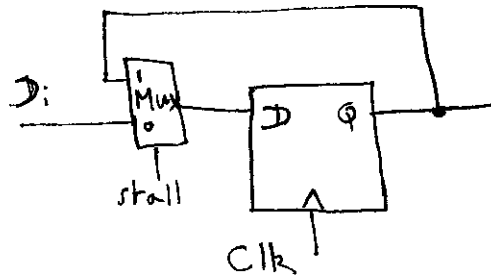
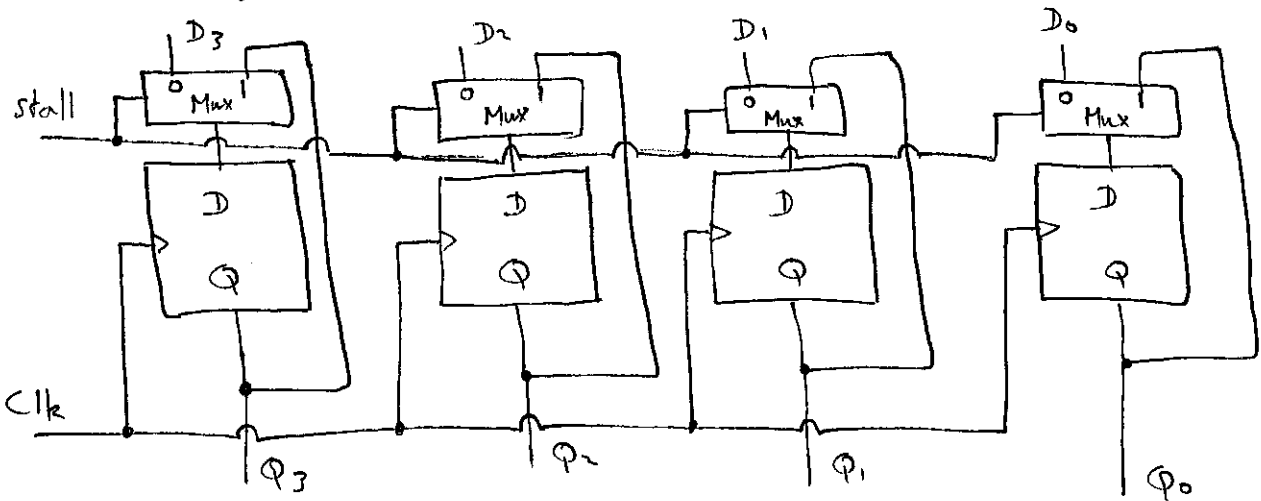


Exercice 1

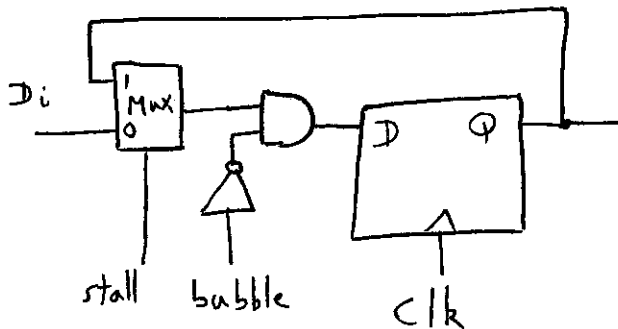
(1.1) Afin de ne pas entraîner de dérive de l'horloge, il ne faut pas rajouter de portes au signal d'horloge initial. Il faut donc re-mémoriser à chaque fois la même valeur en cas de "stall", au moyen d'un multiplexeur:



Cela donne le circuit suivant :



(1.2) Injecter une bulle revient à placer une donnée toujours à zéro, au lieu des autres données. Le circuit élémentaire est donc :



Il était également possible de mettre la porte AND au niveau de l'entrée "0" du multiplexeur (mais dans ce cas, il faut $stall=0$ quand $bubble=1$).

Exercice 2

②

(2.1) " Lorsque code est égal à Jxx (instruction de branchement) et que Bch (décision de branchement) est vrai, alors l'adresse de la prochaine instruction à exécuter sera valC (constante immédiate de l'instruction). Sinon (condition "1" toujours vraie), ce sera valP (adresse de l'instruction immédiatement suivante). "

(2.2) Ce sont celles qui occasionnent un déroutement :

- Jxx (déjà vue)
- CALL : valeur immédiate
- RET : valeur lue en mémoire dans la pile

(2.3) PC = [

code == Jxx && Bch : val C;

code == CALL : val C;

code == RET : val M;

1 : val P;

]

valeur immédiate, comme Jxx
adresse de retour lue à l'étage M

Exercice 3

Pour qu'il soit économiquement intéressant de "pipe-liner" une fonction logique, il faut :

- 1) Que le calcul de cette fonction se répète dans le temps;
- 2) Que la fonction puisse être découpée en sous-fonctions qui s'enchaînent (se calculent l'une après l'autre);
- 3) Que ces sous-fonctions aient approximativement le même coût.

Exercice 4

③

(4.1) Considérons la première fonction: $t_1 = \text{parité}(k_1, m_1, m_2, m_4) = 0$.

On construit la table de Karnaugh:

		$m_2 m_4$			
t_1		00	01	11	10
	00	0	1	0	1
$k_1 m_1$	01	1	0	1	0
	11	0	1	0	1
	10	1	0	1	0

Cette table "en damier" ne peut pas être simplifiée! C'est la signature de la fonction XOR.

On a donc: $t_1 = k_1 \oplus m_1 \oplus m_2 \oplus m_4 = 0 \iff k_1 = m_1 \oplus m_2 \oplus m_4$.

De façon symétrique:

$$k_2 = m_1 \oplus m_3 \oplus m_4$$

$$k_3 = m_2 \oplus m_3 \oplus m_4$$

(4.2) Sur la base de la question précédente, nous savons que ces fonctions sont des XOR, et que la table de Karnaugh ne nous aidera pas. On a donc directement:

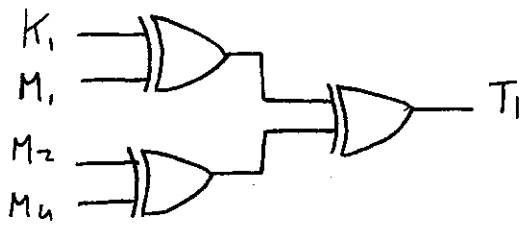
$$T_1 = K_1 \oplus M_1 \oplus M_2 \oplus M_4$$

$$T_2 = K_2 \oplus M_1 \oplus M_3 \oplus M_4$$

$$T_3 = K_3 \oplus M_2 \oplus M_3 \oplus M_4$$

(4.3) On peut utiliser un multiplexeur câblant la fonction logique considérée, ou alors directement des portes XOR.

Par exemple :



(4.4) Il faut inverser (on utilisera pour cela également un XOR)

le fil correspondant au numéro codé par T_1, T_2, T_3 .

Seuls les fils correspondant à M_1, M_2, M_3 et M_4 nous intéressent.

Les erreurs sur K_1, K_2 et K_3 ne remettent pas en cause la transmission.

On utilise pour cela un décodeur :

