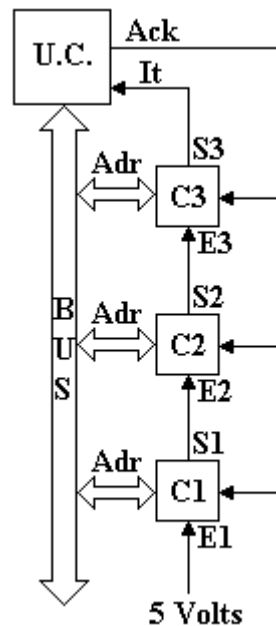


**Exercice I: (06 points)**

Une deuxième forme de dispositif daisy chain servant à la détection et de reconnaissance de la cause d'interruption est décrit dans le schéma de la figure suivante.



Le signal **It** avertit l'UC qu'un signal d'interruption survient, émis par l'un des trois contrôleurs. L'UC dispose d'une bascule **BI** d'interruption. **BI** est une bascule de type JK. Contrairement au Daisy chain vu en cours, le signal **Ack** est envoyé en parallèle à tous les contrôleurs.

**Questions:**

- 1- Représenter sur un schéma, dans le cas du système décrit plus haut, les entrées de la bascule **BI** d'interruption de l'UC.
- 2- Supposons que les contrôleurs C2 et C3 ont émis simultanément un signal d'It. Comment l'UC peut-elle, dans ce cas, recevoir **uniquement le numéro de l'It la plus prioritaire**?
- 3- Donner le schéma interne détaillé du système d'IT **au niveau de la cellule C2**.

**Exercice II (6 points):**

La MC d'un ordinateur est de 1Méga mots. On veut pouvoir exécuter sur cette machine des programmes stockés en mémoire auxiliaire et pouvant atteindre 16 Méga mots. On appellera cet espace la mémoire virtuelle. On suppose que l'UC ne peut exécuter qu'un seul programme à la fois. On découpe pour cela logiquement la MC en parties de tailles égales appelées pages. Pour pouvoir être exécutés, les programmes sont aussi découpés logiquement en pages de taille égale à celle des pages de la MC. Une instruction est référencée par son adresse en mémoire virtuelle (numéro de la page en mémoire virtuelle et déplacement dans cette page) (voir figure suivante).

Numéro de la page en MV	Déplacement dans la page
23	9 8 0

On appellera cette adresse l'adresse virtuelle de l'instruction. Le chargement du programme de la mémoire auxiliaire vers la MC se fait par page. Au départ, la mémoire est remplie avec les premières pages du programme. Si, pendant l'exécution du programme, l'UC ne trouve pas la prochaine instruction à exécuter en MC, elle libère une page en MC. L'UC procède ensuite au chargement dans la page qui vient d'être libérée - à partir de la mémoire auxiliaire - de la page contenant l'instruction à exécuter. On remarquera que les pages du programme ne sont alors plus disposées séquentiellement en MC.

Pour retrouver en MC l'instruction à exécuter, l'UC a besoin de l'adresse réelle en MC de cette instruction. Elle utilise pour cela un mécanisme de conversion des adresses virtuelles en adresses réelles. Ce mécanisme est formé d'une table de pages (TP) associée au programme en cours d'exécution. Chaque élément de TP correspond à une page du programme et est formé de plusieurs champs.

### **Questions:**

- 1- Trouver la **taille de la page**, et le **nombre de pages de la MC et du programme**.
- 2- On veut utiliser une mémoire associative pour accélérer la conversion des adresses. Représenter sur un schéma les différents champs de cette mémoire en précisant la taille de cette mémoire.
- 3- Indiquer sur un schéma le principe de calcul de l'adresse réelle d'une instruction.

### **Exercice III (8 points):**

On désire réaliser une mémoire vive de 64 **Mega Octets** divisée en 2 **modules** entrelacés avec un **degré d'entrelacement D = 4** (l'entrelacement se fait à l'intérieur de chaque module). Cette mémoire est réalisée avec des **boîtiers de 2 Mega Octets**.

On désire réaliser la logique de génération des signaux de sélection des boîtiers qui composent la mémoire précédente par un circuit unique (circuit qui génère automatiquement en sortie tous les signaux  $\overline{CS}$  et qui a en entrée les bits nécessaires du bus d'adresses global).

Pour vous aider, vous commencerez par donner un numéro à tous les boîtiers mémoire en associant à chaque boîtier un numéro unique.

### **Questions:**

- 1- Représenter la table de vérité du circuit qui génère les signaux  $\overline{CS}$  des boîtiers de la mémoire vive **en indiquant clairement les entrées et les sorties**.
- 2- Représenter le schéma du deuxième module du circuit. Représenter le circuit de la logique de génération des signaux  $\overline{CS}$  en utilisant seulement un circuit ROM et **aucun autre circuit** (représenter clairement le contenu de la ROM, ses entrées et ses sorties).

---

**\*\*\* Bonne chance \*\*\***