

contrôleur de tube cathodique

David Ashton (Australie)

Dans les années 80, afficher du texte sur des écrans était une tâche herculéenne, loin de la facilité du *plug-and-play* d'aujourd'hui. Pensons à ceux qui manipulaient les contrôleurs de tubes cathodiques (CRTC), ces héros méconnus du début de l'informatique.

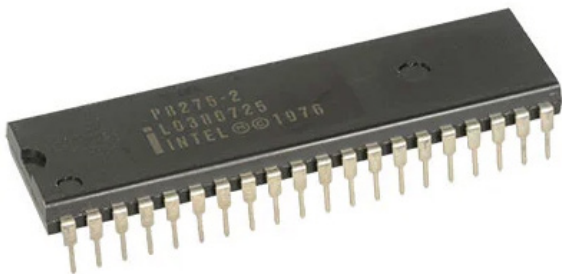


Figure 1. Intel 8275 CRTC. (Source : Jameco Electronics)

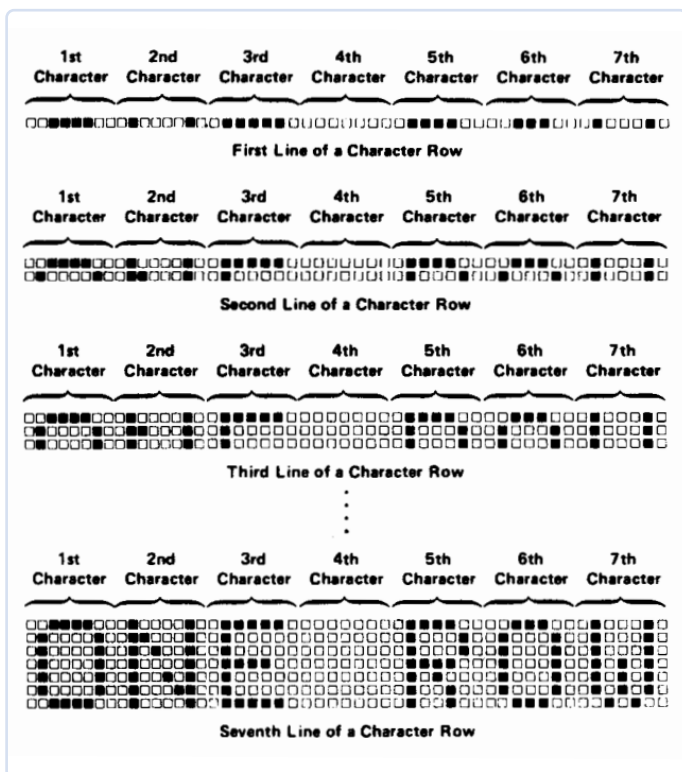


Figure 2. Constitution d'un texte à partir de lignes de pixels - avec une matrice de 5x7 caractères. (Source : Intel)

De nos jours, afficher du texte ou même des graphiques ou des films avec votre microcontrôleur est... eh bien, peut-être pas un jeu d'enfant, mais reste assez facile. Vous pouvez utiliser l'un des afficheurs LCD répandus, ou si votre microcontrôleur est un peu plus haut de gamme comme un Raspberry Pi, il suffit de brancher un moniteur HDMI. Il existe des bibliothèques de logiciels qui permettent d'afficher assez facilement le contenu souhaité.

Pensez donc aux concepteurs de microprocesseurs des années 80. Les afficheurs LCD étaient encore rudimentaires, adaptés principalement pour les calculatrices. On était restreint à l'utilisation de LED ou de quelques afficheurs à sept segments, multiplexés afin d'économiser les précieuses lignes d'E/S. Les moniteurs CRT étaient disponibles, mais comment générer tous ces signaux vidéo avec leurs timing et niveaux précis ?

