

**Objectif :** Application de la démarche d'analyse d'un circuit séquentiel permettant de représenter la fonctionnalité d'un circuit à l'aide de modèles de représentation algébrique, tabulaire, graphique.

### Exercice I. Analyse d'un système séquentiel synchrone transition based (Mealy)

On souhaite établir le comportement d'un circuit séquentiel synchrone dont le symbole fait état d'une entrée de donnée  $X$ , de l'horloge  $clk$  et d'une sortie  $Z$ . Une solution d'implémentation apparaît sur la figure ci-dessous. Le bloc diagramme est construit à partir de cellules logiques de nature séquentielle (bascules JK) ou combinatoire (portes ET, inverseurs, NXOR).

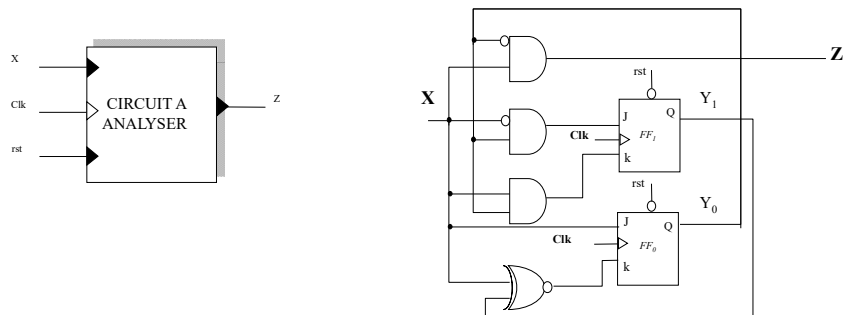


Figure 1: Symbole et structure du circuit à analyser

Le chronogramme ci-dessous montre un exemple de comportement du circuit en réponse à des stimuli d'horloge. Le pas de simulation est défini à 20 ns (période d'horloge).

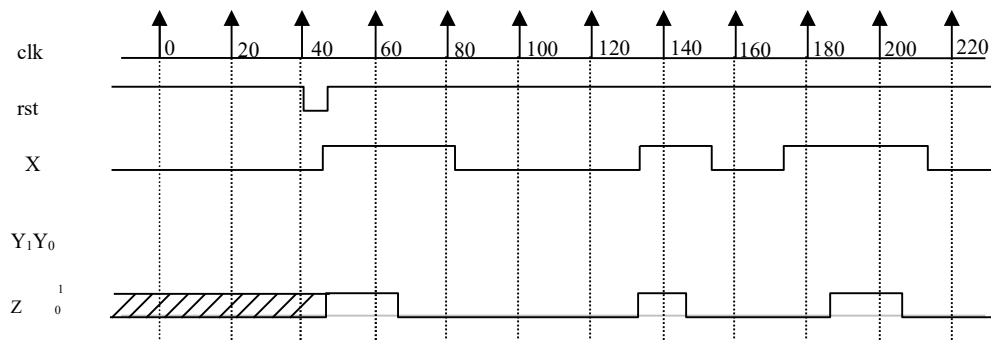


Figure 2: Chronogramme du circuit synchrone

**Questions :** 1. Développez la démarche d'analyse d'un circuit séquentiel en recherchant successivement :

- Les équations d'entrée des bascules (équations d'excitation).
- Les équations des états suivants
- L'équation de la sortie  $Z$
- La table et le diagramme des transitions

