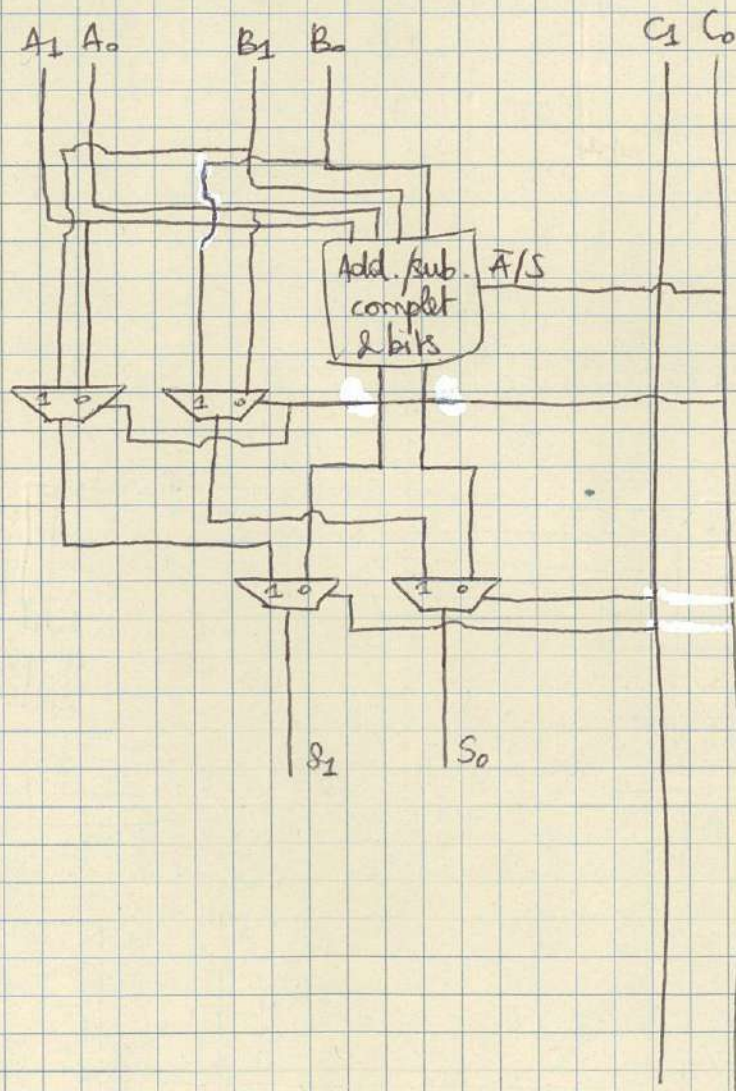


Si $\bar{A}/S = 1$, c'est à-dire que l'on souhaite soustraire A et B, on prend le complément de B, auquel on ajoute 1. On l'ajoute ensuite à A.

3. Si $C_1 = 0$, on effectue les opérations d'addition ou soustraction en fonction de C_0 ($\bar{A}/S = C_0$).

Si $C_1 = 1$, on laisse passer le premier opérande pour $C_0 = 0$ et le deuxième pour $C_0 = 1$.



