

TD 1

Représentation et codages des nombres , Circuits

Exercice 1 : Arithmétique en complément en 2

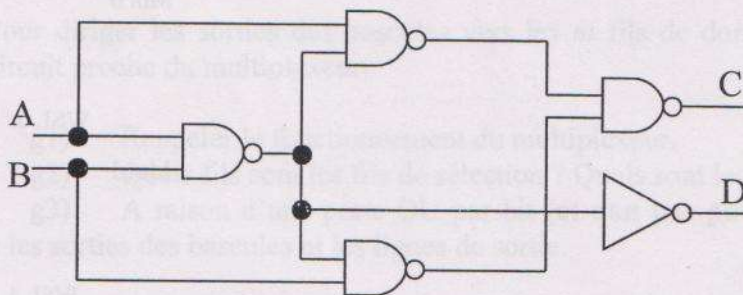
a) Rappeler le principe de l'arithmétique en complément à 2 sur n bits. Quel est l'intervalle des nombres représentables avec n bits ?

Pour cet exercice, $n = 5$, on travaille en complément à 2. Soient les nombres $X_1 = 01100$, $X_2 = 01101$, $X_3 = 10111$, $X_4 = 10100$.

- b) donner les valeurs décimales de ces 4 nombres
- c) effectuer les opérations **binaires** $X_2 - X_1$, $X_1 - X_2$, $X_1 + X_3$, $X_1 + X_2$, $X_3 + X_4$ en profitant de la représentation en complément à 2 pour la soustraction.
- d) Quelle explication donnez-vous aux résultats des deux dernières additions ?
- e) Soient 2 nombres binaires a et b et s leur somme. On nomme a_{n-1} , b_{n-1} et s_{n-1} les bits de signe de a , b et s . soit V la quantité binaire définie par $V = a_{n-1} \overline{b_{n-1}} \overline{s_{n-1}} + a_{n-1} b_{n-1} s_{n-1}$. Quelle est la valeur de V pour les 5 opérations ? Quelle est sa signification ?

Exercice 2 :

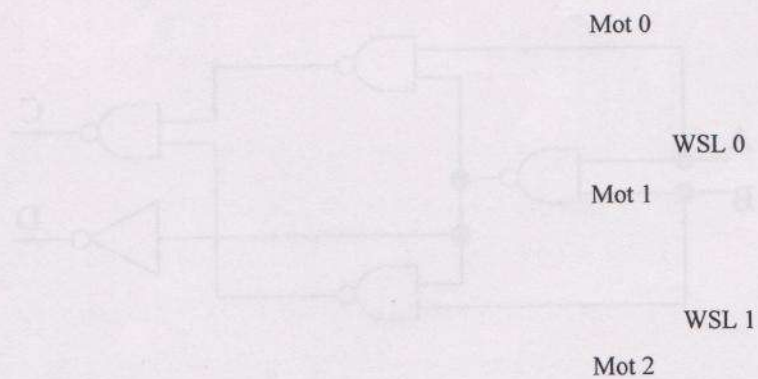
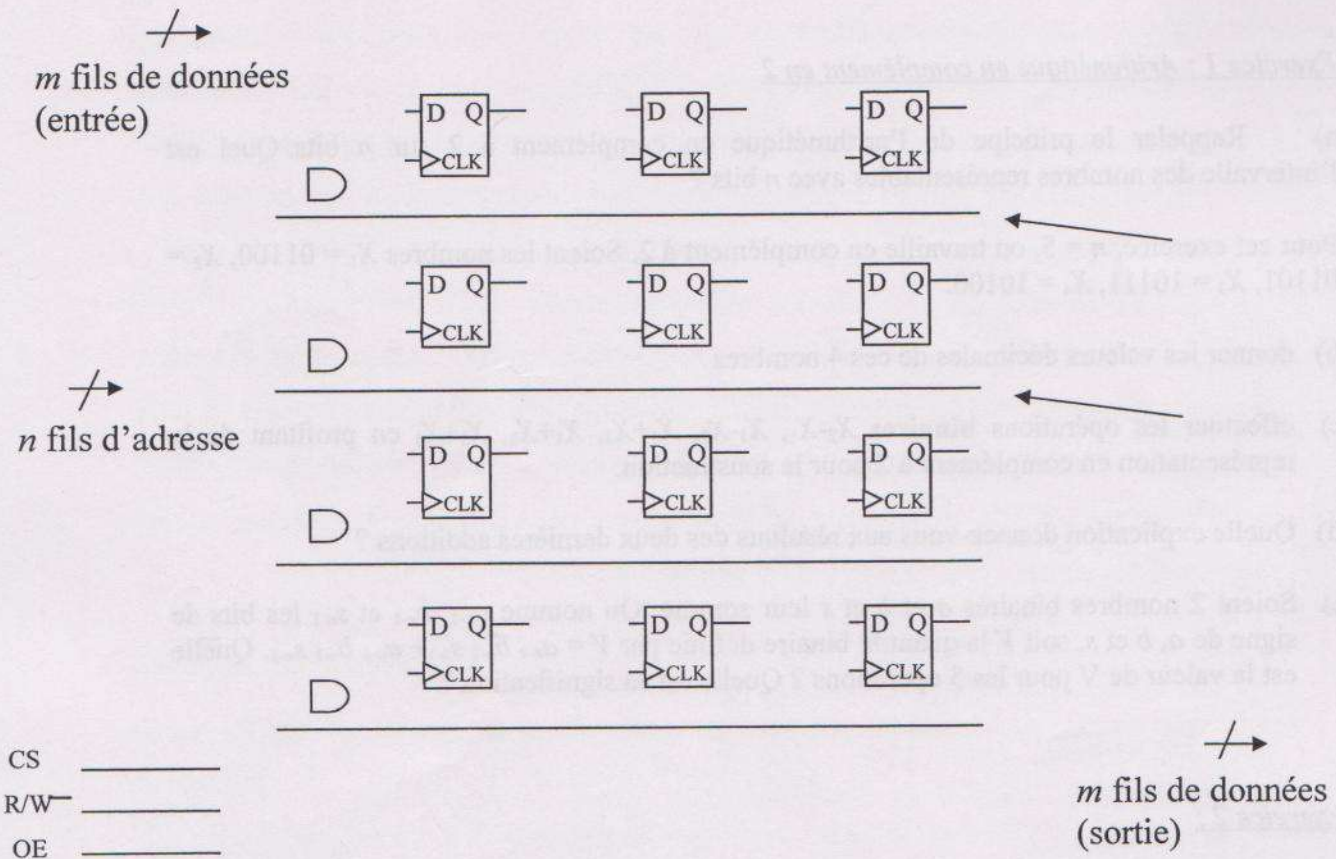
Déterminer la fonction logique réalisée par le système ci-dessous

