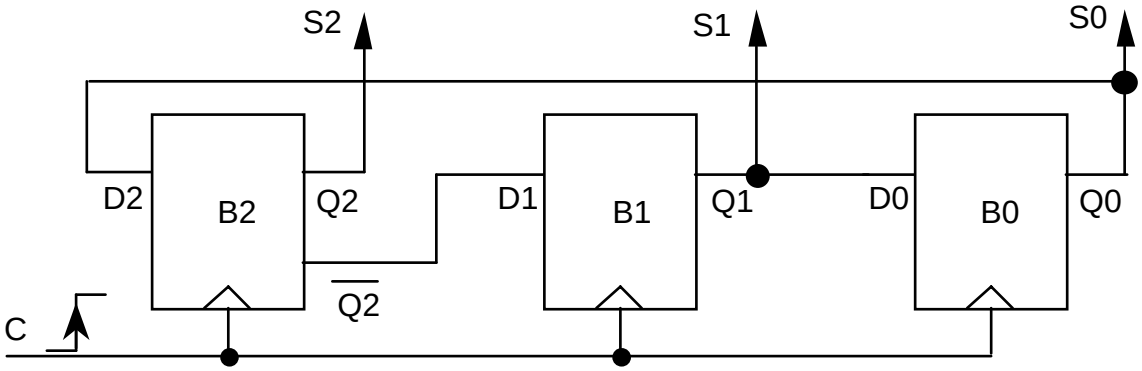
PHY 568

PC « systèmes numériques synchrones & synthèse des automates »

Exercice A : chronogrammes

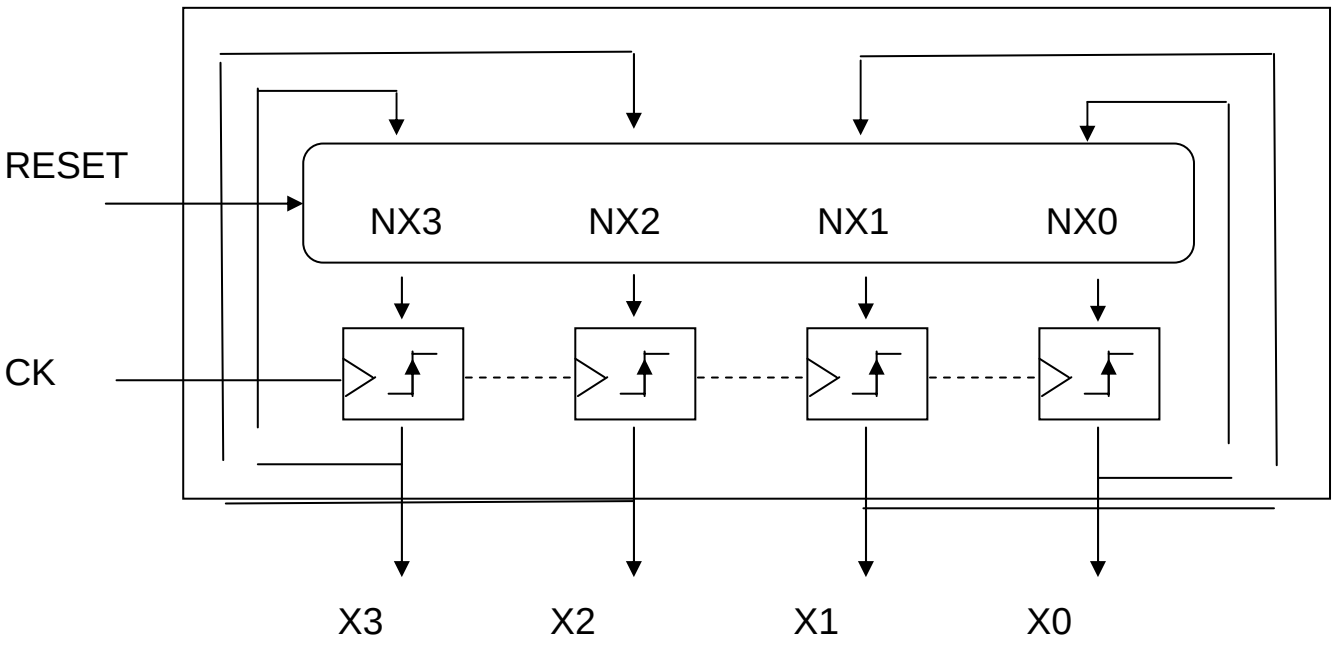
On considère le montage suivant composé de 3 bascules :



