

Microprocesseur 32 bits à hautes performances : la structure du NS 32032

Le NS 32032 de National Semiconductor, présenté comme le premier « vrai » microprocesseur 32 bits disponible sur le marché, associe un bus de données externe de 32 bits à une architecture interne de 32 bits. De ce fait, on dispose maintenant de tous les composants matériels et logiciels nécessaires à la création d'un système 32 bits ayant les performances de gros ordinateurs à des prix de microprocesseur (*).

Le NS 32032 fait son entrée sur le marché en bonne compagnie : en effet, il existe déjà toute une série d'éléments compatibles (unités centra-

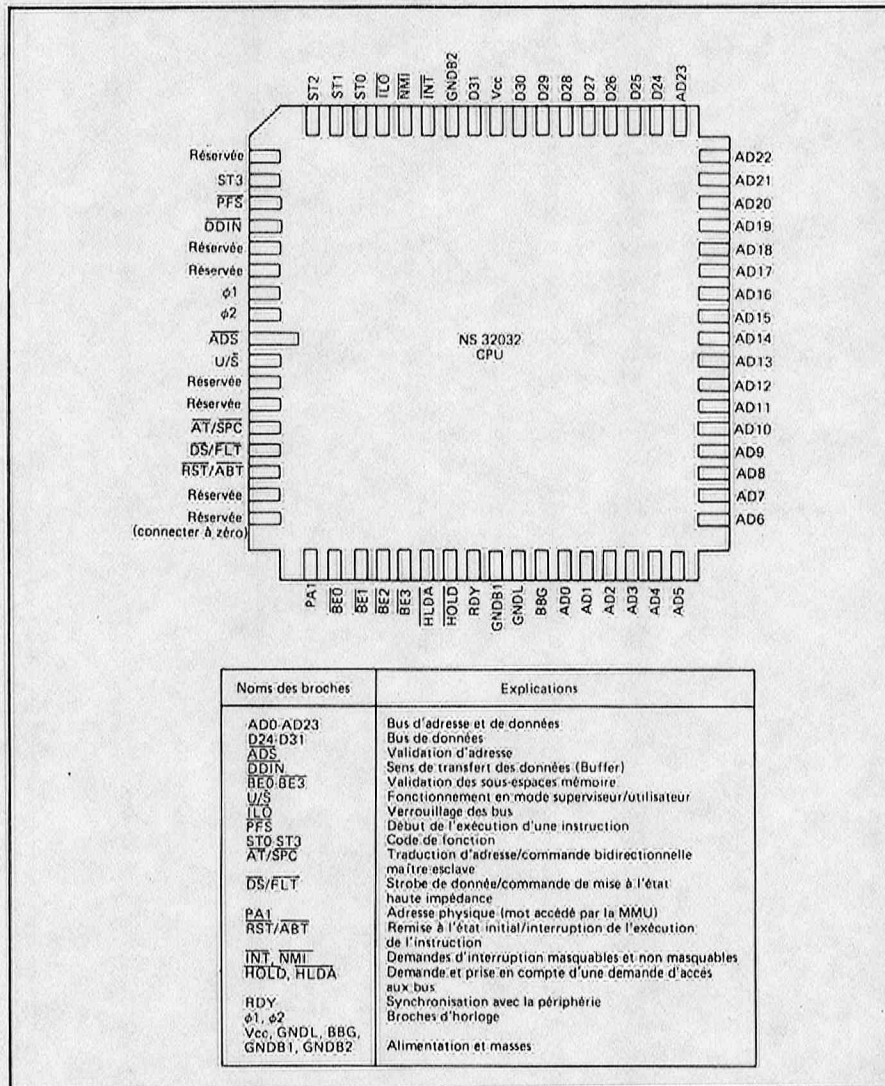
les, processeurs esclaves, périphériques) et un ensemble de logiciels croisés NSX-16 qui permet de travailler aussi bien en langage évolué qu'en

langage assembleur, avec dévermineur symbolique et plusieurs utilitaires. Un logiciel résidant et le système d'exploitation Unix sont également disponibles avec le Genix de National (4.1 BSD Unix) et le progiciel GCS 16.

En ce qui concerne le matériel, le 32032 peut être associé à l'unité arithmétique NS 16081 (FPU), l'unité de gestion mémoire NS 16082 (MMU), l'unité de séquençage (timer) NS 16201 (TCU), l'unité de commande des interruptions NS 16202 (ICU). Par ailleurs, on sait que les unités centrales NS 16032 et NS 08032 d'architecture interne 32 bits, présentent une totale compatibilité logicielle, mais que leurs bus de données externes sont sur 16 et 8 bits respectivement.

