
ARCHITECTURE DES ORDINATEURS

Devoir Surveillé 2

1 heure 20 minutes
sans documents

- N.B.** : - Travaillez calmement. Ne bâclez pas vos réponses : il vaut mieux traiter correctement moins de questions que tout faire de travers.
- Les réponses aux questions doivent être argumentées et aussi concises que possible.
- Le barème est donné à titre indicatif.

Question 1

(12 points)

On considère l'algorithme suivant, qui réalise la multiplication de deux nombres entiers non signés a et b :

```
unsigned int  a, b, p;  
a = ...;  
b = ...;  
p = 0;  
while (a != 0) {  
    if ((a % 2) != 0)  
        p += b;  
    a = a >> 1;  
    b = b << 1;  
}
```

(1.1) (1 point)

Sur votre copie, exécutez « à la main » cet algorithme, avec les valeurs initiales $a = 5$ et $b = 3$. Pour ce faire, représentez dans un tableau l'évolution des valeurs des variables a , b et p à l'entrée dans la boucle et au début de chaque nouveau tour de boucle, jusqu'à la fin de celle-ci.

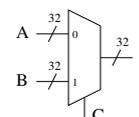
On veut implanter cet algorithme au sein d'un circuit numérique synchrone, c'est-à-dire piloté par une horloge, susceptible d'être implanté au sein d'un processeur CISC à mots de 32 bits. Pour cela, vous allez devoir construire et assembler les briques logiques nécessaires.

(1.2) (1 point)

Écrivez l'équation logique d'un multiplexeur à deux entrées A et B sur un bit, et un bit de contrôle C , qui renvoie sur sa sortie S la valeur A si $C = 0$ et B si $C = 1$. Dessinez le schéma de câblage de ce multiplexeur, au moyen de portes logiques.

En vous appuyant sur le schéma précédent, concevez et dessinez le schéma de câblage d'un multiplexeur à un bit de contrôle C , mais prenant des entrées A et B sur 32 bits.

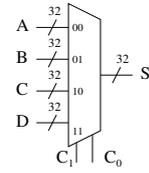
À partir de maintenant, et même si vous n'avez pas répondu aux questions précédentes, vous pouvez représenter un multiplexeur 32×1 sous la forme du circuit ci-à côté, comprenant deux entrées A et B sur 32 bits, une entrée C sur un bit et une sortie S sur 32 bits.



(1.3) (1 point)

En vous appuyant sur le circuit précédent, dessinez le schéma de câblage d'un circuit multiplexeur à quatre entrées A, B, C et D et deux bits de contrôle C_1C_0 .

À partir de maintenant, et même si vous n'avez pas répondu aux questions précédentes, vous pouvez représenter un multiplexeur 32×2 sous la forme du circuit ci-à côté, comprenant quatre entrées A, B, C et D sur 32 bits, deux entrées C_0 et C_1 sur un bit chacune et une sortie S sur 32 bits.

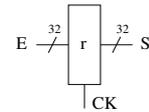


(1.4) (1 point)

Dessinez le schéma de câblage d'un circuit qui, à partir de bascules D, crée un registre permettant de mémoriser un mot de 32 bits lors d'un front montant d'horloge.

N.B. : On ne demande pas de dessiner le câblage interne d'une bascule D.

À partir de maintenant, et même si vous n'avez pas répondu aux questions précédentes, vous pouvez représenter un registre nommé « r » sous la forme du circuit ci-joint, comprenant une entrée E sur 32 bits, une entrée CK et une sortie S sur 32 bits.

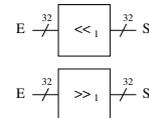


(1.5) (1 point)

Dessinez le câblage d'un circuit décaleur non signé à un bit vers la droite, qui prend en entrée un mot de 32 bits $E = e_{31}e_{30} \dots e_1e_0$ et renvoie en sortie un mot de 32 bits $S = s_{31}s_{30} \dots s_1s_0$, tel que la valeur de S soit la valeur de E décalée d'un bit vers la droite.

Faites de même pour un circuit décaleur non signé à un bit vers la gauche.

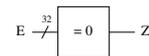
À partir de maintenant, et même si vous n'avez pas répondu aux questions précédentes, vous pouvez représenter des circuits décaleurs à un bit vers la gauche et vers la droite sous la forme des circuits ci-à côté, comprenant une entrée E sur 32 bits et une sortie S sur 32 bits.