Q1 On demande de compléter le chronogramme fourni en annexe, en supposant que la valeur initiale contenue dans les bascules est 000.

Exercice B : compteur 4 bits

On souhaite réaliser un compteur cyclique de 0 à 15. Ce compteur est réalisé avec 4 bascules et un bloc de logique combinatoire suivant le schéma ci-dessous. Ce compteur possède un signal d'initialisation synchrone, RESET, qui force la valeur 0000 dans les quatre bascules.



Q2 Ecrire les expressions Booléennes simplifiées pour les signaux NX0, NX1, NX2, NX3, en fonction des signaux X0, X1, X2, X3, et RESET

Q3 Proposer un schéma en portes logiques pour la partie combinatoire de ce compteur. On n'utilisera que des portes logiques XOR, NOR, NAND et INVERSEUR.

Q4 Déterminer la chaîne longue dans ce schéma, et en déduire une borne inférieure pour le temps de cycle. On utilisera les données suivantes

- Tp (INV) = 0.5 ns
- Tp(NAND) = 1 ns
- Tp(NOR) = 1 ns
- Tp(XOR) = 2 ns
- Tsu(FF) = 1 ns
- Th(FF) = 0 ns
- Ta(FF) = 2 ns

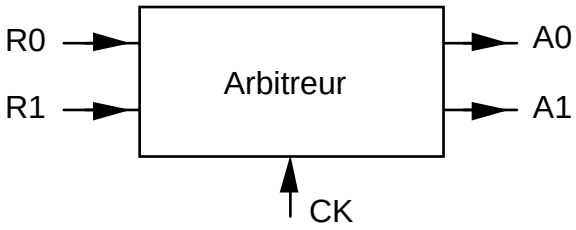
On considère que la valeur du skew est de 500 ps.

Exercice C : allocateur de ressource

On considère deux entités partageant une même ressource (par exemple deux processeurs P0 et P1 utilisant le même bus de communication pour accéder à la mémoire). On cherche à réaliser un « allocateur de ressource » équitable, se comportant comme un automate de Moore.

Chacun des processeurs dispose d'un signal de requête Ri, actif à l'état haut, pour demander à utiliser le bus. Les signaux de requête (R0 pour P0 et R1 pour P1), sont indépendants et peuvent être actifs simultanément.

En réponse, l'allocateur alloue le bus à l'un des deux processeurs et l'indique par l'intermédiaire d'un signal Ai actif à l'état haut (A0 pour P0 et A1 pour P1).
Quand un processeur a fini d'utiliser le bus, il le signale en remettant le signal Ri à zéro. Quand le bus est libéré par un processeur, il ne peut pas être immédiatement re-alloué, il faut attendre au moins un cycle avant de l'allouer de nouveau.
Pour garantir un arbitrage équitable, on applique la règle suivante : s'il y a deux requêtes simultanées, le processeur, qui a obtenu le bus le plus récemment, est le moins prioritaire.



On définit les quatre états de l'automate de Moore :

WAIT0	:	l'arbiteur attend une requête, le processeur 0 est prioritaire.
WAIT1	:	l'arbiteur attend une requête, le processeur 1 est prioritaire.
ALLOC0	:	le bus est alloué au processeur 0. L'arbiteur attend la libération du bus.
ALLOC1	:	le bus est alloué au processeur 1. L'arbiteur attend la libération du bus.