Le bus externe de données sur 32 bits du 32032, dont le brochage est indiqué en **figure 1**, est le principal élément qui contribue à l'amélioration des performances. Pour un temps donné d'accès en mémoire, la largeur du bus fournit un taux d'utilisation supérieur à celui de systèmes 16 bits. De plus, elle offre un autre avantage : en permettant l'accès du chemin de l'UC vers la mémoire aux opérandes et aux instructions, le bus est disponible plus souvent, ce qui augmente l'efficacité en multitraitement.

Fig. 1 - Brochage du NS 32032 et signification des mnémoniques



Organisation 32 bits

L'avantage du NS 32032 ne réside pas seulement dans la largeur du bus de données. En fait, il s'agit d'une extension logique de l'organisation 32 bits à toute la famille NS 16000.

Outre le bus externe de données, le 32032 présente d'autres composantes internes sur 32 bits : registres, unité

(*) Cet article, écrit par le directeur des applications NS 16000 de National Semiconductor, a été traduit de l'anglais par Evelyne Sensier.

arithmétique et logique (ALU), chemins internes de données, déplacements, etc. Les autres unités centrales que l'on trouve actuellement sur le marché n'ont qu'une partie de leur architecture sur 32 bits, l'autre étant sur 16 bits. Ces unités 32/16 bits ne peuvent exécuter qu'un nombre réduit d'opérations 32 bits, et imposent des limitations sur les structures de données supérieures à 64 K octets.

La première limitation provient de la configuration de l'ALU qui, lorsqu'elle fonctionne sur 16 bits, est associée à un additionneur 16 bits pour le calcul des adresses. En conséquence, une partie seulement du jeu d'instructions accepte des opérandes 32 bits.

La deuxième limite découle du choix architectural des déplacements 16 bits pour les modes d'adressage par pointeur. Le NS 32032 ne présente naturellement aucune de ces limitations.

L'examen de son jeu d'instructions illustre bien la configuration 32 bits. Toutes les instructions, portant sur des nombres entiers, acceptent des opérandes au format de l'octet, du mot ou du double mot. En d'autres termes, l'architecture est pleinement symétrique. Elle comprend également les opérations à virgule flottante, et en particulier lorsqu'on utilise l'unité arithmétique 16081, toutes les instructions flottantes acceptent des opérandes à simple ou double précision.

Un examen plus approfondi du jeu d'instructions met également en évidence le degré élevé de symétrie du mode d'adressage. L'architecture définit une machine générale à deux adresses, qui assure une grande liberté de choix des modes d'adressage pour les opérandes d'instructions entières et flottantes. Elle favorise ainsi les opérations de mémoire à mémoire, ce qui présente un avantage certain pour toute application programmée en langage évolué.

Déplacements et manipulations de données

Dans le mode d'adressage par pointeur, une adresse mémoire est formée de deux éléments : une adresse de base (qui se trouve habituellement

Instruction	Temps d'exécution (μ s)		Explications
	Registre à registre	Mémoire à mémoire	
MOVFL	2,6	5,3	Fait passer une opérande de la précision simple à la double précision.
MULF, MULL	4,8 à 6,2	8,5 à 12,0	Multiplication de précision simple, double.
DIVF, DIVL	8,9 à 11,9	12,6 à 17,7	Division de précision simple, double.

Notes : Tous les temps sont donnés pour un fonctionnement à 10 MHz. C'est avec les opérations registre à registre qu'on peut obtenir les vitesses les plus élevées. Les opérations mémoire à mémoire utilisent un mode d'adressage relatif pour les deux opérandes ; cela est caractéristique du code engendré par HLL pour une procédure portant sur des paramètres passés par référence. Ces chiffres comprennent également les temps de lecture et d'écriture des opérandes.

Exemples de temps d'exécution d'instructions flottantes

dans un registre) et un déplacement (qui fait normalement partie de l'instruction) se rapportant à cette adresse. Le choix initial de la valeur du déplacement détermine implicitement la taille des données qui peuvent être manipulées en mémoire, sans avoir à modifier le registre en cours d'exécution. En d'autres termes, cet adressage uniforme n'est pas une condition suffisante pour permettre des opérations directes sur des ensembles de données de taille arbitraire.

