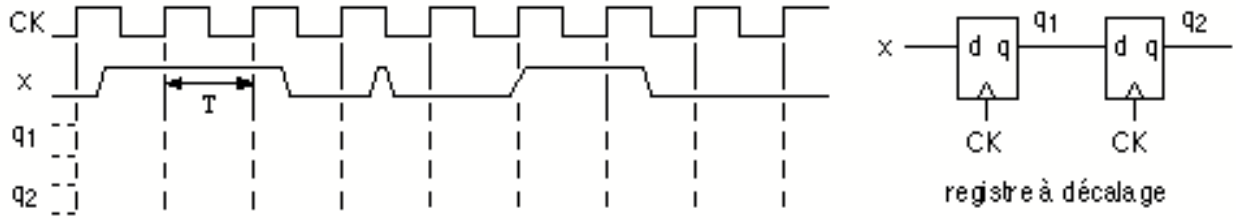
ES102/PC6 : énoncé et corrigé

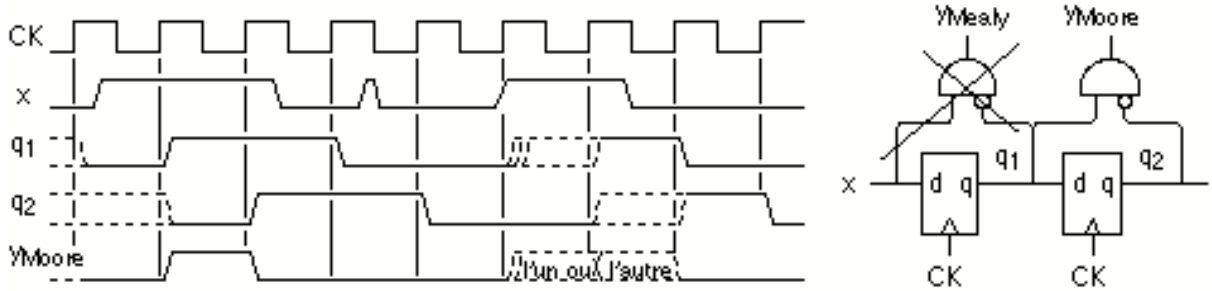
1) Jeu de bascules

1a) Compléter le chronogramme ci-dessous correspondant au montage sur le côté.



Une bascule D échantillonne son signal d'entrée sur un petit intervalle de temps situé autour du front montant de l'horloge, puis transfère rapidement la donnée échantillonnée vers sa sortie, avec des délais petits par rapport à la période d'horloge mais pas négligeables toutefois. D'où le chronogramme suivant (où la dernière ligne y_{Moore} concerne la question suivante). Le *glitch* de x entre deux tops d'horloge successifs passe inaperçu sur q_1 (a fortiori sur q_2).

En pointillés sont représentés les comportements incertains. Au tout début, on ignore quelle est la valeur en sortie de bascule, résultant du passé. Plus loin, lorsque le signal x transite en même temps que le front montant de l'horloge, la première bascule hésite avant de se décider, de façon imprévisible, entre un 0 ou un 1.



La deuxième bascule présente une période de retard sur la première : q_2 est le passé de q_1. Ceci se vérifie en temps continu (mêmes petits délais par rapport aux tops d'horloge) et a fortiori en temps discret (valeurs aux tops d'horloge uniquement).

L'association de bascules D en série est appelée *registre à décalage*. A chaque top d'horloge, la donnée qui était présente en sortie d'une bascule se retrouve donc décalée spatialement en sortie de la suivante. Cela réalise une pile binaire FIFO (first in, first out).

1b) Equiper ce registre à décalage d'une sortie de type Moore qui passe à 1, pendant une seule période d'horloge, dès que *x* a effectué une transition montante. Tracer le chronogramme.

