Examen NSY104

Architectures des systèmes informatiques
Session octobre 2013 - durée : 2h30
Documents papiers autorisés

Exercice 1 : (3 points)

1) Supposons qu'un cache possède un temps d'accès (latence de cache-hit) de 10 ns et un taux de miss de 7 %. Une modification apportée au cache ferait baisser son taux de miss a 3 % mais ferait monter la latence du cache-hit a 18 ns . Dans quelles conditions cette modification pourrait-elle offrir de meilleures performances (temps d'accès moyen a la mémoire plus court) ?

Exercice 2 : (5 points)

On donne un code assembleur pour le micro-processeur LC-3, qui ajoute 5 aux 100 premiers éléments d'un tableau A.

`R0`	compteur (=100 au debut, 0 a la fin)
`R1`	adresse case tableau
`R2`	valeur case tableau
`tableauA`	le tableau
`NbCases`	nombre de cases

```
          LEA R1, tableauA
          LD R0,NbCases
boucle:   LDR R2,R1,0
          ADD R2,R2,5
          STR R2,R1,0
          ADD R1,R1,1
          ADD R0,R0,-1
          BRp boucle
          NOP
```

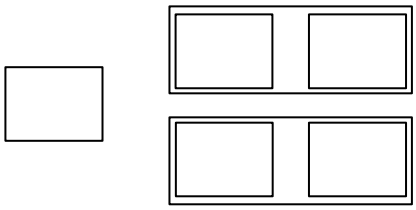
1) En utilisant un pipeline de 5 étages, donnez le diagramme d'exécution de votre code en identifiant les aléas structurels, de données et de contrôle.

2) Optimisez le programme assembleur et proposez des modifications matérielles simples pour limiter ces aléas.
3) Donnez le diagramme d'exécution après ces optimisations.

LEA	*Load Effective Address :* charge dans l'un des registres généraux la somme du compteur de programme et d'un offset codé dans l'instruction.
LD	Charge dans l'un des registres généraux le mot mémoire à l'adresse égale à la somme du compteur de programme et d'un offset codé dans l'instruction.
LDR	Charge dans l'un des registres généraux le mot mémoire à l'adresse égale à la somme du registre de base et d'un offset codé dans l'instruction.
ADD	Permet de faire l'addition de deux valeurs 16 bits.
STR	Range le contenu du registre à l'adresse égale à la somme du registre de base et d'un offset codé dans l'instruction.
NOP	*No operation*.
Brp	*Branch if positive* : si l'accumulateur est >= 0, met le PC à la valeur indiquée.

Exercice : (12 poits)

1) Complétez le schéma fonctionnel suivant en identifiant les composants et en complétant par les connexions nécessaires. De quel type d'architecture s'agit-il ?

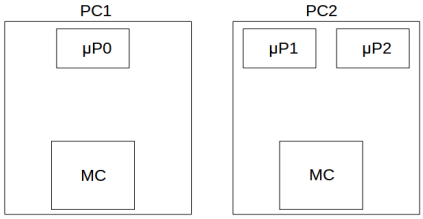


2) En plus des composants que vous avez dans la question précédente, vous disposez également de trois mémoires cache ayant les caractéristiques suivantes :

	Capacité	Coût/bit	Temps d'accès (T_{acc})	Taux d'échec ($Taux_{ech}$)	Pénalité d'échec (T_{ech})
C1	2^{16}	0,04	10ns	0,999	30ns
C2	2^{8}	0,1	1ns	0,980	3ns
C3	2^{10}	0,08	5	0,800	15ns

On vous demande de réaliser une mise à niveau de votre architecture tout en garantissant le meilleur rapport performance/prix. Quelle(s) modification(s) apporteriez-vous en utilisant les caches disponibles ?

