Chapitre 3
La logique séquentielle

Le chapitre précédent présentait les bases de la logique booléenne, appelée aussi la logique combinatoire. En logique combinatoire, l'état des sorties est une fonction logique de l'état des entrées.

Le présent chapitre aborde l'étude des systèmes de logique séquentielle. En logique séquentielle, l'état des sorties est une fonction de l'état des entrées et de l'état du système, ce qui implique qu'une même combinaison d'entrée ne générera pas toujours les mêmes sorties.

Pour analyser les systèmes séquentiels, nous avons la possibilité de recourir à trois approches. Dans le cas des systèmes séquentiels simples, il est possible d'obtenir une séquence de fonctionnement, même recourant à la logique combinatoire avec certaines limitations. Une autre approche possible pour les systèmes séquentiels simple, c'est la méthode de Huffman. Enfin, pour des systèmes plus complexes, des méthodes intuitives dites aussi géométriques existent, mais ne seront que partiellement couvertes ici.

C'est trois approches sont remplacées en automatisation par le GRAFCET qui est une méthode fort aisée d'analyser un système séquentiel. Le GRAFCET sera abordé au chapitre suivant, puisque le présent chapitre nous présentera les deux premières approches présentées au paragraphe précédent.

1 L'utilisation de l'approche par la logique combinatoire pour les systèmes séquentiels simples

Pour les systèmes séquentiels simples, la première approche envisageable, c'est l'utilisation de la logique combinatoire. *Cette approche n'est exploitable que si, pour <u>une combinaison</u> donnée des entrées, il n'y a <u>qu'un seul</u> ensemble de valeurs possibles pour les sorties.*

Pour résoudre un tel système, on utilise la démarche suivante :

- Dénombrer tous les états possibles ;
    - Établir un diagramme des phases ;
    - Établir un diagramme des transitions ;
- Construire la table de vérité du système ;
- Trouver les équations logiques des actionneurs.

1.1 Dénombrement de tous les états possibles

Pour l'analyse d'un système séquentiel, quelle que soit l'approche choisie, il est nécessaire de dénombrer le nombre d'états possibles dans lequel le système peut être. Ce dénombrement sera obtenu en traduisant toutes les conditions de l'énoncé en un *diagramme des phases* ou dans un *graphe des transitions*. Pour chaque état possible identifié, on indique le niveau logique de chaque entrée et de chaque sortie.

Dans le cas de l'approche par la logique combinatoire, il est important de noter qu'il ne peut y avoir deux états distincts ayant la même combinaison d'entrée. Si cela se produit, il faudra choisir une des autres approches (Huffman ou intuitives).

Pour bien comprendre comment faire ce dénombrement, prenons l'exemple de l'automatisme suivant.

Exemple

Soit l'automatisme nommé « plateau tournant » montré en figure 3-1. Le but de cet automatisme est de faire tourner un plateau de 1/8^{ième} de tour suite à l'appui sur un bouton par l'opérateur.

Cet automatisme est constitué de deux vérins commandés par deux distributeurs simple action « V » et « W ». On y retrouve aussi trois capteurs : « a » qui détecte le déverrouillage du plateau, « b » qui détecte la rotation du plateau de 1/8^{ième} de tour et « m » qui est le bouton de démarrage.

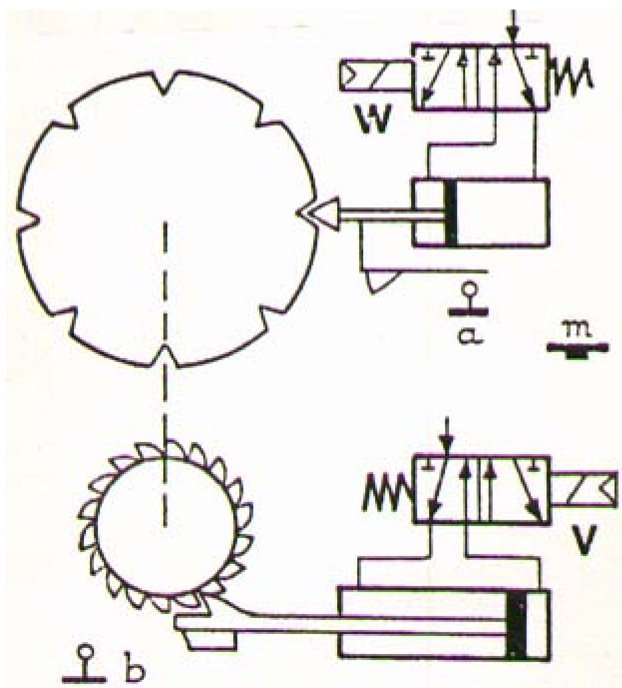


Figure 3-1 : Plateau tournant.