C'était la tâche du CRTC, ou contrôleur de tube cathodique. Il s'agissait de puces périphériques, souvent dotées de 40 broches (**figure 1**), proposées par les principaux fabricants de microprocesseurs, pour faciliter l'affichage de texte sur les moniteurs CRT. Ces derniers nécessitent que les données des pixels soient écrites à l'écran, ligne par ligne, utilisant généralement jusqu'à 600 lignes. Pour un signal TV, la ligne présente des niveaux de luminosité variables, mais pour l'affichage de texte, les niveaux sont soit hauts (pour afficher un point), soit éteints (pas de point). De cette manière, il est possible d'afficher des motifs de points pour composer des caractères et des symboles lisibles (voir **figure 2**).

Ces puces étaient utilisées avec un générateur de caractères (CG) ROM. Ceux-ci avaient généralement sept lignes d'adresse de caractères, qui étaient pilotées par le code ASCII du caractère à générer, et, pour chaque code ASCII, ils émettaient le modèle de pixel pour

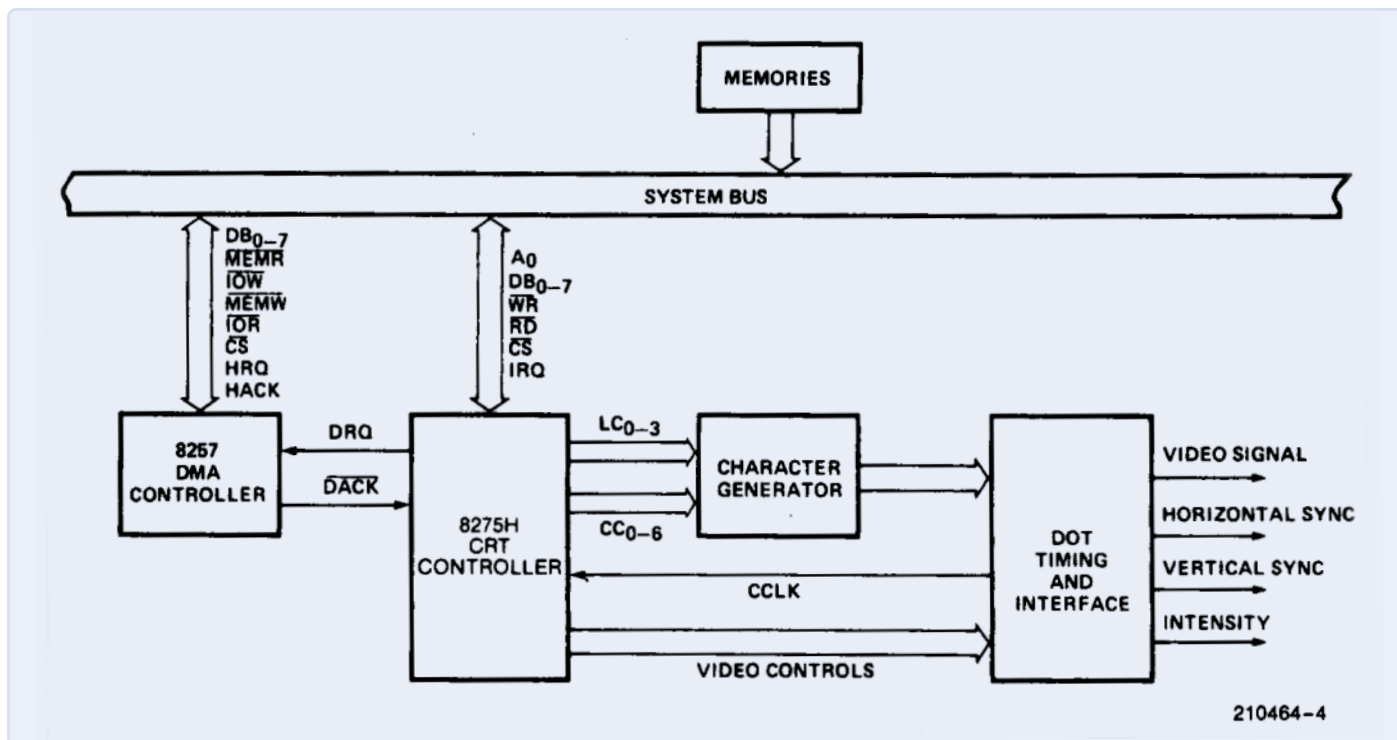


Figure 3. Schéma fonctionnel d'un CRTC. (Source : Intel)

