

# La programmazione della scheda grafica VGA

Per poter affrontare la programmazione delle schede grafiche è indispensabile conoscere, anche se a grandi linee, i motivi che hanno spinto produttori hardware di computer a costruire schede grafiche sempre più potenti. In questi ultimi anni il mondo dei personal computer è stato caratterizzato da profondi cambiamenti hardware e software.

**V**i siete mai chiesti come facciano i programmi di gestione della grafica ad implementare funzioni avanzate ad una velocità di esecuzione elevata, anche su hardware a basso livello? Penso che a molti programmatore sia venuta, prima o poi, la voglia di realizzare degli effetti speciali grafici o addirittura dei veri e propri videogiochi utilizzando un linguaggio ad alto livello come il Turbo Pascal, il C, ecc.

Nonostante questi linguaggi consentano di accedere alle modalità grafiche delle VGA, attraverso comandi built-in (basic) o librerie (C, TPascal), esistono casi in cui la velocità di risposta della scheda video diventa il vero e proprio collo di bottiglia del sistema. Animazioni video o visualizzazioni in real-time di immagini grafiche durante un'acquisizione da scanner o convertitori AD (Analogico-Digitale) ad alta velocità richiedono tempi di risposta strettissimi che le funzioni standard non possono fornire. Si pensi che molte di queste librerie implementano le proprie primitive grafiche facendo un uso pesante dei servizi forniti dalla INT 10 del bios delle schede VGA stesse (che va a sostituire quello del firmware dell'elaboratore). Tali servizi implementano routine che prima di espletare il proprio compito compiono una serie interminabile di controlli, inizializzazione dei registri e temporizzazioni che, aggiunte ai tempi di accesso della ROM BIOS (centinaia di nS), rendono a volte inaccettabile anche il tempo impiegato per la semplice rigenerazione di un'immagine a 24 bit.

Per poter implementare delle buone funzioni grafiche è indispensabile conoscere in maniera approfondita l'architettura della propria scheda grafica e il modo per poterla programmare. Sul mercato esistono numerose marche di schede grafiche, per cui la conoscenza approfondita del funzionamento di tutte è quasi impossibile.

Comunque una conoscenza di base può essere d'aiuto, soprattutto se si fa riferimento allo standard VGA che ormai è disponibile sulla grande maggioranza dei

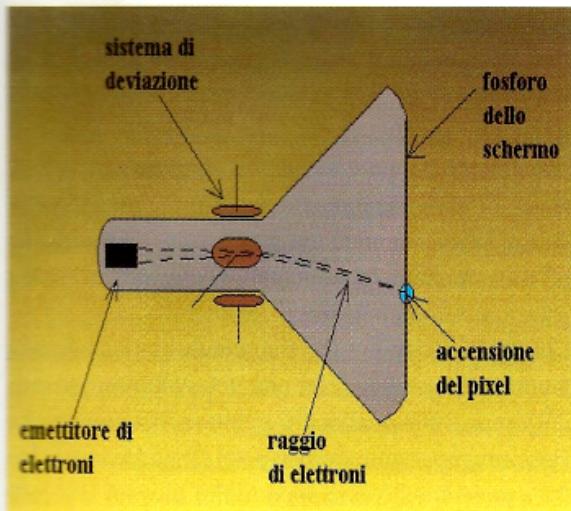
PC. In questo speciale cercheremo di spiegare il funzionamento generale di una scheda grafica e di fornire i dati necessari per poter programmare lo standard VGA.

Questo articolo deve essere considerato come una base per poter affrontare in un domani un programma che faccia diretto accesso ai registri interni delle schede compatibili VGA. Dal momento che l'argomento è molto vasto, in questa sede forniremo solo alcuni esempi pratici per dar spazio alla teoria. Nei prossimi numeri verranno presentati dei listati in linguaggio Assembly e Turbo Pascal che permetteranno di realizzare delle funzioni grafiche avanzate attraverso la programmazione diretta della scheda VGA. La comprensione dettagliata dei futuri articoli sarà vincolata alla conoscenza delle informazioni presentate in questo speciale. A questo punto siamo pronti per iniziare ad analizzare il mondo delle schede video.

## EVOZIONI DELLA COMPUTER GRAPHICS

Per poter affrontare la programmazione delle schede grafiche è indispensabile conoscere, anche se a grandi linee, i motivi che hanno spinto produttori hardware di computer a costruire schede grafiche sempre più potenti. In questi ultimi anni il mondo dei personal computer è stato caratterizzato da profondi cambiamenti hardware e software. Nell'ambito della computer graphics sono avvenute radicali evoluzioni nel trattamento computerizzato delle immagini. Chiunque, oggi, avendo a disposizione un semplice PC, con una scheda grafica VGA è in grado di produrre delle accattivanti immagini o grafici a livello anche professionale. La computer graphics è nata con lo scopo di migliorare i rapporti esistenti tra uomo e macchina. Essa è sorta intorno agli anni sessanta negli Stati Uniti d'America. Una delle prime definizioni di questa disciplina è quella data dalla Encyclopedia of Science and Technology: "the process of pictorial communication between men and computers" ossia il

Figura 2 -  
Funzionamento del  
CRT



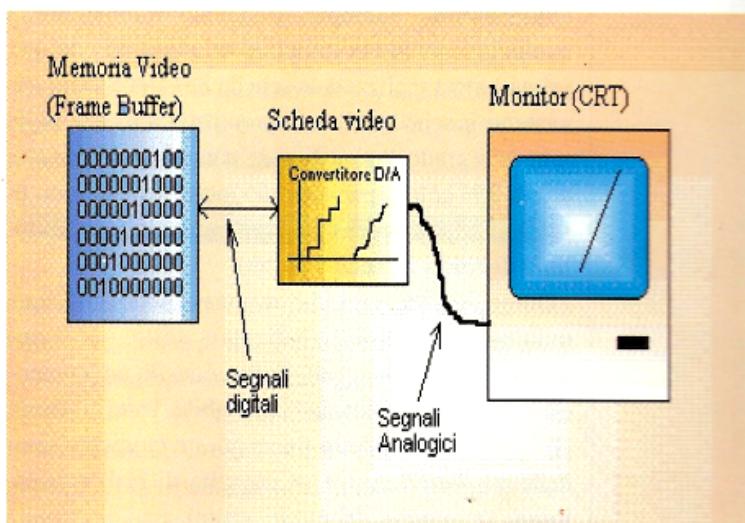
processo di comunicazione tramite immagini tra uomini e computers. Le prime ricerche di questa nuova scienza sono da ricondursi nel Lincoln Laboratory dove il ricercatore Sutherland progettò un sistema interattivo che consentisse ad un utente di comunicare con l'elaboratore TX-2 attraverso l'uso di una penna luminosa per il tracciamento di particolari figure sullo schermo CRT (Cathode Ray Tube = Tubo a Raggi Catodico). Tale sistema venne chiamato Sketchpad (1962) e fu una prima dimostrazione pratica di come fosse possibile migliorare l'interfaccia uomo-macchina. Dopo questo progetto ne iniziarono molti altri che devono essere considerati basilari per l'evoluzione che ha subito questa disciplina. E' d'obbligo ricordare la fondazione del C.T.G. (Computer Technique Group) avvenuta nel 1967 a Tokio della quale fecero parte persone molto giovani (intorno ai venti anni) con una formazione prevalentemente tecnica. Il C.T.G. ebbe un grande successo anche grazie alla collaborazione della I.B.M. ed in particolare dello Scientific Data Center in Tokio con il quale realizzarono una delle immagini più famose di computer graphics la "Running Cola in Africa". Una svolta radicale nel mondo dei computer e della grafica si ebbe quando sul mercato iniziarono a comparire i primi circuiti integrati che consentirono una riduzione del costo dell'hardware con un rapporto di circa 3000:1!

