Architecture des ordinateurs

par Christophe TOMBELLE
IMT Lille Douai

Livret élève

Tous droits de traduction, de reproduction
et d'adaptation réservés pour tous pays.
© IMT Lille Douai, 2017

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les "copies ou reproductions strictement réservées à l'usage du copiste et non destinées à une utilisation collective" et d'autre part, que les analyses et que les courtes citations dans un but d'exemple et d'illustration, "toute représentation intégrale ou partielle, faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite" (alinéa 1er de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du code pénal.

IMT Lille Douai
École Mines-Télécom
IMT-Université de Lille
Cité scientifique, rue G. Marconi - BP 20145
59653 Villeneuve d'Ascq Cedex
Tél. : 03.20.33.55.77 Fax : 03.20.33.55.99
http://www.telecom-lille.fr/ http://imt-lille-douai.fr/

Table des matières

Préambule 5
Séquence 1 Le matériel 7
Séquence 2 Le matériel : travail dirigé 8
Séquence 3 Le logiciel : modèle de programmation 9
Séquence 4 Le logiciel : représentation des données 10
Séquence 5 Le logiciel : structuration des données 12
Séquence 6 Le logiciel : structuration du code 13
Séquence 7 Le logiciel : procédures et fonctions 15
Séquence 8 Le logiciel : paramètres et variables locales 16
Séquence 9 Le matériel : gestion des entrées-sorties 18
Séquence 10 Les logiciels d'exploitation 20
Séquence 11 Approfondissement du parallélisme d'exécution 22
Séquence 12 Performances du matériel 23

Préambule

Infos pratiques

- Coordinateur IMT Lille-Douai : **Christophe TOMBELLE**
  e-mail : christophe.tombelle@imt-lille-douai.fr

Télégroupe

Dates en ligne sur Whippet « Systèmes Numériques »

Intervenant

Christophe TOMBELLE

Ressources

Voir le portail de « Systèmes Numériques » sur Whippet

- un support de cours
- un livret élève
- séquences TuttelNet
- autres ressources

Séquence 1
Le matériel

Objectifs

- Connaître la structure et le fonctionnement d'un système à microprocesseur simple.
- Connaître le rôle et la structure de chacun de ses principaux composants.
- Connaître les fondements technologiques et le rôle des signaux d'un système à microprocesseur.
- Connaître le vocabulaire associé au domaine.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s1/frame_a23_s01.htm
Support de cours

Exercice

Exercice 1.1.

Quelle est l'entité qui établit une correspondance entre une plage d'adresses et une ressource du système ?

Quel bus est-il bidirectionnel 3 états ?

Quelle entité émet-elle l'information lors d'une lecture ? d'une écriture ? d'une entrée ? d'une sortie ?

Quel est l'avantage de l'architecture en bus ? Comment s'appellent les instructions d'entrée et de sortie ?

Quel signal distingue-t-il les accès IO des accès mémoire ?

Quelle différence de comportement distingue-t-elle une case mémoire d'un registre d'entrée-sortie ?

Comment calcule-t-on la taille de l'espace adressable ?

A priori, quelle est la capacité d'une mémoire comportant 10 signaux d'adresses et 8 signaux de données ?

Séquence 2
Le matériel : travail dirigé

Objectifs

- Connaître les fondements de l'exécution d'un programme.
- Approfondir le fonctionnement d'un périphérique
- Se familiariser avec les ordres de grandeur utilisés en environnement microprocesseur.
- Etre capable d'effectuer des calculs de performances.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s2/frame_a23_s02.htm
Support de cours.

Exercice

Faire les exercices proposés par la séquence en ligne.

Exercice 2.1.

Pour quels besoins le processeur accède-t-il à la mémoire en lecture ?

Quels sont les 2 types d'informations circulant sur le bus de données ?

Quels ordres de grandeur retenez-vous pour une opération d'entrées-sorties en termes de :

- durée pour émettre 20 caractères à 9600 bps
- nombre d'instructions exécutées pendant cette durée par un processeur de 50 Mips

Comment appelle-t-on la portion de puissance du processeur utilisée pour le compte du système d'exploitation ?