ce caractère, stocké sous forme de lignes et de colonnes. Les formats de caractères courants étaient de 5×7 points de haut, et de 7×9 pour une meilleure résolution. Le CRTC sélectionnait la ligne d'un caractère en utilisant un plus grand nombre de lignes d'adresse (appelées nombre de lignes).

Affichage de caractères, ligne par ligne

Le CRTC commence par transmettre à la CG-ROM le code du premier caractère, à l'adresse correspondant à la ligne 1, pour obtenir la rangée supérieure de pixels. Il change ensuite l'adresse à celle du deuxième caractère, et obtient la rangée supérieure de pixels. Cette opération se répète pour chaque caractère jusqu'à la fin de la ligne. Le CRTC modifie alors l'adresse pour passer à la deuxième ligne de pixels et réitère le processus pour chaque caractère de la ligne. Tout au long de ce processus, les pixels du CG sont transmis en série au moniteur. Ainsi, une ligne complète de caractères est affichée sur l'écran. Ensuite, une nouvelle ligne de caractères est chargée dans la mémoire du CRTC, qui répète le processus pour une autre ligne de caractères. Il était possible de programmer jusqu'à 80 caractères × 64 lignes, et le CRTC génèrait également les signaux de synchronisation horizontale et verticale. La plupart des CRTC disposaient uniquement de la mémoire nécessaire pour stocker deux lignes de codes de caractères. L'une était traitée pendant que l'autre était chargée. En général, on utilisait l'accès direct à la mémoire (DMA) - ce qui permettait de réduire la charge sur le microprocesseur (figure 3).

Ils intégraient également des fonctions supplémentaires comme la détection de pointeur lumineux. Ainsi, un stylo ou un pistolet (pour les jeux vidéo) doté d'une résistance ou d'un transistor sensible à la lumière pouvait servir à enregistrer les coordonnées à l'écran où le pointeur était dirigé. La position du curseur pourrait être programmée et un bloc de caractères entier s'illumine pour afficher le curseur. Les CRTC offraient aussi la possibilité de confi-

gurer des attributs spécifiques pour chaque caractère, permettant par exemple de les souligner, de les afficher à l'envers, en surbrillance ou en mode clignotant.

J'ai travaillé sur des systèmes de terminaux qui utilisaient des CRTC, mais je n'ai jamais eu l'occasion de les programmer. Ils simplifiaient certes le travail du concepteur de système, mais leur configuration nécessitait encore un effort pour les configurer en fonction du format de caractère et d'écran utilisé, et pour programmer l'accès DMA.

La prochaine fois que vous éprouverez des difficultés à utiliser un écran LCD, qu'il soit textuel ou graphique, pensez aux concepteurs d'antan qui devaient maîtriser les subtilités des CRTC !

240058-04



À propos de l'auteur

David Ashton est né à Londres, a grandi en Rhodésie (aujourd'hui Zimbabwe), a vécu et travaillé au Zimbabwe et vit aujourd'hui en Australie. Depuis son plus jeune âge, il a été passionné par l'électronique. La Rhodésie n'étant pas un centre majeur pour l'électronique, il a dû apprendre à s'adapter et à trouver des solutions créatives. Il a acquis très tôt des compétences en adaptation, substitution et recherche de composants, des compétences dont il est toujours fier. Il a dirigé un laboratoire d'électronique mais a principalement travaillé dans les télécommunications.

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).