Exercice 4 : Organisation interne d'une mémoire

Exemple sur une mémoire composée de 4 mots de 3 bits, à l'aide de bascules D. Chaque bascule comprend les broches D, Q et CLK.

- a) Combien de fils d'adresse sont nécessaires pour adresser les 4 mots ? Quel circuit logique utiliser pour cela ?

b) combien de fils de données sont nécessaires pour les 3 bits de chaque mot ?



- c) En considérant le schéma précédent, comportant n fils d'adressage, réaliser le circuit logique permettant l'adressage entre ces fils et les lignes de sélection de mot (WSL).
- d) Ce boîtier mémoire dispose de 3 fils de commande, délivrant les signaux suivants :

CS : Chip Select, sélection de circuit. Ce signal permet d'activer ce circuit parmi d'autres circuits mémoire éventuellement présents dans l'ordinateur. On considérera qu'il est toujours actif.

OE : Output Enable, activation de sortie. Ce signal permet, s'il est inactif, de mettre les m lignes de données en sortie en état de haute impédance, donc de les isoler du reste des circuits. Cela peut être utile pendant une opération d'écriture.

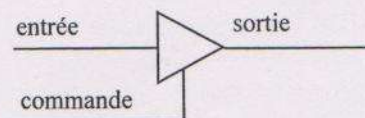
$R/\overline{W}$: Read / $\overline{\text{Write}}$, lecture au niveau 1, écriture au niveau 0.

On dispose en plus de chaque ligne de sélection de mot d'une porte ET pour valider l'écriture dans un mot particulier.

- d1) Quelle combinaison des signaux de commande pilote ces portes ?
- d2) Sur quelle entrée des bascules cette combinaison sera-t-elle appliqué ?
- d3) Matérialiser le chemin de ces signaux sur le schéma.

En fait, la porte d'écriture est aussi conditionnée par l'adresse du mot que l'on veut écrire.

