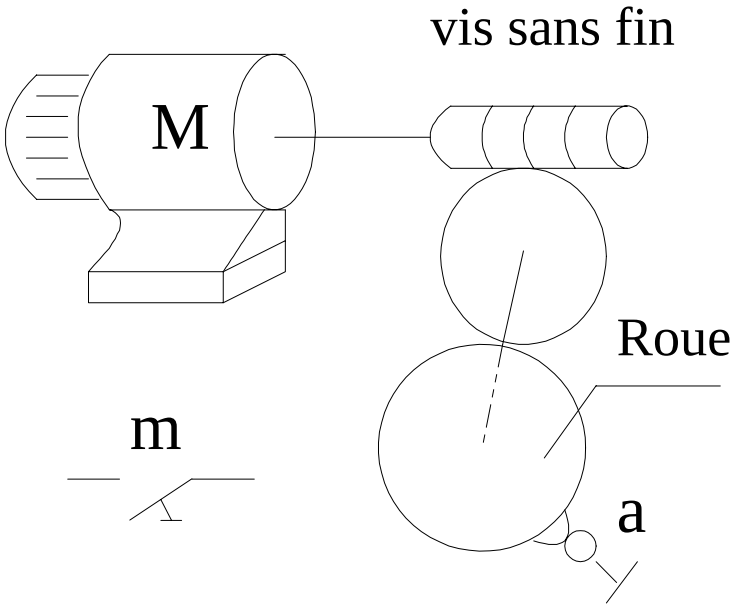
TABLE DE MATIERES

Chapitre I

Systèmes logiques séquentiels

A. Synthèse des systèmes séquentiels: Méthode d'Huffman

A.1. Etude d'un exemple



Un plateau tournant est calé sur l'axe d'un réducteur à roue et vis sans fin qui lui même entraîné par un moteur électrique **(M)**. Le plateau circulaire comporte un bossage qui agit sur un contact électrique **(a)** à la manière d'une came. Le fonctionnement de ce système est le suivant:

-Au repos, le bouton poussoir **(m)** n'est pas actionné, le contact **(a)** est appuyé par le bossage du plateau circulaire.

-Quand on appuie sur le bouton poussoir **(m)**, on désire que le moteur se mette à tourner entraînant ainsi le plateau circulaire par l'intermédiaire d'un réducteur.

-L'appui sur le bouton poussoir **(m)** est maintenu jusqu'à ce que le bossage du plateau circulaire libère le contact électrique **(a)**.

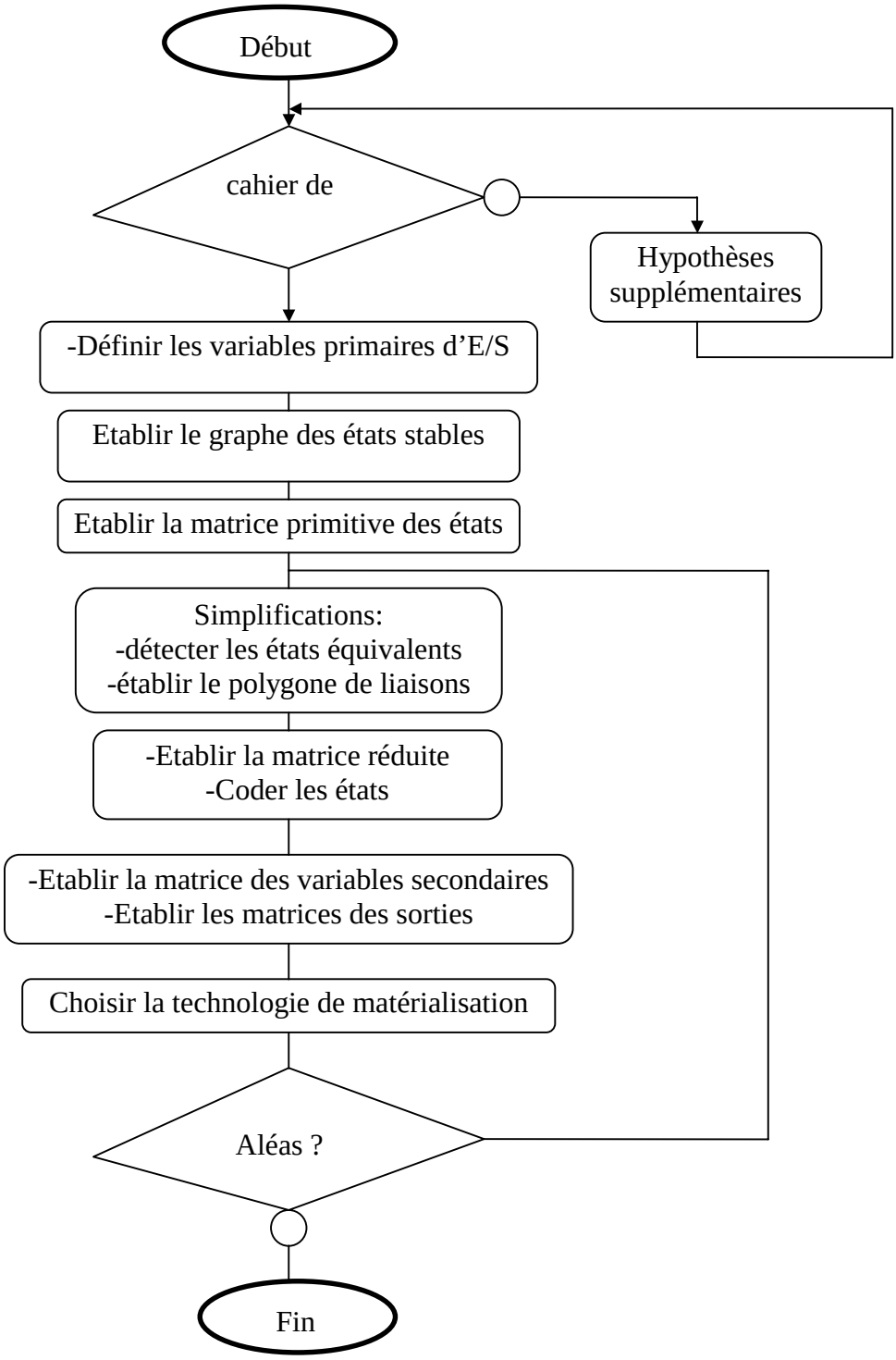
-Lorsque le contact électrique **(A)** est libéré, on relâche le bouton poussoir **(m)** et le plateau continue à tourner.

-Lorsque le plateau fait un tour, on désire que le moteur s'arrête.

On demande de :

1. Matérialiser cet automatisme par un logigramme.
2. On suppose que l'action sur le bouton poussoir m est fugitive (m sera relâché aussitôt qu'il est actionné), refaire l'étude de cet automatisme, conclure.

A.2 Méthode d'Huffman



B. Synthèse et mise en œuvre des bistables

B.1. Définition

Un bistable ou (bascule, mémoire, FLIP-FLOP) est un système séquentiel élémentaire, qui comporte 2 entrées et 2 sorties, possédant deux états

une stable et l'autre instable (si $m = a = 1$ on peut avoir à la sortie des cas totalement différent).

B.2. Bistables asynchrones

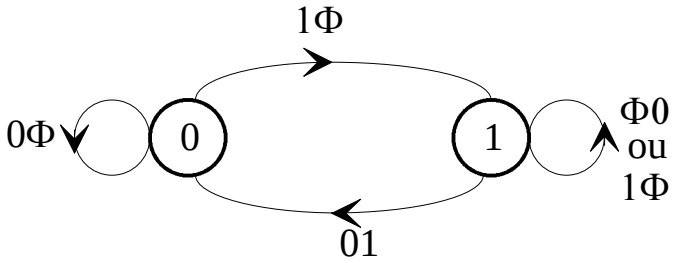
B.2.1. Bistable asynchrone à marche prioritaire

q(t) / ***ma***	**0**	**1**	
00	0	1	
01	0	0	
11	1	1	
10	1	1	
			Q(t + τ)

$$Q(t+\tau) = m + \bar{a}.q = \overline{\bar{m} / (\bar{a} / q)} = \overline{m \downarrow (a \downarrow \bar{q})}$$

t : Situation présente. $t + \tau$: Situation futur.

Diagramme de fluence:

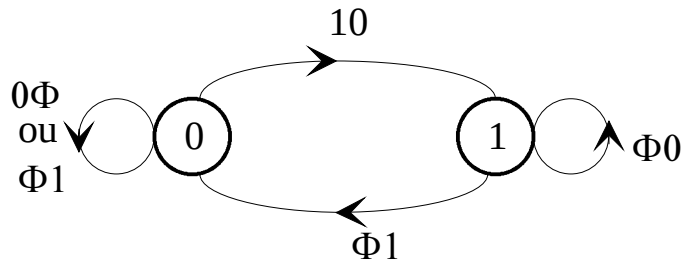


B.2.2. Bistable asynchrone à arrêt prioritaire :

q(t) / ***ma***	**0**	**1**	
00	0	1	
01	0	0	
11	0	0	
10	1	1	
			Q(t + τ)

$$Q(t+\tau) = m.\bar{a} + \bar{a}.q = (m / \bar{a}) / (\bar{a} / q)$$

Diagramme de fluence:



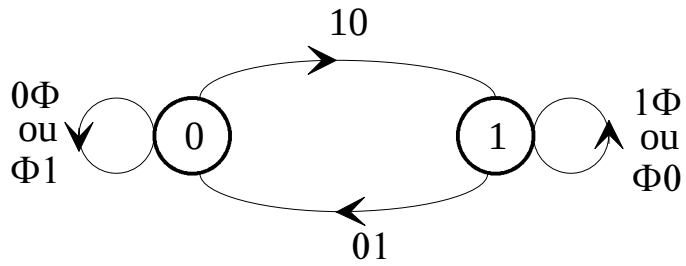
B.2.3. Bistable asynchrone à prioritaire à l'état intérieur

q(t) / *ma*	**0**	**1**	
00	0	1	
01	0	0	
11	0	1	
10	1	1	
			Q(t + τ)

$$Q(t+\tau) = m.\bar{a} + \bar{a}.q + m.q = (m/\bar{a})/(\bar{a}/q)/(m/q)$$

$$= \left[\overline{(m/\bar{a})/(\bar{a}/q)}\right]/(m/q)$$

Diagramme de fluence:

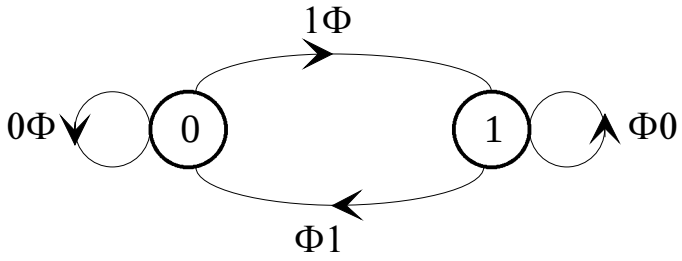


B.2.4. Bistable asynchrone à prioritaire au changement d'état

q(t) / *ma*	**0**	**1**	
00	0	1	
01	0	0	
11	1	0	
10	1	1	
			Q(t + τ)

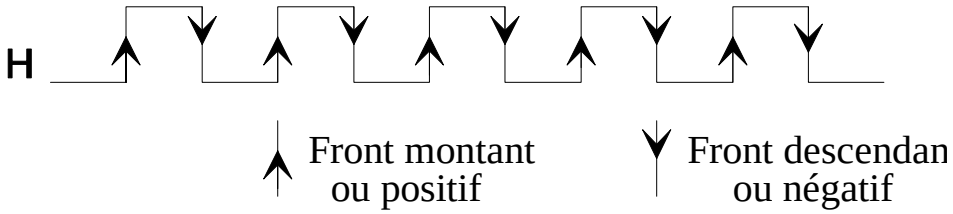
$$Q(t+\tau) = m.\bar{q} + \bar{a}.q = (m/\bar{q})/(\bar{a}/q)$$

Diagramme de fluence:



B.3. Bistables synchrones

Dans une bascule synchrone, l'exécution de l'ordre n'intervient qu'avec un signal de synchronisation. Ce signal est appelé horloge ″**H**″ ou clock pulse ″**Cp**″ ou timing ″**T**″.



