola e catalogandola nei 16 Mbyte di logica equivalente e visibile da una base di 32 Mbyte. Il limite del MEMC, perlomeno nella sua versione attuale, è quello di imporre, al sistema tutto, l'indirizzamento di non più di 4MByte di User-memory. Tale memoria viene usualmente mappata in 128 pagine (sempre relativamente alla zona della RAM) ciascuna delle quali è siglata con un descrittore d'ingresso, in modo che la traslazione da logica a fisica possa avvenire con un accesso il più rapido possibile.

Il rapporto che intercorre fra memoria fisica e paginatura "pre-logica" è di tipo proporzionale. Più aumenta la memoria più diminuisce il numero delle pagine. Ne consegue, come è facilmente evidenziabile dalla figura 2, l'aumentata densità di ogni singola pagina.

I limiti del MEMC nella sua versione attuale sono, come già accennato, i limiti del sistema RISC-OS, malgrado simile sistema, sia già in grado di gestire fino a 256 pagine. Al riguardo sul Programmer's Reference Manual si ipotizza una nuova versione del MEMC in grado di "leggere" fino a 8 Mbyte di memoria.

determinato tasto che viene premuto, verrà "preso" dal Keyboard Driver e trasportato nel Keyboard Handler il quale a sua volta provvederà a convertirlo in una forma definitiva. O come carattere ASCII-equivalente o come un tasto-funzione, un cursore oppure un comando effettivo.

Stabilitane la forma infine, il carattere assegnato verrà immagazzinato nel buffer di tastiera, dove resterà in "attesa" fino a quando attraverso lo Stream System non si provvederà a copiarlo ed eseguirlo. Trasformandosi nel carattere (se il valore ASCII è equivalente a quello di un certo carattere stampabile) oppure eseguendo la determinata operazione codificabile.

Una caratteristica peculiare del Character Input System sono gli *Events*. Ovvero particolari stati che si generano ogni volta che ci si trova innanzi ad una interruzione del sistema. Con una spiegazione sommaria (come dicevamo nell'introduzione) il RISC-OS usa gli *Event* per indicare che è accaduto qualcosa. Il sistema dispone di 18 differenti tipi di eventi riscontrabili.

Nel caso specifico del C.I.S. se ne verificano cinque. Quello relativo all'avvenuto riempimento dell'input-buffer, l'inserimento di un carattere nello stesso buffer, il premere o il rilasciare un tasto, un errore occorso sulla seriale e la condizione di *escape* relativa al tasto stesso.

Dalla tastiera (Keyboard Driver) alla seriale, il sistema dell'input dei caratteri, sempre gestito con il procedimento della bufferizzazione è svolto attraverso l'operatività del **Serial Driver** che sarà addetto al controllo dell'input-buffer. La sua azione sarà quella di provvedere allo stop della trasmissione dei dati ogni qualvolta che il buffer sarà riempito; a tal punto da non avere più spazio per accumulare altri caratteri. Grossa importanza nella gestione della seriale è ovviamente quella rivestita dal settaggio dei *Baud rate* e dei codici di controllo.

Ecco, facciamo che il Serial Driver si è accorto dell'avvenuto riempimento del "nostro" buffer e fermiamoci qui per questa prima escursione nel RISC-OS.

Dopo esserci dedicati a tracciare per grandi linee la necessaria programmazione degli argomenti inerenti il corso e l'avvio della discussione sul Kernel con l'overview sui sistemi per l'Input e l'Output dei caratteri, il prossimo mese ci sposteremo immediatamente sull'importante argomento del Video Display Unit che oltre a completare il discorso iniziato con i sistemi appena descritti amplierà la discussione indirizzandola verso il completamento dell'argomento Kernel e le prime escursioni verso quelli che sono i moduli di estensione. 