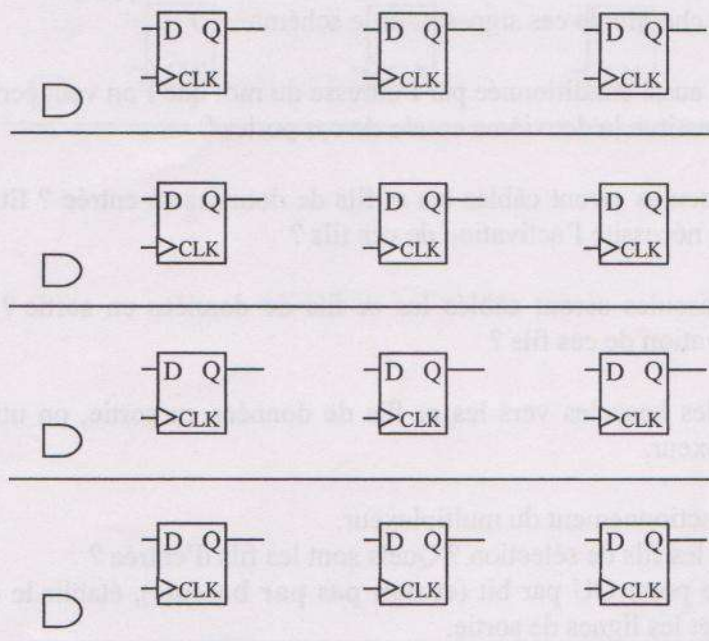
- d4) Quel signal constitue la deuxième entrée de ces portes ?
- e) Sur quelle broche des bascules seront câblés les m fils de données en entrée ? Etablir ce câblage. Quelle opération nécessite l'activation de ces fils ?
- f) Sur quelle broche des bascules seront câblés les m fils de données en sortie ? Quelle opération nécessite l'activation de ces fils ?
- g) Pour diriger les sorties des bascules vers les m fils de données en sortie, on utilise un circuit proche du multiplexeur.
 - g1) Rappeler le fonctionnement du multiplexeur.
 - g2) Quels fils sont les fils de sélection ? Quels sont les fils d'entrée ?
 - g3) A raison d'une porte OU par bit (**et non pas par bascule**), établir le câblage entre les sorties des bascules et les lignes de sortie.
- h) Pour votre information : câblage de la sortie. Les m lignes de sortie ne sont actives que si les trois signaux de commande CS, OE et $R/\overline{W}$ sont actifs. On utilise en fait un circuit élémentaire, qui est un buffer non inverseur dont voici le schéma :



Lorsque la commande est à l'état 0 (inactif), ce circuit agit comme un interrupteur ouvert et la sortie est comme déconnectée (haute impédance). Ce circuit agit de plus comme un amplificateur de puissance du signal.

i) Combien cette mémoire comporte-t-elle de bascules D ? de portes ET ? de portes OU ? En supposant une extension de cette mémoire à 1 ko (1024 mots de 8 bits), combien faut-il de lignes d'adresse ? de données ? de bascules D ? de portes ET ? de portes OU ?

Schéma à remplir (deuxième version)