Par exemple, un microprocesseur disposant de 24 bits d'adresse peut adresser de façon uniforme 16 M octets. Or, cette uniformité est compromise (et un logiciel supplémentaire doit être écrit) si les déplacements sont limités par exemple à 16 bits et qu'il faille manipuler des structures de données supérieures à 64 K octets.

Des structures de données de cette taille deviennent de plus en plus courantes au fur et à mesure que les microprocesseurs envahissent ce qui auparavant relevait exclusivement du domaine des gros ordinateurs. Les applications graphiques en sont un exemple pratique.

Une image graphique de 1 024 x 1 024 pixels stockée sous la forme la plus compacte, occupe 128 K octets de mémoire. Le logiciel d'application graphique qui doit effectuer une translation ou une rotation de cette image doit accéder à la totalité de l'image (résidant dans la Ram). Si on ne dispose pas d'un éventail de déplacements suffisant pour balayer cette structure, le programmeur (ou le compilateur) est obligé de recourir à un palliatif comme la gestion de multi-

ples registres de base (ainsi que la possibilité de reconnaître le moment de leur utilisation) ; il en résulte un programme plus important, et donc une exécution et une compilation plus lente, à un coût de production et d'entretien plus élevé, avec sûrement une fiabilité moins grande.

La configuration du 32032 libère l'ingénieur de ces restrictions grâce à des déplacements par octet, mot simple ou double. On obtient ainsi une densité de code plus importante et une vitesse d'exécution plus rapide. En fait, avec des déplacements sur 32 bits, la taille de la structure des données est limitée, non par le déplacement, mais seulement par l'espace d'adressage mémoire de l'unité centrale.

Exécution d'opérations flottantes

Les opérations en arithmétique flottante prennent de plus en plus d'importance dans tous les domaines, y compris dans le marché du poste de travail. C'est la fonction du 16081, processeur esclave que l'on associe au 32032. En tant que tel, le 16081 reçoit des instructions et des opérandes de l'UC. Il fonctionne de telle manière qu'il sera possible plus tard d'incorporer cette fonction à l'UC elle-même.

Le 16081 exécute les quatre fonctions arithmétiques de base, la conversion entre données entières et flottantes, et la comparaison entre opérandes flottantes. Sont également acceptés les formats à simple et double précision

spécifiés par la norme 10.0 de l'IEEE p 754.

Le tableau de la page précédente donne quelques exemples de temps d'exécution avec différentes longueurs d'opérandes et différents modes d'adressage.

Gestion mémoire et système d'exploitation

Actuellement, un nombre de plus en plus grand d'applications évoluées à base de microprocesseurs requiert l'implantation d'un système d'exploitation. Dans de nombreux cas, Unix apparaît comme le système d'exploitation de référence. Cette tendance se poursuivra probablement au fur et à mesure que de nouvelles versions d'Unix (système V et 4.2 BSD) seront disponibles. Le 32032 est un excellent support des systèmes d'exploitation car il offre un environnement de type Vax par l'intermédiaire de son unité de gestion mémoire (MMU) 16082.

Cette MMU dispose de deux modes de fonctionnement : mode utilisateur ou superviseur. En général, le mode superviseur permet d'exécuter toutes les instructions avec n'importe quel opérande, alors que le mode utilisateur rejette certaines instructions ou opérandes. Par exemple, l'essai en mode utilisateur visant à modifier le registre de base d'interruption (pointeur dirigé vers les descripteurs pour accéder au programme de traitement d'interruption/déroutement) déclenche une trappe d'instruction « illégale », et redonne la main au mode superviseur. Ainsi, on interdit aux programmes utilisateur d'accéder à une ressource globale (système).

Dans la pratique, les programmes d'application utilisent le 32032 en mode utilisateur, et passent en mode superviseur dans des conditions de commande précises, définies par les appels du superviseur (SVC), les déroutements (traps) ou les interruptions. En passant en mode superviseur, le programme utilisateur transfère la commande au système d'exploitation, prévenant ainsi tout dommage

imprévisible pouvant survenir aux ressources sensibles du système.

La MMU est toujours au courant du mode de fonctionnement de l'UC, afin que les différences de privilège d'accès en mémoire puissent être établies entre les programmes utilisateur et superviseur.

L'installation d'un système d'exploitation à mémoire virtuelle avec demande de page (DPVM) est possible grâce à la MMU. Dans les systèmes DPVM, mémoire virtuelle et mémoire physique sont chacune divisées en pages de format déterminé. Un programme et l'espace de données correspondant sont implantés (par le système d'exploitation) sur un certain nombre de pages (qui ne sont pas obligatoirement contiguës) de la mémoire physique.