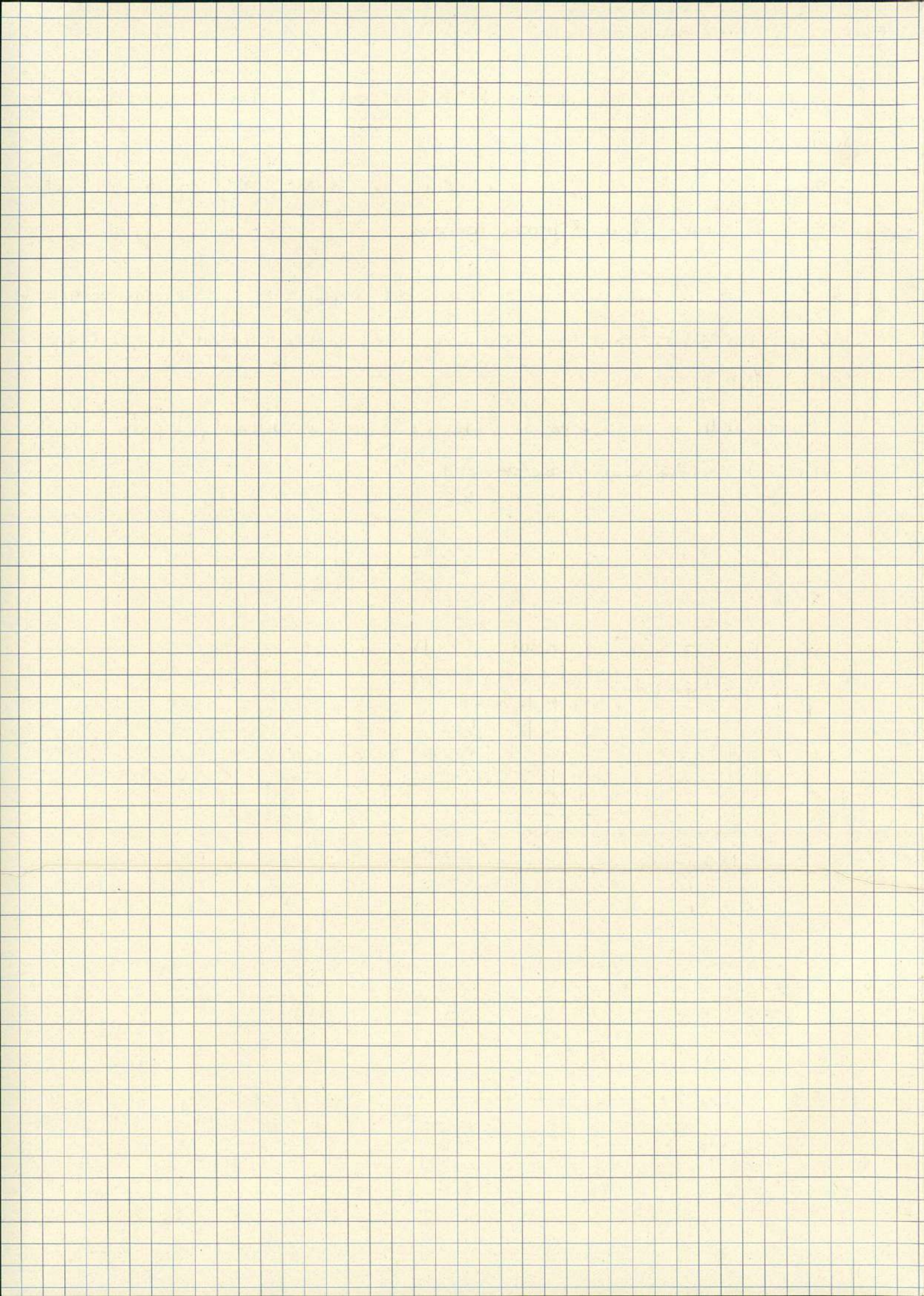
Exercice 2 - Mémoire centrale (1)

1. Chaque barrette a une capacité de 128 Mo soit 2^{27} octets, avec des mots de 8 bits.

Puisqu'il y a 2^{27} mots, il y a ~~28~~ broches adresse, et puisqu'un mot est constitué de 8 bits, il a 8 broches données.

2. On veut, avec les barrettes vues précédemment, implémenter une mémoire centrale de 256 mots de 8 bits. On a donc besoin de deux barrettes : un bit pour le Chip Select suffit.

Le bus adresse sera donc de 29 bits (28 pour les adresses, 1 pour le CS), et le bus de données sera encore de 8 bits.



NOM COMBETTE

Prénom Elise

Promo 2018 LL A

Date 03/06/2015

MATIÈRE DE d'ADO

Exercice 4 - Processeur

5

1.	Nombre d'accès mém.	Informations concernées
Fetch	1 lecture.	Instructions
Decode	0	Aucun.
Execute	1 lecture ou 1 écriture, ou rien.	Données

2. les instructions ADD, SUB, MUL, DIV et CMP mettent à jour le CCR, tandis que BRZ le consulte.

3. 0 LDA #1.

1 STA 1002.

2 LDA 1000.

3 CMP #1.

4 BRZ 12.

5 LDA 1002.

6 MUL 1000.

7 STA 1002.

8 LDA 1000.

9 SUB #1.

10 STA 1000.

11 BRA 3.

12 STOP.

Exercice 3 - Mémoire cache

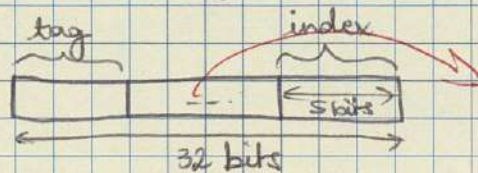
(35)

4. Les caches associatifs sont ^{plus} moins efficaces que les caches à correspondance directe mais sont également moins coûteux; en effet l'associativité permet d'utiliser moins de comparateurs: le cache est moins performant mais possède moins de composants onéreux. *Cartreuse*

5. On organise un cache en lignes de mots pour se servir de la propriété de localité spatiale d'un programme: par exemple si l'on souhaite utiliser un tableau, dont les "cases" sont donc contiguës les unes aux autres, on utilisera alors moins de lignes de cache et on accèdera plusieurs fois à la même ligne, permettant de faire un hit et gagner du temps d'accès mémoire.

1. Le cache est composé de 8 entrées, c'est-à-dire 8 lignes de 4 mots de 2 octets chacun. Sa capacité est donc de $8 \times 4 \times 2 = 64$ octets.

2. Le cache interprète une adresse mémoire émise par le processeur grâce au tag et à l'index de cette dernière. Ici le tag n'a pas d'utilité puisque l'on a un seul bloc.



Quant à l'index, on a besoin de 2 bits pour le choix du mot ($4 = 2^2$ mots par ligne) et 3 bits pour le choix de la ligne ($8 = 2^3$ entrées): il est donc de 5 bits.

Le tag sert à choisir le bloc dans le cache, l'index sert à choisir le mot dans l'entrée.