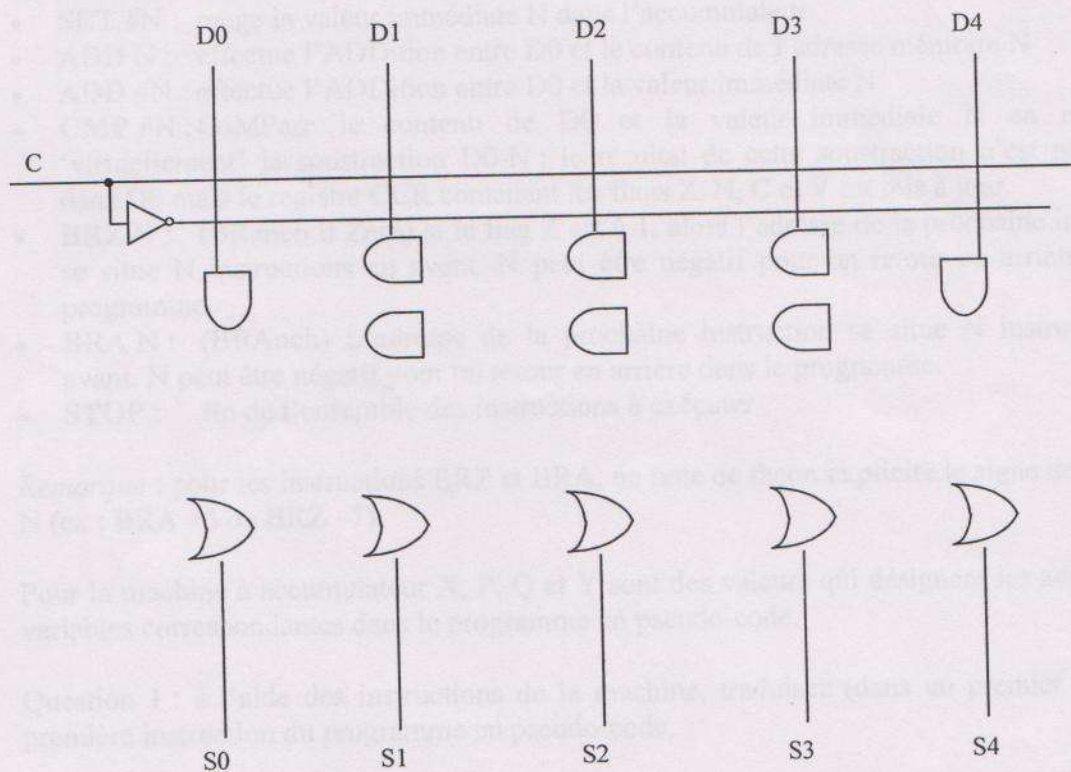


CS _____
 R/W _____
 OE _____

Exercice 5 : circuit à décalage, décalage logique et décalage arithmétique

Le décaleur est un système arithmétique de type MSI qui réalise un décalage de position de 1 bit, à droite ou à gauche, sur les n bits d'un mot d'information. Soit un circuit comprenant 5 entrées (D0 à D4) et 5 sorties (S0 à S4). On dispose d'un signal de commande C indiquant le sens du décalage. C=0 pour la gauche, C=1 pour la droite.

- En représentation non signée, quelle est l'opération arithmétique réalisée par un décalage à gauche (on fera des hypothèses sur la valeur de D0) ? par un décalage à droite ?
- Comment pourrait-on traiter le cas où D0 = 1 pour le décalage à gauche ?
- Etablir le câblage de ce circuit à partir de l'ébauche fournie ci-dessous.
- On passe en représentation signée. En cas de décalage à droite, qu'advient-il du bit de signe d'après le schéma ? comment traiter ce cas ? comment câbler cette solution ?
- Nous venons de voir les spécificités des opérations de décalage dites arithmétiques. A votre avis, de quelle manière (matérielle ou logicielle) sont réalisés les traitements particuliers des bits de signe et d'extension ? Justifiez votre réponse.



TD 2

Circuits, Chemin de données

Exercice 1 : chemin de données, registres, instructions.

Soit le programme suivant écrit en pseudo-code :

```
X := P+Q;  
Si X=4      Alors Y = 6;  
            Sinon Y = Q+2;
```

On suppose aussi que la machine à accumulateur du cours (dont le schéma est donné ci-dessous) possède les instructions suivantes :

- **STA N**: (STore Adress) range le contenu de l'accumulateur D0 à l'adresse mémoire de valeur N.
- **LDA N**: (Load from Address) range le contenu de l'adresse mémoire N dans D0.
- **SET #N**: range la valeur immédiate N dans l'accumulateur
- **ADD N**: effectue l'ADDition entre D0 et le contenu de l'adresse mémoire N
- **ADD #N**: effectue l'ADDition entre D0 et la valeur immédiate N
- **CMP #N**: CoMPare le contenu de D0 et la valeur immédiate N en effectuant 'virtuellement' la soustraction D0-N : le résultat de cette soustraction n'est pas stocké dans D0 mais le registre CCR contenant les flags Z, N, C et V est mis à jour.
- **BRZ N**: (BRanch if Zero) si le flag Z est à 1, alors l'adresse de la prochaine instruction se situe N instructions en avant. N peut être négatif pour un retour en arrière dans le programme.
- **BRA N**: (BRAnch) L'adresse de la prochaine instruction se situe N instructions en avant. N peut être négatif pour un retour en arrière dans le programme.
- **STOP**: fin de l'ensemble des instructions à exécuter

Remarque : pour les instructions BRZ et BRA, on note de façon **explicite** le signe de la valeur N (ex : BRA +3 ou BRZ -7).

Pour la machine à accumulateur X, P, Q et Y sont des valeurs qui désignent les adresses des variables correspondantes dans le programme en pseudo-code.

Question 1 : à l'aide des instructions de la machine, traduisez (dans un premier temps) la première instruction du programme en pseudo-code.