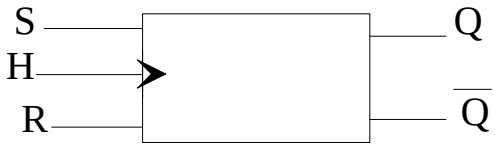
Symboles

- Horloge agissant pendant toute la durée du niveau haut "H"
- Horloge agissant pendant toute la durée du niveau bas "L"
- Horloge agissant au voisinage de la transition "L → H"
- Horloge agissant au voisinage de la transition "H → L"

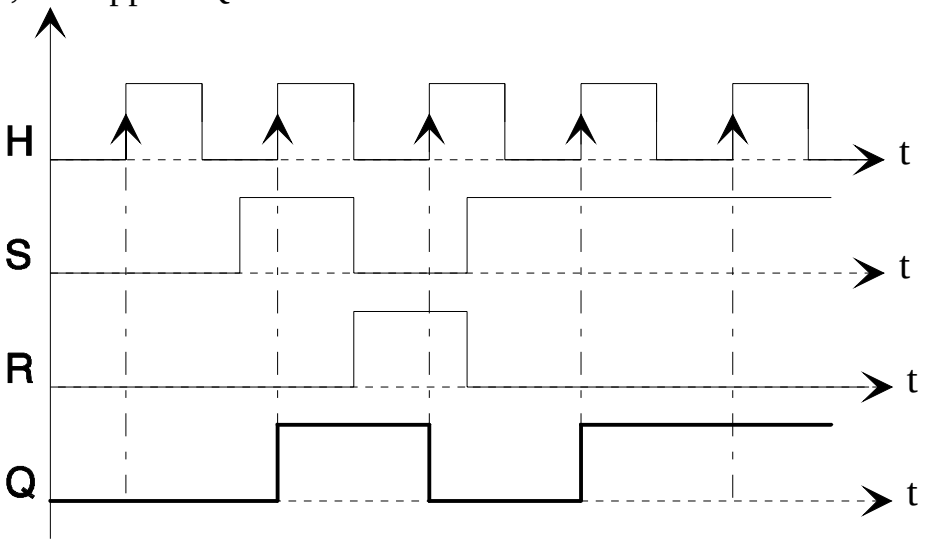
Exemples

1. Bascule RSH à front montant et chronogrammes

S: SET (Mise à 1) et R: RESET (remise à zéro)

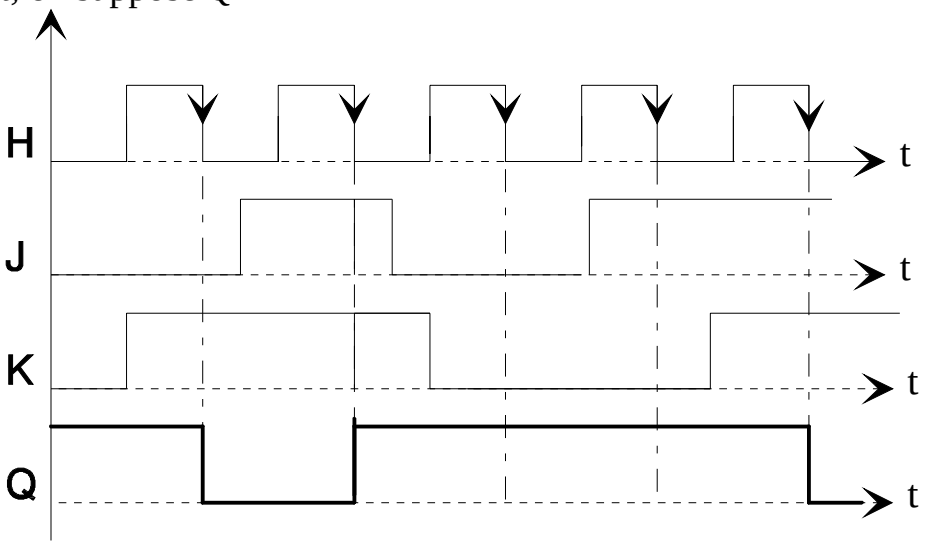


Au départ, on suppose *Q = 0*



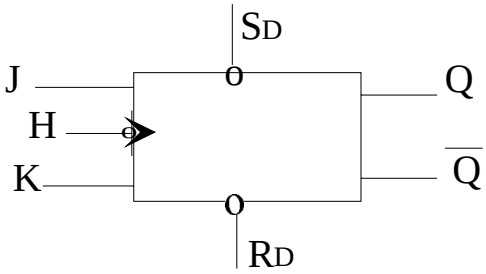
2. Bascule JKH à front descendant et chronogrammes

Au départ, on suppose *Q = 1*



Remarque

En général, les bascules synchrones comportent deux entrées supplémentaires dites asynchrones, qui sont prioritaires par rapport aux entrées synchrones.



S_D : entrée asynchrone de forçage à 1 (PRESET)
R_D : entrée asynchrone de forçage à zéro (CLEAR)

* Si $S_D = 0$ et $R_D = 1 \Rightarrow Q = 1 \; \forall$ J, K et H
* Si $S_D = 1$ et $R_D = 0 \Rightarrow Q = 0 \; \forall$ J, K et H

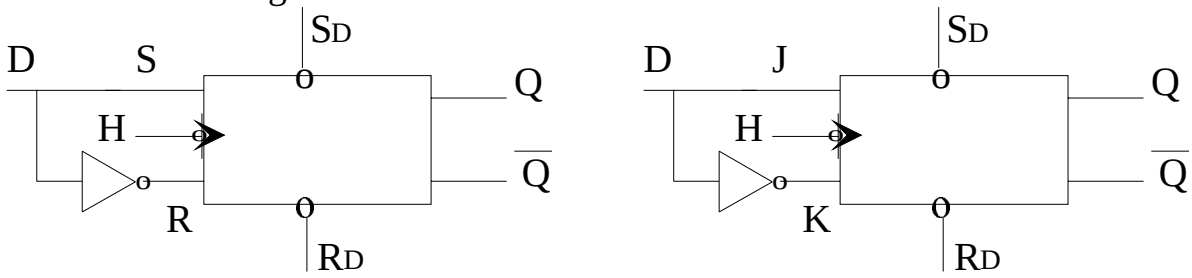
La bascule est commandé par J, K et H seulement dans le cas $S_D = R_D = 1$.

B.3.1. Bistables dérivées

On utilise coramment d'autres fonctions séquentielles qui sont des applications ou des cas particuliers des fonctions élémentaires RS et JK.

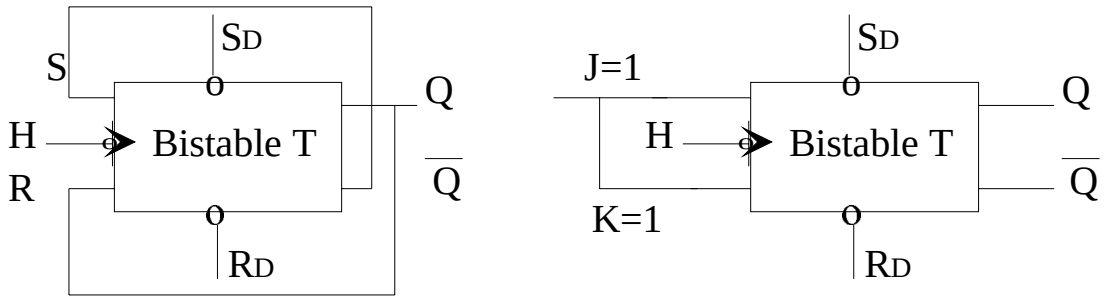
a. Bistable D

C'est typiquement la mémoire unitaire, dont la seule fonction est l'enregistrement et la conservation d'un bit 1 ou 0.

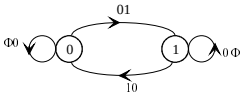
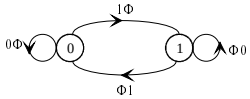


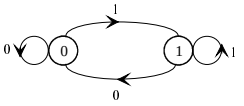
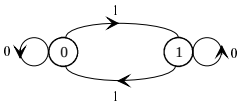
b. Bistable T

Elle n'a qu'une seule entrée T, qui est en fait l'entrée de synchronisation sur laquelle on envoie des impulsions de commande. sa sortie change de valeur à chaque impulsion arrivant sur H. Ce changement peut s'effectuer soit sur le front montant soit sur une front descendant, selon la réalisation de l'opérateur.

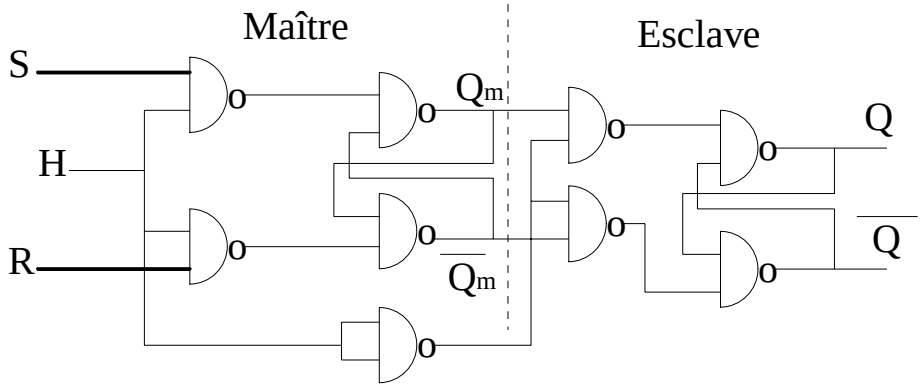


B.3.2. Récapitulation

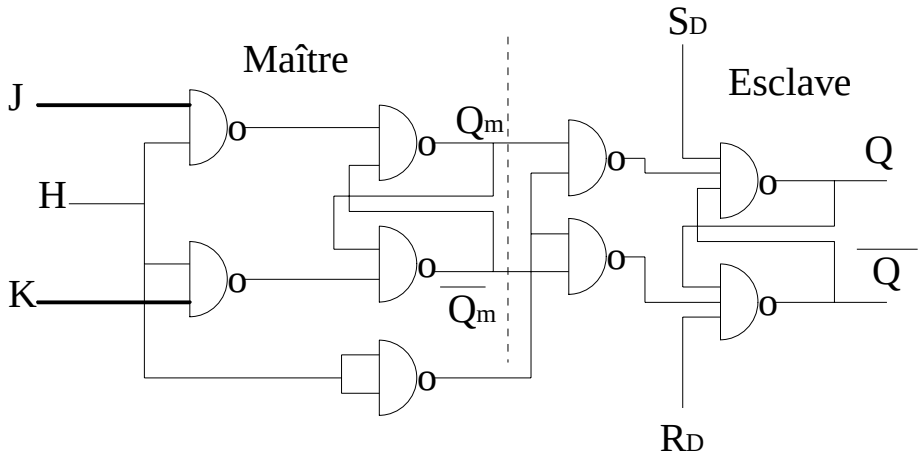
Type	Table de vérité			Table d'éxcitation			Diagramme de fluence	Equation
	S	**R**	Q_{n+1}	**S**	**R**			
	0	0	Q_n	1	0	ε		
RS	1	0	1	0	1	δ		$Q_{n+1} = S + \overline{R}.Q_n$
	0	1	0	0	ϕ	μ_0		
	1	1	--	ϕ	0	μ_1		
	J	**K**	Q_{n+1}	**J**	**K**			
	0	0	Q_n	1	ϕ	ε		
JK	1	0	1	ϕ	1	δ		$Q_{n+1} = J\overline{Q_n} + \overline{K}.Q_n$
	0	1	0	0	ϕ	μ_0		
	1	1	$\overline{Q_n}$	ϕ	0	μ_1		

Type	Table de vérité		Table d'éxcitation			Diagramme de fluence	Equation
	D	Q_{n+1}	**D**	Q_n			
	0	0	1	0	ε		
D	1	1	0	1	δ		$Q_{n+1} = D$
			0	0	μ_0		
			1	1	μ_1		
	T	Q_{n+1}	**T**	Q_n			
	0	0	1	0	ε		
T	1	1	1	1	δ		$Q_{n+1} = \overline{TQ_n} + \overline{T}.Q_n$
			0	0	μ_0		
			0	1	μ_1		

B.3.3. Bistable RS Maître esclave



B.3.4. Bistable JK Maître esclave avec entrée de forçage



Fonctionnement statique d'un bascule JK

Mode	S_D	R_D	H	J	K	Q_{n+1}	Remarque
Asynchrone	0	1	x	x	x	1	μ_1
	1	0	x	x	x	0	μ_0
	0	0	x	x	x	--	instable
Synchrone	1	1	↓	0	0	$\mathbf{Q_n}$	Maintien
	1	1	↓	0	1	0	δ
	1	1	↓	1	0	1	ε
	1	1	↓	1	1	$\overline{\mathbf{Q_n}}$	Commutation
	1	1	0 ou 1	x	x	$\mathbf{Q_n}$	Maintien

C. Synthèse et mise en œuvre des compteurs

C.1. Introduction

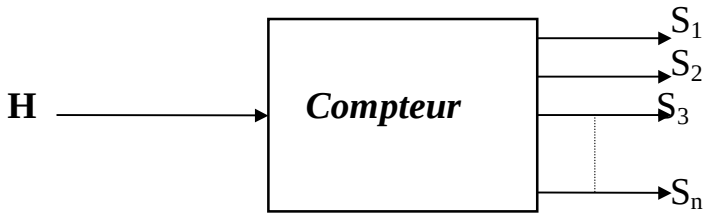
Un compteur est un dispositif destiné à enregistrer le résultat d'un comptage d'impulsions. Le nom de compteur désigne, généralement un nombre binaire croissant en fonction des impulsions d'entrée.

