Examen Architecture des Ordinateurs – 6 Janvier 2020

Tous documents autorisés. Calculatrices interdites. Durée 2h

Pour toutes les questions, une explication **concise** est nécessaire pour que la réponse soit prise en compte.

1. Fonctions booléennes et circuits combinatoires [10 pts]

On cherche à construire un circuit effectuant la multiplication de deux nombres entiers positifs codés sur 2 bits.

Q1. Regardons d'abord le cas simple de deux nombres codés sur 1 bit chacun.

- Sur combien de bit(s) il faut coder le résultat afin de ne pas avoir de dépassement
- Donner l'expression algébrique du résultat

Q2. Regardons maintenant le cas de deux nombres codés sur 2 bits chacun. Justifier pourquoi il faut 4 bits pour contenir le résultat afin de ne pas avoir d'overflow (impossibilité de représenter le résultat)

Q3. Écrire la table de vérité, les deux nombres sur deux bits en entrée seront notés a_1a_0 et b_1b_0 ; et les bits de sortie $c_3c_2c_1c_0$.

Pour calculer les expressions algébriques des différents bits de résultat, on utilisera la façon « classique » de faire une multiplication :

$$\begin{array}{r} \\ \\ \times \\ \hline \cdot d \cdot d \\ a \cdot c \cdot c \\ \hline \end{array}$$

Q4. Donner une expression pour c_0 en vous aidant éventuellement de la question **Q1**.

Q5. Ensuite, calculer l'expression de c_1 à partir de la somme (ou logique) de deux termes (produit de variables d'entrée).

Q6. Calculer c_2 et c_3 en faisant attention à bien prendre en compte les retenues éventuelles.

Q7. En supposant que le temps de propagation d'une porte logique (ET,OR,XOR,AND, NOT) soit de τ (quelque-soit le nombre d'entrée de la porte). Quel est le temps de propagation de la multiplication entre les entrées et les sorties.

On veut construire un indicateur p nous indiquant si le résultat de la multiplication va être pair ou non en fonction des variables d'entrée a1,a0,b1,b0.

Q8. Donner la forme conjonctive normale (FCN) de p.

Q9. A l'aide d'une table de Karnaugh, donner une expression réduite de cet indicateur p (la table de Karnaugh est demandée).

Q10. En utilisant un multiplexeur 8 vers 1, implémenter la fonction correspondant à l'indicateur p. Vous indiquerez comment les différentes entrées du multiplexeur (les entrées, et les variables de sélection) sont câblées à l'aide des entrées a1,a0,b1,b0.

2. Circuits séquentiels et Bascules [6 pts]

On souhaite réaliser un compteur-décompteur automatique par 8. Il part de 0, compte jusqu'à 7 et décompte jusqu'à 0 et puis recommence :

$0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 7 \rightarrow 6 \rightarrow \dots \rightarrow 0 \rightarrow \dots \rightarrow 7 \rightarrow \dots$

Q11. On commence par établir le fonctionnement d'un compteur-décompteur par 8 à l'aide d'une variable E. On sera en mode compteur quand E=0, et en mode décompteur quand E=1. Donner le nombre de bascules nécessaire et donner la table de vérité.

On cherche maintenant à faire changer l'état de E lorsque l'on **arrive** sur l'état 0 ou l'état 7 à l'aide d'une quatrième bascule Qs.

Q12. On veut définir une variable A, qui sera à 1 lorsque l'on se trouve dans l'état 6 en mode compteur ou dans l'état 1 en mode décompteur (sinon A sera à 0). Donner l'expression de A en fonction de Q2, Q1, Q0 et E.

Q13. On veut maintenant trouver le câblage Ds de la quatrième bascule afin que la bascule change d'état ($0 \rightarrow 1$ ou $1 \rightarrow 0$) lorsque l'on passe de l'état 6 à l'état 7 en mode compteur ou lorsque l'on passe de l'état 1 à l'état 0 en mode décompteur. Donner l'expression de Ds en fonction de Qs et de A afin de finaliser le circuit final.

Q14. Comment câbler la variable E afin de réaliser la bascule ?

3. Pipeline [4 pts]

Soit le jeu d'instructions suivant : (on dispose des registres entiers R0 à R31 et des registres flottants F0 à F31).

<u>Instructions sur les flottants :</u>	
lf Fdest, adresse	Charge un flottant depuis l'adresse (l'adresse est au format N(R) avec N un immédiat et R un registre entier
sf Fsrc, adresse	Stocker un flottant dans l'adresse
fadd Fdest, Fsrc1, Fsrc2	$F_{dest} \leftarrow F_{src1} + F_{src2}$ (addition flottante simple précision)
fsub Fdest, Fsrc1, Fsrc2	$F_{dest} \leftarrow F_{src1} - F_{src2}$ (soustraction flottante simple précision)
fmul Fdest, Fsrc1, Fsrc2	$F_{dest} \leftarrow F_{src1} * F_{src2}$ (multiplication flottante simple précision)
fdiv Fdest, Fsrc1, Fsrc2	

	$F_{dest} \leftarrow F_{src1} / F_{src2}$ (division flottante simple précision)
<u>Instructions sur les entiers :</u>	
lw Rdest, adresse	Charge un entier depuis l'adresse
sf Rsrc, adresse	Stocke un entier dans l'adresse
add Rdest, Rsrc1, Rsrc2	$R_{dest} \leftarrow R_{src1} + R_{src2}$ (addition avec des entiers)
addi Rdest, Rsrc1, Imm	$R_{dest} \leftarrow R_{src1} + Imm$ (addition avec des entiers)
<u>Branchements :</u>	
bne Ra, Rb, Label	Si $Ra \neq Rb$: goto Label
bneq R, label	Si $R \neq 0$: goto Label

Les additions, soustractions et multiplications flottantes sont pipelinées. Une nouvelle instruction peut démarrer à chaque cycle. Les latences sont de 2 cycles pour **lf** et fournies par les résultats de l'exercice 1 pour les multiplications et additions flottantes. La division a une latence de 15 cycles. Elle n'est pas pipelinée : une nouvelle division ne peut commencer que lorsque la division précédente est terminée. On utilisera les latences suivantes :

<u>Instructions</u>	<u>Latence</u>	<u>Pipelinée ?</u>
lf	2	oui
fadd, fsub	4	oui
fmul	6	oui
fdiv	15	non

On considère la code suivante :

```
int N=100
float X[N], Y[N], a;
for (int i=0 ; i<N ; i++)
    Y[i]= X[i] * a;
```

On supposera que la variable 'a' est déjà initialisée dans le registre F31. Le tableau X est implémenté à l'adresse 0x10000000 et Y à 0x10000400. La variable R1 est initialisée à 0x10000000.

Q15. Ecrire le code assembleur du coeur de la boucle sans aucun déroulage en optimisant le nombre de cycles. Vous ferez attention à spécifier pour chaque instruction sur quel cycle elle démarre. Combien de cycles par itérations obtient-on ?

Q16. Effectuer un déroulage d'ordre 2, toujours en optimisant le nombre de cycles et en spécifiant le cycle sur lequel démarra chaque instruction. Combien de cycles par itérations obtient-on ?

Q17. Combien de cycles par itération obtient-on si au lieu d'une multiplication on a une division ? Que faire pour améliorer la situation ?