Une table de conversion d'adresse (de la mémoire virtuelle à la mémoire physique) est établie par le système d'exploitation. Le contenu de la table ne fournit pas seulement l'information sur la conversion d'adresse, mais indique également la présence ou

Fig. 2 - Schéma simplifié des interconnexions entre un 32032 et un séquenceur/timer 16201

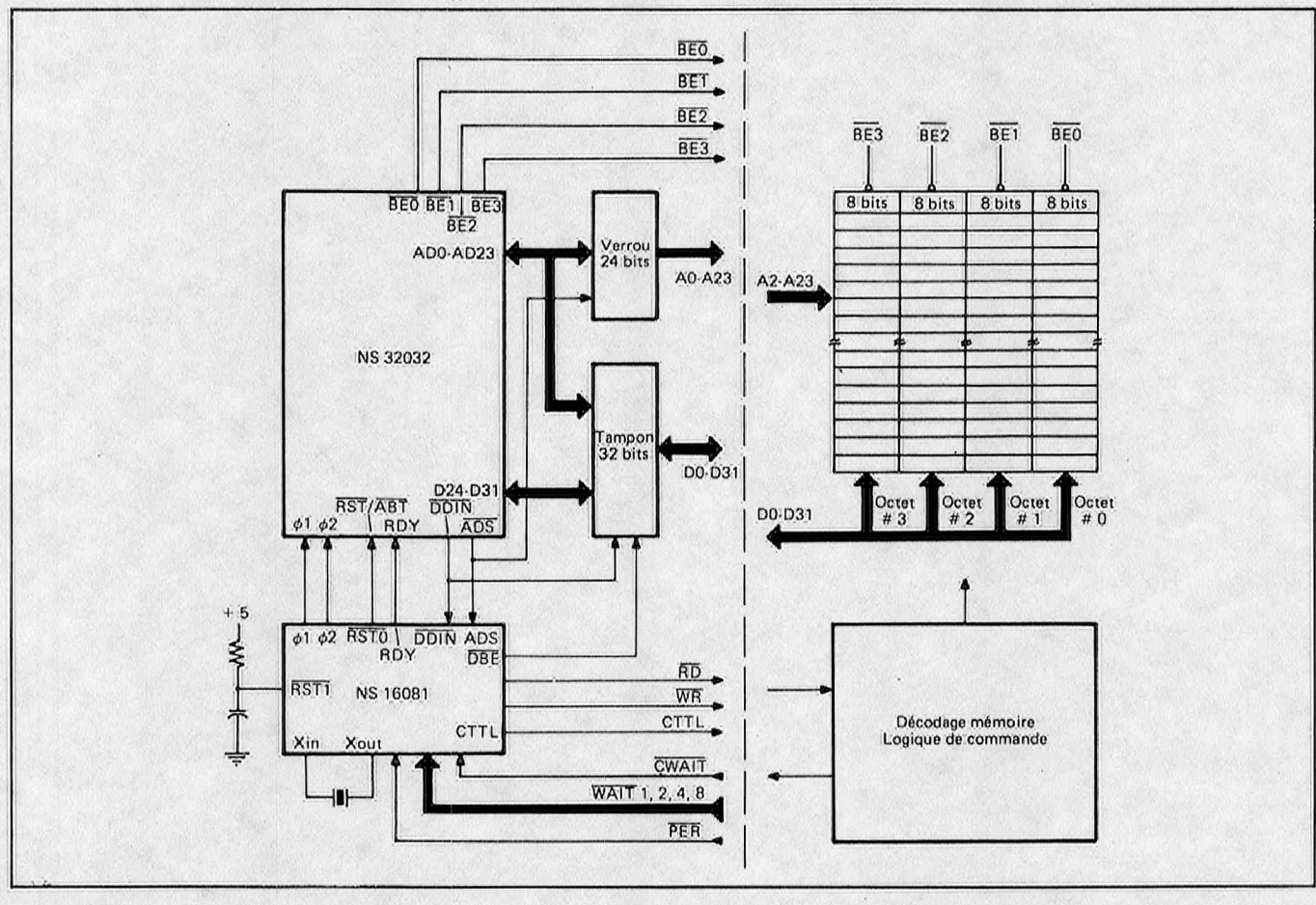


Fig. 3 - Cycle lecture du montage représenté en figure 2

l'absence d'une page virtuelle donnée dans la mémoire physique. Cela permet de réaliser l'échange de pages en cours d'exécution du programme.