Un compteur est en général, constitué de deux parties essentielles :

* un ensemble de n bascules dont le nombre est fonction du modulo du compteur m tel que : $2^{n-1} < m \leq 2^n$.

* Un réseau combinatoire qui relie entre les différentes bascules de manière à caractériser le comptage, c'est à dire à obtenir après chaque impulsion, les sorties correspondantes au code choisi.

C.2. Modélisation



C.3. Compteurs asynchrones

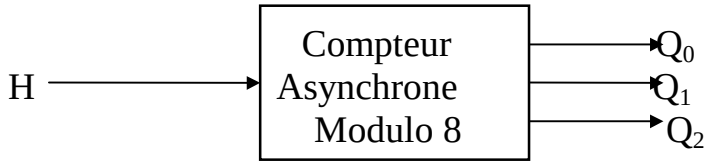
Les impulsions s'appliquent à l'entrée de la première bascule, de poids faible. Sa sortie est appliquée à l'entrée horloge de la bascule suivante, et ainsi de suite jusqu'à la dernière bascule.

Le modulo est un nombre indiquant le nombre d'état compté, ou bien, le nombre par lequel la fréquence est divisée à la sortie de la dernière bascule.

Un compteur modulo 2^n contient les 2^n états possibles. Il compte jusqu'à (2^n-1) avant d'être initialisé.

C.3.1. Exemple : compteur modulo 8

En utilisant une méthode systématique, on va faire la synthèse d'un compteur asynchrone modulo 8 dans le code binaire naturel.



a. Table des états

Décimal	Q_2	Q_1	Q_0	$H_0 = H$	$H_1 = Q_0$	$H_2 = Q_1$
0	0	0	0	**H** ↘		
1	0	0	1 ↓	**H** ↘	**Q_0** ↘	
2	0	1	0	**H** ↘		
3	0	1 ↓	1 ↓	**H** ↘	**Q_0** ↘	**Q_1** ↘
4	1	0	0	**H** ↘		
5	1	0	1 ↓	**H** ↘	**Q_0** ↘	
6	1	1	0	**H** ↘		
7	1 ↓	1 ↓	1 ↓	**H** ↘	**Q_0** ↘	**Q_1** ↘
0	0	0	0			

b. Table de succession des états

Q_2 \ $Q_1 Q_0$	**00**	**01**	**11**	**10**	
0	001	010	100	011	
1	101	110	000	111	
					$Q_i(t + \tau)$

Q_2 \ $Q_1 Q_0$	**00**	**01**	**11**	**10**	
0	xxε	xεδ	εδδ	xxε	
1	xxε	xεδ	δδδ	xxε	
					$Q_i(t + \tau)$

Remarque

* Pour déterminer l'expression de J, on prend seulement les enclenchements (ε), les déclenchements (δ) et les maintient à 1 (μ_1).

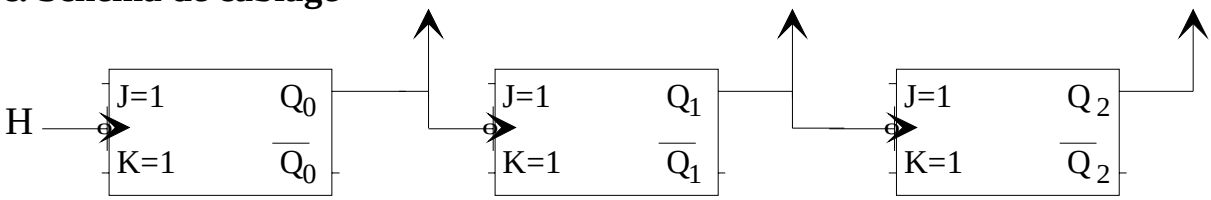
* Pour déterminer l'expression de K, on prend seulement les déclenchements (δ), les enclenchements (ε) et les maintient à 0 (μ_0).

Donc, on a : $J_0 = J_1 = J_2 = 1$ & $K_0 = K_1 = K_2 = 1$

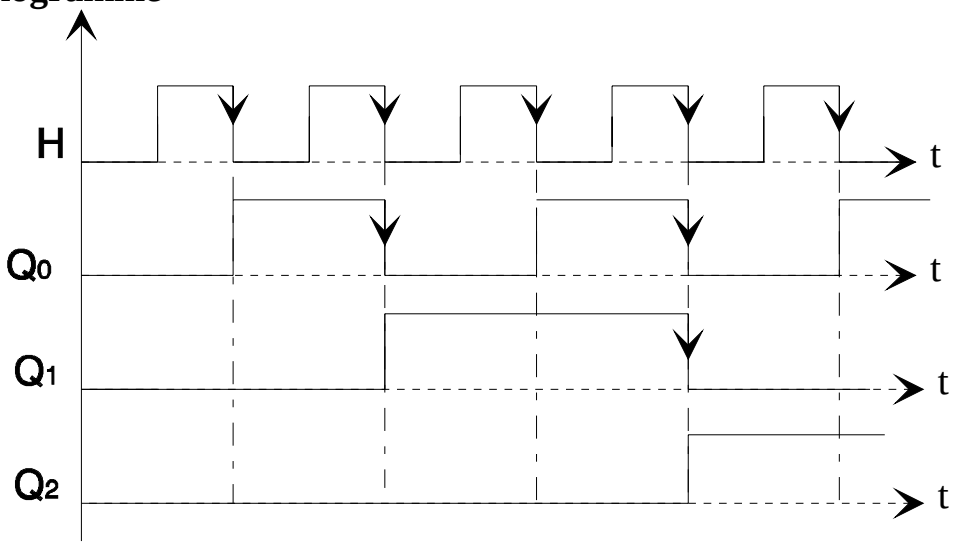
Pour la synthèse d'un compteur asynchrone de capacité une puissance de 2 on a:

$$J_i = K_j = 1 \quad \forall \ i \ et \ j$$

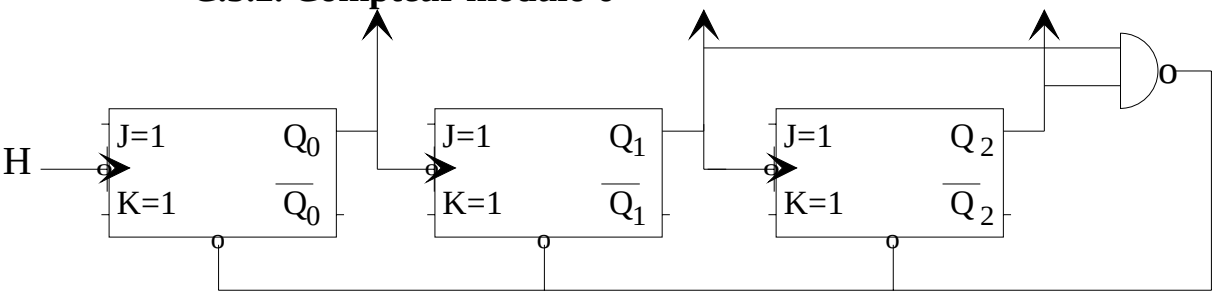
c. Schéma de câblage



d. chronogramme



C.3.2. Compteur modulo 6



C.4. Compteurs synchrones

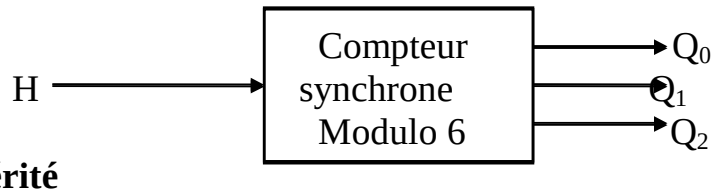
Dans le compteur synchrone, toutes les bascules sont commandées en même temps par le même signal d'horloge. le basculement des différents étages s'effectue donc au même moment puisque l'on dispose d'une horloge commune. On élimine ainsi l'inconvénient rencontré dans les compteurs asynchrone, dans lequel les temps de commutation s'ajoutaient et rendaient impossible le comptage en fréquence élevée.

Pour réaliser un compteur, on procède de la manière suivante :

* définition de la capacité maximale du compteur: c'est à dire le plus grand nombre binaire que le compteur peut afficher. Ceci permet de déterminer le nombre de bascules à utiliser.
* choix du code binaire à utiliser.

* A l'aide des tables des transitions des bascules utilisées, déterminer les valeurs d'entrée de chacune des bascules qui définissent l'état du compteur à chaque impulsion d'avancement.
* Porter les valeurs des entrées dans un tableau de Karnaugh afin de déterminer les équations simplifiées à réaliser sur chacune des entrées.

C.4.1. Mode de fonctionnement unique



a. Table de vérité

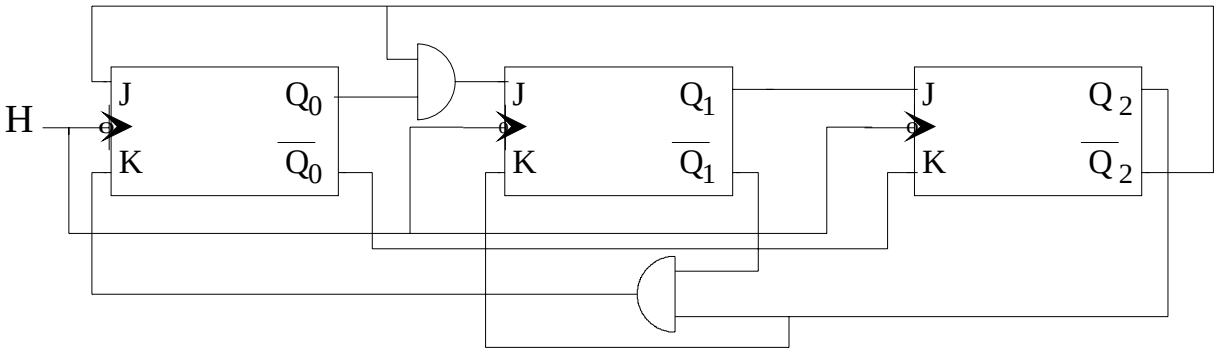
Impulsion	Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
0	0	0	0	0	0	1
1	0	0	1	0	1	1
2	0	1	1	1	1	1
	0	1	0			
	1	1	0			
3	1	1	1	1	0	1
4	1	0	1	1	0	0
5	1	0	0	0	0	0

b. Table de succession des états codées

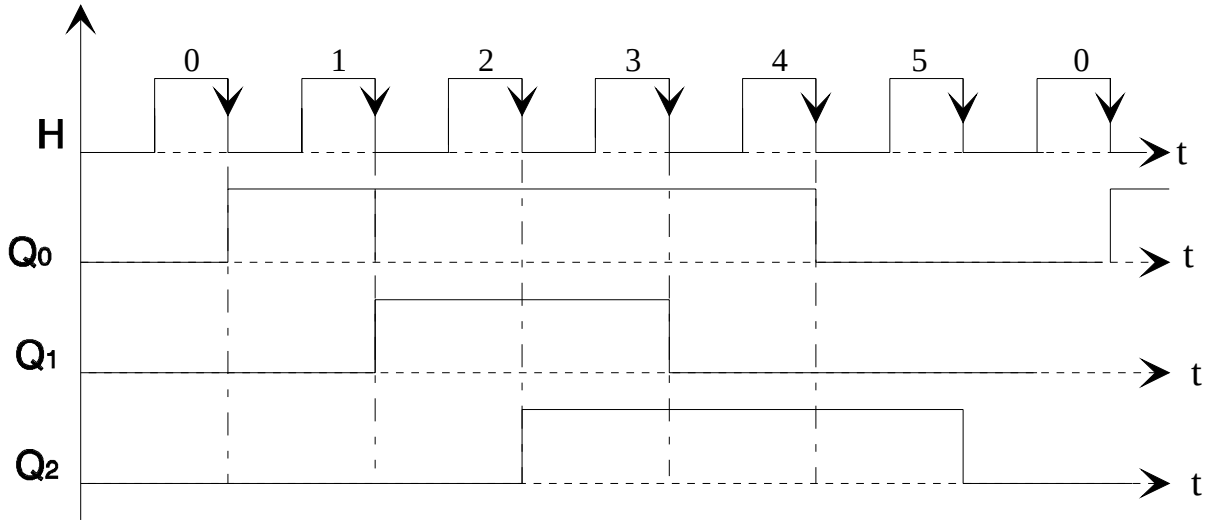
S.F / **S.A**	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
000	μ_0	μ_0	ε
001	μ_0	ε	μ_1
011	ε	μ_1	μ_1
010	--	--	--
110	--	--	--
111	μ_1	δ	μ_1
101	μ_1	μ_0	δ
100	δ	μ_0	μ_0