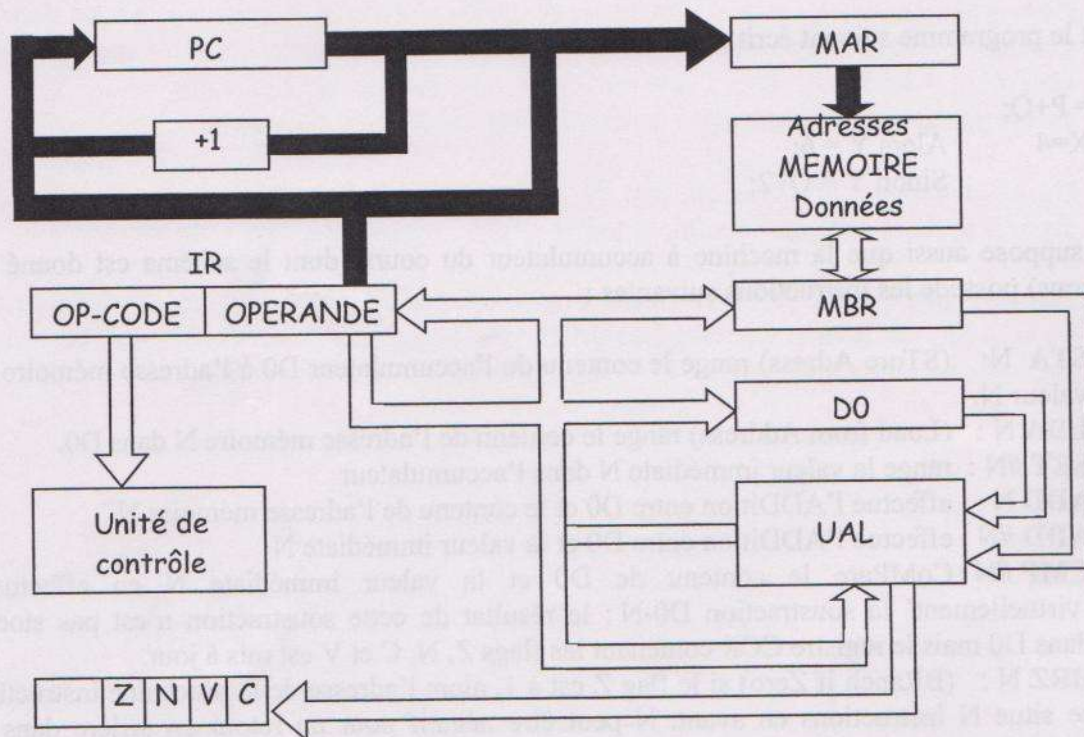
Question 2 : traduisez l'ensemble du programme en pseudo-code.

Question 3 : rappelez le chemin de données utilisé lors de la phase Fetch . Est-il nécessaire de le faire pour toutes les instructions de la machine ? argumentez votre réponse.

Question 4 : détaillez le chemin de données pour l'ensemble des instructions de la machine correspondant aux lignes suivantes du programme en pseudo-code :

X := P+Q ; Si X=4 ; Y := Q+2 ;

Question 5 : Indiquez les modes d'adressage (accès aux valeurs d'après les opérandes) utilisés dans votre programme.



TD 3

Réalisation du chemin des données d'une machine RISC

On cherche à réaliser une machine RISC très simple, pour cela, on doit d'abord déterminer le chemin des données de la même manière que pour la petite machine CISC vue en cours

Le processeur que l'on veut schématiser possède 4 registres de données D0 à D3, et il s'agit d'une machine à chargement/rangement.

Question 1) *rappelez le principe d'une machine à chargement/rangement.*

Voici le jeu d'instructions sommaire dont on veut disposer pour débiter : les instructions ont en général 2 opérandes, et lorsqu'on lit une instruction, celle-ci se présente systématiquement sous la forme : CODE, opérande source, opérande destination. Par exemple, ADD opérande1, opérande2 effectuera : $\text{opérande2} = \text{opérande1} + \text{opérande2}$; MOVE opérande1, opérande2 effectuera l'affectation $\text{opérande2} = \text{opérande1}$, etc...

Question 2) *indiquez tous les types possible pour les opérandes des différents types d'instructions listés ci-dessous :*

Instruction de transfert (dont le nom est MOVE)

Ex : MOVE opérande1, opérande2

Instructions arithmétiques et logiques ADD, SUB, MUL, DIV, OR, AND, XOR, NOT

Ex : ADD opérande1, opérande2

Instruction de comparaison (dont le nom est CMP)

Ex : CMP opérande1, opérande2

L'instruction de comparaison agit de la manière suivante : pour savoir si deux valeurs sont égales, on peut effectuer la soustraction $\text{opérande2} = \text{opérande1} - \text{opérande2}$: si le résultat est nul, c'est que les deux valeurs sont égales, et cela sera signalé par le flag Z, qui sera positionné à 1. Cependant, l'opérande2 est modifiée par le calcul. L'instruction CMP effectue la soustraction sans affecter le résultat à l'opérande2, mais modifie tout de même les flags.

Instructions de sauts (BRA, BRZ, BRN, BRC)

BRA : BRanch Always : branchement inconditionnel

BRZ : BRanch on Zero : branchement si le flag Z est à 1

BRN : BRanch on Negative : branchement si le flag N est à 1

BRC : BRanch on Carry : branchement si le flag C (retenue) est à 1

Ex : BRA opérandel

Question 3) *Pendant le traitement d'une instruction par un processeur, lors de quelles phases peut-il y avoir des accès à la mémoire ?*

Lorsque l'on utilise une machine RISC, il faut séparer les différents accès mémoire intervenant lors de phases distinctes. Ainsi en phase FETCH, on sait que l'on accède forcément à une instruction qui doit être amenée dans le registre IR du processeur : on considère alors que l'on accède à une mémoire contenant seulement les instructions.

