

N7_SN_1A

Architecture des ordinateurs - Semestre 5

BE-TP4 - Circuits séquentiels - Algorithmes cablés

L'objectif est d'implanter l'algorithme de tri par sélection dans un circuit logique.

Vous aurez le choix entre :

1- Tri décroissant : le plus moins difficile

noté sur 8 points.

Pour I de **1 à N** Faire

Calculer Max et Indice_Max de Tab($I..N$)

Echanger Tab(I) et Tab(Indice_Max)

Fin Pour

1B- Peut être amélioré en évitant le dernier passage dans la boucle (I de 1 à $N-1$) : **+0,5**

2- Tri croissant en utilisant le Min,

noté sur 9 points.

Pour I de **1 à N-1** Faire

Calculer Min et Indice_Min de Tab($I..N$)

Echanger Tab(I) et Tab(Indice_Min)

Fin Pour

Cela nécessite de réaliser un circuit Min qui sera une simple adaptation du circuit Max réalisé en TP3.

3- Tri croissant en utilisant le Max,

noté sur 10 points.

Pour I de **N à 2, pas=-1**, Faire

Calculer Max et Indice_Max de Tab($1..I$)

Echanger Tab(I) et Tab(Indice_Max)

Fin Pour

Cela nécessite de réaliser un module supplémentaire pour gérer la boucle avec pas=-1

Vous **avez le choix** entre les algorithmes indiqués dessus, et vous devrez effectuer un **travail préparatoire avant de venir en TP4** (sinon vous n'aurez pas le temps de finir).

Vous adopterez la même démarche qu'en TD3/TP3.

1- Graphes d'états, et module etats_tri_tab

- **A faire avant de venir en TP4**

2 points

- **Temps estimé : 1 heure**

Adaptez (détaillez sur le modèle de celui de cal_max) l'algorithme choisi en sachant que l'on peut accéder qu'à un seul mot mémoire à un instant donné.

Dessinez le graphe d'états de l'algorithme de tri choisi. Veillez à choisir des **noms significatifs**, car cela vous facilitera l'implantation et la relecture de vos modules.

Vous **devez déposer votre travail** préparatoire dans votre espace shdl.fr dans un fichier (module) portant le nom etats_tri_tab.

- **Algorithme adapté** sur le modèle du calcul du max :

```

Ad_courante <- Ad1
Max <- Tab(Ad_courante)
Ad_Max <- Ad_courante
Ad_courante <- Ad_courante + 1
Tant Que Ad_Couante <= Ad2 Faire
    Si Tab(Ad_courante) > Max Alors
        Max <- Tab(Ad_Courante)
        Ad_Max <- Ad_courante
    Fin Si
    Ad_courante <- Ad_courante + 1
Fin TantQue

```

- **Graphe d'états** représenté sous la forme suivante (exemple : etats_cal_max) :

Transition	Condition	opérations
* -> INITAD	/cal	aucune
INITAD -> INITMAX	cal	adCour <- ad1
INITMAX -> MAJMAX	cal	max <- Tab(ad1) adMax <- ad1
MAJMAX -> MAJMAX	cal*/adCour_sup_ad2	Si max > Tab(adCour) max <- Tab(adCour) adMax <- adCour adCour <- adCour + 1
MAJMAX -> FINI	cal*adCour_sup_ad2	Si max > Tab(adCour) max <- Tab(adCour) adMax <- adCour
FINI -> FINI	cal	aucune

Votre enseignant vous fera un retour en début de séance du TP4.

Réalisez le module etats_tri_tab et testez le au début de la séance du TP4.

2 points

2- Module tri_tab

Réalisez le module tri_tab qui effectue le tri d'un tableau (stocké dans une RAM interne au module, entre les adresses ad1 et ad2)

```

module tri_tab (rst, clk, cal, ad1[7..0], ad2[7..0] :
etats[N-1..0],      // les N états fournis par etats_tri_tab
calMax,             // le signal donnant l'ordre du calcul du max (calMin pour algo2)
adCour[7..0]        // adresse courante
etats_max[3..0],    // les 4 états de cal_max (etats_min et cal_min pour algo2)
                    // on peut écrire : etats_max[3..0] = FINMAX & MAJMAX & INITMAX & INITAD
max[31..0],         // la valeur du max courant (min pour algo2)
adMax[7..0],        // l'adresse du max courant (adMin pour algo2)
adRam[7..0],        // l'adresse à l'entrée de la RAM
inRam[31..0],       // la donnée à l'entrée de la RAM
wRam,              // le signal d'écriture dans la RAM
outRam[31..0])      // la donnée à la sortie de la RAM

```

Pour pouvoir vérifier le résultat dans la RAM, on fera de sorte de pouvoir lire le contenu de l'adresse entrée sur ad1 sur la sortie outRam lorsque cal = 0.

Ce qui revient écrire :

```
adRam[7..0] = cal* ...           // opérations sur la RAM lors du tri
              +/cal*ad1[7..0]    // pour lire la ram quand cal=0
```

A- Pour éviter des difficultés de mise au point, vous **devez réaliser un module intermédiaire** qui implante le séquençement et la gestion de l'adresse adCour, sans calcul du max (ou min) et sans RAM :

```
module tri_tab_int (rst, clk, cal, ad1[7..0], ad2[7..0], maxOK : etats[X..0], calMax, adCour[7..0])
// minOK et calMin pour algo2
// Remplacer X par la bonne valeur
```

Le signal maxOK (minOK) permet d'indiquer (manuellement dans un premier temps) que le calcul du max (min) du sous-tableau courant est terminé.

Testez ce module selon le scénario suivant :

- Mettre l'entrée rst à 1 puis à 0
- Mettre ad2 à 00000100 et toutes autres entrées à 0
- Cliquez sur clk plusieurs fois et vérifiez que rien ne se passe (car cal=0)
- Mettre cal à 1 et cliquez sur clk plusieurs fois : le circuit reste dans l'état de calcul du max (min), et calMax (calMin) reste à 1. Vérifiez que adCour=0 (adCour=00000100 pour l'algo 3)
- Mettre maxOK à 1 (minOK) et cliquez sur clk jusqu'au retour à l'état de calcul du max (min) et calMax=1 (calMin). Vérifiez que adCour a été incrémenté (décrémenté pour l'algo 3)
- Laissez maxOK à 1 (minOK) et cliquez sur clk jusqu'au retour à l'état de calcul du max (min) et calMax=1 (calMin). Vérifiez que adCour a été incrémenté (décrémenté pour l'algo 3)
- Répéter l'opération précédente jusqu'à adCour=ad2 (adCour=ad1 pour l'algo 3). Vérifiez que l'état final du tri a été atteint et que plus rien n'évolue même en cliquant sur clk.

Bloquez ce module.

2 à 3 points

B- Réalisez le module tri_tab en :

- reprenant une copie du code précédent
- y ajoutant le circuit cal_max (ou cal_min), une RAM, et tout ce qu'il faut pour réaliser l'opération : Echanger Tab(I) et Tab(Indice_Max). Bien identifier :
 - les cas où wrRam = 1
 - les termes de l'équation de adRam (les cas d'accès à la Ram en lecture ou écriture)
 - les termes de l'équation de inRam (ce qu'on écrit dans la RAM)

Bloquez ce module.

2 à 3 points

Vous pouvez poursuivre votre travail dans un autre module tri_tab_suite, qu'il faudra **bloquer au plus tard le samedi de la semaine de ce TP à 23 heures.**