2. Complétez le chronogramme de la figure ci-dessous en ajoutant l'évolution de l'état interne.

## Exercice II : Analyse d'un système asynchrone en mode pulsé

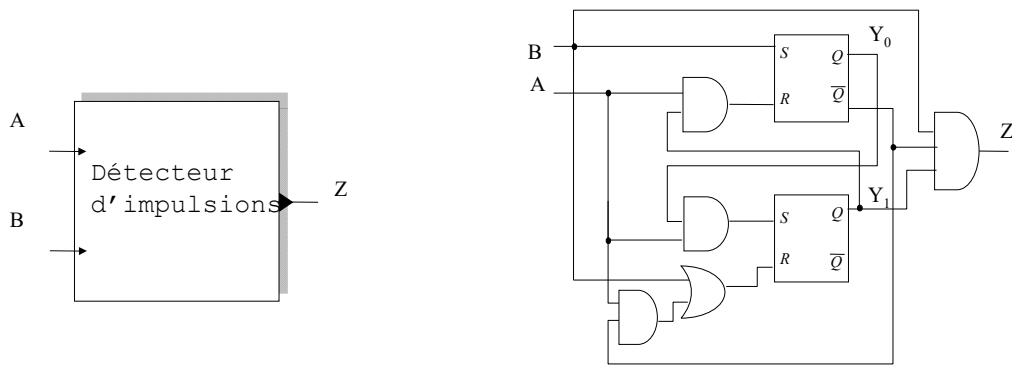


Figure 3: Symbole du circuit de détection d'impulsions

Le circuit dont la structure apparaît sur la figure ci-dessous est un circuit permettant de détecter une séquence d'impulsions de type BAA. Le système transmet alors la valeur de l'entrée B sur sa sortie Z. Dans un tel système pulsé, il ne peut y avoir de recouvrement des impulsions. Le chronogramme ci-dessous trace l'évolution temporelle des signaux internes et du signal en sortie.

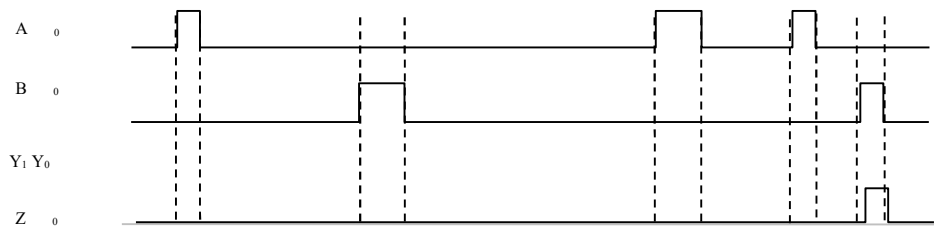


Figure 4: Chronogramme du circuit pulsé

**Questions :** A.. Déterminez les équations d'entrée des éléments mémoires du circuit.

**B.** A partir de ces équations **et** de l'équation générale du latch RS, déterminez les équations des états suivants. Donnez l'équation de la sortie Z.

**C.** Dressez le modèle de représentation du comportement à partir d'une table de transition. On définira les états suivants/sorties pour chacune des quatre combinaisons possibles du vecteur d'entrée (en colonnes) et des quatre combinaisons possibles du vecteur d'états (en ligne).

**D.** Complétez le chronogramme ci-dessus en faisant évoluer l'état interne  $Y_1 Y_0$  en fonction des combinaisons d'entrée. On supposera le vecteur d'état fixé initialement à 00.

**E.** Que peut on dire du vecteur d'état si on observe attentivement le chronogramme entre 2 impulsions ? Sachant d'autre part que l'hypothèse d'absence de recouvrement temporel entre 2 impulsions en entrée s'applique aux systèmes pulsés, proposez une forme plus compacte de la table des transitions et dessinez le diagramme de transitions correspondant.

**Objectif : Synthèse de systèmes séquentiels en mode horloge ou en mode pulsé**

## Exercice I : Synthèse d'un compteur DCB

On souhaite concevoir un compteur synchrone 4 bits comptant en DCB, soit de 0000 à 1001, de manière cyclique.