Les sorties du registre d'état sont notées Q1, Q0. On choisit le codage suivant :

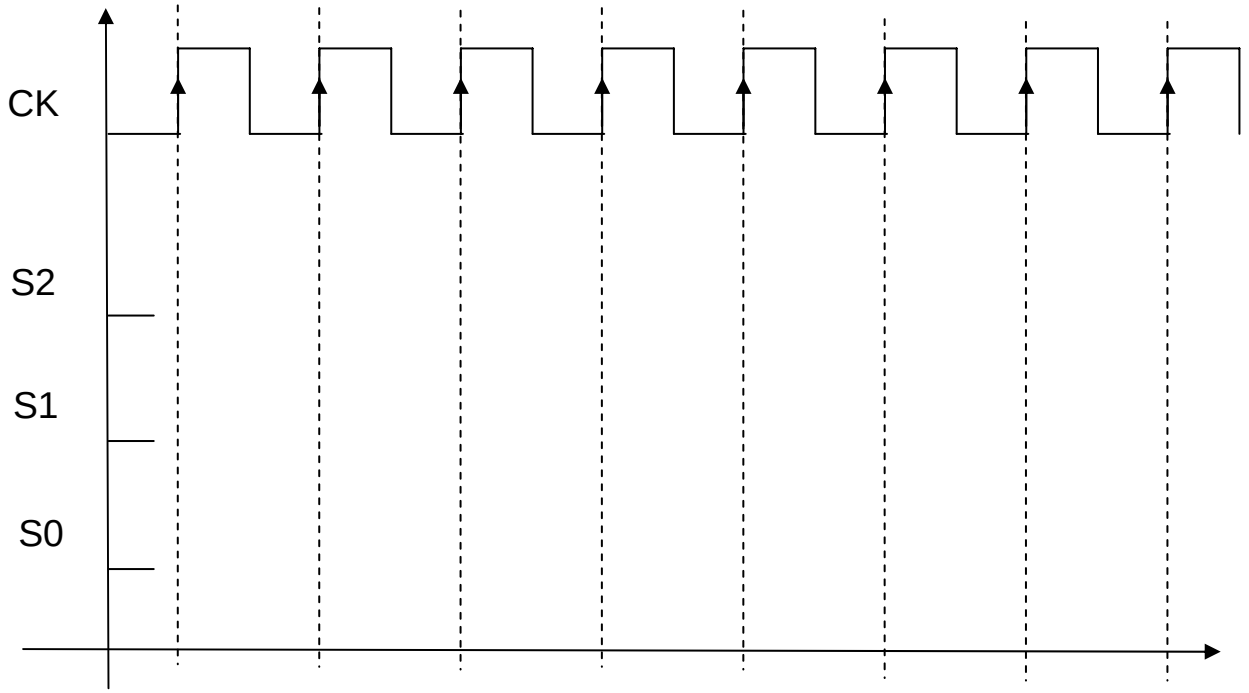
Etat	Q1	Q0
WAIT0	0	0
ALLOC0	0	1
ALLOC1	1	0
WAIT1	1	1

Q5 Compléter le diagramme d'état fourni en annexe en étiquetant les transitions avec les expressions booléennes dépendant de R1 et R0, et en associant à chaque état les valeurs des sorties A0 et A1.