Séquence 3
Le logiciel : modèle de programmation

Objectifs

- Introduire les notions liées au logiciel de bas niveau.
- Prendre conscience des spécificités de ce type de logiciel.
- A plus court terme, se familiariser avec l'assembleur du 8086.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s3/frame_a23_s03.htm
Support de cours.

Exercice

Exercice 3.1.

Combien vaut la différence entre les adresses de 2 mots consécutifs ?

Pour les processeurs Intel : quel est l'ordre de stockage mémoire utilisé ? Où est stocké le poids faible / poids fort ?

Peut-on parler de "l'adresse d'un registre" tel que AX ou SI ?

Quelle est la taille des registres du 8086 ?

Quelle est la taille de AH ? de AL ?

Quelle est la taille d'un indicateur (ou flag) ?

Quels sont les noms des registres sécables du 8086 ?

De quelle taille l'espace adressable du 8086 est-il ?

A quelle autre adresse une adresse offset est-elle relative ? Autrement dit, à l'intérieur de quelle entité une adresse offset a-t-elle un sens ?

De quelle taille sont les adresses physiques du 8086 ? Même question pour les adresses offset, pour les adresses segment et pour les adresses segmentées.

Comment calcule-t-on une adresse physique à partir d'une adresse segmentée ?

Quelle est la taille d'un mot, d'un double mot, d'un paragraphe ?

Quelle est la taille maximale d'un segment du 8086 ?

De quel nombre les adresses physiques de début de segment sont-elles multiples ?

Sous Turbo-Debugger, à quelle condition la notation cs:0100 est-elle équivalente à la notation 427C:0100 ?

Avec quel registre segment sont associés les registres offsets suivants :

- BX, SI, DI
- BP
- SP
- IP

Etendre la réflexion sur la notion d'adresses : adressage des secteurs d'un disque dur.

Séquence 4
Le logiciel : représentation des données

Objectifs

- Connaître les principaux modes d'adressage.
- Etudier quelques instructions d'un microprocesseur
- Mettre en œuvre quelques modes d'adressage.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s4/frame_a23_s04.htm
Support de cours.

Exercices

Coder en langage d'assemblage les 4 premières lignes de la procédure proc du TP2 :
http://wapiti.enic.fr/commun/ens/peda/modules/A23/Tp2/Tp2.html

Exercice 4.1. Modes d'adressage

Dans quel mode d'adressage l'adresse de l'opérande est-elle codée dans le code machine ?

Quel mode d'adressage ne peut-il pas être utilisé comme opérande gauche ?

Dans quel mode d'adressage la combinaison de registres fournissant l'adresse de l'opérande est-elle codée dans le code machine ?

Dans quel mode d'adressage l'opérande est-il codé dans le code machine ?

Quel est le mode d'adressage qui ne fait appel à aucune adresse effective ?

On considère la notation `[BX+SI+depl]`.

A quel segment accède ce mode d'adressage ?

Comment appelle-t-on ce mode d'adressage ?

Comment qualifie-t-on le registre `BX` ?

Comment qualifie-t-on le registre `SI` ?

Quel autre mode d'adressage devient ce mode d'adressage si on supprime la partie `SI+depl` ?

Quel autre mode d'adressage devient ce mode d'adressage si on supprime la partie `BX+SI` ?

Quel autre mode d'adressage devient ce mode d'adressage si on supprime la partie `BX` ?

A quel segment accède le mode d'adressage `[BP+SI+depl]` ?

Exercice 4.2. Instructions

Quelles sont les instructions qui permettent de charger un couple de registres depuis un pointeur far ?

Quelle est l'instruction qui récupère dans un registre l'adresse offset issue d'un mode d'adressage direct ou indirect ?

L'adresse effective est-elle une adresse segment, une adresse offset ou une adresse segmentée ?

Dans l'instruction `ADD opg,opd` où est stocké le résultat ?

