ADO DE.

QCM:

1) La mémoire centrale d'un ordinateur :
- peut être composée de plusieurs circuits mémoire indépendants.

2) Les instructions exécutées par un processeur :
- sont stockées dans un registre interne du processeur avant exécution.

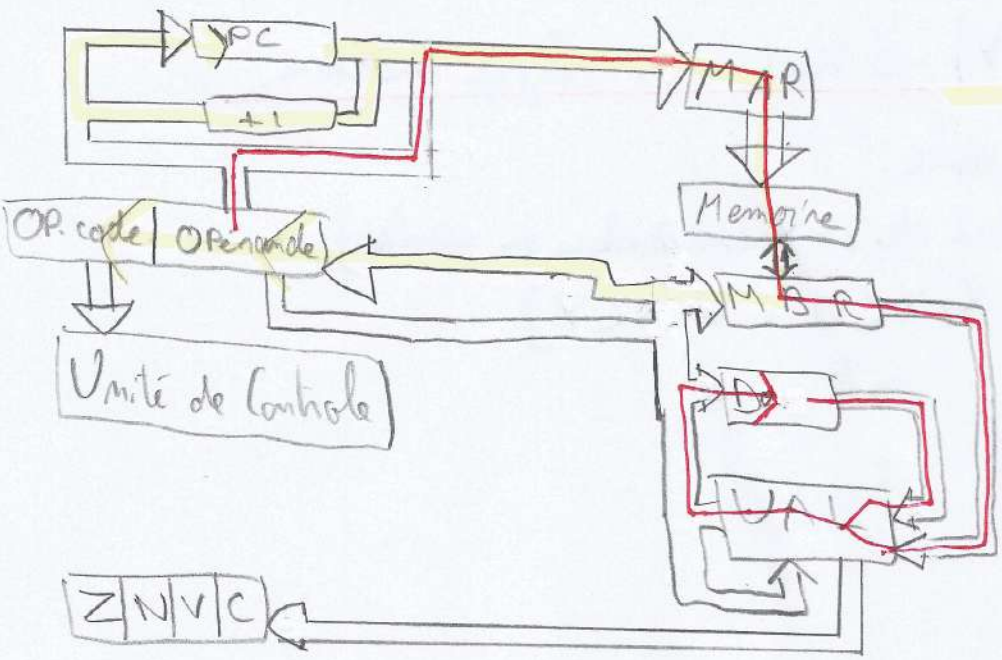
3) Un programme est une suite ordonnée d'instructions. Lors de l'exécution d'un programme :
- Certaines de ces instructions peuvent être exécutées plusieurs fois.

4) Une mémoire cache :
- a une taille inférieure à celle de la mémoire centrale.

5) Lors d'un cache miss :
- Le mot mémoire cherché est chargé dans le cache et envoyé au processeur.

II Processeur:

Schéma Proco + Mémoire centrale.



"add 1520"
↳ rechercher l'info

"add 1520"
↳ effectuer l'action.

Nommer et décrire le nom des 3 étapes qu'un processeur enchaine pour traiter une instruction

Fetch : Le processeur lit en mémoire la prochaine instruction à exécuter et la range dans le registre interne IR.

Decode : Le processeur décode l'instruction et établit les chemins de données que vont emprunter ses opérandes et l'opération (UAL) qui va éventuellement lui être attribuée.

Execute L'instruction est effectivement exécutée : les opérandes suivent les chemins qui ont été établis et l'opération est effectuée.

Définir ce qu'est le mode d'adressage d'une opérande d'instruction. Donner le nom et la signification de deux modes d'adressages courants

- Le mode d'adressage d'un opérande définit comment interpréter le champ OPERANDE, i.e. où trouver l'opérande final de l'instruction
  - Mode d'adressage immédiat : le champ OPERAnde contient l'opérande final
  - Mode d'adressage direct mémoire : Le champ Opérande contient l'adresse en mémoire de l'opérande final.
  - Mode d'adressage registre : Le champ opérande contient le n° de registre dans lequel se trouve l'opérande final.