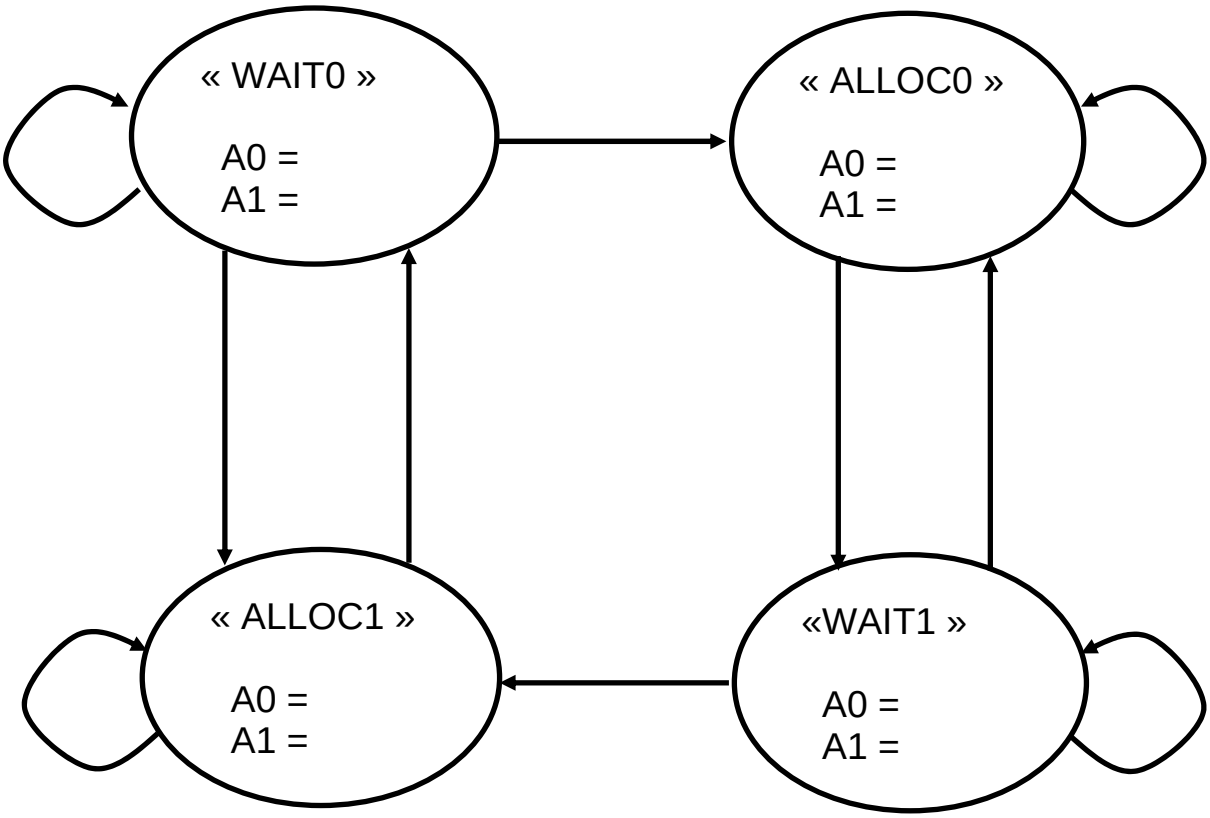
Q6 Remplir la table de vérité représentant l'état futur NQ1, NQ0 ainsi que les sorties A1 et A0, en fonction de R1, R0, Q1 et Q0, et donner les expressions booléennes simplifiées de A1, A0, NQ1, NQ0

Q7 Proposer un schéma en portes logiques réalisant l'allocateur de ressource.

Annexe A : Chronogrammes



Annexe B : Allocateur de Ressource



Etat	Q1	Q0	R1	R0	NQ1	NQ0	A1	A0
'WAIT0'	0	0	0	0				
'WAIT0'	0	0	0	1				
'WAIT0'	0	0	1	0				
'WAIT0'	0	0	1	1				
'ALLOC0'	0	1	0	0				
'ALLOC0'	0	1	0	1				
'ALLOC0'	0	1	1	0				
'ALLOC0'	0	1	1	1				
'ALLOC1'	1	0	0	0				
'ALLOC1'	1	0	0	1				
'ALLOC1'	1	0	1	0				
'ALLOC1'	1	0	1	1				
'WAIT1'	1	1	0	0				
'WAIT1'	1	1	0	1				
'WAIT1'	1	1	1	0				
'WAIT1'	1	1	1	1				