Y	Y <sup>+</sup>	Z <sub>3</sub> Z <sub>2</sub> Z <sub>1</sub> Z <sub>0</sub>
E0	E1	0000
E1	E2	0001
E2	E3	0010
E3	E4	0011
E4	E5	0100
E5	E6	0101
E6	E7	0110
E7	E8	0111
E8	E9	1000
E9	E0	1001
E10-E15	--	----

**Figure 1: Symbole et table d'états du circuit de comptage DCB**

On notera qu'il s'agit d'un système autonome (pas d'entrée synchrone). La figure ci-dessus fournit également un diagramme d'état caractérisant le fonctionnement attendu du circuit.

*Questions :*

- Donnez la table de transistions en considérant que chaque état sera codé par la sortie, c'est-à-dire par le code DCB: E0 :0000, E1 :0001, ..., E9 :1001. On présentera la table de transition poids fort à gauche. Les combinaisons non utilisés (de 1010 à 1111) conduiront à des états suivants indéterminés.
- Dressez la table d'excitation des entrées. La technologie ciblée sera celle des bascules JK.
- Simplifiez les équations des entrées des éléments mémoires.
- Donnez une implémentation du circuit à base portes élémentaires ET/OU/NON (nombre d'entrées quelconque). On ajoutera une commande asynchrone *rst* permettant de forcer le circuit dans l'état initial 0000 à la mise sous tension ou en cas d'anomalie.
- Modifiez le symbole et la table des états pour définir un compteur BCD avec enable. Lorsque l'entrée *enable* vaut 1, le compteur s'incrémentera au front d'horloge. Dans le cas contraire, il restera figé à la valeur courante.

## Exercice II : Contrôle automatique d'une barrière de péage

On souhaite réaliser un système de péage automatique sur le modèle de la figure ci-dessous.

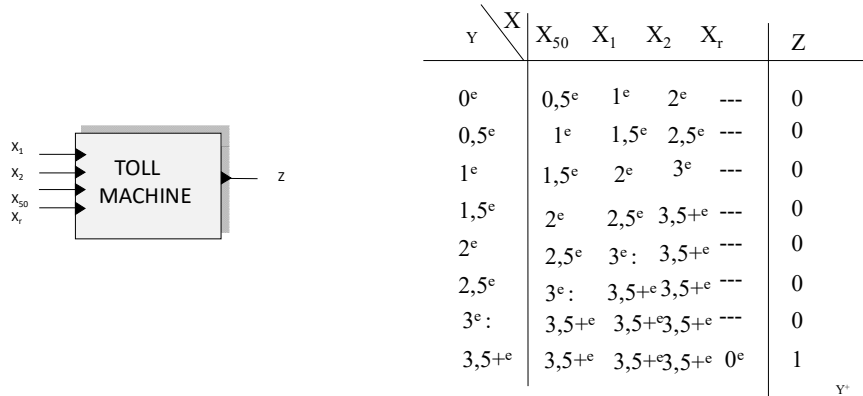


Figure 2: Symbole et fonctionnement du circuit de contrôle de la barrière

Le prix est fixé à 3 euros 50 et la machine accepte les pièces de 1 €, 2 € et 50 cts . Le système automatique accepte les pièces séquentiellement (même si elles sont jetées simultanément) et génère une des 3 impulsions  $x_1$ ,  $x_2$  et  $x_{50}$  selon le type de pièce (1 €, 2 € ou 50 cts respectivement). Le circuit séquentiel produit une sortie à niveau  $Z=1$  (permettant par exemple l'ouverture de la barrière) lorsque la valeur perçue totalise 3 € 50 ou plus. Lorsque la voiture est passée, une impulsion  $x_r$  est générée automatiquement et repositionne le système dans son état initial. Tout sur paiement est profitable au dépositaire de la machine.

La figure ci-dessus fournit une table d'états caractérisant le comportement attendu du circuit à concevoir.