La table comporte en outre la description du degré de protection de la mémoire (total, uniquement à la lecture, aucun accès) pour chaque page mémoire. La MMU conserve, sur une mémoire associative incorporée, les trente-deux dernières réaffectations d'adresse utilisées, réalisant ainsi la conversion d'adresse en 100 ns.

Étant donné la taille des pages et le nombre d'entrées associatives, on obtient dans la pratique un taux de 98 % d'utilisation de la mémoire cache. Quand la conversion demandée n'est pas disponible dans la mémoire-cache, la MMU accède aux tables résidant en mémoire en son nom propre, afin d'obtenir la conversion et

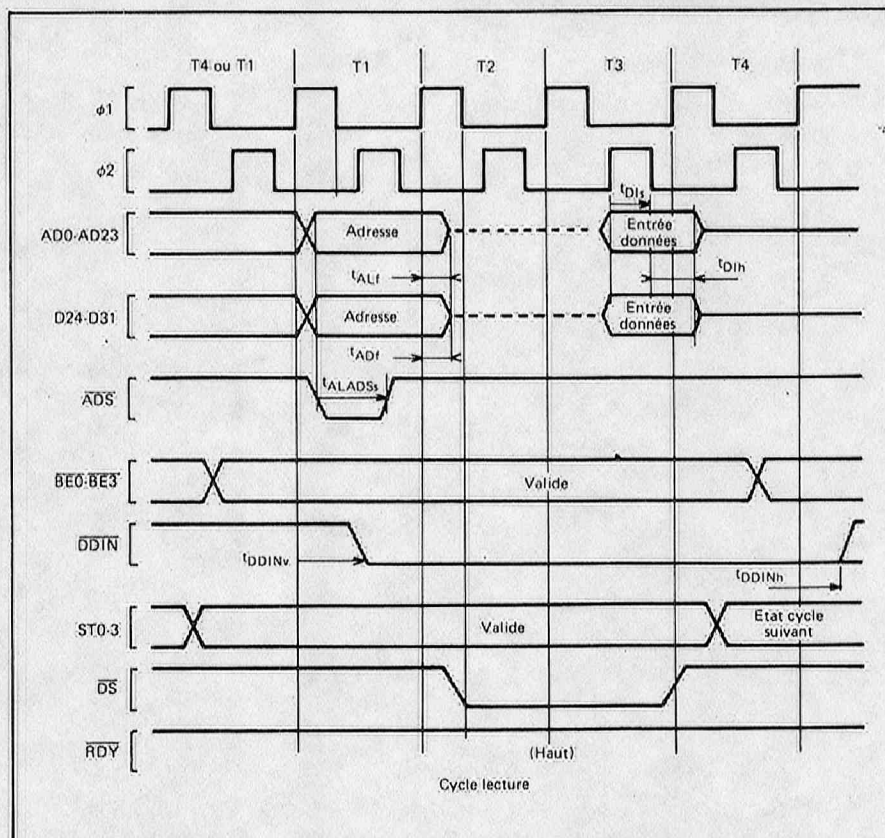
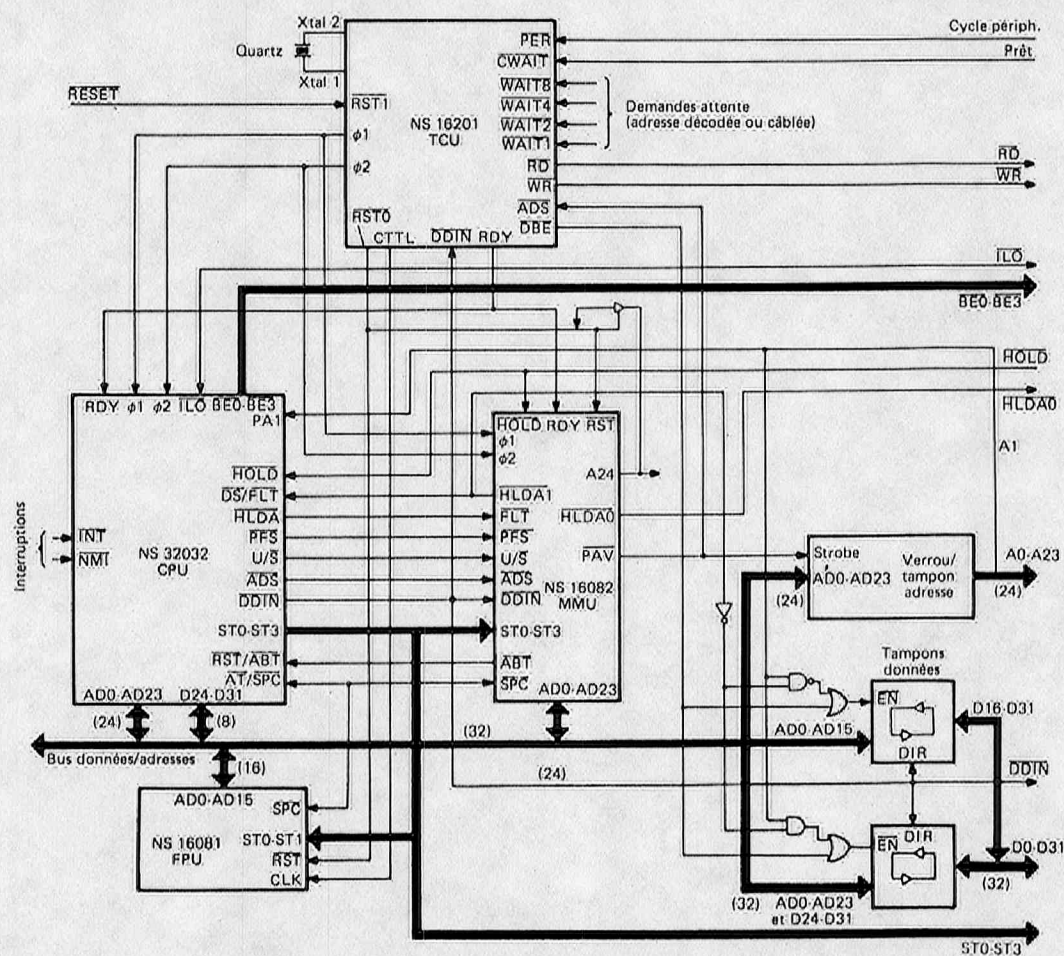