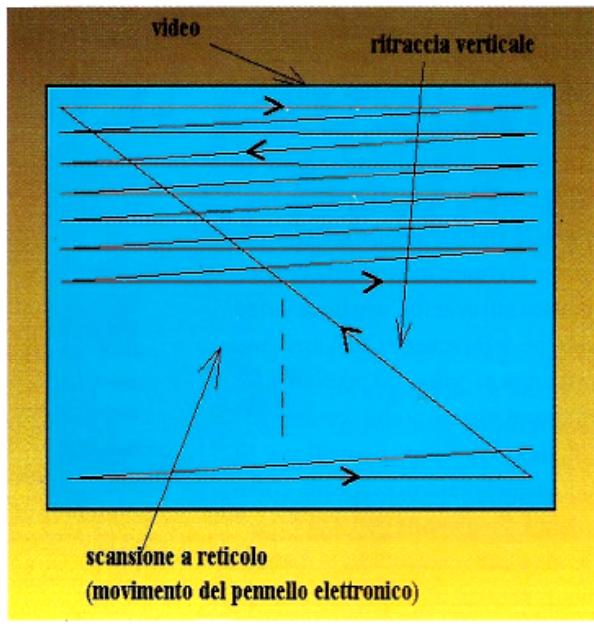
Con il passare del tempo iniziarono a comparire i primi personal computer che avevano delle capacità grafiche molto limitate. Nel 1981 fu presentato il primo PC dall'I.B.M., esso offriva due tipi di adattatori video (schede grafiche): l'MDA (Monochrome

Display Adapter) ossia l'adattatore video monocromatico e il CGA (Color Graphics Adapter) ossia l'adattatore colore/grafica. L'MDA era stato progettato per poter lavorare su un video monocromatico dell'I.B.M. ed in particolare per le modalità di testo alfanumeriche con 80x25 caratteri per pagina (80 colonne e 25 righe), mentre il CGA era stato progettato per poter lavorare in modalità grafica e in modalità alfanumerica attraverso l'uso di un video RGB. Le capacità grafiche della scheda CGA arrivano ad una risoluzione massima di 640x200 pixel mentre il numero di colori massimi visualizzabili contemporaneamente era di 4 solo con una risoluzione di 320x200 pixel. Quindi in quel periodo la scheda video CGA rappresentava il gioiello di punta dell'I.B.M. nel mondo dei PC. Molte altre case costruttrici si dedicarono alla costruzione di schede video, tra queste va ricordata la scheda monocromatica Hercules (HGC), presentata nel 1982, in grado di superare il limite imposto dalla MDA avendo la possibilità di visualizzare, sullo stesso monitor, sia testo che immagini grafiche ad una risoluzione di 720x348 pixel.

Successivamente anche questa scheda grafica fu migliorata e si arrivò nel 1986 ad avere la scheda Plus Hercules (HGC+) che consentiva di personalizzare il set di caratteri alfanumerici basati su RAM, a differenza delle schede MDA e HGC che si basavano su un set di caratteri residenti in ROM e quindi non modificabile. Nel frattempo I.B.M. lavorò sulla scheda CGA tentando di apportare dei miglioramenti che furono presentati nel 1985 con la scheda EGA (Enhanced Graphics Adapter = Adattatore Grafico Avanzato) la quale consentiva di lavorare in modo testo e grafico con una

Figura 1 -  
Schema generale  
del funzionamento  
di un sistema  
grafico interattivo\*





**Figura 3 - Linee di scansione del video**

integrato nei modelli 25 e 30 del PS/2, capace di lavorare con 256 colori in modalità grafica 320x200 grazie al convertitore digitale analogico DAC montato sulla scheda video. Nel 1987 l'I.B.M. realizzò la scheda VGA (Video Graphics Array=Matrice Video Grafica Integrata) che è diventata uno standard di fatto nel mondo dei PC. Questa scheda grafica ha delle prestazioni che superano quelle della MCGA in quanto essa è in grado di visualizzare 256 colori su una tavolozza di 262144 colori, con una risoluzione massima di 640x480 pixel. Attualmente le schede grafiche hanno raggiunto prestazioni notevolmente migliori rispetto allo standard VGA. Varie case costruttrici hanno migliorato la risoluzione grafica delle proprie schede fornendo valori elevati di 800x600 pixel (standard SuperVGA) oppure di 1024x768 pixel (VGA estesa). L'I.B.M. ha cercato di importare la risoluzione di 1024x768 pixel nelle sue schede; per prima è stata realizzata la scheda 8514/A e successivamente la scheda XGA. Esistono in commercio delle schede in grado di visualizzare contemporaneamente più di 256 colori, per esempio molte schede sono in grado di visualizzare contemporaneamente 16,7 milioni di colori (schede a 24 bit).

Queste schede grafiche avanzate sono utilizzate quando si vogliono realizzare delle immagini realistiche, per le quali l'adozione di un numero elevato di colori risulta indispensabile. Tutte le schede che abbiamo analizzato fino a questo momento sono delle periferiche prive di capacità di elaborazione locale su scheda. In questi ultimi anni le schede

risoluzione di 640x350 pixel con 16 colori.

Tramite questa scheda si riuscì ad ottenere un notevole salto di qualità nell'ambito delle stazioni grafiche su PC. Un altro passo in avanti fu fatto dall'I.B.M. con la scheda MCGA che rappresentava un sottosistema di visualizzazione

grafiche hanno subito un radicale cambiamento nel loro progetto. Dall'utilizzo di programmi grafici ad alte prestazioni si è notato che l'elaborazione di immagini ad alta risoluzione aveva dei tempi di esecuzione troppo elevati. Per minimizzare questi tempi di elaborazione si è pensato di inserire nella stessa scheda grafica un processore dedicato che effettuasse tutte le principali operazioni grafiche: rotazioni, traslazioni, zoom, ecc.

In questo modo la CPU e la scheda grafica possono lavorare in parallelo con una notevole diminuzione dei tempi di elaborazione. Sarebbe molto interessante riuscire a programmare questi tipi di schede video, ma dal momento che esse sono molto giovani una loro documentazione dettagliata non è ancora disponibile per cui l'utilizzo è riservato a pochi privilegiati. A questo punto possiamo iniziare con la descrizione della scheda grafica VGA.

### Cos'è UNA SCHEDA GRAFICA?

Fondamentalmente una scheda grafica non è altro che un componente hardware del computer in grado di convertire i segnali digitali provenienti dalla memoria video (tabella nella quale vengono registrati i pixel tramite una codifica a bit) in segnali utilizzabili dal video (figura 1). In particolare essa consente di pilotare il raggio luminoso dello schermo, di manipolare i segnali di sincronizzazione per le frequenze di tracciamento, ecc.

I tipi di video che possono essere collegati ad un personal computer sono: un video digitale (TTL), un video composito e un video analogico RGB. Il video digitale è un video con gli ingressi digitali (ossia dei segnali elettrici aventi tensioni 0 oppure 5 Volt, compatibili quindi con lo standard TTL). Tipicamente un video digitale ha da una a sei linee di colore in ingresso; per cui quando una di queste linee diventa attiva ( $On = 5 V$ ) il colore corrispondente viene visualizzato sullo schermo. Questi tipi di monitor hanno il grosso limite dei colori, infatti il numero massimo di colori disponibili è strettamente legato al numero di linee fisiche in ingresso al video. In particolare il numero massimo di colori visualizzabili su un video digitale è  $2^n$ , dove n indica il numero di linee dei colori. A causa del limite del numero di colori questi tipi di video non sono più utilizzati. Un esempio di scheda video compatibile con questi tipi di monitor è la scheda EGA che come sappiamo è stata superata

da tempo. Il secondo tipo di video fondamentale è quello composito. Questo monitor ha una sola linea di ingresso per cui i video composti sono tutti monocromatici e sono caratterizzati da una risoluzione grafica limitata.