`AL` contient `5A`. Que contient `AL` après l'instruction `AND AL,0F` ? après 4 instructions `SHR AL,1`

Exercice 4.2. Nombres signés

Dans la représentation des nombres entiers signés, quel est le bit affecté d'un poids négatif ?

Comment écrit-on le nombre `-1` codé sur un octet ? codé sur 2 octets ?

On considère des opérandes signés sur 16 bits. Que donne le calcul `32767 + 1` ?

Séquence 5
Le logiciel : structuration des données

Objectifs

- Connaître l'usage que font les langages évolués des modes d'adressage d'un microprocesseur.
- Approfondir les notions de pointeur, d'enregistrements et de tableaux.
- A plus court terme, être capable de coder, en assembleur 8086, l'accès à une donnée structurée.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s5/frame_a23_s05.htm
Support de cours.

Exercices

Faire les exercices proposés dans la séquence en ligne.

Faire la suite du TP2

Sujet en ligne de TP: http://wapiti.enic.fr/commun/ens/peda/modules/A23/Tp2/Tp2.html

Corrigé en ligne de TP: http://wapiti.enic.fr/commun/ens/peda/modules/A23/Tp2/Tp2_Corrige.html

Séquence 6
Le logiciel : structuration du code

Objectifs

- Connaître les fondements technologiques sur lesquels s'appuie la notion d'algorithme : comment le processeur prend des décisions sur la suite à donner au programme selon l'analyse qu'il fait des données qu'il traite,
- Comprendre comment un compilateur génère du code assembleur à partir d'un texte en langage évolué.
- A plus court terme, être capable d'implanter un algorithme simple en langage d'assemblage.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s6/frame_a23_s06.htm
Support de cours

Exercices

Exercice 6.1.

Qu'est-ce qu'une étiquette ?

A l'aide de quelles instructions un processeur prend-il des décisions ?

Quelles sont les instructions qui existent en version intra-segment et inter-segment ?

Qu'est-ce que du code mort ?

Pour éviter le code mort, de quoi doit-on faire suivre un saut inconditionnel ?

A quoi reconnaît-on une boucle au niveau du code assembleur ?

Exercice 6.2.

Sujet en ligne du TP : http://wapiti.enic.fr/commun/ens/peda/modules/A23/Tp3/Tp3.html

Corrigé en ligne du TP : http://wapiti.enic.fr/commun/ens/peda/modules/A23/Tp3/Tp3_Corrige.html

Exercice 6.3.

On considère l'algorithme suivant qui, dans une chaîne de caractères terminée par '\0', remplace les virgules par des points-virgules.

```
/*    i=0; */
/*    while (chaine[i]!='\0') { */
/*          if (chaine[i]==',') */
/*                chaine[i]=';'; */
/*          i++; */
/*    } */
```

Le coder en langage d'assemblage en respectant la méthodologie proposée.

Dans un premier temps, coder la structure de l'algorithme : sauts conditionnels, inconditionnels et étiquettes selon le schéma type d'un "si alors sinon" et d'un "tant que". Laisser des "trous" pour le reste. Adopter une démarche systématique. Par exemple, un "sinon" correspond systématiquement à `jmp fsiXX` suivi d'une étiquette `elseXX`.

Dans un deuxième temps, coder le test des expressions booléennes puis les actions. Sortir de la boucle les éventuelles instructions réexécutées inutilement (invariants). Ecrire un programme permettant de tester l'algorithme.

Exercice 6.4. Réflexion

En quoi une instruction `JMP BX` ou `JMP [BX+depl]` peut-elle aider à implanter une structure de cas plus efficacement ?

Séquence 7
Le logiciel : procédures et fonctions

Objectif

- Connaître les fondements technologiques sur lesquels s'appuie la notion de sous-programme, élément fondamental de la structuration du code dans les langages évolués.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s7/frame_a23_s07.htm
Support de cours.

Exercices

Exercice 7.1. Pile

Que pointe le registre SP ?