Question 4) *Comment pouvez-vous schématiser, le plus simplement possible, le chemin des données de la phase FETCH, en utilisant les registres PC, IR ainsi que la mémoire contenant les instructions ?*

Question 5) En reprenant les types d'opérande listés dans la question 2, indiquez quelles sont les valeurs qui peuvent être présentes à l'entrée de l'UAL. *Schématisez ce chemin des données en matérialisant les 4 registres individuellement ainsi que l'UAL.*

Question 6) toujours en reprenant la question 2, d'où peuvent provenir les valeurs que l'on peut stocker dans des registres de données ? *Complétez le chemin des données correspondant.*

Question 7) Les accès en mémoire sont possibles en phase EXECUTE. *Indiquez à quelle occasion ces accès ont lieu. On considère dans ce cas que ces accès se font dans une partie de la mémoire différente de celle où sont stockées les instructions : une petite mémoire de données. Ajoutez cette mémoire sur le schéma et indiquez le chemin des données relatif à cette mémoire de données.*

Question 8) Gestion du registre PC : on utilise, pour la gestion du registre PC, un simple circuit additionneur à matérialiser sur le schéma. *Quelles valeurs peuvent se trouver en entrée de ce circuit additionneur ?* complétez le schéma : vous avez une petite machine RISC !

Question 9) Utilité de l'unité de contrôle : Sans entrer dans le détail du fonctionnement de cette unité, on veut décrire son rôle dans la gestion du chemin des données. Considérons toutes les versions possibles de l'instruction de transfert MOVE : *pour chacune de ces versions, indiquez où doivent être dirigées les valeurs contenues dans les champs opérande1 et opérande2 de ces instructions. Ces chemins sont-ils bien distincts ?* Sachant que l'unité de contrôle est un séquenceur câblé, qui ne se base que sur la partie OPCODE de l'instruction pour diriger les différentes valeurs dans le microprocesseur, *combien de codes différents sont nécessaires pour les différentes versions de l'instruction de transfert ?*

Au total, combien de codes différents faut-il pour toutes les versions des instructions listées à la question 2 ? En déduire le nombre de bits du champ OPCODE d'une l'instruction.

Question 10) En allant un peu plus dans le détail, on peut décrire le fonctionnement d'une instruction en indiquant le rôle exact des opérandes. Prenons l'exemple de l'instruction : MOVE adresse, registre : on peut la décrire de la façon suivante :

L'instruction, pointée par le registre PC, est lue dans la mémoire d'instructions et transférée dans le registre IR.

La partie code est une version de l'instruction de transfert, elle est dirigée vers l'unité de contrôle (le séquenceur) pour organiser le chemin des données.

La première opérande est une adresse, elle est dirigée vers un circuit décodeur pour indiquer où se trouve effectivement la donnée en mémoire, une opération de lecture de la mémoire de données est faite (c'est l'unité de contrôle qui gère cela).

La deuxième opérande est le numéro d'un registre, qui sera la destination de l'opération : un circuit décodeur permet de sélectionner effectivement, parmi les 4, celui avec lequel aura lieu l'opération. une opération d'écriture dans ce registre est effectuée.

L'UAL n'est pas sollicitée par cette instruction, les flags ne sont pas mis à jour.

Décrire de cette manière les trois instructions suivantes :

ADD #302, D3 : (addition d'une valeur immédiate)

CMP D2,D4

BRN -24

Question 11) On vous demande d'écrire un petit programme en assembleur, utilisant les instructions de cette machine, pour trouver la valeur maximale parmi toutes les valeurs en mémoire situées entre l'adresse 1000 et l'adresse 1100. Si vous essayez d'écrire ce programme, à quelle difficulté vous trouvez-vous confronté ? Quel mode d'adressage serait utile pour pouvoir traiter ce genre de problèmes ? Quelle partie du chemin de données est en cause ? Proposez une modification du schéma obtenu pour permettre l'utilisation de ce mode d'adressage. En supposant que l'on dispose maintenant de ce mode d'adressage, choisissez une syntaxe pour ce type d'opérande et écrivez le programme demandé.

TD 4 Architecture générale Processeurs / Mémoire

Exercice 1 : gros boutisme et petit boutisme

Considérons deux ordinateurs possédant des mots mémoire de 32 bits. L'un adopte la représentation gros-boutiste, l'autre la représentation petit-boutiste.