$J_0 = \overline{Q_2}$; $K_0 = Q_2\overline{Q_1}$, $J_1 = \overline{Q_2}Q_0$; $K_1 = Q_2$, $J_2 = Q_1$; $K_2 = \overline{Q_0}$

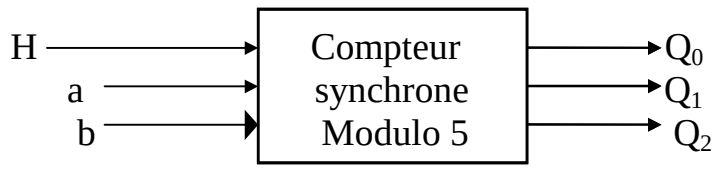
c. Schéma de câblage



d. chronogramme



C.4.2. Mode de fonctionnement multiple



a / b	0	1
0	**Remise à zéro R.A.Z.**	**Comptage**
1	**Décomptage**	**Arrêt**

a. Table de succession des états

S.F.

ab / S.A.	00	01	11	10
0	0	4	0	1
1	0	0	1	2
2	0	1	2	3
3	0	2	3	4
4	0	3	4	0

b. Table de succession des états codées

ab / $Q_2^n\ Q_1^n\ Q_0^n$	00	01	11	10
000	000	100	000	001
001	000	000	001	010
011	000	010	011	100
010	000	001	010	011
110	--	--	--	--
111	--	--	--	--
101	--	--	--	--
100	000	011	100	000

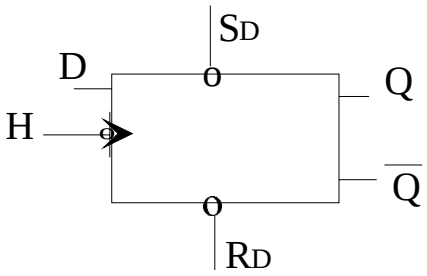
D. Synthèse et mise en œuvre des registres

D.1. Définition

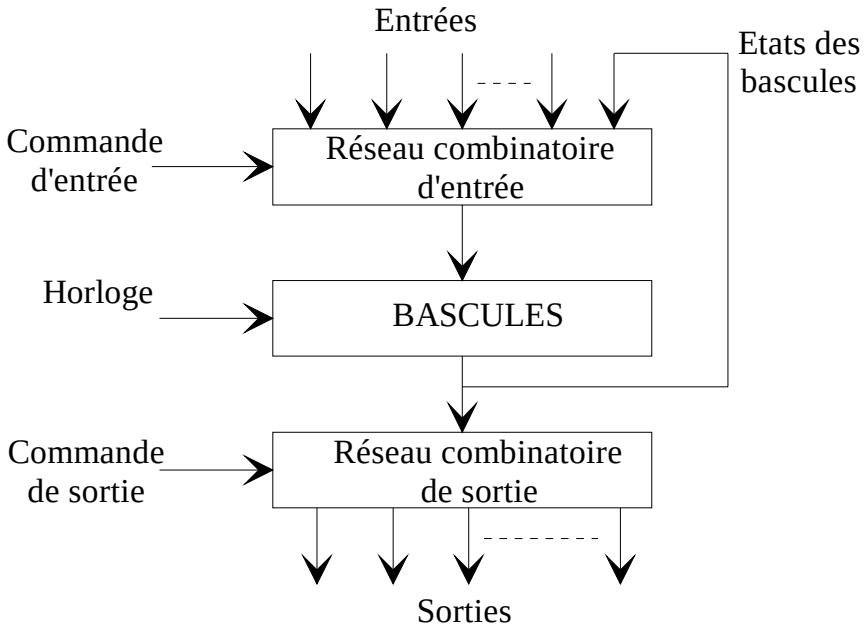
Un registre est un système séquentiel synchrone permettant le stockage des données binaires jusqu'à leur traitement ou leur expédition: c'est une mémoire formé par des bistables.

Il est spécifié par : * le nombre de bits de sortie,
* le mode de fonctionnement.

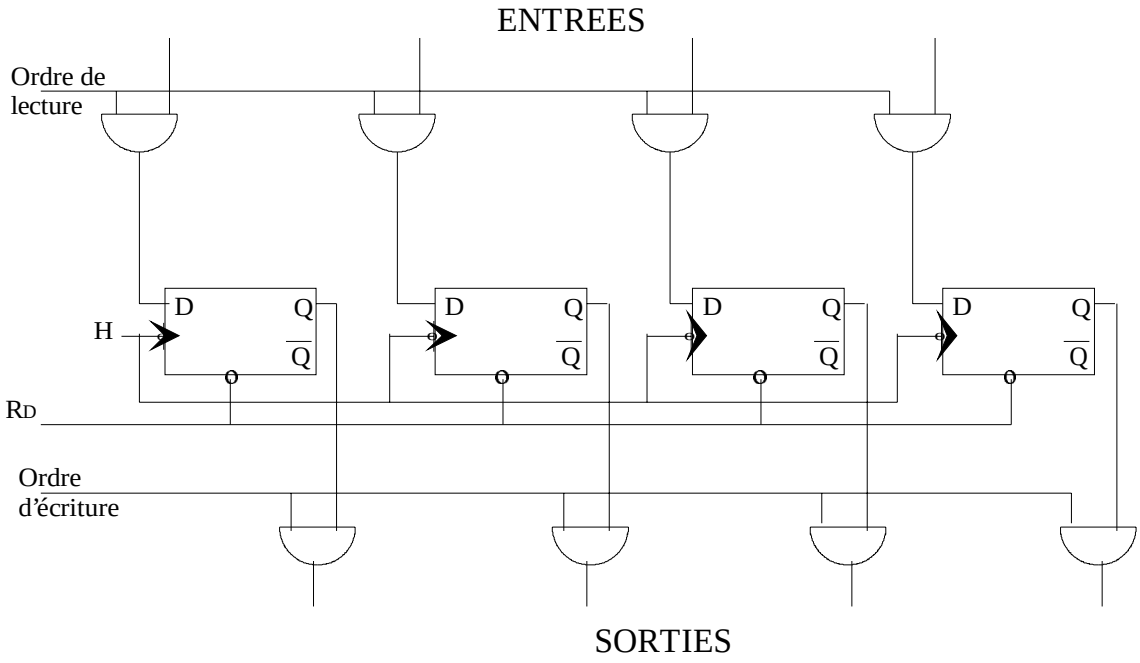
D.2. Registre élémentaire: $Q^{n+1} = D$



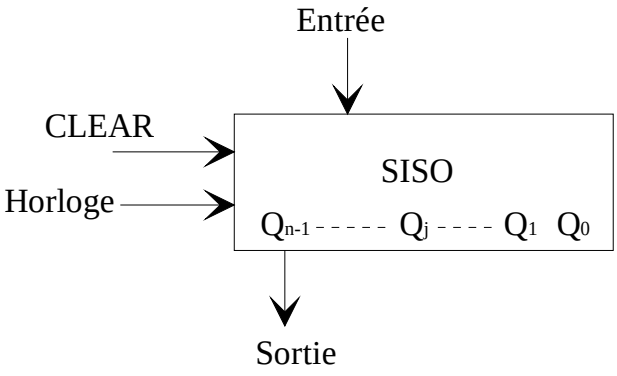
D.3. Structure d'un registre à écriture et lecture // **(PIPO)**



Exemple de registre PIPO



***D.4. Registre à écriture et lecture série : décalage* (SISO)**



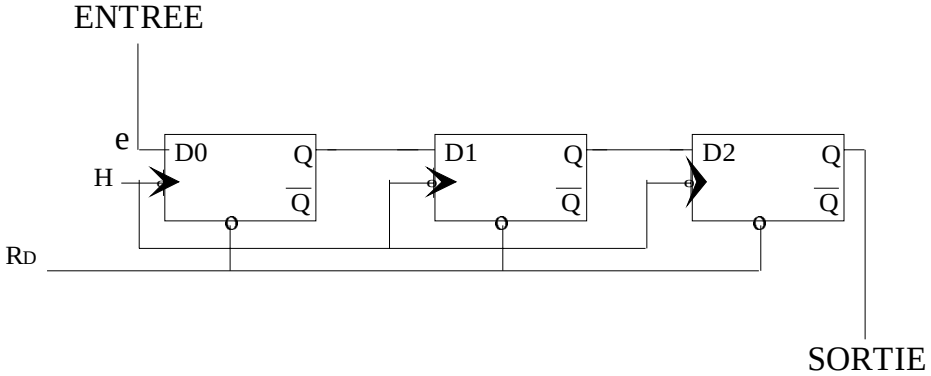
Exemple d'un registre SISO à décalage droite

a. Table de succession des états

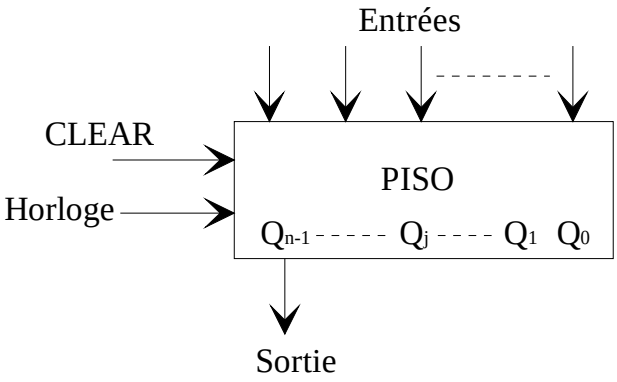
$e\ Q_0^n$ / $Q_1^n\ Q_2^n$	**00**	**01**	**11**	**10**
00	000	010	110	100
01	000	010	110	100
11	001	011	111	101
10	001	011	111	101

$$D_0 = e,\ D_1 = Q_0,\ \ D_2 = Q_1$$

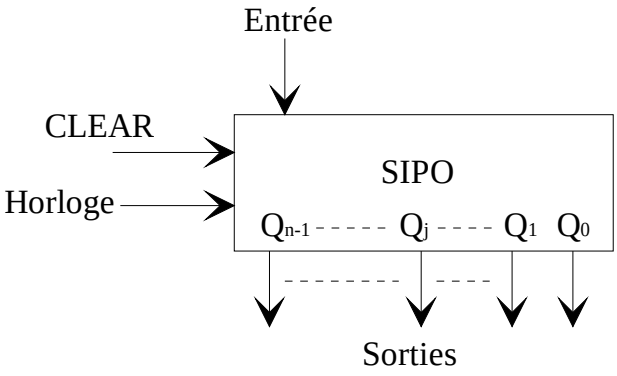
b. Schéma de câblage



***D.5. Registre à écriture // et lecture série :* (PISO)**

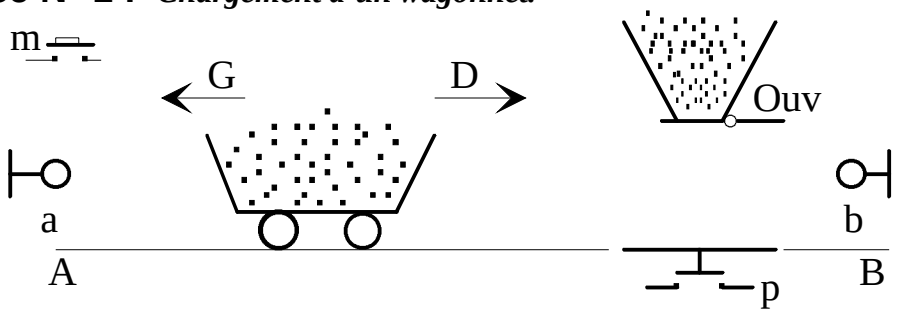


***D.6. Registre à écriture série et lecture // :* (SIPO)**



Série n°1

Exercice N° 1 : *Chargement d'un wagonnet.*

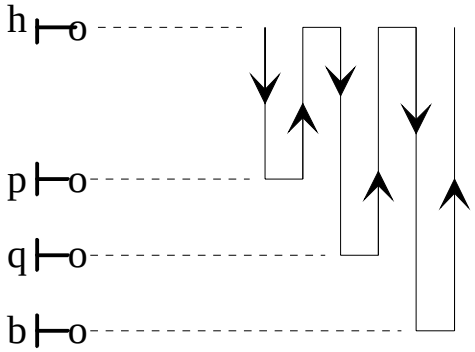


Un wagonnet peut se déplacer entre deux points *A* et *B*. En *A*, un opérateur peut demander le chargement du wagonnet par l'appui fugitif sur un bouton poussoir marche (m). Le wagonnet va jusqu'au point *B*. Lorsqu'il y arrive, le chargement s'effectue par l'ouverture d'une trémie. Dès que le chargement est terminé, la trémie se referme et le wagonnet revient jusqu'au *A* où sa charge est utilisée. Il repartira quand un nouveau chargement sera demandé par l'opérateur. A l'état initial, le wagonnet est en attente au point *A*.