Une telle sortie y_{Moore} doit dépendre combinatoirement des sorties de bascules D, c'est-à-dire des bits d'état (ici q_1 et q_2), mais pas des entrées (ici *x*). Donc y_{Moore} ne peut changer de valeur que juste après chaque top d'horloge. En ce sens, l'exigence « dès que » ci-dessus est une spécification imprécise, qui ne peut être satisfaite *stricto senso* : pour passer à 1, y_{Moore} devra forcément attendre le prochain top d'horloge suivant la transition montante de *x*.

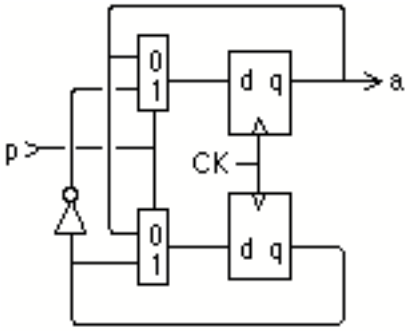
Détecter que *x* vient de passer de 0 à 1, d'un top d'horloge au suivant, équivaut à détecter le motif (q_2=0) et (q_1=1). En effet, q_1 est la valeur de *x* qui vient d'être échantillonnée, tandis que q_2 est celle qui l'a été au top d'horloge précédent. D'où $y_{Moore} = q_2' \cdot q_1$.

Dans le cas indésirable où le signal x transite au top d'horloge, y_{Moore} finira par passer à 1, soit à la période suivante, soit à celle d'après (voir le chronogramme), mais pas les deux. Ce cas montre que le caractère non synchrone d'une entrée est parfois supportable.

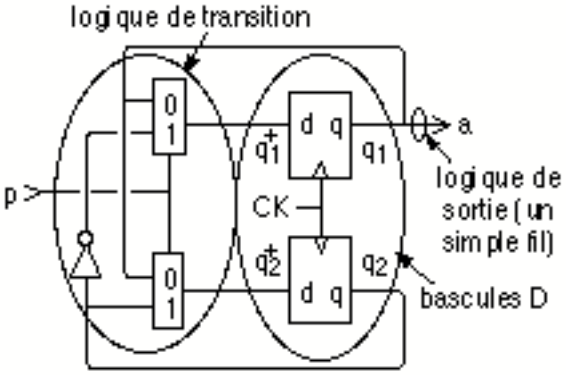
Pour être plus réactif, on pourrait renoncer à la contrainte Moore et brancher ce même montage en amont, sur x et q_1, pour produire une sortie de type Mealy notée y_{Mealy}. Mais y_{Mealy} subirait alors combinatoirement tout caprice de *x*, sortant du cadre synchrone. Comme indiqué en cours, cette pratique est à proscrire, dans un souci de sûreté de fonctionnement.

2) Analyse d'un petit circuit séquentiel

2a) Sur le montage ci-contre, identifier bascules D, bits d'état, état, état futur, signaux d'entrées, bloc combinatoire de transition, signaux de sortie, bloc combinatoire de sortie.



Les bits d'état sont les sorties des bascules D, notées q_1 et q_2 sur le schéma ci-dessous. Leur couple (q_2, q_1) constitue l'état **q**. Il y a un signal d'entrée, *p*, et un de sortie, *a*. L'état futur est la valeur à venir $\mathbf{q}^+ = (q_2^+, q_1^+)$ de **q**, après le prochain top d'horloge. Elle est fournie à l'entrée des bascules D par le bloc combinatoire de transition avant le prochain top d'horloge.



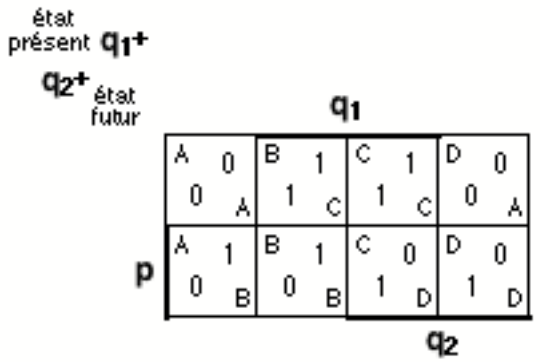
2b) Exprimer fonctions de transition et de sortie.