L'automatisme fera la séquence suivante suite à l'appui sur le bouton m :

- Déverrouillage du plateau par rentrée du vérin W ;
- Rotation du plateau de 1/8^{ième} de tour par sortie du vérin V ;
- Verrouillage du plateau par sortie du vérin W ;
- Réarmement du système de rotation par entrée du vérin V.

Si on examine plus en détail ce qui se passe en tenant compte de l'état des capteurs, on obtient la séquence détaillée suivante :

- Machine au repos : entrées a = 0, b = 0, m = 0 / sorties W = 0 et V = 0 ;
- Appui sur le bouton m et déverrouillage du plateau : entrées a = 0, b = 0, m = 1 / sorties W = 1 et V = 0.

Si l'opérateur relâche le bouton m avant que le vérin W ait complété son mouvement, nous aurons la combinaison a = 0, b = 0, m = 0 qui correspond à l'état machine au repos. Comme dans cet état W = 0, alors le vérin W re-verrouille le plateau et aucune rotation ne s'est produite.

Il faut donc que le bouton m soit actionné par l'opérateur jusqu'à ce que le vérin W ait complété son mouvement, i.e., jusqu'à ce que a = 1. Donc, la séquence détaillée serait :

- Plateau déverrouillé et rotation du plateau : entrées a = 1, b = 0, m = 1 / sorties W = 1 et V = 1 ;
- Relâchement du bouton m et rotation du plateau : entrées a = 1, b = 0, m = 0 / sorties W = 1 et V = 1.

Maintenant, pour le reste de la séquence, on assumera quelle se fera quel que soit l'état du bouton m. Ainsi, le reste de la séquence sera :

- Plateau tourné et re-verrouillage du plateau : entrées a = 1, b = 1, m = X / sorties W = 0 et V = 1;
- Plateau verrouillé et réarmement du vérin de rotation : entrées a = 0, b = 1, m = X / sorties W = 0 et V = 0 ;
- Machine de nouveau au repos.

1.2 Établissement d'un diagramme des phases

Ce diagramme est un outil qui sert à dénombrer facilement le nombre d'états et leur enchaînement dans le temps. Il montre l'évolution du niveau logique des entrées et des sorties dans le temps. Le diagramme des phases du plateau tournant est montré à la Fig. 3-2.

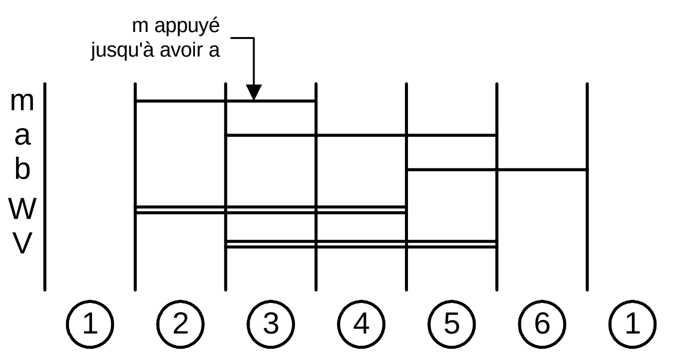


Figure 3-2 : Diagramme des phases du plateau tournant.

On retrouve à gauche de ce diagramme la liste de toutes les entrées et de toutes les sorties du système que l'on analyse. Puis on retrouve l'évolution de l'état logique de toutes ces variables dans le temps. Lorsque le niveau logique d'une variable est 0, on ne dessine rien. Lorsque le niveau logique d'une variable est 1, on trace un trait horizontal montrant la durée de ce niveau logique 1. Pour distinguer les variables d'entrées des variables de sorties, on dessine pour chaque entrée un trait horizontal simple (exemple : m) et pour chaque sortie un trait horizontal double (exemple : W).

À chaque fois qu'une ou plusieurs variables changent d'état, on trace un trait vertical. Cela facilite énormément la visualisation des états possibles du système. Enfin, on fait la numérotation des états en inscrivant dans un cercle le numéro de l'état correspondant à chaque zone entre deux traits verticaux.