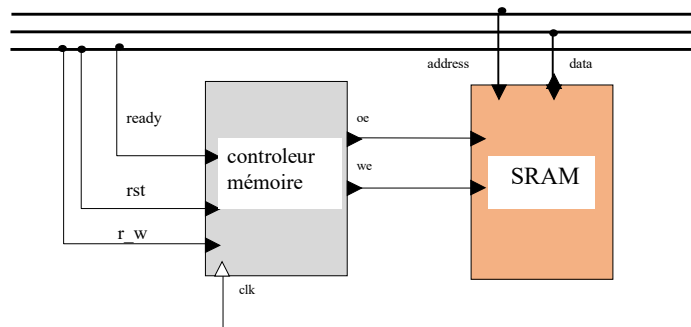
**Questions : A.** Dressez la table des transistions en adoptant un codage séquentiel des différents états: 0<sup>e</sup> : 000, ..., 3,5<sup>e</sup> : 111.

**B.** Donnez les tables d'excitation des éléments mémoire pour un circuit dont la technologie ciblée sera celle des latches RS NOR.

**C.** Simplifiez les différentes équations des entrées mémoire en remarquant qu'aucune simplification n'est possible entre entrées. On pourra donc dresser les tables pour chaque entrée et combiner ensuite les sous tables avec l'opérateur OU. On ne fournira pas de schéma d'implémentation.

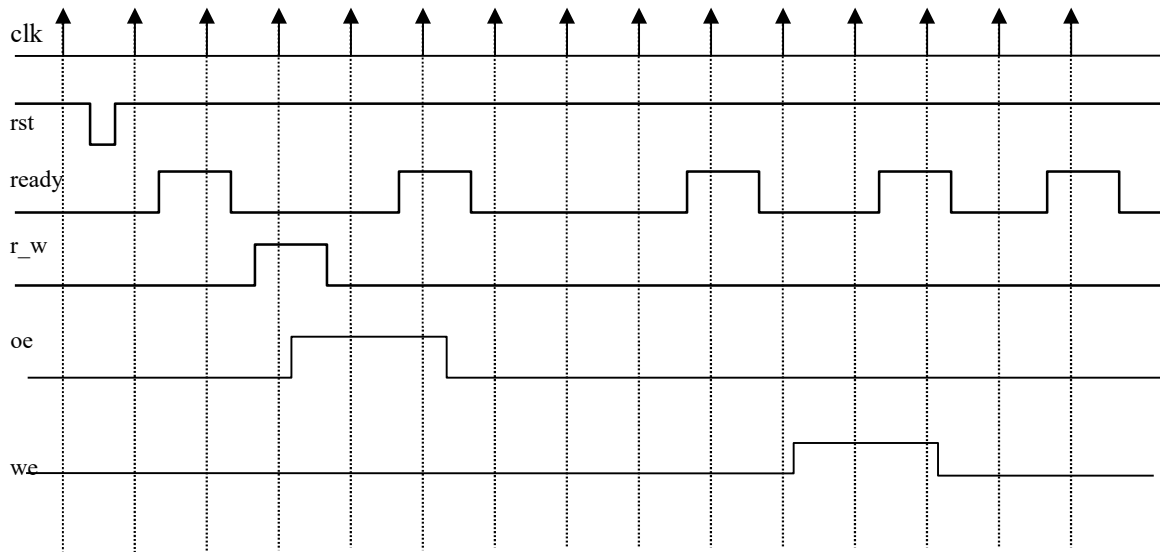
*Objectif : Application des techniques de codage une bascule par état et codage par le sorties*

On souhaite concevoir un contrôleur mémoire sur le modèle du bloc diagramme ci-dessous. Son rôle est de gérer l'accès(partage) d'une mémoire à partir de processeur multiples depuis un bus.