Di solito essi vengono utilizzati come monitor nei centri di videoscrittura dove l'effetto rilassante del colore monocromatico facilita il lavoro dell'utente in fase di immissione dei testi. La CGA è una delle poche schede video compatibili con un video composito ed uno digitale (TTL). L'ultimo tipo di video che ci è rimasto da analizzare è quello analogico RGB. Questo video è quello più utilizzato attualmente poiché offre delle prestazioni molto elevate sia per il numero di colori visualizzabili sullo schermo e sia per la risoluzione grafica.

I video analogici RGB sono caratterizzati da tre linee di ingresso analogiche per il colore; queste linee corrispondono ai colori di base rosso, verde, blu (RGB= Red Green Blue). I colori vengono realizzati attraverso l'unione dei tre colori fondamentali RGB, per cui attribuendo un valore di tensione su ogni linea, almeno in teoria, si possono creare infiniti colori.

A questo punto è possibile comprendere l'enorme sviluppo dei video analogici RGB nel mercato dei personal computer. In pratica il limite fisico di colori visualizzabili sullo schermo dipende dalle capacità della scheda grafica e non dal monitor. Come è facile intuire le schede VGA sono compatibili con i tipi di video analogici RGB. Dopo aver analizzato le differenze tra i principali tipi di video affronteremo il problema di come avviene la visualizzazione reale delle informazioni sul video, in altri termini cercheremo di spiegare come faccia il monitor a visualizzare i punti sullo schermo.

## FUNZIONAMENTO DEL CRT

La parte fondamentale di un video è il tubo a raggi catodici (figura 2). Questo tubo non è altro che un involucro nel quale viaggia un fascio di elettroni che consente di accendere i relativi pixel sullo schermo.

I pixel vengono visualizzati sullo schermo quando il fascio di elettroni urta sulla superficie del video (che generalmente è realizzata con il fosforo).

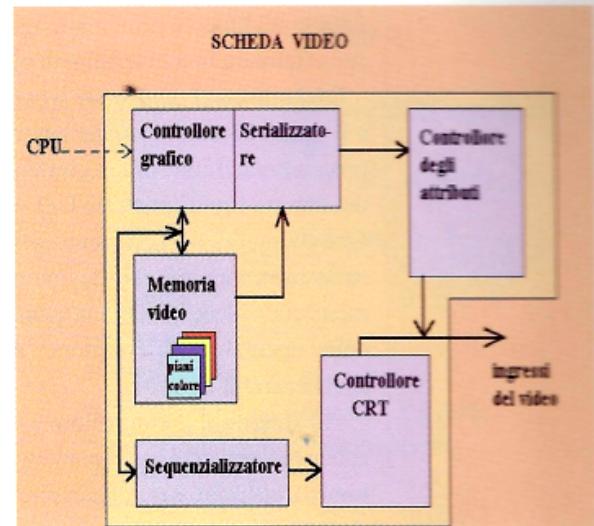
La luminosità della zona colpita dal raggio di elettroni decade rapidamente, per questo motivo è indispensabile ripetere in maniera rapida e frequente l'accensione dei pixel. In pratica per non provocare

lampeggiamenti delle immagini sul video (fenomeno flickering) è necessario ripetere le operazioni di tracciamento delle immagini sul video almeno 30 volte al secondo.

Semplificando al massimo, il fascio di elettronni (pennello elettronico) viene prodotto da un CATODO (cannone elettronico) e 'sparato' verso l'ANODO rappresentato dallo schermo, polarizzato tramite una forte tensione di EAT (Extra High Tension). La traiettoria del pennello elettronico viene regolata da bobine di deflessione, attraverso un campo elettromagnetico, controllato dai circuiti di deflessione orizzontale e verticale del monitor a colori. Tutto questo meccanismo permette di 'accendere' il singolo pixel di coordinate x, y del cinescopio (figura 3). Il pixel (un'area molto piccola di fosforo), colpito dal fascio di elettroni emette fotoni, quindi un effetto luminoso più o meno accentuato. Il fascio infatti inizia nella posizione in alto a sinistra dello schermo e si sposta attraverso il video, descrivendo una linea orizzontale. Arrivato alla fine dello schermo esso si riporta all'inizio della riga successiva, tramite un segnale di sincronizzazione orizzontale e così via. Quando il pennello elettronico arriva in fondo allo schermo, viene riportato nell'angolo in alto a sinistra per iniziare una nuova scansione. Tramite questo metodo, ad una frequenza elevata, è possibile ottenere delle immagini sullo schermo ad una risoluzione elevata.

## ARCHITETTURA GENERALE DI UNA SCHEDA VIDEO

Avevamo definito una scheda grafica come una unità hardware in grado di convertire le informazioni provenienti dai programmi in segnali video, utilizzabili dai monitor. A questo punto analiziamo l'architettura generale di una scheda video, evidenziandone le parti peculiari. Lo schema generale di una scheda grafica può essere rappresentato tramite la figura 4. In essa si



**Figura 4 - Architettura della scheda video**

notano sei blocchi principali, ognuno dei quali serve per effettuare una determinata operazione.

Iniziamo con l'analizzare il controllore grafico che è il primo blocco nel quale passano i dati inviati dal programma. Il controllore grafico è situato tra il flusso dei dati tra l'unità centrale CPU e la memoria video. Questo integrato VLSI viene utilizzato per manipolare le informazioni che devono essere registrate in memoria video, avendo la possibilità di effettuare delle operazioni di rotazione, di AND, di OR, di EXOR sui bit dei dati.

All'interno del controllore grafico sono presenti anche dei registri Latch (mantengono l'informazione logica), utilizzati per poter essere combinati con i dati che arrivano dal processore (figura 5). Utilizzando opportunamente questo integrato si possono realizzare degli effetti speciali notevoli con pochi comandi. La memoria video rappresenta un'altra unità fondamentale della scheda video.

In essa vengono registrati, sotto forma di bit, le informazioni visualizzate sullo schermo; in un certo senso possiamo considerare il contenuto della memoria video come una copia dei pixel sullo schermo. La memoria video viene suddivisa in piani diversi per poter ottenere una gestione più flessibile delle informazioni. Di solito sono implementati 4 piani (LAYER) che hanno delle dimensioni variabili a seconda della capacità delle schede (di solito ogni piano ha una dimensione di 64 Kb).

A seconda del modo di visualizzazione nel quale ci troviamo: testo oppure grafico, le informazioni presenti nei piani della memoria video sono codificate in un modo diverso. In modo grafico, variando la risoluzione dello schermo varia il tipo di codifica nei piani, questo dipende fondamentalmente dal numero di colori visualizzabili contemporaneamente sullo schermo. La maggiore difficoltà che s'incontra in fase di programmazione delle schede video consiste proprio nella gestione delle informazioni nella memoria video. Più avanti faremo degli esempi di codifica delle informazioni nella memoria video nel

modo testo e grafico per la VGA.

Un'altra unità fondamentale nell'architettura delle schede video è il serializzatore. Il serializzatore è situato fisicamente nello stesso circuito integrato del controllore grafico e svolge il compito di serializzare i dati provenienti dalla memoria video per inviarli al video CRT, dopo essere stati opportunamente decodificati. Questo componente funzionale della scheda è poco utilizzato dal programmatore che in pratica ha bisogno solo di sapere che i dati vengono serializzati a partire dal bit più significativo. Il flusso dei dati tra il controllore grafico e la memoria video viene gestito attraverso l'unità hardware chiamata sequenzializzatore.

Questa unità controlla le sequenze di tutte le parti funzionali della scheda video, in particolare genera il clock dei caratteri, gestisce la frequenza di refresh del video, effettua la temporizzazione dei cicli di lettura e scrittura in memoria video, ecc.