(1.6) (1 point)

Construisez un circuit de test de nullité, prenant une valeur d'entrée $E = e_{31}e_{30} \dots e_1e_0$ sur 32 bits et muni d'une sortie Z sur un bit. $Z = 1$ si $E = 0$ et $Z = 0$ si $E \neq 0$.

À partir de maintenant, et même si vous n'avez pas répondu aux questions précédentes, vous pouvez représenter le circuit testeur de nullité sous la forme du circuit ci-à côté, comprenant une entrée E sur 32 bits et une sortie Z sur 1 bit.



(1.7) (2 points)

En vous inspirant des circuits précédents, construisez un circuit synchrone piloté par un fil d'horloge CK et un fil de contrôle C, de telle sorte que :

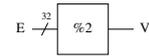
- si $C = 1$, le registre du circuit est chargé avec la valeur sur 32 bits fournie sur l'entrée A du circuit ;
- si $C = 0$, la valeur du registre est décalée à droite d'un bit à chaque top d'horloge, et lisible sur la sortie S du circuit ;
- lorsque la valeur de sortie $S = 0$, le fil de sortie T du circuit vaut 1, sinon $T = 0$.

N.B. : Mettez en œuvre une solution qui ne crée pas de dérive de l'horloge !

(1.8) (1 point)

Comment réaliser en pratique, de façon simple, le test « $(a \% 2) \neq 0$ » de l'algorithme, permettant d'ajouter ou non à p la valeur de b ? Pour ce faire, supposez que la variable a est câblée sous la forme d'une valeur d'entrée $E = e_{31}e_{30} \dots e_1e_0$.

À partir de maintenant, et même si vous n'avez pas répondu aux questions précédentes, vous pouvez représenter le circuit testeur de parité sous la forme du circuit ci-à côté, comprenant une entrée E sur 32 bits et une sortie V sur 1 bit qui vaut 1 si le résultat du test est vrai.



(1.9) (2 points)

En vous inspirant des circuits précédents, ainsi que d'un circuit additionneur sur 32 bits mis à votre disposition, réalisez un circuit synchrone piloté par un fil d'horloge CK et un fil de contrôle C , tel que :

- si $C = 1$, les registres a et b du circuit sont chargés avec les valeurs d'entrée A et B fournies en entrée, et le registre p est chargé avec la valeur 0 ;
- si $C = 0$, le circuit modifie les valeurs de a , b et p à l'image de l'exécution d'un tour de boucle de l'algorithme présenté en début de problème ;
- lorsque le calcul est terminé, le fil de sortie T du circuit passe à 1.

N.B. : Mettez en œuvre une solution qui ne crée pas de dérive de l'horloge !

(1.10) (1 point)

Modifiez le circuit précédent afin que, tant qu'un calcul n'est pas terminé, il ne soit pas possible d'insérer de nouvelles valeurs dans le circuit.

Question 2 (4 points)

Écrivez le code $\gamma 86$ d'un programme qui implémente la fonction C de prototype suivant :

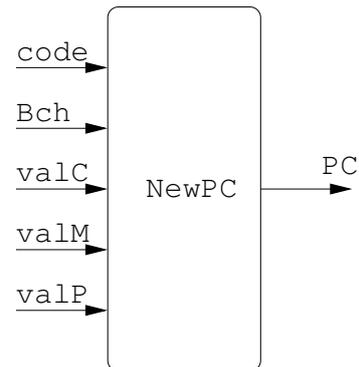
```
void swap (int * p, int * q);
```

qui effectue l'échange des deux valeurs pointées par p et q . Ainsi, si à l'entrée de la fonction on avait $*p = x$ et $*q = y$, alors au retour de la fonction on doit avoir $*p = y$ et $*q = x$.

N.B. : Veillez au bon usage des registres que vous employez !

Question 3 (4 points)

La figure ci-contre montre les entrées du circuit combinatoire `NewPC` chargé de calculer l'adresse de la prochaine instruction que le processeur $\gamma 86$ séquentiel devra charger (l'entrée `Bch` signifie « *Branch Flag* »). Le code HCL tronqué décrivant ce circuit est donné sous le schéma.



(3.1) (1 point)

Expliquez le code HCL fourni.

(3.2) (1 point)

Quelles sont les instructions nécessitant un traitement spécifique dans `NewPC` ?

(3.3) (2 points)

Complétez ce code pour traiter les quelques cas non gérés actuellement. Pour chaque ligne ajoutée, expliquez précisément le choix de la valeur résultat.

```
PC = [
  code == Jxx && Bch : valC ;
  ...
  1 : valP ;
]
```