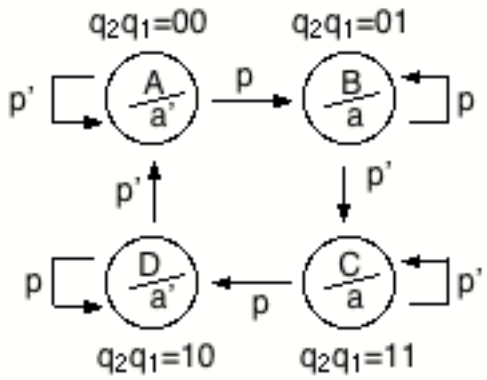
La fonction de transition *f* fournit l'état futur (q_2^+, q_1^+) en fonction de l'état courant (q_2, q_1) et de l'entrée *p*. C'est une fonction booléenne *vectorielle* : $q_2^+=pq_2+p'q_1$ et $q_1^+=pq_2'+p'q_1$.

La fonction de sortie *g* est une fonction booléenne simple (car à valeur binaire *a*). La logique de sortie étant ici réduite à un simple fil, on a trivialement : $a=g(q_2,q_1)=q_1$.

2c) Abstraire le circuit en un automate, c'est-à-dire exprimer les ensembles d'entrées *X*, d'états *Q* et de sorties *Y*, puis dessiner le diagramme d'état correspondant.

X et Y répertorient les configurations en entrée et sortie. Au lieu des valeurs 0 et 1, utilisons une notation fonctionnelle (fonctions caractéristiques) montrant les variables en jeu : *X*={p', p}, *Q*={$q_2'q_1'$, $q_2'q_1$, q_2q_1', q_2q_1} et *Y*={a', a}. Afin de se mettre dans l'état d'esprit d'une synthèse, les éléments de *Q* sont en outre symbolisés ci-dessous par des lettres majuscules, en suivant un ordre suggéré par le diagramme d'état.





Avant d'élaborer le diagramme d'état, on représente f, c'est-à-dire l'état futur, sous la forme d'un tableau de Karnaugh fonctionnel *vectoriel* : y apparaissent simultanément q_2^+ – en bas à

gauche de chaque case – et q_1^+ – en haut à droite. Mieux encore, on fait apparaître dans les deux autres coins les valeurs symboliques de **q** et de **q**$^+$. Cette façon de faire, claire et efficace en général, est **fortement recommandée**. Chaque case correspond ici à une transition sur le diagramme d'état (dans le cas général, une transition pourra correspondre à plusieurs cases).

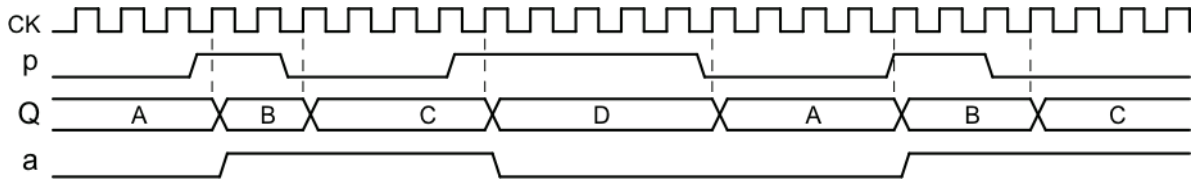
2d) Quelle propriété intéressante présente l'encodage des états ?

Il est adjacent : d'un état à son successeur, le code change d'au plus un bit. Cette propriété est ici obtenue tout simplement en utilisant le code de Gray sur 2 bits pour un cycle de 4 états.

2e) Comprendre le comportement de cet automate par analyse du diagramme d'état, l'illustrer par un chronogramme et en trouver un usage typique.

Le diagramme d'état montre que la sortie *a* ne change pas lorsque l'entrée *p* reste ou passe à 0. En revanche, *a* change si *p* passe de 0 à 1. Mais *a* ne change plus si *p* demeure ensuite à 1. Ce comportement est celui d'un télérupteur commandé par un bouton-**p**oussoir, alimentant une **a**mpoule. D'où le chronogramme ci-dessous, où l'on comprendra précisément la représentation de l'état Q ainsi que la synchronisation de Q et *a* par rapport à l'horloge CK.