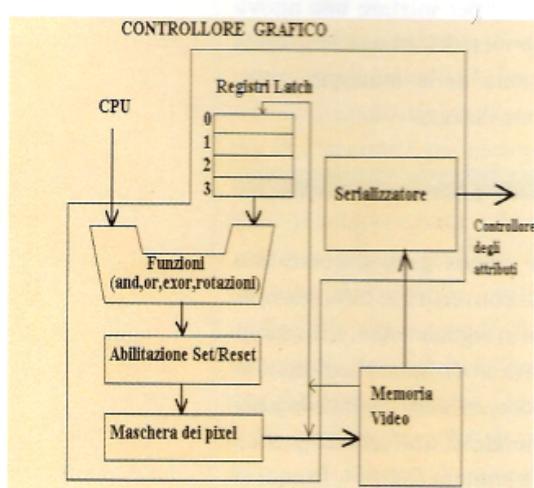
Dal punto di vista del programmatore la funzionalità più importante del sequenzializzatore è quella di controllare le operazioni in scrittura dei dati dal controllore grafico alla memoria video. In pratica all'interno di questa unità è presente un registro di abilitazione ai piani che consente di decidere su quale dei 4 piani disponibili in memoria video sia possibile scrivere (figura 6). L'utilizzo del registro di abilitazione ai piani risulterà molto comodo quando si dovranno effettuare delle operazioni di scrittura sulla memoria video e quindi per l'accensione di pixel sullo schermo. Un'altra unità fondamentale della scheda grafica è il controllore degli attributi.

Questa unità consente di gestire i colori visualizzabili sullo schermo. La parte fondamentale del controllore degli attributi è costituita da una tabella (Look-up Table) nella quale vengono memorizzati i colori da visualizzare per ogni numero selezionato. In pratica ogni volta che si vorrà modificare un colore si dovrà sostituire l'elemento corrispondente nella Look-up Table. Una caratteristica fondamentale di questa tabella è che una modifica ad un colore si ripercuterà immediatamente sullo schermo.

Per esempio, se sul video è visualizzata una figura con il colore rosso e il programmatore modifica il contenuto della tabella del controllore degli attributi sostituendo all'indirizzo del colore rosso il colore giallo, la figura sullo schermo cambierà immediatamente lo "stato" dal rosso al giallo.

Programmando opportunamente la Look-up Table si

**Figura 5 - Schema del controllore grafico**



possono realizzare degli effetti di animazione: modificando i colori con quelli dello sfondo, si potranno oscurare le figure e quindi provocare l'effetto di movimento. Nei prossimi numeri analizzeremo in maniera più approfondita questo argomento riferendoci al controllore degli attributi della VGA: il DAC video (Digital to Analog Converter). L'ultima unità che ci è rimasta da analizzare è il controllore CRT. Questa parte della scheda grafica consente di impostare le informazioni necessarie per poter visualizzare lo schermo. In particolare, questo controllore gestisce la temporizzazione del raster del CRT, definisce il formato dei dati visualizzati sullo schermo, la forma del cursore, la posizione del cursore, ecc.

Molti dei registri presenti nel controllore CRT sono di poco interesse per il programmatore, anche perché una modifica non prevista dal costruttore del valore dei registri più critici, come quello di fine visualizzazione orizzontale può provocare dei danni al video (naturalmente non esplosioni o cose del genere!). Dal punto di vista pratico l'utilizzo del controllore CRT può essere di enorme aiuto nella realizzazione di finestre sullo schermo o di effetti di visualizzazione particolari.

Dopo aver analizzato sinteticamente le varie parti funzionali di una generica scheda video, possiamo esaminare i modi con i quali vengono codificate le informazioni nella memoria video.

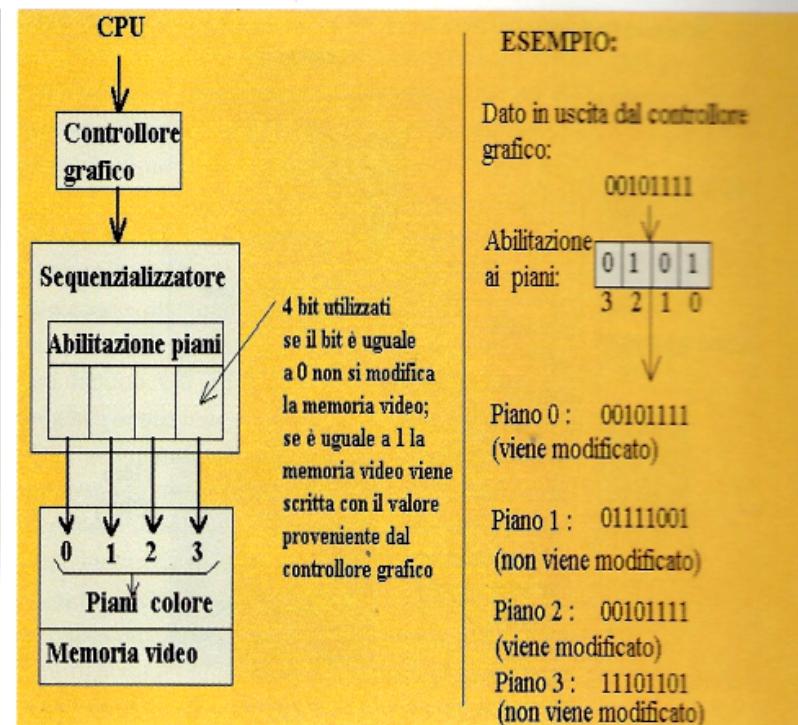
### MEMORIA VIDEO VIRTUALE E REALE

Fino a questo punto abbiamo parlato di memoria video e abbiamo detto che essa è situata fisicamente a bordo della scheda grafica. Nella realtà la memoria video fisica (nella scheda) non viene utilizzata realmente dal programmatore.

Per poter scrivere nella memoria video della scheda il programmatore utilizzerà delle aree riservate della memoria principale (RAM). L'insieme di queste aree riservate nella memoria principale viene chiamato memoria video virtuale (figura 7).

In pratica ogni qual volta il programmatore vorrà modificare il contenuto della memoria video reale (cioè quella presente nella scheda video) dovrà effettuare delle scritture in memoria principale, cioè nella memoria video virtuale.

Molti di voi si staranno chiedendo come mai si devono complicare le cose effettuando tutte queste operazioni per poter accendere dei pixel o visualizzare



dei caratteri sullo schermo! In realtà in questo modo le cose, dal punto di vista pratico, si semplificano notevolmente, dal momento che basterà effettuare una scrittura nella memoria video virtuale (quindi nella memoria principale) per poter visualizzare istantaneamente i risultati sul video.

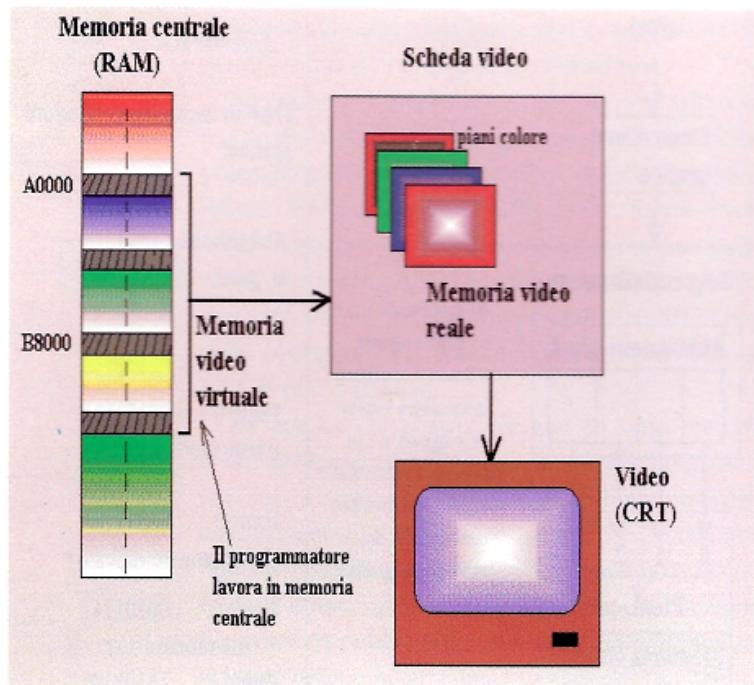
E' intuitibile a questo punto che le zone di memoria principale riservate per le schede grafiche (memoria video virtuale) sono gestite da una circuiteria dedicata che consente di trasferire le informazioni direttamente nelle memorie della scheda grafica. Da ora in poi quando parleremo di memoria video intenderemo sempre quella virtuale, dal momento che è quella più utilizzata in fase di programmazione.