**Figure 1: Bloc diagramme du système**

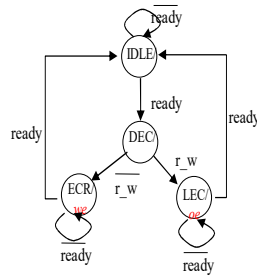
Le principe de fonctionnement du contrôleur est le suivant : lorsqu'un processeur du système souhaite accéder à la mémoire, il initialise une transaction en affirmant le signal *ready*. Un cycle d'horloge plus tard, le signal *r\_w* est affirmé si l'opération demandée est une lecture, opération induisant *oe*=1. Ce signal est infirmé si l'opération demandée est une écriture, opération induisant *we*=1. Chaque cycle de lecture ou écriture se termine par une nouvelle affirmation du signal *ready*.



**Figure 2: Spécification de fonctionnement du contrôleur mémoire**

On notera que le demandeur n'est pas dans l'obligation de maintenir le signal *r\_w* au delà de la deuxième période d'horloge. La figure 3 donne un diagramme des états du contrôleur mémoire.

*Questions : A. S'agit il d'une machine de moore ou de Mealy ? Justifiez.*



**Figure 3: Diagramme des états finis du controleur**

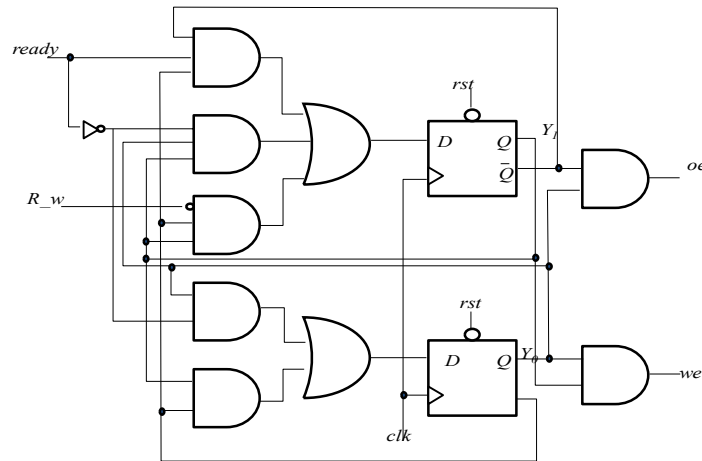
**B.** Un codage binaire utilisant les affectations suivantes : IDLE :00 LEC :01 DEC :10 ECR :11 aurait typiquement coinduit aux équations et au schéma ci-dessous

$$D_1 = \overline{Y_1} \overline{Y_0} \text{ready} + Y_1 Y_0 \text{ready} + Y_1 \overline{Y_0} r_w$$

$$D_0 = Y_0 \text{ready} + Y_1 \overline{Y_0}$$

$$we = Y_1 Y_0$$

$$oe = \overline{Y_1} Y_0$$



**Figure 4: bloc diagramme du circuit pour un codage binaire**

Calculez la surface (nombre de transistors) et la fréquence maximale d'horloge. On prendra 1 ns comme temps de propagation pour une porte inverseur (2 transistors) et 2.4 ns/6 transistors pour les portes ET ou OU 2 entrées. On ajoutera 0.4 ns/2 transistors par entrée supplémentaire. Une FF aura 3.5ns de Tco, 0.1 de temps de setup et comptera 20 Transistors. Un temps d'input delay de 3.5 ns sera appliquée sur les entrée.

**C.** Proposez une solution à partir d'un codage des états de la machine utilisant une bascule par état (one hot encoding). Le bloc diagramme sera déduit directement à partir du diagramme des états.. Évaluez les performances de cette solution.