Comment évolue l'adresse présente dans SP lors d'un empilement ? d'un dépilement ?

Quels sont les segments mis en jeu par l'instruction `push [uneVariable]` ? Précisez pour chacun s'il s'agit d'un accès en lecture ou en écriture. Mêmes questions avec l'instruction `pop [uneVariable]`.

A quel endroit du segment de pile la pile est-elle implantée ? Donc, avec quelle adresse initialise-t-on SP pour initialiser une pile vide dans un segment de pile de 32 ko ?

Quelle est la taille d'un emplacement de pile ?

Exercice 7.1. Sous-programme

Quelles sont les instructions qui existent en version intra-segment et inter-segment ?

Quelle instruction empile une adresse de retour ?

Quelle instruction dépile une adresse de retour ? Dans quel(s) registre(s) ?

Par quel type d'entité est transmis le résultat d'une fonction au programme appelant ?

Quels sont les segments mis en jeu par les instructions d'appel et de retour de sous-programme ? Pour quel usage ?

Séquence 8
Le logiciel : paramètres et variables locales

Objectifs

- Connaître les mécanismes de passage de paramètres par valeur ou par pointeur,
- connaître les mécanismes d'accès aux variables locales et aux paramètres,
- être capable d'interfacer un sous-programme assembleur avec paramètres et une application en langage C.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s8/frame_a23_s08.htm
Support de cours.

Exercices

Exercice 8.1.

Dans quelle entité les paramètres et les variables locales sont-ils stockés ?

Dans quel segment est située cette entité ?

Un microprocesseur possède-t-il une instruction d'appel de sous-programme avec arguments ?

Quelles instructions réservent-elles l'emplacement pour les paramètres ? Où sont-elles situées ?

Quelle instruction réserve-t-elle l'emplacement pour les variables locales ? Où est-elle située ?

Relier par un trait les instructions ayant un rôle inverse. A quoi correspond la valeur `XX` ?

`call proc`	`mov sp,bp`
`push bp`	`pop bp`
`mov bp,sp`	`ret`
`sub sp,XX`	

L'adresse d'une variable locale est-elle connue lors de la compilation ?

Quel mode d'adressage permet-il d'accéder aux paramètres et variables locales ?

Quel est le signe du déplacement pour l'accès aux variables locales ? aux paramètres ?

A quoi correspond `[BP]` ? `[BP+2]` ? `[BP+4]` pour un sous-programme inter-segment ?

En langage C, quel est l'argument empilé en premier ?

A quoi sert un `add sp,XX` situé à la suite d'une instruction `call` ? A quoi correspond `XX` ?

Un **paramètre** `int tab[N]` est-il un pointeur ou un tableau ? Pourquoi ?

Construire le contexte de pile de la procédure `void incr( int *pVariable, int step );` on travaillera systématiquement avec des adresses far.

Estimer l'espace occupé sur la pile par la procédure `hanoi( int n, int source, int dest )` et comportant les variables intermédiaires `int tmp, result;` on supposera que la procédure est compilée dans sa version inter-segment.

Exercice 8.2. Réflexion

Réfléchir au mécanisme d'appel de méthodes virtuelles des langages à objets.

Séquence 9
Le matériel : gestion des entrées-sorties

Objectifs

- Savoir comment fonctionnent les interruptions et le DMA sur le plan matériel et logiciel,
- savoir ce qu'est une interruption logicielle, une exception,
- connaître les usages des différents modes de gestion des entrées-sorties,
- connaître le vocabulaire associé à ces notions.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s9/frame_a23_s09.htm
Support de cours.

Exercices

Quel est le rôle d'un pilote de périphérique ?

Exercice 9.1. Polling

Quel type de registre (et de quelle entité) doit-on lire pour savoir si un événement s'est produit ?

A quelle condition le polling est-il acceptable ?

Exercice 9.2. Interruptions

Distinguer les termes "numéro de vecteur", "vecteur", "table des vecteurs", "sous-programme d'interruption"

Quelle est la taille d'un vecteur d'interruption ? A quelle adresse est situé le vecteur dont le numéro hexa est 21 ?