### MEMORIA VIDEO IN MODO TESTO

La memoria video in modo testo viene utilizzata come mezzo per poter memorizzare i caratteri con i relativi colori presenti di volta in volta sullo schermo. L'indirizzo base per le modalità di visualizzazione testo a colori con 80x25 caratteri per pagina è posizionato nel segmento di memoria B800 con uno spiazzamento pari a 0 (in pratica inizia nell'indirizzo B800:0000, in esadecimale).

Ogni carattere viene memorizzato nella memoria video attraverso il suo codice ASCII, quindi ogni carattere richiede 1 byte. In realtà ogni carattere

Figura 6 - Schema del sequenzializzatore



**Figura 7 - Gestione della memoria video**

presente sullo schermo viene archiviato in memoria video attraverso 2 byte, uno per il codice ASCII e uno per l'attributo. L'attributo di un carattere nei modi a colore non è altro che una informazione contenente il colore dello sfondo e quello di primo piano.

Ogni singolo attributo viene generato attraverso un byte formato secondo la figura 8. In questa figura è possibile notare che i colori vengono generati attraverso l'unione dei tre colori fondamentali RGB. In totale è possibile ottenere 16 colori diversi per il primo piano e 8 colori diversi per lo sfondo. I 16 colori di primo piano vengono generati oltre che dai tre bit RGB da un bit di intensità che in pratica aumenta la luminosità del colore. Il bit più significativo del byte degli attributi è utilizzato per attivare la modalità di lampeggiamento del carattere (un uno in questa posizione attiverà il lampeggiamento del carattere).

Queste informazioni che abbiamo analizzato per generare un carattere (codice ASCII + attributo) vengono memorizzate in sequenza nella memoria video, a partire dall'indirizzo B800:0000. In pratica nelle posizioni pari saranno presenti i codici ASCII dei caratteri visualizzati sullo schermo e nelle posizioni dispari gli attributi.

Il primo indirizzo nella memoria video (avente 0 per offset) corrisponde all'angolo in alto a sinistra dello schermo. Le posizioni dello schermo quindi vengono gestite in memoria video in maniera sequenziale, ovvero una pagina di testo nella memoria video

corrisponde ad un vettore di 4000 byte. Lo spiazzamento in memoria video, corrispondente ad una determinata posizione dello schermo, può essere ricavato dalla seguente relazione, se si considera l'angolo in alto a sinistra di coordinate 0,0:

$$\text{offset} = 160 * Y + 2 * X$$

L'utilizzo della memoria video testo si può rivelare utile in tutte quelle applicazioni che richiedono una velocità di esecuzione elevata in fase di stampa sul video. Utilizzando il linguaggio di programmazione Turbo Pascal si può sfruttare il vettore predefinito MEM che consente di accedere ad una locazione di memoria specificando il segmento e lo spiazzamento corrispondente. A livello hardware le informazioni presenti nella memoria video reale (nella scheda) vengono gestite per piani. Tutti i codici ASCII dei caratteri presenti sullo schermo vengono memorizzati nel piano 0 della memoria video reale, gli attributi nel piano 1 e la tabella di generazione dei caratteri nel piano 2 (figura 9). La tabella di generazione dei caratteri, invece di essere memorizzata su ROM viene caricata sul piano 2 della memoria video reale. Questa gestione risulta notevolmente flessibile, dal momento che sono disponibili 8 mappe di caratteri per la VGA, con un massimo di 256 caratteri per mappa. Possono essere attivi uno o due mappe di caratteri, per un totale di 512 caratteri visualizzabili sullo schermo. Se sono attivi due insiemi di caratteri, il bit 3 del byte degli attributi indica da quale mappa prelevare la codifica del carattere. Un registro del sequenzializzatore è utilizzato per selezionare l'insieme attivo dei caratteri.

Ogni carattere ha un'ampiezza fissa di 8 pixel (è possibile anche ottenere una larghezza di 9 pixel, per i modi non standard), mentre la lunghezza varia da 1 a 32 pixel, selezionabile mediante un registro di output. I caratteri presenti nella tabella del generatore di caratteri vengono codificati tramite una bit mapped. In pratica ogni byte rappresenta (nel modo standard) una riga di pixel che forma il carattere.

Quando un bit è attivo lo sarà anche il corrispettivo pixel del carattere. In questo modo la creazione di caratteri personalizzati può essere effettuata modificando i valori presenti nella tabella dei caratteri in maniera molto semplice ed efficace. Per maggiori informazioni sulla memoria video testo può essere utile la lettura di un mio articolo apparso sul numero 3 della nostra rivista ("Esplorando la memoria video").

## MEMORIA VIDEO IN MODO GRAFICO

La memoria video a livello grafico ha un indirizzo di memoria diverso da quello per le modalità di testo. Infatti essa ha inizio all'indirizzo A000:0000 nel quale sono codificate le informazioni relative ai pixel dello schermo. La codifica di queste informazioni dipende dal numero di colori che si utilizzano per la modalità di visualizzazione. Fondamentalmente il numero totale di bit necessari per memorizzare tutti i colori di un pixel è dato dal logaritmo in base 2 del numero di colori. Ad esempio, nel caso di una modalità di visualizzazione monocromatica sarà necessario 1 bit per ogni pixel, dal momento che si hanno a disposizione solo 2 colori (0 = pixel spento, 1 = pixel acceso). Volendo utilizzare una modalità di visualizzazione a 16 colori, per esempio il modo 640x480 pixel della VGA standard, saranno necessari 4 bit per codificare ogni singolo pixel. Nella modalità a 256 colori ogni pixel corrisponde ad 8 bit ossia ad un byte. Per ogni modalità con diverso numero di colori la memoria video sarà organizzata in un modo diverso. Esistono fondamentalmente due tecniche per poter scrivere in memoria video. Queste due tecniche sono: la codifica per pacchetti di pixel e quella per piani di colore (figura 10). La VGA utilizza soprattutto quella a piani di colore, tanto che la memoria video viene organizzata in piani diversi, comunque è possibile anche utilizzare una codifica per pacchetti.

Con la codifica per piani ogni singolo pixel è dato dalla somma dei bit di ogni porzione di piano, mentre in quella a pacchetti il colore del pixel viene memorizzato tramite dei bit in sequenza memorizzati nel piano 0 della memoria video. La decisione del tipo di scrittura da usare in memoria video deve essere effettuata attraverso l'utilizzo di un registro del modo presente nel controllore grafico.

A questo punto per poter fornire degli esempi pratici di come sia possibile scrivere in memoria video, possiamo analizzare i registri presenti nella scheda VGA.