Fig. 4 - Schéma d'interconnexion d'un ensemble avec 32032, processeur esclave et MMU



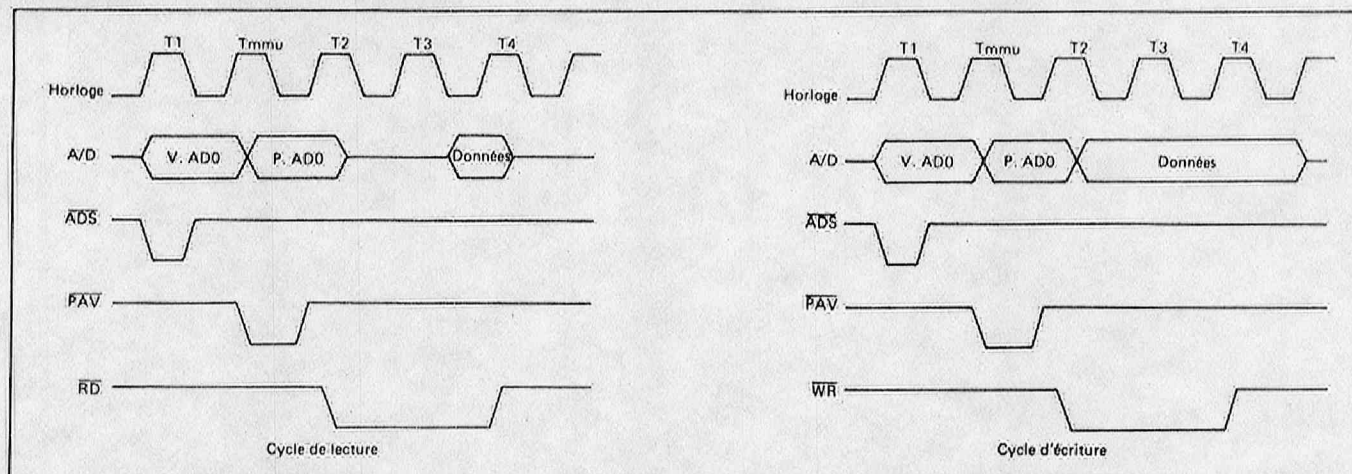


Fig. 5 - Diagramme des temps du bus avec adresse virtuelle figurant dans le tampon de conversion d'adresse

remettre sa mémoire-cache à jour. Au cours de cet accès, les bits d'utilisation de pages sont remis à jour dans les tables, ce qui permet au système d'exploitation de prendre une décision rationnelle avec tous les éléments nécessaires quand un transfert de page est demandé.