Représenter le fonctionnement de cet automatisme par la méthode d'Huffman.

Exercice N° 2 : *Usinage d'une pièce.*

Sur une perceuse, on désire réaliser l'usinage suivant de manière automatique (seuls la mise en place et le retrait de la pièce seront manuels).



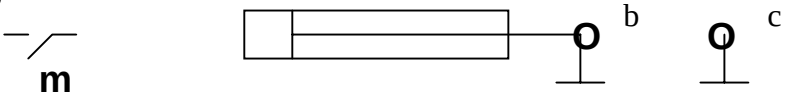
Les différents positions de la broche seront détectées au moyen de quatre capteurs : h (position haute), b (position basse), p et q (position intermédiaires).

Le début de l'usinage aura lieu lors de l'appui fugitif sur un bouton poussoir marche (m).

La bâti de la perceuse, la rotation de la broche assurée par un moteur asynchrone triphasé.

En appliquant la méthode d'Huffman, décrire le fonctionnement de cette perceuse.

Exercice N° 3



Lorsqu'on appui et on relâche le bouton poussoir m, la tige d'un vérin effectue un seul mouvement de va et vient et s'arrête. L'appui sur m au cours du mouvement n'a aucune influence sur le déroulement du cycle.

Déterminer les équations correspondantes du système.

Exercice N° 4

La commande d'ouverture d'une porte S = 1 est assuré par les séquences suivantes :

ab	00	01	11	10	00
S	0	0	0	0	1

Pour mettre S = 0 il suffit d'appuyer soit sur a soit sur b.
Faire la synthèse de cet automatisme par la méthode d'Huffman.

Exercice N° 5

Faire la synthèse d'un compteur BCD_{8421} avec des bascules D. Fonctionnement au front montant.

Exercice N° 6

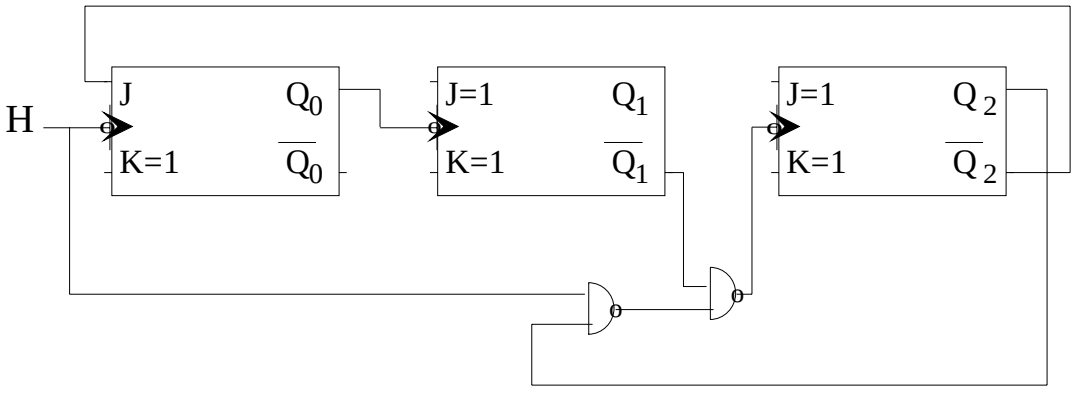
On désire diviser par 60, la fréquence d'un signal d'horloge H. On utilise à cette fin, un compteur asynchrone matérialisé par des bistables JKH↓ et des portes NAND à deux entrées. Faire la synthèse de ce système et proposer une réalisation.

Exercice N° 7

Réaliser un compteur qui compte une décade (10 impulsions) dans le code binaire naturel. Utiliser des bascules JKH↓.

Exercice N° 8

Soit le compteur asynchrone suivant :

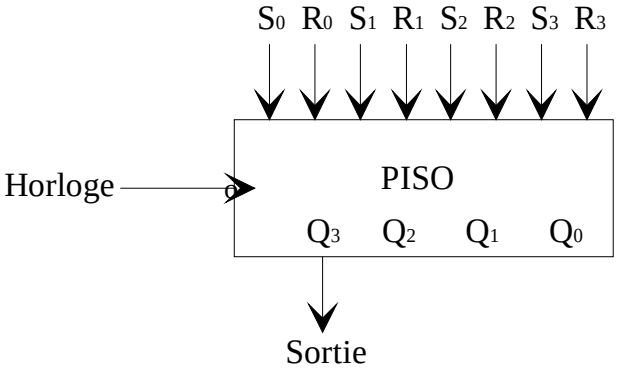


1. Tracer le diagramme temporel des sorties à partir de l'état 000. De quel modulo est ce compteur.

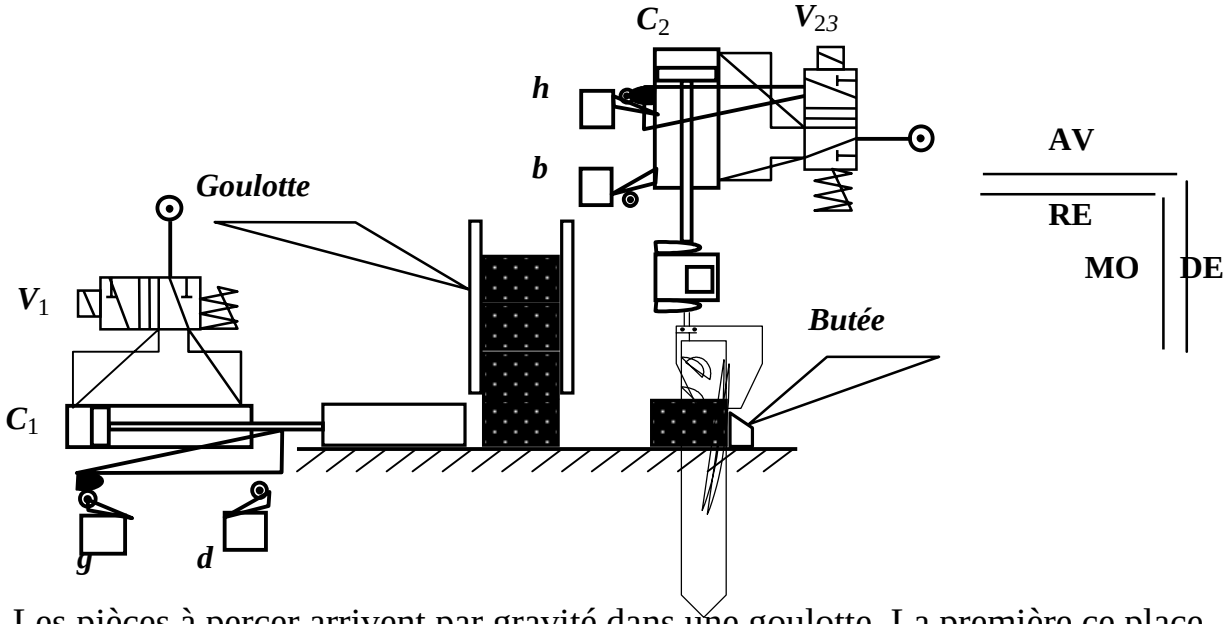
2. En déduire un compteur modulo 9.

Exercice N° 9

Faire la synthèse d'un registre à entrées parallèles-sorties séries (PISO). Utiliser les entrées de prédisposition S **(Set)** et R **(Reset).**



Exercice N°10 : poste de perçage automatisé : Cycle en L



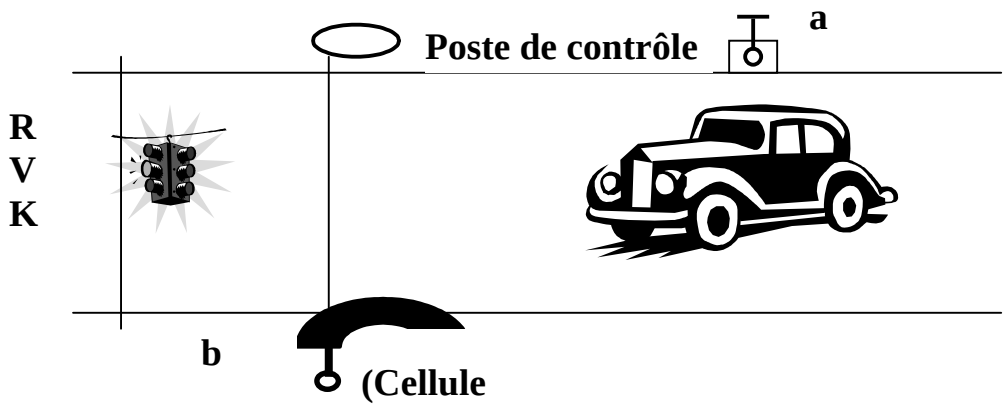
Les pièces à percer arrivent par gravité dans une goulotte. La première ce place devant la tige du vérin C_1 en position initiale (g actionné). Si la perceuse est au point haut, la pièce est poussée et maintenue serrée en position de perçage. La perceuse qui est en rotation permanente, descend, effectue le perçage et remonte. La première pièce est éjectée et la suivante dont l'avance était contrariée par la garde du vérin vient se mettre en place pour permettre la reprise du cycle dès la rentrée de la tige du vérin C_1 pour actionner le capteur g.

Retrouver, par la méthode d'Huffman, les équations de commande suivantes :

$$\begin{cases} X = d.h + x.\overline{b} \\ V_1 = g.\overline{h} + x \\ V_2 = \overline{x} + \overline{g} \end{cases}$$

Exercice N° 11

Pour emprunter une autoroute à péage, l'automobiliste doit s'arrêter au poste de contrôle, déposer une pièce de monnaie dans un panier placé à cet effet. Le feu rouge disparaît alors que le feu vert s'allume. L'automobiliste peut alors emprunter l'autoroute.



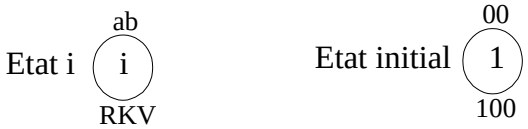
Le fonctionnement de ce poste de péage est le suivant:

L'automobiliste s'arrête, dépose une pièce qui agit sur le passage sur le contact « **a »** (impulsion: la variable « **a** » passe à 1 puis revient à 0) et le feu vert s'allume (V=1); il avance, interceptant le faisceau de la cellule b (b=1) qui maintient le feu vert; lorsque la voiture n'intercepte plus le faisceau de cellule, le feu rouge s'allume (R=1) et le vert s'éteint.

Si l'automobiliste s'avance par distraction devant la cellule b sans payer, la Klaxon retentit (K=1) et le feu rouge reste allumé (R=1). Le conducteur fait marche arrière pour se placer face au panier a, le Klaxon retentit, le feu reste rouge. Dès que la pièce est introduite dans le panier, le feu vert s'allume (V=1) et le conducteur peut continuer sa route dans les conditions normales spécifiées dans le paragraphe précèdent.

Dans le cas où la pièce de monnaie n'ayant pas été déposée, le Klaxon est actionné par la cellule b et l'automobiliste ne peut reculer parce que la voiture suivante le gêne. Il descend de voiture et met la pièce; alors le feu vert s'allume, le Klaxon s'arrête et il peut reprendre sa route.

On demande de faire la synthèse de cet automatisme en utilisant la méthode d'Huffman.



Chapitre II

GRAFCET

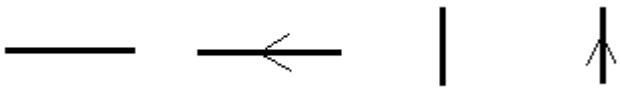
I - Introduction

Le Grafcet est un outil graphique de définition pour l'automatisme séquentiel, en tout ou rien. Mais il est également utilisé dans beaucoup de cas combinatoires, dans le cas où il y a une séquence à respecter mais où l'état des capteurs suffirait pour résoudre le problème en combinatoire. Il utilise une représentation graphique. C'est un langage clair, strict mais sans ambiguïté, permettant par exemple au réalisateur de montrer au donneur d'ordre comment il a compris le cahier des charges. Langage universel, indépendant (dans un premier temps) de la réalisation pratique (peut se "câbler" par séquenceurs, être programmé sur automate voire sur ordinateur).