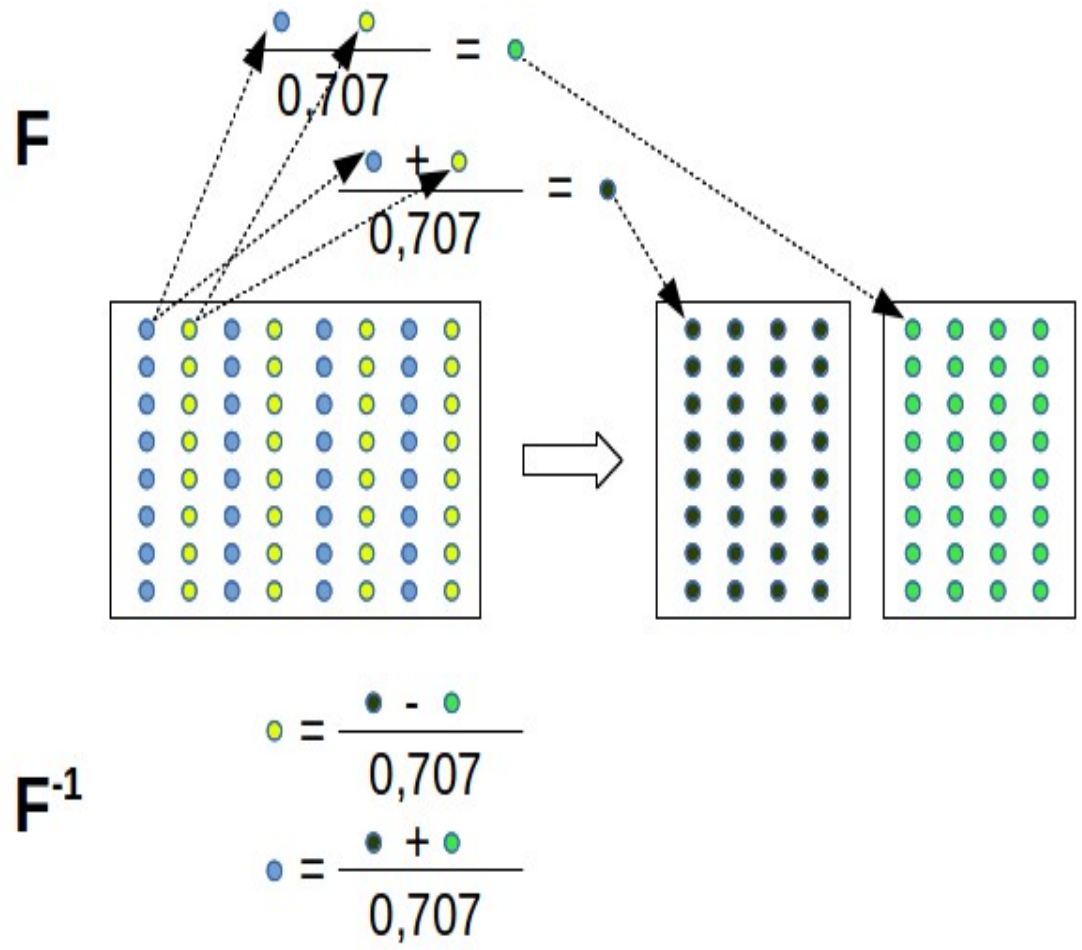
3) Pour monter en performance de calcul, vous disposez désormais des deux machines décrites par le schéma ci-dessous :



Votre mission consiste à ajouter le/les caches qu'il faut (à choisir parmi le tableau de la deuxième question) pour atteindre les meilleures performances sur un *cluster* formé de PC1 et PC2. Que proposez-vous comme solution ?

4) Vous disposez désormais de deux machine identiques PC1 et PC2. Vous allez proposer une solution logicielle efficace pour faire du traitement d'images. Plus précisément, pour une image X 3D (matrice réelle de taille 128 x 128 x 10), vous allez calculer la transformé en ondelettes (F) et la transformée en ondelettes inverse (F^{-1}). Autrement dit, après avoir appliqué la transformée F et son inverse F^{-1}, vous devez retrouver exactement la matrice X : $X = F^{-1}(F(X))$.

Comme transformée F, on considérera la transformée de Haar 2D expliquée dans le schéma ci-dessous :



Autrement dit, pour effectuer la transformation F, on commence par séparer les colonnes paires et impaires. Ensuite, pour une ligne donnée l, on calcule un coefficient d'approximation (a) et un coefficient de détail (d) de la façon suivante :

```
a[k,l] = (image[2k,l] + image[2k+1,l]) / 0,707
d[k,l] = (image[2k,l] - image[2k+1,l]) / 0,707
```

Pour effectuer la transformation F^{-1}, on calcule les coefficients suivants :

```
image[2k,l] = (a[k,l] +d[k,l]) / 0,707
image[2k+1,l] = (a[k,l] - d[k,l]) / 0,707
```

On va vérifier si $X = F^{-1}(F(X))$. Pour cela, on va effectuer la transformation 2D et son inverse pour chaque coupe 2D de X (matrice 128x128).

Proposez un modèle de programmation parallèle efficace pour résoudre ce problème. Tous les choix doivent être justifiés.

5) Implémentez votre solution.