Z = min (X, Y) → traduire en langage machine.

```
LDA | X      1 : espace.
CMP | Y      {calcul de X-Y sans stockage du résultat}
BRN | +3     {Si X-Y < 0 i.e. X < Y}
LDA | Y
STA | Z
```

7) Donner nbre accès mémoire par proco ... L lecture, * DE ADO

	FETCH		Decode		Execute	
	Accès	mot	Accès	mot	accès	mot
LDA X	1L	1	0	-	1L	D
CMP Y	1L	1	0	-	1L	D
BRN +3	1L	1	0	-	0	-
LDA Y	1L	1	0	-	1L	D
STA Z	1L	1	0	-	1L	D

* E écriture, I instruct°, D donnée

III Cache mémoire

Nommer et définir la propriété des programmes s'exécutant sur un calculateur sans laquelle un cache mémoire serait inutile. Expliquer ce qu'il se passerait dans un cache si elle n'était pas vérifiée.

Localité temporelle : Lorsqu'un programme accède à un mot mémoire, il est très probable qu'il y accède à nouveau très rapidement

Sans cette propriété, on chargerait dans le cache des mots mémoire qui ne seraient jamais réaccédés par la suite : le cache serait encombré de mots inutiles, il ne contiendrait jamais celui que l'on cherche, rendant le cache contre-productif.

Donner deux exemples (1 instruct°, 1 pour autre donnée) montrant intuitivement pq les programmes ont cette capacité.

Instructions : Les programmes comprennent de nbreuses boucles : après avoir été accédé lors de la 1ère itération, les instructions du corps de boucle seront très probablement accédées à nouveau très rapidement, car les boucles sont en général itérées ⊕ d'1 fois.

Données : Les variables locales d'une fct°, notamment les variables de boucle, sont très souvent accédées durant un bref laps de temps correspondant au temps d'exécut° de la fonction.

Localité Spatiale

Lorsqu'un programme accède à un mot mémoire, il est très probable qu'il accède également à des mots voisins très rapidement.

Lors d'un miss, on charge alors dans le cache non seulement le mot manquant mais aussi les mots voisins afin de minimiser le nbre de miss à venir.

Données : Les progs. contiennent souvent des tableaux ou chaines de caractères balayés du début à la fin. qd 1ère lue, on charge tout.

Instruct°s : Les programmes contiennent des séquences linéaires d'instruct°. qd on accède à la 1ère instruct° d'une séquence, les autres le seront aussi très rapidement.

C = tps d'accès à la mémoire cache.
m = _____ centrale
h = taux de hit du cache
t_m = temps moyen d'accès à la mémoire centrale perçu par le Proco.

Expliquer les cas trouvées pour h = 0 (0%) et h = 1 (100%)

Formule : $\boxed{t_m = hC + (1-h)(C+m) = C + (1-h)m}$

Pour h = 0 : $t_m = C + m$

h = 0 → tous les accès sont des miss, il faut d'abord accéder au cache pour s'en rendre compte (durée : C/2) puis accéder à la mémoire centrale pour rapatrier le mot cherché dans le cache (durée m) puis finalement le rapatrier dans le CPU (durée : C/2)

h = 1 : $t_m = C$

h = 1 → tous les accès sont des hit. Il suffit d'accéder au cache pour lire le mot cherché (durée : C). La mem. n'est jamais sollicitée.

h = 1 est un taux idéal qui ne peut jamais être atteint.

DE ADO

Expliquer pourquoi la politique de remplacement Least Recently Used (LRU) ne peut pas être appliquée dans un cache à correspondance directe.

Dans un cache à correspondance directe, la ligne dans laquelle ranger un nouveau mot mémoire suite à un miss est déterminée de manière univoque par l'adresse du mot. Le sont donc nécessairement les mots qui se trouvent dans cette ligne qui vont devoir être remplacés, même si ce ne sont pas les moins récemment utilisés.

Puisqu'on ne peut pas, par construct°, choisir la ligne remplacée/évincée, aucune politique de remplacement ne peut être mise en œuvre qu'elle soit LRU, FIFO ou autre.

Localité temporelle pour un site web + comment le navigateur en tire parti.

Il est probable qu'on accède très vite de nouveau à une page web à laquelle on vient d'accéder.

Navigateur va donc stocker la page Web sur le disque dur.
disque dur = cache des sites web.

Localité spatiale pour un site web + comment le navigateur en tire parti. + Pblème

Il est probable qu'on consulte les pages web référencées par celle où on se trouve actuellement.

Navigateur peut pré-charger les pages référencées lors du chargement/consultat° de la page en cours

Le préchargement conduirait à une engorgement inutile du réseau et des serveurs web, car localité spatiale est ☹ bien vérifiée que la temporelle pour les pages web.