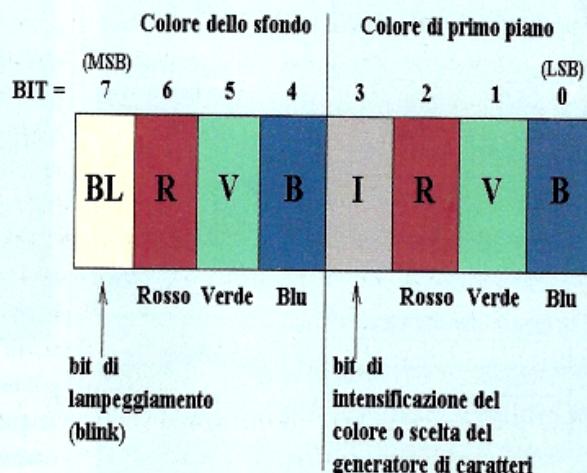
## PROGRAMMARE LA VGA

Come vedremo di seguito, la programmazione della scheda VGA è molto laboriosa, in quanto per poter sfruttare al mas-

simo tutte le possibilità offerte da questa periferica è necessario conoscere in maniera approfondita il funzionamento di tutti i registri interni. Con un po' di pazienza ed una buona organizzazione, alla fine riuscirete facilmente a creare delle vostre funzioni grafiche ad alto livello con un tempo di esecuzione molto basso. La programmazione della scheda VGA, come del resto di qualsiasi altra periferica, viene effettuata utilizzando due istruzioni macchina In e Out che consentono di scrivere e leggere in determinati indirizzi di porta. Gli indirizzi di porta non sono altro che i registri interni delle varie periferiche contenenti informazioni generali sullo stato della unità hardware. Per poter programmare una periferica è indispensabile conoscere tutti i registri interni della scheda con i relativi indirizzi di porta. Nel caso della scheda VGA ci sono numerosi registri, (più di 60) ognuno dei quali è predisposto per una particolare operazione. I registri più utili per il programmatore sono in numero minore. Questi registri vengono divisi logicamente in gruppi a seconda della unità funzionale di appartenenza. Per questo motivo ci saranno i registri del controllore grafico, del controllore degli attributi, ecc. Esistono anche dei registri chiamati esterni poiché non fanno parte delle unità funzionali della scheda (tabella 1). La modifica casuale del contenuto dei registri della scheda può causare dei problemi in fase di visualizzazione dei dati sul video. Nel caso si verifichino tali errori sarà sufficiente resettare il proprio personal computer, in modo da risistemare i valori dei registri in fase di caricamento del sistema operativo. Nella tabella 2

## BYTE DEGLI ATTRIBUTI

Figura 8 -  
Byte degli  
attributi



sono riportati tutti i registri della scheda VGA divisi a seconda della appartenenza ai diversi blocchi funzionali della periferica. Per poter accedere ai registri riportati in tabella 2 è necessario utilizzare gli indirizzi di porta in due fasi distinte. Com'è possibile notare, ogni gruppo di registri appartenenti ad uno stesso blocco funzionale è caratterizzato da due soli indirizzi di porta. Questi due indirizzi di porta vengono utilizzati per selezionare il registro della scheda nel quale si vuole scrivere o leggere e per effettuare l'operazione di lettura o scrittura desiderata. Per poter selezionare il registro su cui lavorare è necessario specificare nel registro indice il numero corrispondente a quel determinato registro. Per poter scrivere o leggere nel registro selezionato è necessario utilizzare il registro dei dati. Per esempio, per poter effettuare un'operazione di scrittura sul registro del modo appartenente al gruppo del controllore grafico dovranno essere utilizzate le seguenti istruzioni Assembly:

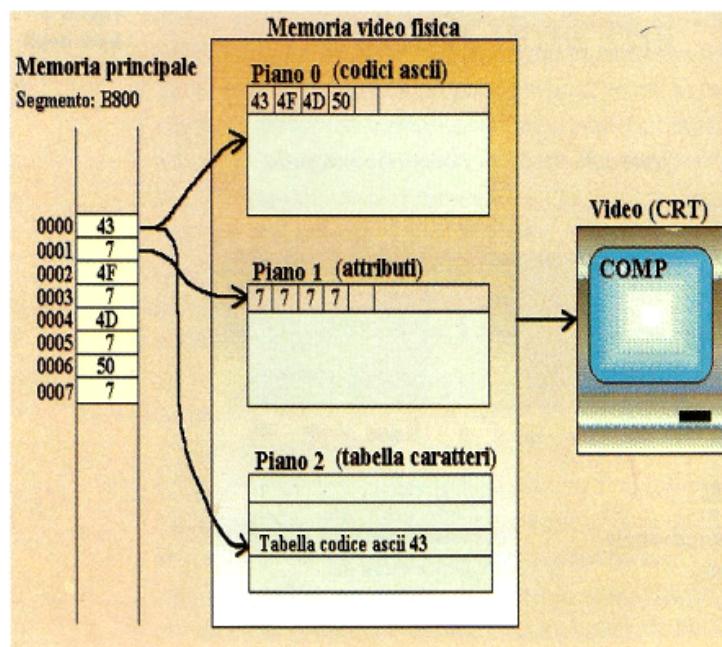
```

MOV DX, 3CEh ; reg. indice
MOV AL, 5      ; reg. modo
OUT DX, AL     ; invia il dato
MOV DX, 3CFh ; reg. dato
MOV AL, 0      ; modo scrittura
OUT DX, AL     ; invia il dato
...

```

**Figura 9 - Memoria video in modo testo**

Arrivati a questo punto possiamo analizzare in maniera più approfondita un esempio pratico di programmazione della scheda grafica VGA.



Uno dei problemi principali che si incontra nella programmazione della scheda VGA è quello dell'accensione di un pixel sullo schermo. Esistono, come abbiamo già visto, due metodi fondamentali per scrivere nella memoria video, quello per piani di colore e quello per pacchetti.

Dal punto di vista pratico la scelta di uno di questi due metodi è strettamente vincolata dal tipo di operazione che si vuole implementare.

Nel nostro esempio analizzeremo un metodo di validità generale per poter effettuare l'accensione di pixel utilizzando il metodo di scrittura per piani. In figura 11 è riportato il listato in linguaggio Assembly (utilizzando come supporto il Turbo Assembler) di un programma dimostrativo che consente di scrivere nella memoria video per piani di colore.

Le prime tre istruzioni macchina vengono utilizzate per poter attivare la modalità grafica VGA 640x480 pixel a 16 colori (l'analisi dettagliata di questa funzione verrà effettuata nel seguito dell'articolo). Dopo queste prime 3 istruzioni seguono quelle principali che consentono di dialogare direttamente con la scheda VGA. Per poter utilizzare il metodo di scrittura in memoria video per piani è necessario programmare opportunamente la scheda, per questo motivo è necessario utilizzare il registro della modalità di scrittura presente nel controllore grafico.

Il registro del modo di scrittura è organizzato nel modo seguente:

BIT	COMMENTO
7	riservato (0)
6	modo VGA 256 colori
5	modo a traslazione
4	modo pari/dispari
3	confronto fra colori (1 abilitato)
2	riservato
1	modo di scrittura
0	" "

I bit interessanti nel nostro esempio sono quelli meno significativi, ossia il bit 0 e il bit 1. A seconda del valore contenuto in questi bit la scheda effettuerà una modalità di scrittura differente in memoria video. I

valori che possono essere presenti in questi due bit sono:

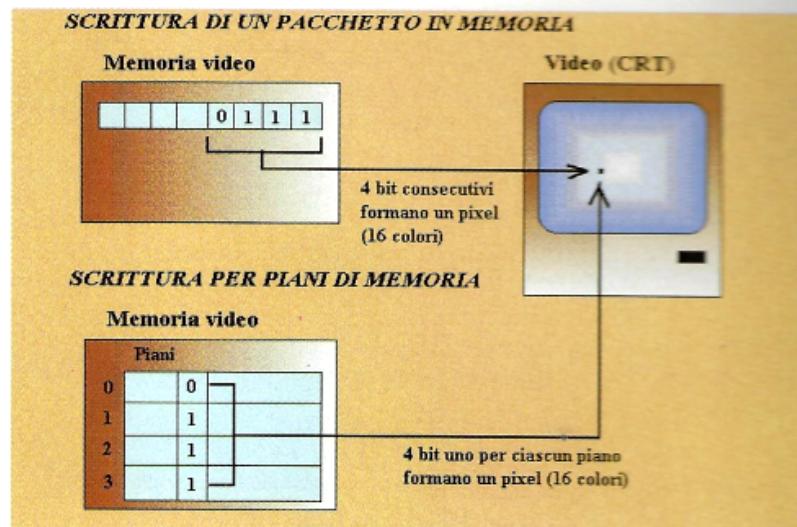
**00 modo scrittura 0**

**01 modo scrittura 1**

**10 modo scrittura 2**

**11 non utilizzato**

Il modo di scrittura 0 corrisponde al metodo diretto, ossia quello di scrittura per piani; il modo 1 corrisponde al metodo di scrittura dai latch, ossia in memoria video vengono scritti i valori presenti nei latch e il dato del processore viene ignorato ed infine il metodo 2 è quello della scrittura a pacchetti. Nel nostro esempio il valore del registro del modo è stato impostato a 0, in modo da utilizzare i piani di colore. Successivamente nel programma per poter effettuare l'operazione di scrittura in memoria video è stato necessario predisporre il valore di alcuni registri del controllore grafico e del sequenzializzatore in modo da filtrare opportunamente il dato proveniente dal processore. Per poter effettuare la scrittura in memoria video è stato necessario utilizzare il registro di maschera dei bit, presente nel controllore grafico, ed il registro della maschera di abilitazione, presente nel sequenzializzatore. Questi due registri servono per poter decidere quali bit dovranno essere modificati in memoria video e in quali piani di colore. Dal momento che il programma che stiamo analizzando lavora con la risoluzione 640x480 in 16 colori, ogni pixel verrà memorizzato con 4 bit. Nel nostro caso la costruzione del pixel avviene tramite l'unione di 4 bit presenti ognuno in un piano di memoria video diverso. Tramite il registro di maschera dei bit abbiamo potuto stabilire quali dei bit del dato da inviare nella memoria video dovevano essere modificati. In pratica con il valore 11111111 inviato a questo registro, ogni bit di ogni piano di colore veniva modificato. Successivamente, per poter azzerare i valori dei bit presenti nei piani di memoria, nel programma si è effettuata una scrittura del valore 0 nella memoria video. La scrittura del valore 0 è molto importante, dal momento che lavorando sui piani di memoria si deve presupporre che essi siano tutti inizializzati ad un valore prefissato. Dopo la scrittura in memoria video il programma continua con l'impostazione del registro di maschera



di abilitazione in scrittura dei piani di colore. Questo registro viene utilizzato per poter scegliere quale dei piani di memoria dovranno essere modificati. Avendo azzerato preventivamente tutti i bit dei piani di memoria l'impostazione del valore da attribuire al registro della maschera di abilitazione in scrittura equivale alla scelta dei colori dei pixel. Infatti nel proseguo del programma, per poter effettivamente accendere i pixel sullo schermo, è sufficiente effettuare una scrittura in memoria specificando quali pixel dovranno essere

**Figura 10 - Scrittura per piani e per pacchetti**

## TABELLA 1

### ELENCO REGISTRI ESTERNI SCHEDA VGA STANDARD

Indirizzo I/O	Registro
3C2	Registro di output generale
	Registro di stato input 0
3DA	Registro per il controllo ausiliario
	Registro di stato input 1
3DB	Azzeramento del latch della penna luminosa
3DC	Imposta il latch della penna luminosa ad 1
3C4, 3C5	Sequenzializzatore
3D4, 3D5	Controllore CRT
3CA, 3CC	Controllore grafico
3C0	Controllore degli attributi
3C3	Abilitazione della scheda VGA
3C6, 3C7	Sistema DAC VGA

**Figura 11 - Programma per l'accensione di 8 pixel**

```

dosseg
.model tiny
.stack
.data
.code

mov ah, 0
mov al, 12h      ; attiva modalità grafica 640x480 16 colori VGA
int 10h

mov dx, 3ceh      ; registro indice del controllore grafico
mov al, 5          ; indice del registro del modo
out dx, al         ; invia il dato
mov dx, 3cfh      ; registro dati del controllore grafico
mov al, 0          ; modalità di scrittura 0
out dx, al         ; invia il dato

mov ax, 0A000h    ; segmento della memoria video
mov es, ax         ; carica il segmento nel registro ES
mov bx, 0          ; spiazzamento in memoria video

mov dx, 3ceh      ; registro indice del controllore grafico
mov al, 8          ; indice del registro maschera sui bit
out dx, al         ; invia il dato
mov dx, 3cfh      ; registro dati del controllore grafico
mov al, 11111111b ; attiva tutti i pixel per la scrittura
out dx, al         ; invia il dato

mov al, 0
mov es:[bx], al

mov dx, 3c4h      ; registro indice del sequenzializzatore
mov al, 2          ; indice del registro maschera di abilitazione
out dx, al         ; invia il dato
mov dx, 3c5h      ; registro dati del sequenzializzatore
mov al, 4          ; numero del colore
out dx, al         ; invia il dato

mov al, 11111111b ; accende tutti i pixel (8 pixel)
mov es:[bx], al   ; scrittura in memoria video

mov ah, 4ch
int 21h
end

```

attivati con il colore specificato tramite il registro di maschera di abilitazione.

Il programma, in definitiva, esegue un tracciamento di 8 pixel di colore rosso nella porzione di schermo in alto a sinistra nella modalità di visualizzazione grafica VGA standard 640x480 a 16 colori.

**IMPOSTAZIONE DEL MODO DI VISUALIZZAZIONE**

Per poter attivare una modalità di visualizzazione grafica o di testo particolare, può essere utile conoscere il funzionamento dell'interrupt 10h del BIOS. Tale interrupt consente di attivare le modalità di visualizzazione standard (tabella 3) e quelle non standard che dipendono, naturalmente, dal tipo di scheda grafica che si ha a disposizione.

Per utilizzare correttamente questo interrupt è necessario assegnare al registro AH il valore 0 e al registro AL il numero del modo di visualizzazione scelta tra i valori presenti nelle tabelle 3 o 4. Per esempio, in Assembler, per poter visualizzare la modalità di visualizzazione grafica VGA 640x480 a 16 colori sono necessarie le seguenti istruzioni:

```

MOV AH, 0
MOV AL, 12h
INT 10h

```

L'utilizzo di tale interrupt può essere molto utile in tutti quei linguaggi che non hanno delle funzioni di libreria per gestire modalità grafiche particolari.

**CONCLUSIONI**

La programmazione delle schede grafiche è un argomento molto lungo da trattare, per questo non è stato possibile spiegare in maniera dettagliata ogni singolo componente di questa interessante periferica. Per questo motivo nei numeri successivi della rivista cercheremo di approfondire di volta in volta gli argomenti che sono stati trascurati in queste pagine. Uno di questi è l'utilizzo delle modalità estese della VGA. Ma nella selva di standard e specifiche fra loro incompatibili il lavoro del programmatore diventa problematico. Una risposta -se pur parziale- a questo problema viene dallo standard VESA BIOS EXENTIONS 1.2 trattato nel prossimo articolo. Per quanto ci riguarda, l'appuntamento è fissato per i prossimi mesi quando torneremo sull'argomento.