La combinaison 32032/16082 offre donc un système de gestion mémoire que l'on pourrait décrire comme étant un « classique de la nouveauté ». Ses ressemblances avec les techniques Vax-11 et IBM 370 simplifient la portabilité d'Unix et de ses logiciels associés ainsi que celle d'autres systèmes d'exploitation. Rappelons, à ce propos, qu'Unix a été adopté par National Semiconductor et que le noyau de Genix est 25 % plus petit que celui de son homologue Vax.

Exemples d'interconnexion

La figure 2 représente un schéma d'interconnexion simplifiée entre un 32032, un 16201 et un bloc mémoire de 32 bits de large, avec temporisation correspondante du cycle mémoire.

Le TCU (16201) est chargé d'exécuter quatre fonctions principales : remise à zéro, génération de signaux d'horloge et d'échantillonnage, synchronisation mémoire et UC. La fonction « Reset » est déclenchée par un trigger de Schmitt. Une horloge externe fournit à la fois des signaux d'horloge de niveau Mos ($\phi 1$ et $\phi 2$) et CTTL, niveau TTL équivalent de $\phi 1$.

Un cycle de bus est lancé par le signal ADS (address strobe) envoyé

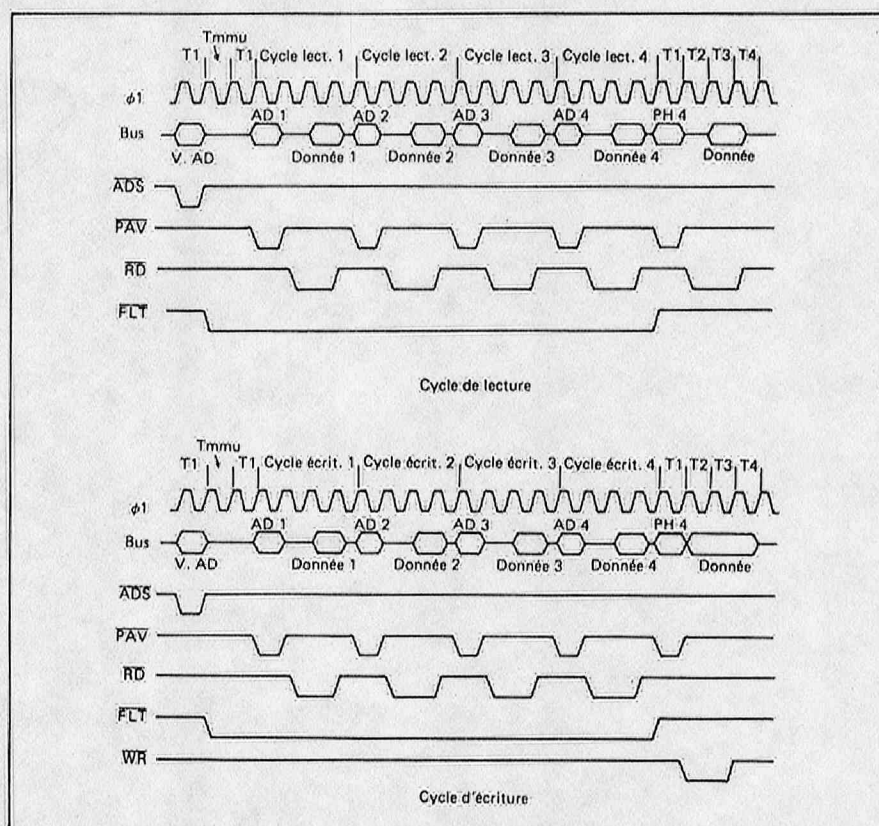


Fig. 6 - Diagramme des temps du bus avec adresse virtuelle ne figurant pas dans le tampon de conversion d'adresse

par l'UC, qui sert également au démultiplexage de l'adresse provenant du bus de données (fig. 3). Le TCU génère des signaux de lecture ou d'écriture mémoire selon l'état du signal de direction des données DDIN émanant de l'unité centrale. Le TCU adapte les vitesses des mémoires, périphériques et UC en insérant des temps d'attente, qui découlent du décodage d'adresse et de la logique de commande appliquée à ses entrées.