Autre façon de voir les choses, si *p* reçoit un signal oscillant à une certaine fréquence *f* (petite par rapport à f_{CK}) la sortie *a* oscillera à une fréquence f/2. Ce montage peut donc aussi servir de *diviseur de fréquence par 2*, entre son entrée et sa sortie (on ne joue pas avec CK ici !).



2f) Si *p* vient d'un bouton-poussoir, alors c'est une entrée asynchrone. Que faire dans ce cas ?

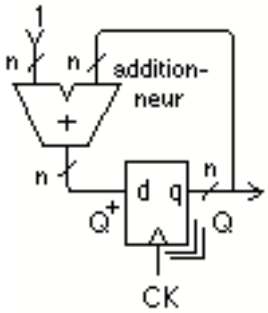
p doit être préalablement synchronisé par une bascule D avant d'attaquer la logique de transition, sinon cela pourrait parfois provoquer une évolution incohérente de q_1 et q_2.

3) Compteur *n* bits

3a) Un compteur *n* bits élémentaire est un circuit séquentiel synchrone dont l'état est un entier sur *n* bits en notation classique, incrémenté (modulo 2^n) à chaque top d'horloge. En proposer une implantation à un niveau numérique, puis au niveau portes.

Pour un compteur, l'état est un entier Q (il est donc d'emblée numérique, situation simplifiée par rapport au cas général d'une synthèse partant d'états symboliques). En outre, on dispose immédiatement d'une loi de transition numérique : $Q^+=(Q+1)\%2^n$. Le diagramme d'état en découle trivialement : c'est un cycle qui chaîne les 2^n états allant de Q=0 à $Q=2^n-1$ par des transitions inconditionnelles, car il n'y a pas d'entrée. L'état Q=0 succède à l'état $Q=2^n-1$. D'autre part, la sortie est l'état Q lui-même.

L'étape suivante habituelle de la synthèse est l'encodage d'état. Ici, l'entier non signé Q est classiquement représenté sur *n* bits $(q_i)_{0\le i<n}$. Ces q_i seront donc les bits d'état, fournis en sortie des bascules D, tandis qu'en entrée se trouveront les valeurs futures q_i^+, que le bloc combinatoire de transition devra fournir un peu avant le prochain top d'horloge. Pour l'implanter, dernière étape de la synthèse, il faut exprimer les $(q_i^+)_{0\le i<2^n}$ en fonction des $(q_i)_{0\le i<2^n}$. Il suffirait en fait d'utiliser un additionneur numérique avec comme opérandes Q et 1, comme montré ci-contre. Mais c'est une solution pour électronicien pressé et riche. Soyons plus économes : additionner Q et 1 revient à additionner Q et 0 avec retenue entrante c_0 de l'additionneur à 1. Utilisant un

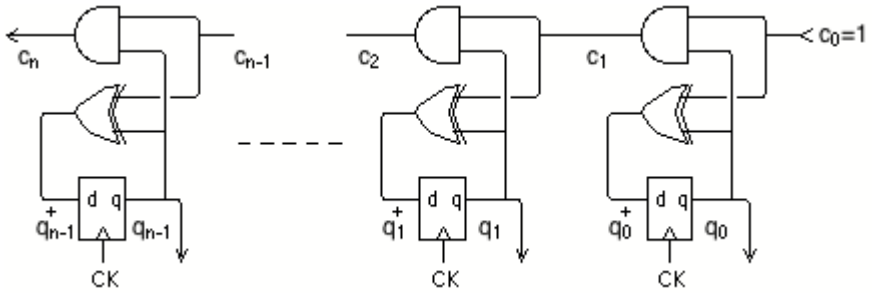


additionneur à retenue propagée, constitué de *n* FAs en série, on aura alors, ∀i (incluant i=0) :

$$q_i^+ = \mathrm{Par}(q_i, 0, c_i) = q_i \oplus c_i \quad \text{et} \quad c_{i+1} = \mathrm{Maj}(q_i, 0, c_i) = q_i \cdot c_i$$

Chaque FA dégénère donc en une porte AND et une porte XOR. D'où la structure de compteur ci-dessous, constituée par l'aboutement de *n* tranches identiques où se combinent les 2 portes ci-dessus avec une bascule D supportant un q_i. Sans oublier que c_0=1.