- a) *rappelez l'organisation des octets dans un mot mémoire avec chacune de ces deux représentations.*
- b) On suppose que ces machines stockent les informations suivantes pour un enregistrement d'une base de données :
- Nom : « Luc Duval », qui est une chaîne de caractères, chaque caractère occupant un octet en mémoire ;
 - Age : 31, qui est un entier stocké dans un mot mémoire ;
 - Numéro de service : 260, qui est aussi un entier (pour info $260 = 1 \times 256 + 4$) stocké dans un mot mémoire.

Donnez les représentations en mémoire de ces données pour chacune des deux machines (pour info : 20 octets, soit 5 mots mémoires, sont nécessaires).

- c) On effectue un transfert de ces données depuis la machine gros-boutiste vers la machine petit-boutiste. Si ce transfert est effectué octet par octet (de l'octet 0 à l'octet 19), *comment sont interprétées les valeurs reçues par la machine petit-boutiste ?*
- d) Puisque la numérotation des octets est inversée entre les deux représentations, on décide, après le transfert, d'inverser l'ordre des octets à l'intérieur d'un mot mémoire. *Quel est le résultat obtenu ?*
- e) *Quelle solution pouvez-vous proposer pour obtenir un transfert correct des données ?*

Exercice 2 : gestion de cache mémoire

Soit un processeur comportant un cache primaire de données (on ne se soucie pas des instructions) de 16 ko. La taille d'un mot mémoire est toujours de 32 bits.

- a) pour se dérouiller les neurones : combien de mots mémoire peuvent être stockés dans ce cache ?

La politique de gestion des transferts entre le cache et la mémoire est la suivante : si le processeur ne trouve pas une donnée dans le cache, il la rapatrie depuis la mémoire vers le cache à la place de la donnée du cache la moins récemment utilisée (algorithme LRU). Un programmeur décide d'utiliser un tableau de N entiers (chaque entier est représenté par un mot de 16 bits), et de lui appliquer une série de P traitements linéaires, depuis le début jusqu'à la fin du tableau (en clair : P boucles *for i allant de 1 à N*).

- b) le temps c d'accès au cache est de 8 ns, le temps m d'accès à la mémoire est de 40 ns. On considère que les temps de traitement des données et d'écriture dans le cache sont négligeables. *Exprimer en fonction de c , m , N et P le temps moyen d'exécution du programme, sachant qu'au départ, le cache ne contient aucune donnée du tableau.* Pour info, N a une très grande influence sur ce temps. Applications numériques :

$$N = 5\,000, P = 10 ; \text{ puis } N = 10\,000, P = 10.$$

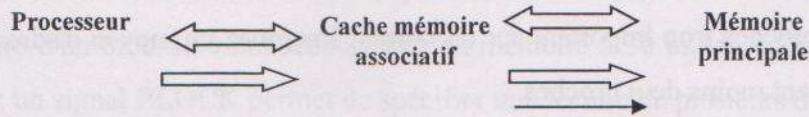
On ajoute un cache de données secondaire de 256 ko, de temps d'accès $c' = 14$ ns. La politique de gestion de ce cache est la même que celle du cache primaire. Si le processeur ne trouve pas la donnée dans le cache primaire, il regarde dans le cache secondaire. Si la donnée s'y trouve, il la recopie dans le cache primaire selon la méthode LRU. Si elle ne s'y trouve pas, le processeur interroge la mémoire principale et recopie cette donnée dans les deux caches selon la méthode LRU.

Exprimer en fonction de c , c' , m , N et P le temps moyen d'exécution du programme, sachant qu'au départ, le cache ne contient aucune donnée du tableau. Recalculer les applications numériques précédentes ainsi que la valeur de ce temps pour $N = 200\,000$, $P = 10$.

- c) calcul moyen de temps d'accès : Le concepteur de l'ordinateur cherche à calculer les temps d'accès à une donnée avec les configurations à 1 puis à 2 caches pour préparer sa plaquette marketing. Pour ce faire, il se place dans le cas où les accès sont faits de manière totalement aléatoire dans la mémoire. *La machine comportant une mémoire principale de 1 Mo, quel est le temps d'accès moyen pour la configuration à 1 cache ? à 2 caches ? Calculez les temps théoriques d'accès aux caches permettant dans ce cas d'améliorer le temps d'accès moyen à la mémoire.*
- d) *Commentez ces résultats en regard des différents éléments vus en cours.*

Exercice 3 : Types de cache

a) A votre avis, quels signaux transitent sur le schéma suivant ?



- b) Quelles sont alors les informations que doit contenir le cache ? En déduire une organisation du cache et la méthode de recherche de données associée. A votre avis, quel est le principal inconvénient de cette solution ?
- c) Une autre organisation du cache consiste à faire en sorte qu'une adresse demandée par le processeur ne puisse être située qu'à un endroit du cache. Comment réaliser cette organisation ? Avec cette solution, plusieurs adresses peuvent être situées au même endroit du cache. Comment les différencier ?
- d) En utilisant des portes logiques et des comparateurs, dressez un schéma de ce type d'organisation.