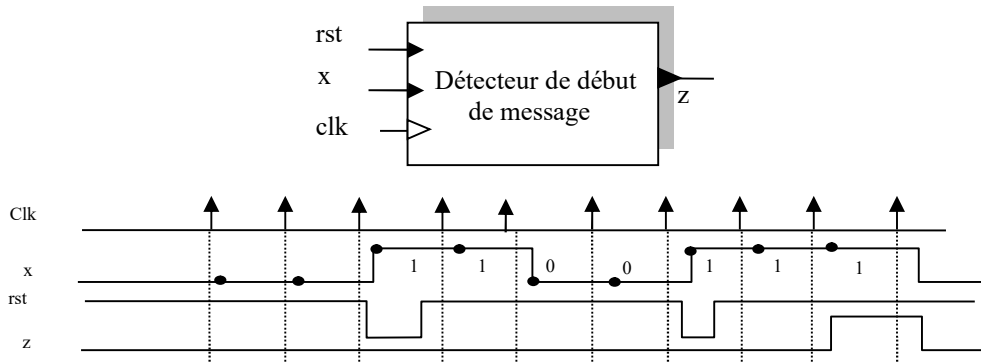
**D.** Dressez la table des états symboliques correspondant au diagramme des états. Proposez un codage des états par les sorties (codage de Moore) de manière à optimiser le temps de positionnement des sorties relativement à l'instant d'arrivée de l'horloge, tout en minimisant le nombre de FF. Effectuez l'évaluation des performances de la solution.

**E.** Concluez sur ces 3 solutions si une contrainte de fréquence maximale (fréquence du bus) est fixée à 100 MHZ.

*Objectif : Etude des règles heuristiques de codage par les adjacences*

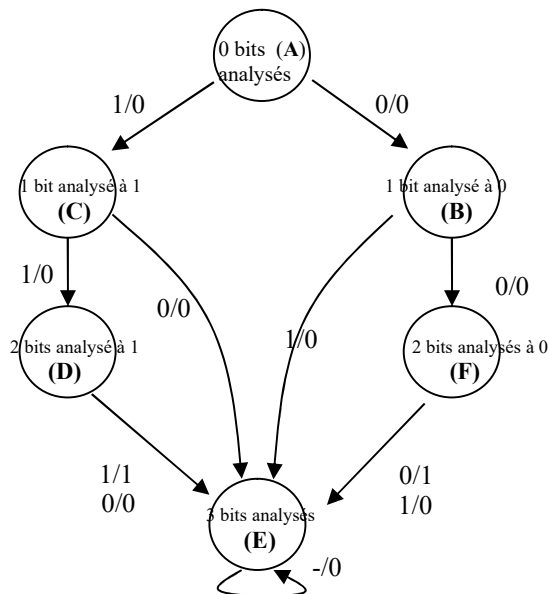
## I Détection d'un début de message

Des nombres entre 0 et 7 exprimés en binaire sont transmis en série sur une ligne x à partir d'un appareil de mesure digital. Le circuit à concevoir aura une sortie  $z=1$  uniquement au moment du 3ème bit du nombre binaire si la séquence des 3 chiffres binaires correspond à 000 ou à 111. Dans le cas contraire,  $z$  vaudra 0.



**Figure 1: Symbole et chronogramme du circuit détecteur de message**

La figure ci-dessus montre deux scénarii possibles. Au front d'horloge suivant le *rst*, le circuit échantillonne un premier chiffre binaire considéré comme le bit de poids fort de la séquence de 3 bits. La recherche de la séquence des 3 '1' s'effectue donc uniquement dans les 3 périodes d'horloge suivant un *rst*.



**Figure 2: Diagramme des états**

*Question* : A. Donnez la forme tabulaire correspondant au diagramme des états (figure 2).

B. Proposez un codage binaire pur de ces états afin d'optimiser la logique combinatoire (codage par les adjacences).

C. Dressez finalement la table des transitions à partir de l'affectation retenue.

## II Vérificateur de parité séquentiel

Lorsque les bits d'un code arrive en série comme dans le cas d'une transmission sur une ligne série, certaines opérations peuvent être avantageusement réalisées sous forme séquentielles plutôt que sous forme combinatoire lorsque les données sont disponibles en parallèle.