Quelle est le nombre maximal de vecteurs d'interruptions pour le 8086 ?

Quelle est la taille de la Table des Vecteurs d'Interruption ?

Par quel dispositif est fourni le numéro de vecteur dans les interruptions matérielles ?

Comment est fourni le numéro de vecteur dans les interruptions logicielles ?

Citez un usage des interruptions logicielles.

Comment est fourni le numéro de vecteur dans les exceptions ?

Quelle précaution un sous-programme d'interruption doit-il prendre pour ne pas perturber le programme interrompu ?

Exercice 9.3. DMA

Citer quelques dispositifs faisant appel au DMA.

Quelle est l'entité qui programme le DMAC et lance la sous-traitance DMA ?

Lors d'une sous-traitance DMA, un contrôleur de DMA exécute de manière matérielle l'équivalent d'un algorithme simple. Ecrire cet algorithme en pseudo-code.

Quelle est l'entité matérielle qui synchronise le transfert d'un mot ? Définir le rôle des signaux mis en œuvre par le DMA.

Comment et par où passent les mots transférés ?

Que se passe-t-il en fin de sous-traitance DMA ?

Que fait le processeur pendant le transfert d'un mot ?

Séquence 10
Les logiciels d'exploitation

Objectifs

- Connaître les principaux services rendus par un système d'exploitation,
- connaître les fondements technologiques sur lesquels s'appuient les systèmes protégés, multi-tâches ou à mémoire virtuelle paginée.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s10/frame_a23_s10.htm
Support de cours.

Exercices

Exercice 10.1.

Quel est le périphérique utilisé par la mémoire virtuelle pour l'extension de l'espace mémoire ?

Sur quels dispositifs matériels s'appuie le système d'exploitation pour la protection et la pagination de la mémoire ?

Exercice 10.2. Systèmes protégés

Quels modes d'exécution trouve-t-on dans les systèmes protégés les plus simples ? Comment passe-t-on d'un mode à l'autre ?

Combien a-t-on de modes d'exécution dans le 386 ?

Dans un système protégé, un programme bogué provoque l'arrêt brutal d'une application avec le message "`segmentation fault`". Que s'est-il passé ?

Exercice 10.3. Segmentation - pagination

Selon vous, à quoi peut bien servir `pagefile.sys` dans un système Windows NT ? (ce n'est pas dans le cours)

A quel périphérique fait-on appel pour étendre la mémoire centrale ?

A quel prix, les applications peuvent-ils disposer apparemment de plus de mémoire qu'il n'y a de mémoire physique ?

Comment se fait-il qu'ajouter de la mémoire à un ordinateur puisse améliorer ses performances ?

Pour le 386, quelle est la taille d'une adresse linéaire ? physique ?

Que se passe-t-il lorsqu'une application tente d'accéder à une page logique qui n'est pas chargée en mémoire ? Quel est le traitement correctif qui s'en suit ? Ce traitement est-il pris en charge par l'application ou le système d'exploitation ?

Comment appelle-t-on une allocation de mémoire au fur et à mesure des besoins ?

Mettez en relation les mots **segment** (ou segmentation) et **page** (ou pagination) d'une part et les caractéristiques suivantes :

• de taille fixe, • de taille variable, • supporte le système de protection, • supporte un système d'allocation mémoire transparent pour les applications, • les physiques peuvent être non contigu(e)s et les logiques contigu(e)s, • peuvent être de code de donnés ou de pile,	• ont une taille de quelques kilooctets, • peuvent atteindre les 4Go sur le 386 et successeurs, • ont un bit de présence en mémoire, d'accès et de modification pour les algorithmes qui les remplace, • ont leur image sur disque, • en son absence adresse linéaire=adresse physique, • calcule une adresse linéaire, • calcule une adresse physique.

Exercice 10.4. Parallélisme

Qu'est-ce qu'un processus bloqué ? éligible ? élu ?

Que dit-on d'un processus prêt qui ne reçoit plus le processeur ?