Lorsque $\overline{\text{CWAIT}}$ est actif, les états d'attente sont insérés dans le cycle de bus en cours, jusqu'à ce que le signal disparaisse. Les quatre entrées WAIT n (avec $n = 1, 2, 4$ ou 8) peuvent être pilotées en formant des combinaisons de manière à insérer de zéro à quinze états d'attente dans le cycle de bus. La broche PER permet d'insérer cinq états d'attente et d'ajuster la temporisation des signaux de lecture ou d'écriture pour la rendre compatible

avec les exigences de temps de prépositionnement et de maintien (Hold) des périphériques. Ces trois modes de synchronisation simplifient énormément la tâche de l'ingénieur système en facilitant l'utilisation d'une gamme variée de composants.

L'examen de la mémoire (fig. 2) montre qu'elle est organisée en quatre blocs d'un octet de large. L'ensemble reçoit vingt-deux des lignes d'adresse démultiplexées, A2-A23, alors que les octets reçoivent une validation individuelle issue de BE0-BE3 provenant de l'UC. En effet, la mémoire est adressée par double mot, avec des validations individuelles d'octets à l'intérieur du double mot.

Le 32032 permet l'accès à des octets contigus à l'intérieur du double mot adressé. Cela entraîne dix possibilités différentes d'accès à la mémoire : quatre d'un octet, trois d'un mot, deux de trois octets, et un d'un double mot. Le 32032 n'impose aucune restriction d'alignement de mot ou de double mot aux programmes et aux données ; comme il se doit, on accède aux structures non-alignées par des cycles mémoire multiples. Les utilisateurs peuvent cependant améliorer les performances, au prix de sollicitations plus importantes de la mémoire, en forçant l'alignement des données.

Un 32032 en technologie C-Mos

National Semiconductor a récemment annoncé que la version C-Mos du 32032 se fera vers le milieu de cette année, et que l'évolution vers le 32132 progressera par étapes successives intégrant de plus en plus de fonctions. Le 32132 comportera environ 500 000 transistors et fonctionnera à une vitesse de 30 MHz (à l'horizon 1988), ce qui n'est pas possible en X-Mos si l'on n'introduit pas de refroidissement efficace du circuit. Une grande partie de la surface du silicium (30 % prévoit-on) sera consacrée au test qui devient de plus en plus complexe à mesure que la densité d'intégration s'accroît.

Mise en œuvre des processeurs esclaves

La figure 4 illustre l'association des processeurs esclaves FPU 16081 et MMU 16082 à la combinaison UC/TCU. Les principales connexions avec les processeurs esclaves s'appliquent aux mots de poids faible du bus de données, à tout ou partie des lignes d'état de l'UC ST0-ST3 et au signal bidirectionnel SPC. Il convient juste d'ajouter une remise à zéro et une horloge à la FPU (ou à tout autre processeur esclave susceptible d'être implanté pour augmenter le jeu d'instruction de l'UC).

Outre l'interface esclave de base, la MMU demande des connexions supplémentaires. Elle comprend en entier le bus d'adresses, toutes les lignes d'état (y compris l'indicateur de mode utilisateur/superviseur, U/S) et le signal ADS (address strobe).

Dans un système à gestion de mémoire, l'exécution du cycle de bus se déroule ainsi (fig. 5 et 6) : le 32032 présente l'adresse virtuelle sur AD0-AD23 pendant T1 et signale cet événement à la MMU en envoyant le signal ADS. Ensuite, l'UC met ces lignes à l'état haute impédance. La MMU forme l'adresse physique (habituellement à partir de la mémoire-cache interne) sur les mêmes lignes pendant Tmmu, et informe le TCU de cette action par la ligne « Validation d'Adresse Physique » (PAV) ; cela provoque le démultiplexage de l'adresse par le verrou d'adresse et le démarrage du cycle de bus sous le contrôle du TCU.

Quand la conversion d'adresse ne peut pas se faire directement à partir de la mémoire-cache incorporée à la MMU, celle-ci suspend le cycle de bus, avant d'envoyer PAV, au moyen de la ligne FLT, puis accède aux tables résidant en mémoire par son bus de données 16 bits. Elle reprend le cycle interrompu, en utilisant l'adresse réelle ainsi obtenue. Pendant ces cycles flottants, le tampon supérieur du bus de données (fig. 3) assure l'interfaçage du bus de données 16 bits de la MMU en « ramenant » D16-D31 sur AD0-AD15 lorsque la MMU accède à des mots impairs.

Gary A. Wray