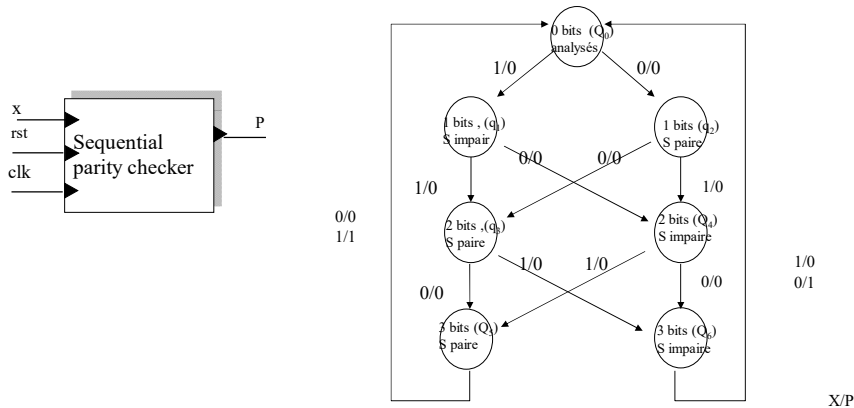


Figure 3: Symbole et description d'un vérificateur de parité séquentiel

On considère que cette suite de bit est parfaitement synchronisée avec l'horloge  $clk$  et que le poids faible arrive en tête de cette séquence. Le circuit "recalcule" la parité  $P$  de l'octet et doit produire une sortie  $P=1$  si la parité (considérée comme paire) est incorrecte,  $P=0$  sinon. Par conséquent, seules les suites de 4 bits contenant 0,2 ou 4 bits à 1 fournissent une sortie  $P$  à 0.

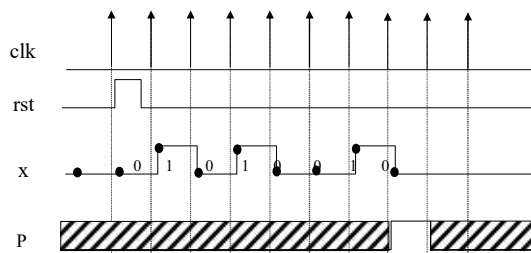


Figure 4: Exemple de chronogramme pour la vérification de la parité de 52h

Le chronogramme ci-dessus montre un exemple de réponse du circuit matérialisant une parité incorrecte. On notera que, bien que positionnée à 0, la valeur de  $P$  n'est réellement significative que sur les 4<sup>ème</sup> cycle d'horloge.

**Questions :** A. Donnez la forme tabulaire correspondant au diagramme des états (figure 3).

B. Donnez un codage binaire permettant d'optimiser la logique combinatoire (codage par les adjacences) de la table des états correspondant au diagramme de la figure ci-dessus.

C. Dressez ensuite la table des transitions à partir de ce codage.



*Obectifs : Réduction des machines séquentielles*

Soit le circuit dont la table d'état est la suivante:

$y \backslash x$	0	1	2	3
1	6/0	2/0	1/0	1/0
2	6/0	3/0	1/0	1/0
3	6/0	9/0	4/1	1/0
4	5/1	6/0	7/1	8/0
5	5/1	9/0	7/1	1/0
6	6/0	6/0	1/0	1/0
7	5/1	10/0	7/1	1/0
8	6/0	2/0	1/0	8/0
9	9/0	9/0	1/0	1/0
10	6/0	11/0	1/0	1/0
11	6/0	9/0	4/1	1/0

**Figure 1. Table d'états à réduire**

**Questions :** Donnez une forme minimale équivalente de ce circuit en utilisant la méthode de partitionnement en classes d'équivalence :

1. Donnez une première partition de l'ensemble des états sur la base des sorties
2. Vérifiez que les états suivants des éléments d'une classe se situent tous dans la même classe
3. Si tel n'est pas le cas , repartitionnez l'ensemble en isolant les états qui ne peuvent etre équivalents. Vérifiez à nouveau les classes des états suivants.
4. Lorsque les conditions d'équivalence sont vérifiées, formez la table d'état minimale à partir de ces classes d'équivalence.