Quelle est l'entité qui alloue le processeur aux processus ?

Quels systèmes garantissent-ils un traitement dans un intervalle de temps borné ?

Qu'appelle-t-on famine d'un processus ?

Quel est le terme qui qualifie une allocation du processeur aux processus prêts selon un ordre "chacun son tour" ?

En ordonnancement temps-réel, à quelles conditions un processus moins prioritaire peut-il obtenir le processeur ?

Comment qualifie-t-on une gestion du processeur telle qu'un processus peut se voir retirer le processeur d'autorité ?

Séquence 11
Approfondissement du parallélisme d'exécution

Objectif

- Mieux comprendre comment fonctionne l'exécution de plusieurs processus par un seul microprocesseur.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s11/frame_a23_s11.htm
Support de cours.

Exercices

Exercice 11.1.

Dans une commutation de contexte quels sont les registres qui ne sont pas stockés sur la pile ? Où sont-ils stockés ?

De quel dispositif matériel a-t-on besoin pour implanter un ordonnanceur préemptif ?

De quelles étapes le traitement de l'interruption générée par ce dispositif est-il composé ?

Habituellement un sous-programme retourne au programme appelant. En quoi le sous-programme de cette interruption échappe-t-il à la règle ?

Comment un ordonnanceur peut-il allouer le processeur alors qu'il en a besoin lui-même ?

Exercice 11.2.

Prolonger la réflexion sur la conception d'un système multi-tâche. Que faut-il faire pour ajouter un nouveau processus aux processus déjà actifs ? Quels sont les paramètres à passer à la procédure d'ajout ?

Quand un processus termine son exécution, que faut-il faire pour qu'il ne participe plus à l'ordonnancement ? Comment récupérer les ressources qu'il utilisait ?

On veut pouvoir gérer des états "bloqué", éligible, élu. Réfléchir aux structures de données nécessaires et à leur gestion. Comment un processus passe-t-il de l'état bloqué à l'état éligible ? de l'état éligible à l'état bloqué ? de l'état éligible à l'état élu ? de l'état élu à l'état éligible ? de l'état élu à l'état bloqué ?

Réfléchir à un moyen pour un processus de s'endormir (une forme de blocage) et de se faire réveiller par un autre (une forme de déblocage).

Comment gérer des priorités ? Comment ordonnancer des processus éligibles de même priorité ?

Séquence 12
Performances du matériel

Objectifs

- Connaître les technologies d'amélioration des performances d'un ordinateur ou d'un processeur,
- connaître les principaux types d'architecture de processeur,
- connaître l'architecture d'un système à hautes performances.

Ressources

Séquence en ligne :
https://whippet.telecom-lille.fr/pluginfile.php/107/mod_resource/content/2/rainet/A23/s12/frame_a23_s12.htm
Support de cours

Exercices

Exercice 12.1.

Avec un processeur RISC, quelles sont les instructions qui accèdent à la mémoire ?

En quoi consiste le principe du "registre à registre" des microprocesseurs RISC ?

En quoi le compilateur a-t-il un rôle déterminant dans les architectures RISC ?

Pour quel type de systèmes un nombre de registres important peut-il se révéler un inconvénient ?

Pourquoi les ordinateurs actuels ont-ils besoin de mémoire cache ?

Exercice 12.2. Réflexion

Réfléchir à une nouvelle architecture PROCESSEUR+RAM qui prolongerait le principe du pipe-line jusqu'à l'intérieur de la RAM.

Au sens premier du terme, la mémoire cache est située entre le processeur et la mémoire centrale. La notion de cache étant relative, on peut l'étendre à des dispositifs de plus en plus lents. Lesquels ?

Le Web utilise une hiérarchie de mémoires caches. A quels endroits selon vous ?

En quoi la gestion de la mémoire virtuelle paginée est-elle similaire à celle d'une mémoire cache ?

A part le processeur ou la carte mère, connaissez-vous d'autres dispositifs électroniques comportant une forme de mémoire cache.