II - définitions

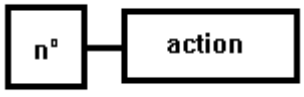
Un Grafcet est composé d'étapes, de transitions et de liaisons.

Une LIAISON est un arc orienté (ne peut être parcouru que dans un sens). A une extrémité d'une liaison il y a UNE (et une seule) étape, à l'autre UNE transition. On la représente par un trait plein rectiligne, vertical ou horizontal. Une verticale est parcourue de haut en bas, sinon il faut le préciser par une flèche. Une horizontale est parcourue de gauche à droite, sinon le préciser par une flèche.

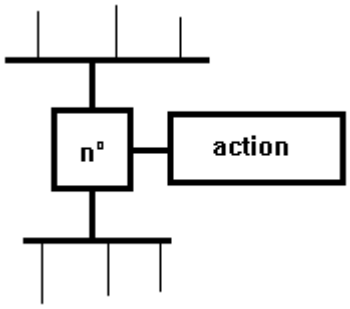


Une ETAPE correspond à une phase durant laquelle on effectue une ACTION pendant une certaine DUREE (même faible mais jamais nulle). L'action doit être stable, c'est à dire que l'on fait la même chose pendant toute la durée de l'étape, mais la notion d'action est assez large, en particulier composition de plusieurs actions, ou à l'opposé l'inaction (étape dite d'attente).

On représente chaque étape par un carré, l'action est représentée dans un rectangle à gauche, l'entrée se fait par le haut et la sortie par le bas. On numérote chaque étape par un entier positif, mais pas nécessairement croissant par pas de 1, il faut simplement que jamais deux étapes différentes n'aient le même numéro.



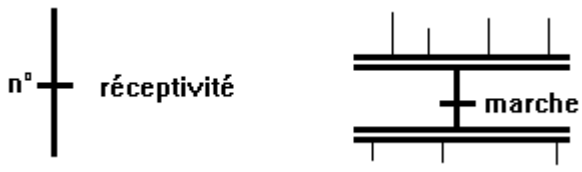
Si plusieurs liaisons arrivent sur une étape, pour plus de clarté on les fait arriver sur une barre horizontale, de même pour plusieurs liaisons partant de l'étape. Cette barre horizontale n'est pas une nouvelle entité du Grafcet, elle fait partie de l'étape, et ne représente qu'un "agrandissement" de la face supérieure (ou inférieure) de l'étape. On accepte de remplacer cette barre par un point si cela ne crée aucune ambiguïté.



Une étape est dite active lorsqu'elle correspond à une phase "en fonctionnement", c'est à dire qu'elle effectue l'action qui lui est associée. On représente quelquefois une étape active à un inótant donné en dessinant un point à l'intérieur.

Une TRANSITION est une condition de passage d'une étape à une autre. Elle n'est que logique (dans son sens Vrai ou Faux), sans notion de durée. La condition est définie par une RECEPTIVITE qui est généralement une expression booléenne (c.à.d avec des ET et des OU) de l'état des CAPTEURS.

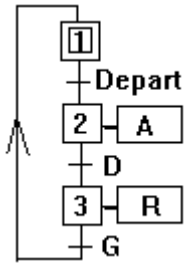
On représente une transition par un petit trait horizontal sur une liaison verticale. On note à droite la réceptivité, on peut noter à gauche un numéro de transition (entier positif, indépendant des numéros d'étapes). Dans le cas de plusieurs liaisons arrivant sur une transition, on les fait converger sur une grande double barre horizontale, qõi n'est qu'une représentation du dessus de la transition. De même pour plusieurs liaisons partant sous une transition.



III - exemple simple

Supposons un chariot pouvant avancer (A) ou reculer (R) sur un rail limité par deux capteurs G et D, Un cahier des charges pourrait être : Quand on appuie sur le bouton DEPART, on avance jusqu'en D, puis on revient. Ce C.d.C. est incomplet et imprécis. La réalisation du Grafcet permet de remarquer : Que fait-

on avant l'appui de DEPART, jusqu'où revient-on, quelles sont les conditions initiales ? On réécrit le C.d.C. en 3 phases : Attendre jusqu'à l'appui de DEPART, avancer jusqu'en D, reculer jusqu'en G, attendre à nouveau DEPART et recommencer. On suppose le chariot initialement en G (sinon faire un cycle l'amenant en G).



IV - règles d'évolution

La modification de l'état de l'automatisme est appelée évolution, et est régie par 5 règles :

R1 : Les étapes INITIALES sont celles qui sont actives au début du fonctionnement. On les représente en doublant les côtés des symboles. On appelle début du fonctionnement le moment où le système n'a pas besoin de se souvenir de ce qui c'est passé auparavant (allumage du système, bouton "reset",...). Les étapes initiales sont souvent des étapes d'attente pour ne pas effectuer une action dangereuse par exemple à la fin d'une panne de secteur.

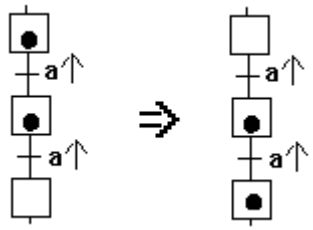
R2 : Une TRANSITION est soit validée, soit non validée (et pas à moitié validée). Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives (toutes celles reliées directement à la double barre supérieure de la transition). Elle ne peut être FRANCHIE que lorsqu'elle est validée et que sa réceptivité est vraie. Elle est alors obligatoirement franchie.

R3 : Le FRANCHISSEMENT d'une transition entraîne l'activation de TOUTES les étapes immédiatement suivante et la désactivation de TOUTES les étapes immédiatement précédentes (TOUTES se limitant à 1 s'il n'y a pas de double barre).

R4 : Plusieurs transitions SIMULTANEMENT franchissables sont simultanément franchies (ou du moins toutes franchies dans un laps de temps négligeable pour le fonctionnement). La durée limite dépend du "temps de réponse" nécessaire à l'application (très différent entre un système de poursuite de missile et une ouverture de serre quand le soleil est suffisant).

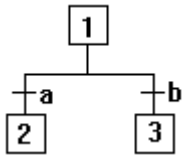
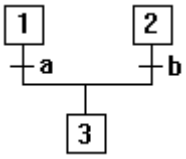
R5 : Si une étape doit être à la fois activée et désactivée, elle RESTE active. Une temporisation ou un compteur actionnés par cette étape ne seraient pas

réinitialisés. Cette règle est prévue pour lever toute ambiguïté dans certains cas particuliers qui pourraient arriver dans certains cas :

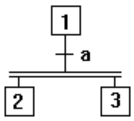


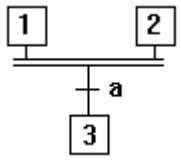
La partie COURS s'arrête ici. Toute autre règle que vous auriez pu entendre autre part ne fait pas partie du Grafcet. Il faudra TOUJOURS que votre Grafcet vérifie ce qui a été dit ci dessus (sinon ce n'est pas du Grafcet). Je tiens à préciser que le Grafcet devra être mis en oeuvre (câblé ou programmé) et donc une traduction de ce Grafcet en un schéma ou une suite d'instructions sera nécessaire. Le résultat de cette traduction, même s'il ressemble quelquefois à un Grafcet, ne peut pas imposer de nouvelles règles au Grafcet (qui dirait par exemple que le cas proposé après la règle 5 est interdit en Grafcet)

V - Configurations courantes

divergence en OU :	Convergence en OU :
 si 1 active et si a seul, alors désactivation de 1 et activation de 2, 3 inchangé. si a et b puis 1 active alors désactivation 1, activation 2 et 3 quel que soit leur état précédent. (règle 4)	 Si 1 active et a sans b, alors activation de 3 et désactivation de 1, 2 reste inchangé Si 1 et 2 et a et b alors 3 seule active

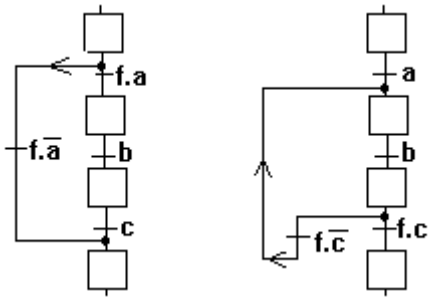
On appelle BARRE DE OU la barre symbolisant les entrées / sorties multiples d'étapes.

Divergence en ET :	Convergence en ET :
	

si 1 active et si a, alors désactivation de 1 et activation de 2 et 3.	 Si 1 active seule et a alors aucun changement. Si 1 et 2 et a, alors activation de 3 et désactivation de 1 et 2.

On appelle couramment BARRE DE ET la double barre, mais attention ce n'est pas une entité à part mais une partie d'une transition.

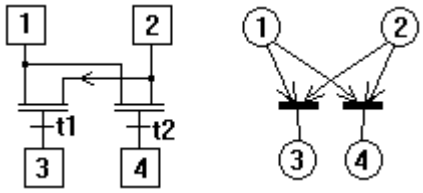
Détaillons également le saut avant (si a alors ...) et les boucles (répéter ... jusqu'à c). Ce sont les deux seules possibilités avec des OU: il ne peut y avoir de divergence en ou après une transition



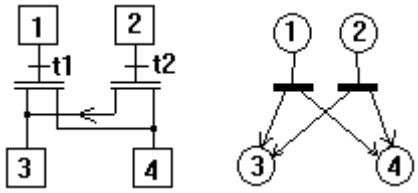
Passons maintenant à quelques problèmes plus complexes (tirés de "Comprendre et maîtriser le Grafcet, Blanchard, ed. Capadues"):

1- soient 4 étapes 1 à 4 et deux transitions de réceptivité t1 et t2. Construire la portion de Grafcet réalisant : Quand 1 ET 2 actifs alors
si t1 passer en 3 (et désactiver 1 et 2),
si t2 passer en 4 (et désactiver 1 et 2),
sinon rester en 1 et 2

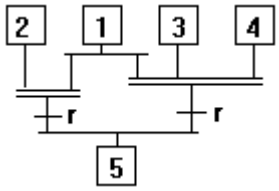
La solution ci-dessous est accompagnée d'une représentation de type "réseau de Petri" pour bien montrer où doivent se placer les convergences et divergences (à quoi doit être reliée 1?, à quoi doit être reliée t1? ...). En fait on trouve la solution facilement en analysant les cas d'évolution (quand franchit t'on t1 ?). Il faut souligner que l'ajout d'une étape intermédiaire n'est pas une bonne solution car tout passage d'une étape dure un laps de temps (donc discontinuité sur les sorties = aléa technologique)..



2 - Problème du même ordre : Quand (étape 1 et t1) OU (étape 2 et t2) alors passer en 3 ET 4:



3 - si {étape 1 et [étape 2 ou (étapes 3 et 4)]} et transition t alors activer l'étape 5 (et désactiver les autres).

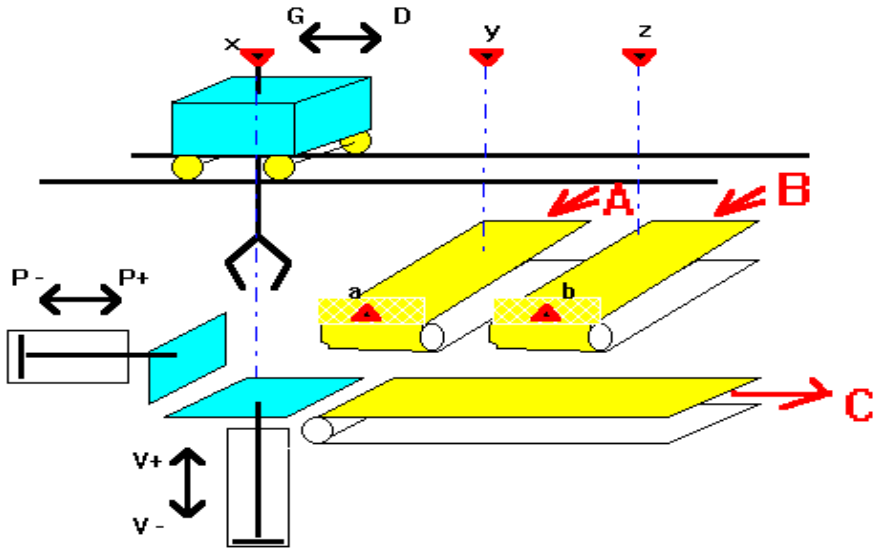


VI Cas génériques

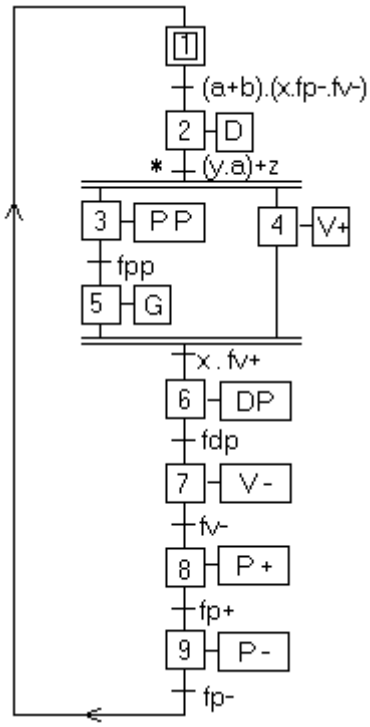
Nous traitons ici des exemples génériques, c'est à dire que les problèmes évoqués ici se posent assez souvent, et la méthode utilisée pour les résoudre pourra être réutilisée.