## TABELLA 2

## ELENCO REGISTRI INTERNI DELLA SCHEDA VGA STANDARD

## - CONTROLLORE CRT

Indice (3D4) Dato (3D5)

0	Totale orizzontale
1	Limite visualizzazione orizzontale
2	Inizio blank orizzontale
3	Fine blank orizzontale
4	Inizio ritraccia orizzontale
5	Fine ritraccia orizzontale
6	Totale verticale
7	Overflow
8	Preset di scansione di riga
9	Max. linee di scansione
A	Inizio cursore
B	Fine cursore
C	Indirizzo di inizio (MSB)
D	Indirizzo di inizio (LSB)
E	Posizione del cursore (MSB)
F	Posizione del cursore (LSB)
10	Inizio ritraccia verticale
11	Fine ritraccia verticale
12	Fine attivazione visualizzazione verticale
13	Offset (larghezza linea logica)
14	Posizione di sottolineatura
15	Inizio blank verticale
16	Fine blank verticale
17	Controllo del modo
18	Confronto tra linee

## - SEQUENZIALIZZATORE

Indice (3C4) Dato (3C5)

0	Registro di reset
1	Modo del clock
2	Abilitazione per la scrittura sui piani di colore
3	Selezione del generatore di caratteri
4	Modo di memoria

## - CONTROLLORE GRAFICO

Indice (3CE) Dato (3CF)

0	Set/reset
1	Abilitazione set/reset
2	Confronto fra colori
3	Selezione di funzione e rotazioni di dati

4	Selezione del piano in lettura
5	Modo di scrittura
6	Funzioni varie
7	Disabilitazione del colore
8	Maschera sui bit

## - CONTROLLORE DEGLI ATTRIBUTI

Indice (3C0) Dato (3C0)

0	Registro 0 della tavolozza dei colori
1	" 1 "
2	" 2 "
3	" 3 "
4	" 4 "
5	" 5 "
6	" 6 "
7	" 7 "
8	" 8 "
9	" 9 "
A	" 10 "
B	" 11 "
C	" 12 "
D	" 13 "
E	" 14 "
F	" 15 "
10	Registro di controllo del modo
11	Colore del bordo dello schermo
12	Abilità i piani di colore
13	Panning orizzontale
14	Registro di selezione del colore

## TABELLA 3

## MODALITA' DI VISUALIZZAZIONE STANDARD

Num.	Modo	Adattatore	Risoluzione	Colori
00	Testo	CGA	320x200	16
		EGA	320x350	16
		MCGA	320x400	16
		VGA	360x400	16
01	Testo	CGA	320x200	16
		EGA	320x350	16
		MCGA	320x400	16
		VGA	360x400	16
02	Testo	CGA	640x200	16
		EGA	640x350	16
		MCGA	640x400	16
		VGA	720x400	16
03	Testo	CGA	640x200	16
		EGA	640x350	16
		MCGA	640x400	16

		VGA	720x400	16
04	Grafica	CGA/EGA/MCGA/VGA	320x200	4
05	Grafica	CGA/EGA/MCGA/VGA	320x200	4
06	Grafica	CGA/EGA/MCGA/VGA	640x200	2
07	Testo	MDA/EGA	720x350	Mono
		VGA	720x400	Mono
08	Grafica	PCjr	160x200	16
09	Grafica	PCjr	320x200	16
0A	Grafica	PCjr	640x200	4
0B	Riservato			
0C	Riservato			
0D	Grafica	EGA/VGA	320x200	16
0E	Grafica	EGA/VGA	640x200	16
0F	Grafica	EGA/VGA	640x350	Mono
10	Grafica	EGA/VGA	640x350	16
11	Grafica	MCGA/VGA	640x480	2
12	Grafica	VGA	640x480	16
13	Grafica	MCGA/VGA	320x200	256

TABELLA 4

#### MODALITA' DI VISUALIZZAZIONE NON STANDARD

Num.	Modo	Adattatore	Risoluzione	Colori
14/20		Sono utilizzati dai modi grafici EGA e VGA		
18	Testo	Tseng Labs EVA	132x44	Mono
19	Testo	Tseng Labs EVA	132x25	Mono
1A	Testo	Tseng Labs EVA	132x28	Mono
21	Grafica	Pagina grafica 1 della scheda Hercules		
22	Grafica	Pagina grafica 2 della scheda Hercules		
22		Tseng, Ahead	132x44	
23		Tseng Labs EVA	132x25	
		Ahead Systems EGA2001	132x25	
		ATI EGA Wonder	132x25	
24	Grafica	Tseng Labs EVA	132x28	
25	Grafica	Tseng Labs EVA	640x480	
		VEGA VGA	640x480	16
26		Tseng Labs EVA	80x60	
		Ahead Systems EGA2001	640x480	
27	Grafica	VEGA VGA	720x512	16
29	Grafica	VEGA VGA	800x600	16
2D	Grafica	VEGA VGA	640x350	256
2E	Grafica	VEGA VGA	640x480	256
2F	Grafica	VEGA VGA	720x512	256
30	Grafica	VEGA VGA	800x600	256
36	Grafica	VEGA VGA	960x720	16
37	Grafica	VEGA VGA	1024x768	16
		ATI EGA Wonder	132x44	Mono
40	Grafica	AT&T 6300	640x400	2
		VEGA VGA	80x43	
41	Grafica	AT&T 6300	640x200	16
		VEGA VGA	132x25	

42	Grafica	AT&T 6300	640x400	16
		VEGA VGA	132x43	
43		VEGA VGA	80x60	
44		AT&T 6300	disabilita le uscite VDC e DEB	
		VEGA VGA	100x60	
48	Grafica	AT&T 6300	640x400	
4D		VEGA VGA	120x25	
4E		VEGA VGA	120x43	
Testo	OAK VGA	80x60	16	
4F		VEGA VGA	132x25	
Testo	OAK VGA	132x60	16	
50	Testo	Ahead Systems EGA2001	132x25	Mono
	Testo	OAK VGA	132x25	16
	Grafica	Paradise EGA-480	640x480	16
	Testo	VEGA VGA	80x43	Mono
51	Testo	Paradise EGA-480	80x30	
	Testo	VEGA VGA	132x25	Mono
	Testo	OAK VGA	132x43	16
52		Ahead Systems EGA2001	132x44	Mono
		VEGA VGA	132x43	Mono
	Grafica	OAK VGA	800x600	16
53	Grafica	OAK VGA	640x480	256
54	Grafica	OAK VGA	800x600	256
		Paradise EGA-480	132x43	
		Paradise VGA	132x43	16
		Paradise VGA on multisync	132x43	16
		Taxan 565 EGA	132x43	
55	Grafica	OAK VGA	1024x768	4
		Paradise EGA-480	132x25	
		Paradise VGA	132x25	16
		Paradise VGA on multisync	132x25	16
		Taxan 565 EGA	132x25	
56	Grafica	OAK VGA	1024x768	16
		NSI Smart EGA+	132x43	
		Paradise VGA	132x43	
		Paradise VGA on multisync	132x43	
		Taxan 565 EGA	132x43	Mono
57		NSI Smart EGA+	132x25	
		Paradise VGA	132x25	
		Paradise VGA on multisync	132x25	
		Taxan 565 EGA	132x25	
		NSI Smart EGA+	132x25	
		Paradise VGA	132x25	
		Paradise VGA on multisync	132x25	
		Taxan 565 EGA	132x25	
58	Grafica	OAK VGA	1280x1024	16
		Paradise VGA	800x600	
59	Grafica	Paradise VGA	800x600	
		OAK VGA	1024x768	256
5E	Grafica	Paradise VGA/ VEGA VGA	640x400	
5F	Grafica	Paradise VGA	640x480	
60	Grafica	VEGA VGA	752x410	
61	Grafica	VEGA VGA	720x540	
62	Grafica	VEGA VGA	800x600	
71	Grafica	NSI Smart EGA+	800x600	16
74	Grafica	Toshiba 3100	640x400	
82	Testo	AT&T VDC overlay mode	80x25	Mono
83	Testo	AT&T VDC overlay mode	80x25	
86	Grafica	AT&T VDC overlay mode	640x200	Mono
D0	Grafica	DEC VAXmate AT&T mode	640x400	