Du coup la première tranche (à gauche) pourrait être encore simplifiée, mais on ne le fait pas car elle nous servira bientôt. A l'autre extrémité, la dernière porte AND n'est pas indispensable, mais néanmoins utile car c_n indique lorsque le compteur a fait un tour complet (débordement).



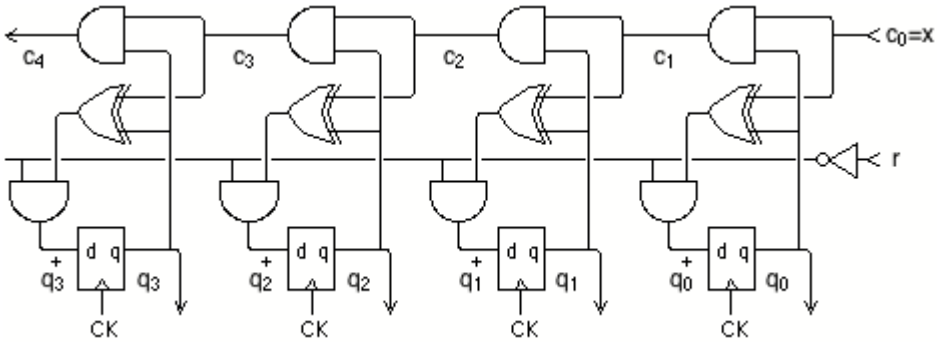
3b) Simuler à la main une rotation complète d'un compteur 2 bits.

Où l'on constate effectivement que, en 4 tops d'horloge successifs, le couple (q_1, q_0) passe successivement de (0, 0) à (0, 1), puis (1, 0), puis (1, 1) avant de revenir à (0, 0).

3c) Les considérations précédentes ont fait intervenir la retenue entrante de l'addition : c_0. Que se passe-t-il si c_0 devient une entrée binaire *x* ?

La loi de transition/évolution numérique devient : $Q^+=(Q+x)\%2^n$. Le compteur se transforme donc en *intégrateur numérique* de signal binaire, modulo 2^n. On parle aussi de « compteur d'événements » où les événements sont les tops d'horloges où x=1. C'est un opérateur fort utile. Le diagramme d'état, cyclique, s'en trouve modifié. Les anciennes transitions, inconditionnelles, sont maintenant conditionnées par x. De nouvelles apparaissent, conditionnées par x', qui bouclent sur chaque état.

Détail sournois : avec cette modification, c_n est devenue une sortie de type Mealy. Sauf si c_0=x est elle-même une entrée synchrone (provenant de bascules D)…

3d) Equiper le compteur d'une remise à 0, en restant au niveau des portes élémentaires.

Une capacité de remise à 0 est indispensable dans bien des cas, en particulier pour du comptage d'événements. Il s'agit d'équiper le compteur d'une entrée *r* (reset) synchrone qui, en passant à 1, mettra l'état futur Q^+ à 0, pour obtenir Q=0 au prochain top d'horloge. On pourrait insérer un multiplexeur juste devant l'entrée de chaque bascule, qui laisserait entrer un 0 si r=1 et l'entrée habituelle sinon : cela aboutit à une porte AND recevant r'. On parvient au même résultat en modifiant la fonction de transition, de $q_i^+ = q_i \oplus c_i$ à $q_i^+ = (q_i \oplus c_i) \cdot r'$. L'implantation est illustrée ci-dessous sur le cas d'un compteur d'événements modulo 16.