1 - priorité

Soit un chariot se déplaçant sur deux rails (action D vers la droite, G vers la gauche). Il comporte une pince pouvant prendre une pièce (PP, fin quand fpp) s'il se trouve sur le tapis A (capteur y) et qu'une pièce est présente (capteur a) (idem en z si b). Puis il retourne en x, pose la pièce (action DP, fin quand fdp) sur le plateaux supposé en position haute (fv+). Celui-ci descend (V-, jusqu'à fv-), un second vérin pousse la pièce (P+, fin quand fp+), puis le pousseur recule en fp-, le plateau remonte en fv+ Le tapis de sortie C est supposé toujours en mouvement. Les tapis A et B sont commandés par des systèmes non traités ici.



Effectuer d'abord un Grafcet linéaire comprenant une seule voie d'arrivée A. Puis l'améliorer en prévoyant les retours des actionneurs en temps masqué (attention toutefois de ne pas endommager le pousseur). Puis prévoir deux tapis d'alimentation A et B (en cas de pièces en a ET b, prendre celle en a). Puis prévoir une priorité tournante (en cas de conflit, prendre la voie qui n'a pas été servie la fois précédente) attention, si plusieurs pièces arrivent sur la même voie et aucune sur l'autre, ne pas bloquer le système. Puis modifier la règle de priorité en donnant en cas de conflit la priorité à celui qui n'en a pas profité lors du dernier conflit.

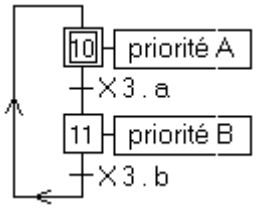


priorité voie A, retour V masqué

Pour gérer la priorité tournante, remplacer la réceptivité de la deuxième transition (notée *) par :

$y.a.(\overline{b}+X10)+z$

qui signifie : arrivé en y avec une pièce en a et soit pas de pièce en b, soit priorité. sinon on continue et quoi qu'il arrive on s'arrête en z (le chariot n'a pas de parachute), en rajoutant un second Grafcet définissant quelle voie est prioritaire :

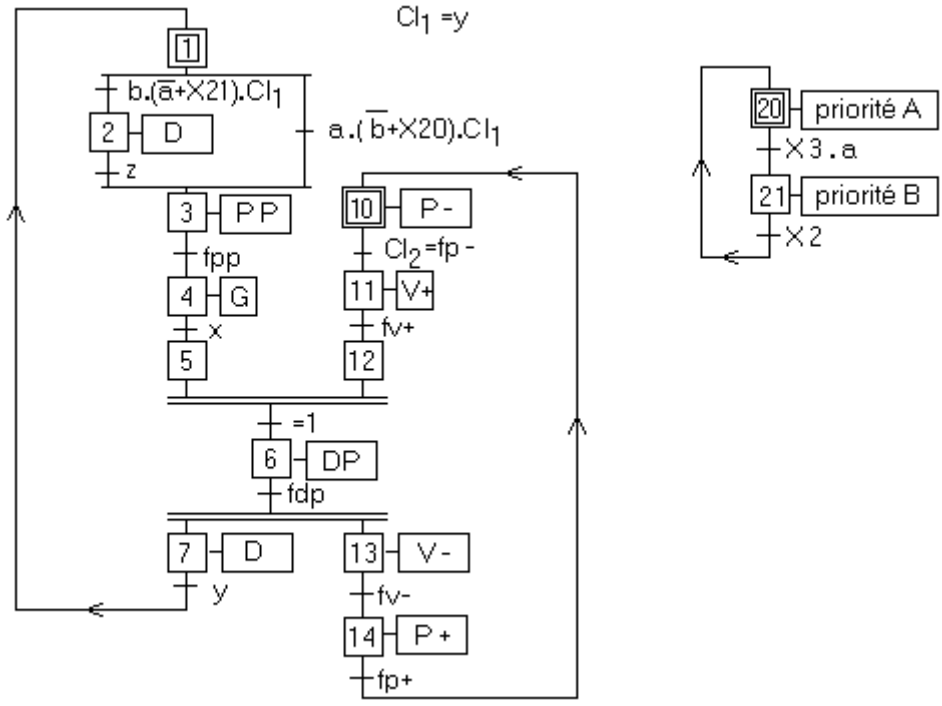


Chaque fois qu'une condition séquentielle (dépendant de ce qui s'est passé auparavant) intervient dans une réceptivité, il vaut mieux ne pas compliquer le Grafcet, mais "calculer" cette condition par un petit Grafcet annexe.

Améliorations :

a) permettre au chariot de rechercher une pièce dès qu'il a posé la précédente : séparer le problème en deux : chariot et partie basse.Prévoir deux Grafcet différents, pouvant évoluer simultanément, mais synchronisés pour le dépose de la pièce (par des Xi ou une ressource)

b) faire attendre le chariot en y plutôt qu'en x (pour améliorer le temps de réponse). Pour la partie basse, l'attente se fait plateau en haut, mais ce ne peut pas être l'état initial (il risque de descendre pendant la nuit). Prendre celà en compte :

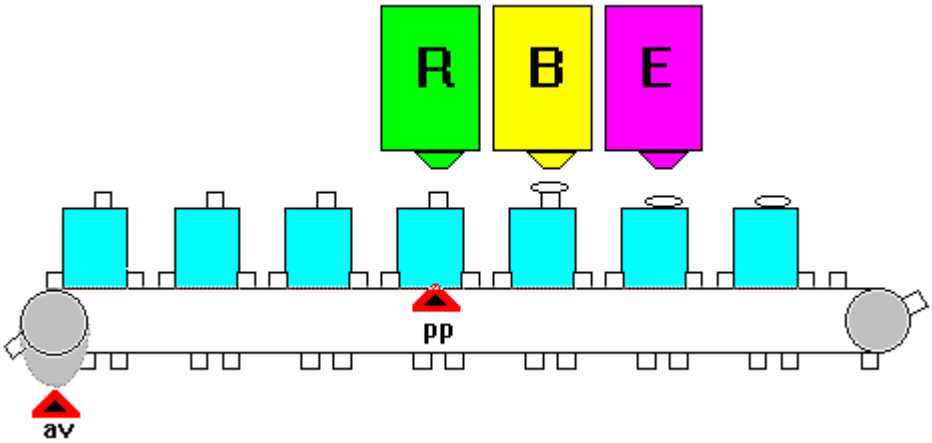


Les deux étapes initiales ne testent que leurs conditions initiales respectives (partie haute ou partie basse).

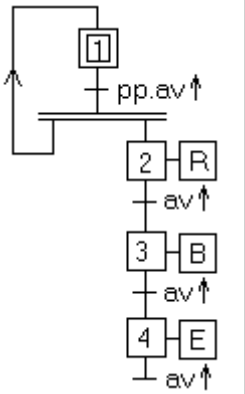
2 - travail à la chaîne

Soit une chaîne de remplissage de bidons d'huile. Un tapis roulant se déplaçant par saccades (cadencé par un système supposé externe à notre Grafcet, s'arrêtant à chaque nouvel appui de la came sur le capteur av) est alimenté manuellement

(de temps en temps il manque des bidons). Trois postes sont prévus : remplissage (R), bouchage (B) et enfoncement (E).



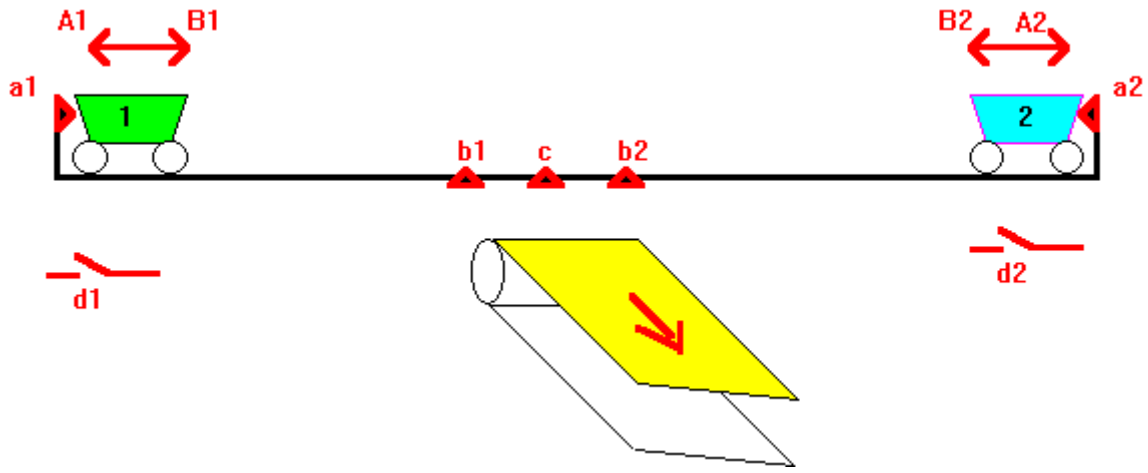
Un seul capteur détecte la présence d'un bidon en début de chaîne : pp. On désire faire les 3 opérations simultanément, sauf s'il n'y a pas de bidon sous le poste. S'il vous semble obligatoire de rajouter des capteurs, vous n'avez RIEN compris au Grafcet puisqu'il vous faut un système combinatoire (il vaut mieux alors câbler en combinatoire chaque poste : avance tapis ET présence bidon => effectuer l'action). On suppose que le tapis est vide lors de l'initialisation.



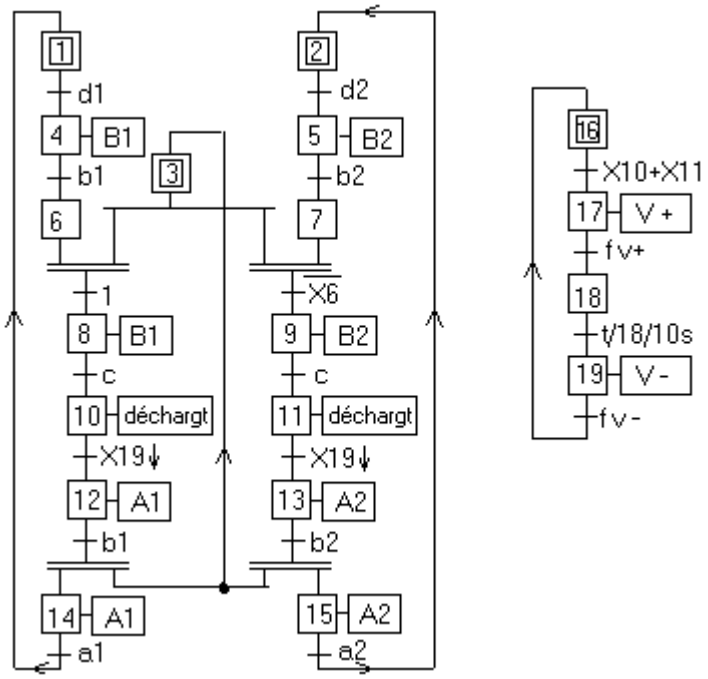
L'étape 1 est constamment active. La dernière transition est appelée "transition puits", mais il était possible de la relier à l'étape 1. En fonctionnement normal, toutes les étapes du Grafcet sont actives. Du point de vue commande, chaque opération comportera plusieurs étapes (R = descendre l'entonnoir, ouvrir le robinet,...) dont une seule sera active à la fois). Chaque activation représente un bidon dans le circuit.