TD 5 : Cache mémoire / Bus et gestion de bus

Exercice 1 : adressage de mémoire

La capacité des circuits mémoire est en augmentation constante. Le nombre de broches nécessaire à l'adressage des mots mémoire suit cette augmentation, et il est souvent gênant d'avoir un nombre de broches trop important sur un boîtier. Imaginez un moyen d'adresser 2^n mots mémoire en utilisant moins de n broches.

Exercice 2 : cache mémoire

A partir des schémas de cache associatif et de cache à correspondance directe vus en cours, faites le schéma complet d'un cache partiellement associatif par 4 (c'est-à-dire comportant 4 caches à correspondance directe). Chaque cache à correspondance directe ne contient qu'une donnée (pas de ligne de cache) à chaque emplacement, et contient 2^8 emplacements.

Exercice 3 : performance de bus

Le bus PC/AT, fonctionnant à une fréquence de 10 MHz, nécessite 4 cycles pour lire un mot mémoire.

- 1) Est-ce un bus synchrone ou asynchrone ? justifiez votre réponse.
- 2) La taille d'un mot mémoire étant de 16 bits, calculez la bande passante de ce bus en bits/s et en octets/s.

Exercice 4 : débit de bus

- 1) Calculez la bande passante de bus nécessaire à un écran VGA couleur (640x480) pour afficher une vidéo à 50 et 60 images/seconde. Chaque pixel se voit attribuer une couleur parmi une palette fixe de 16 couleurs choisies parmi 262 144 possibles.
- 2) Soit un moniteur connectable par le bus USB sur un ordinateur. Le bus USB a un débit de 1,5 Mo/s. Avec un taux de rafraîchissement de 25 images/seconde et des images en noir et blanc, quelle résolution pourrait-on obtenir sur ce moniteur ?

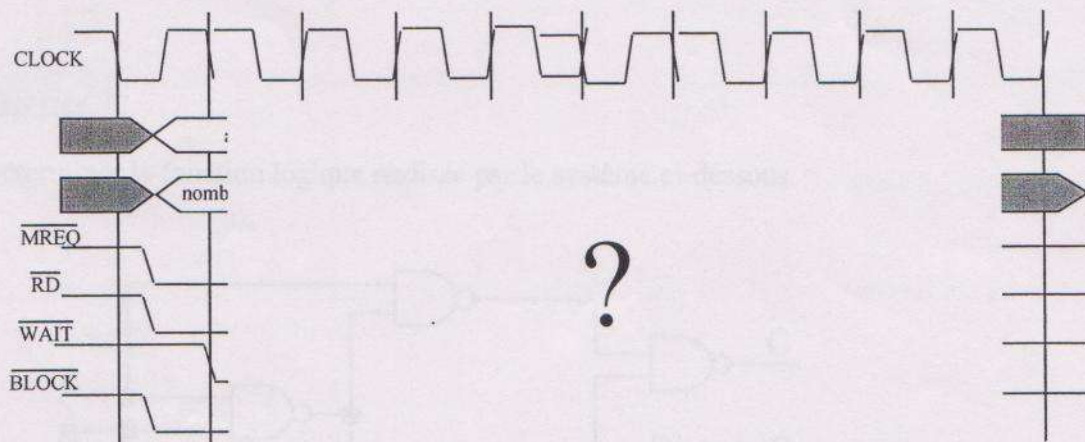
Exercice 5 : chronogramme de bus

Dressez les chronogrammes de bus pour les échanges suivants, en n'omettant aucun signal :

- Lecture mémoire via un bus asynchrone;
- Ecriture mémoire à 60 ns via un bus synchrone à 40 MHz;
- Lecture d'un bloc de 4 données 32 bits en mémoire à 30 ns via un bus synchrone à 100 MHz; un signal BLOCK permet de spécifier une lecture de plusieurs données à la suite : il est positionné par le maître pour signaler à l'esclave que la requête concerne un bloc de données. Le nombre de données à lire est indiqué par le maître via le bus de données.

Lorsque la première donnée est mise sur le bus par la mémoire, le maître désactive ses signaux de commande, chaque donnée est ensuite positionnée sur le bus de données par la mémoire à raison d'une donnée par cycle. On considère que le nombre de données demandé est de 4.

Voici tous les signaux pour le début de ce chronogramme :



- Ecriture en mémoire via un bus asynchrone multiplexé (les adresses et données partagent les mêmes lignes, un signal A/D indique si les adresses ($A/D=1$) sont présentes ou les données ($A/D=0$) sont présentes sur le bus.