Cette méthode utilise au mieux le séquencement du Grafcet, on peut maintenant rajouter des capteurs, mais qui n'auront pour fonction que de vérifier le bon fonctionnement du système. Dans tous les cas similaires, on utilisera cette démarche : faire le Grafcet pour une pièce seule, puis le modifier pour gérer l'ensemble des pièces, en vérifiant bien que jamais une même étape ne corresponde à 2 pièces, on décompose donc le système en tronçons et on ne laisse entrer dans un tronçon que s'il est libre. Exemples : atelier flexible (on suit la pièce pour chaque opération jusqu'au produit fini), montage (monter 2 pièces ensemble correspond à une convergence en ET : de 2 étapes actives on arrive à 1), chariots filo-guidés (si un tronçon est occupé, essayer de le contourner par une voie libre)...

3 - ressource (ou sémaphore)



Au fond du puits de mine ndeg. i, un mineur remplit un chariot X_i. Quand il est plein (le chariot), il (le mineur) appuie sur un bouton d_i. Immédiatement, le chariot se déplace dans la direction B_i jusqu'au poste de déchargement, composé d'un tapis roulant en mouvement continu, et d'un vérin V qui retourne la benne. Si le poste de déchargement est libre, le chariot avance jusqu'au capteur c, est déchargé puis s'en retourne en a_i. Si le poste est occupé, il attend son tour en b_i. Le poste de déchargement, commun à plusieurs voies, n'est utilisable que par une voie à la fois. On l'appelle une "ressource physique". Traiter le cas de 2 voies (pas nécessairement de la même longueur).



Supposer que la ressource est occupée en utilisant le capteur c est IDIOT : et s'il est entre b_i et c ? Et si le temps de freinage l'a arrêté juste à côté de c ? Il faut

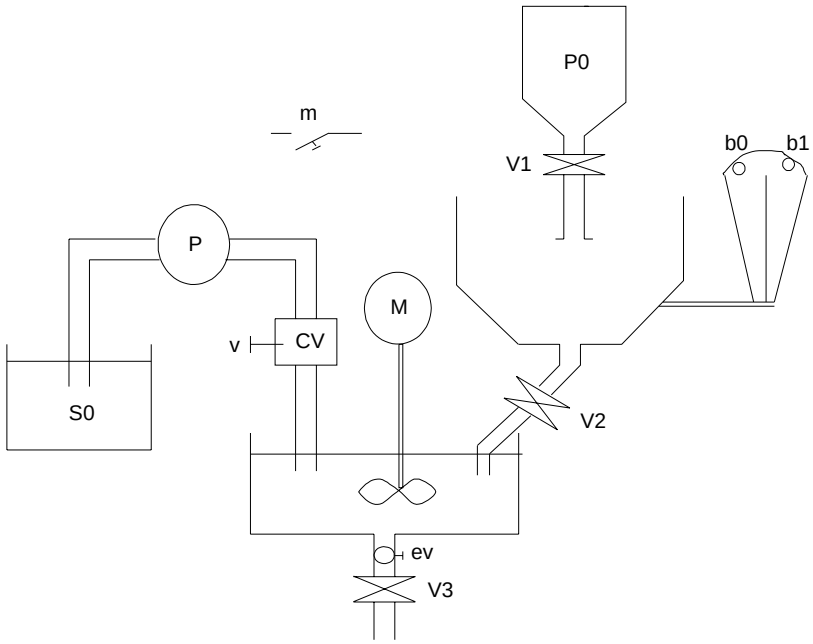
utiliser les facilités séquentielles du Grafcet autant que possible (ne tester un capteur que quand c'est nécessaire). Un capteur ne doit servir que comme condition de passage d'une étape à une autre, mais pas pour vérifier un état du système qui découle du séquencement effectué (par exemple, une transition vérifie la présence d'une pièce, aucune action ne déplace la pièce puis on re-vérifie la présence : Ce n'est censé que si l'on prévoit dans le Grafcet ce qu'il faut faire si la pièce a disparu). Ici, on utilise donc une étape (la ressource), qui est active quand la ressource physique est disponible. Dès utilisation, on la désactive, pour la réactiver quand on libère la ressource physique.

On pouvait également résoudre le problème par des Grafcets séparés (un pour chaque chariot, un pour le déchargement) synchronisés par des Xi. La seule différence est que n'ayant plus de divergence sous l'étape 3, on risque d'oublier de traiter le cas d'arrivée simultanée en b1 et b2, cas arrivant assez rarement pour que l'on ne détecte pas le problème en phase d'essais, mais se produira de temps en temps en fonctionnement réel sans que l'on puisse reproduire le problème lorsqu'un spécialiste sera présent (seule solution : graphe des états accessibles).

Série n°2

Exercice n°1: ***Automatisation d'un poste de mélange***

On veut obtenir la dissolution d'un produit pulvérisant P_0 dans un solvant S_0. Le mélange doit contenir un volume V de solvant S_0 mesuré par un compteur volumétrique CV, et une quantité Q du produit P_0 mesurée par pesée sur une bascule B. Le compteur volumétrique délivre une information **v** telle que (v=0) si le volume débité depuis le début du cycle est inférieur à V et (v=1) si ce volume est supérieur ou égal à V. Le solvant est extrait d'un réservoir par une pompe **P**. Le dosage du produit P_0 avec la bascule B est effectué en commandant deux vannes $\mathbf{V_1}$ et $\mathbf{V_2}$ fermées au repos. On ouvre tout d'abord V_1 (V_1=1) pour que le produit contenu dans la trémie se diverse sur le plateau de la bascule B. Quand la quantité Q voulue est atteinte, un contact $\mathbf{b_1}$ passe de 0 à 1. On laisse alors V_1 se refermer et on ouvre V_2 pour que P_0 se diverse dans le mélangeur. Un contact $\mathbf{b_0}$ est actionné lorsque la bascule B est vide. Une vanne $\mathbf{V_3}$ permet l'évacuation du produit fini. La fin de la vidange du mélangeur est testé par une variable **ev**. Un moteur **M** sert à l'agitation du mélange: il est mis en marche (M=1) en début de cycle et s'arrête lorsque l'évacuation est terminée. Le début du cycle est commandé par un bouton poussoir **m**.



1) Décrire par un GRAFCET le fonctionnement de la PC du poste de mélange.

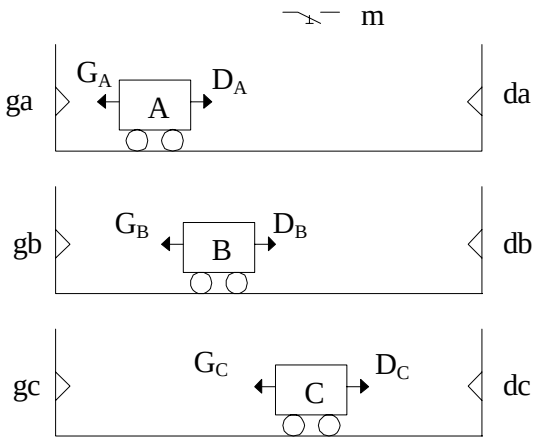
2) Matérialiser ce grafcet par un séquenceur câblé à base de :

-bistables RS,

-compteur programmable.

Exercice n°2:

On considère trois chariots A,B et C qui se déplacent sur des voies indépendantes comme indiqué ci-dessous:



Des contacts de fin de course délivrent des signaux logiques gi=1 ou di=1 quand le chariot i est à gauche, ou à droite, respectivement.

On considère un système de commande qui a pour entrées ga, da, gb, db, gc, dc et m, et pour sorties G_A, D_A, G_B, D_B, G_C, D_C qui correspondent aux déplacements vers la gauche ou la droite comme l'indique la figure ci-dessus.

Au départ les trois chariots sont à gauche. Dés que l'on appuie sur m ils se déplacent tous vers la droite. En arrivant à droite, les chariots A et B s'attendent et repartent vers la gauche dés qu'ils sont arrivés tous les deux à droite, et ceci quelle que soit la position de C. Le chariot C, lui, ira dans tous les cas à droite et ne commencera son retour vers la gauche que lorsque A et B seront tous deux arrivés à droite, ou déjà répartis vers la gauche. Autrement dit, A et B s'attendent mais n'attendent pas C, et C attend A et B.

Un nouveau cycle commencera quand les trois chariots seront à gauche, et le bouton m appuyé.

1°/ Décrire par un GRAFCET le fonctionnement du système de commande.

2°/ Mettre en oeuvre cet automatisme à l'aide d'un séquenceur à base de

-mémoires d'étape,

-bistables RS,

-compteur programmable.

Quel est la meilleure solution ? Justifier votre réponse.

Exercice 3:

On se propose d'étudier la commande séquentielle d'un poste automatisé pour chanfreinage des douilles électriques.

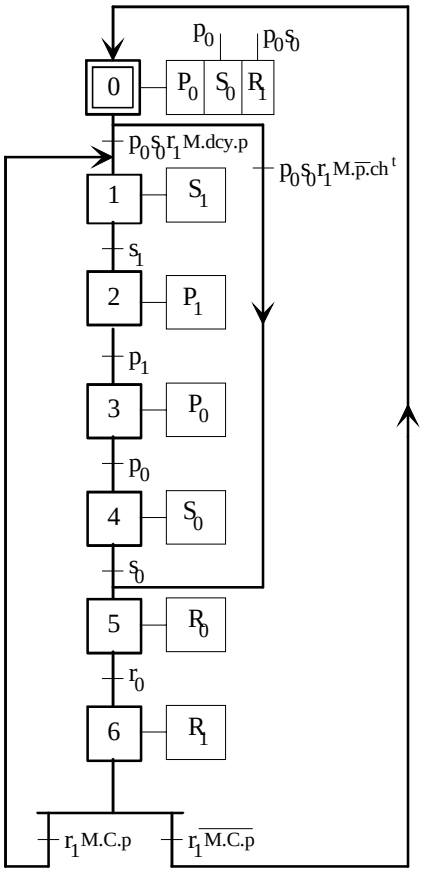


Figure 2. Grafcet décrivant le fonctionnement du système.

M: Commutateur marche-arrêt. *dcy : Bouton départ cycle.*

Le grafcet de la **Figure 2**, décrivant le fonctionnement du système, intègre l'ensemble des modes de marches et d'arrêts suivants :

1. Modes de marche de production

1.a. Marche cycle par cycle : A la fin de chaque cycle, l'information départ ″**dcy**″ doit être donnée pour l'exécution d'un nouveau cycle.

1.b. Marche continue (cycles répétés) : Un commutateur ″**C**″ est sur marche, un sélecteur est sur continu, une action sur ″**dcy**″ commande le fonctionnement en cycle répété. Si ″**C**″ est sur arrêt, où si

on veut changer le mode de marche sur cycle/cycle où si une pièce n'est pas présente, un arrêt est demandé en fin de cycle.

2. Modes de marche de vérification

2.a. Marche de vérification dans l'ordre du cycle : Marche étape par étape (et/et). Blocage de l'évolution des étapes au niveau des réceptivités, la condition de franchissement et l'information (normale ou réglage étape par étape sont nécessaires pour évoluer).

2.b. Marche de vérification dans le désordre : Une information marche réglage désactive toutes les étapes et permet la validation des commandes manuelles. L'information ″**Init**″ désactive les étapes de 1 à 6 et active l'étape initiale qui commande la mise en référence de la P.O.

3. Modes d'arrêts

3.a. Arrêt normal : Arrêt en cours de cycle, blocage de l'évolution a lieu par coupure de la pression. Redémarrage à l'étape d'arrêt au rétablissement de la pression.

3.b. Arrêt d'urgence : Un arrêt d'urgence ″**AU**″ doit être envisagé dans les cas suivants :

* Arrêt normal en fin de mouvement (retour unité de perçage).

* Mise hors énergie du vérin de rotation plateau, une pièce mal engagée peut occasionner un blocage.

En cas d'arrêt d'urgence, la reprise (redémarrage) peut se faire soit à partir de l'étape d'arrêt ou à partir de l'étape initiale.

Travail demandé

I. Mettre en oeuvre cet automatisme à l'aide d'un séquenceur à base de :

I.1. mémoires d'étapes;

I.2. Compteur programmable.

II. Mise en oeuvre du **GEMMA**.

En se servant de la grille (document réponse)

* Identifier les situations possibles du système.
* Définir les rectangles d'états ».
* Etablir les liaisons entre états.
* Spécifier les conditions qui générant ces transitions.

REFERENCES BIBLIOGRAPHIQUES

[1] Moncef GOSSA ; Notes de cours Automatisme & Informatique Industrielle ISEFC 2000.

[2] M. GAUCH; Informatique Industrielle, Troisième partie, Architecture des systèmes numériques, Mai 1997.

[3] M. GAUCH; Informatique Industrielle, deuxième partie, Systèmes logiques séquentiels, bascules, registres et compteurs. Avril 1997.