Le diagramme des phases de la figure 3-2 montre que le système séquentiel de notre système comporte idéalement six états distincts. À l'état 1, aucun capteur n'est actionné et les vérins sont aux repos. À l'état 2, l'opérateur appui sur le bouton « m » et le vérin « W » déverrouille le plateau. À l'état 3, le bouton « m » est toujours actionné, le capteur « a » indique que le plateau est déverrouillé et le vérin « V » est alors activé pour provoquer la rotation. À l'état 4, l'opérateur relâche le bouton « m » pendant que la rotation a lieu. À l'état 5, le capteur « b » détecte que la rotation est complétée et le vérin « W » est désactivé pour re-verrouiller le plateau. À l'état 6, le capteur « a » indique que le plateau est verrouillé et le vérin « V » est désactivé pour réarmer le système de rotation. Enfin, une fois que le réarmement est complété, on retourne à l'état 1. Cela ressemble beaucoup à la description de l'automatisme donné un peu plus tôt.

Il est important de mentionner que dans cet exemple, l'opérateur est obligé d'appuyer sur le bouton « m » tant que le capteur « a » n'est pas actionné. Sans cette obligation, le diagramme des phases montré à la Fig. 3-3 (si l'opérateur à le doigt rapide) s'appliquerait :

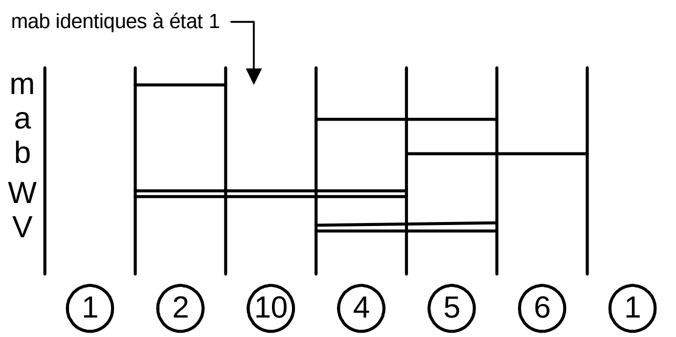


Figure 3-3 : Diagramme des phases du plateau tournant, si l'appui sur « m » est très court.

À ce moment, l'état 10 et l'état 1 auraient eu la même combinaison de variables d'entrées pour des combinaisons différentes des variables de sorties. Nous n'aurions pas pu utiliser l'approche de la logique combinatoire. Donc, dans notre exemple, si l'opérateur n'appui pas assez longtemps sur le bouton « m », l'opération est annulée.

1.3 Le diagramme des transitions

Le diagramme des transitions est un autre outil permettant de mettre en évidence toutes les évolutions possibles entre les divers états de façon graphique. Ainsi, notre exemple aurait le diagramme des transitions montré à la figure 3-4.

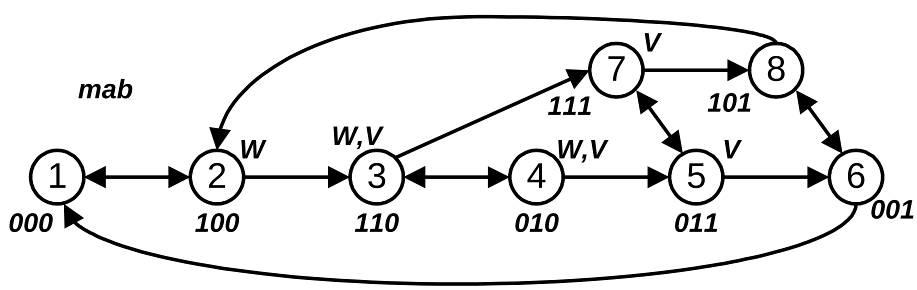


Figure 3-4 : Diagramme des transition du plateau tournant.

Comme on peut le constater, chaque état est représenté par un cercle numéroté (le diagramme comporte donc huit états). À proximité du cercle numéroté, on retrouve l'état logique des variables d'entrées. Par exemple, à l'état 3, m = 1, a = 1 et b = 0, d'où le « 110 » inscrit au-dessous. Toujours à proximité du cercle numéroté, on retrouve la liste des sorties activées dans cet état. Par exemple, à l'état 3, les vérins W et V sont actionnés (W = 1 et V = 1), car on retrouve « W, V » inscrit au-dessus.

Les états sont reliés entre eux par des flèches. Ces flèches peuvent être à simple direction (comme celle liant l'état 4 à l'état 5). Cela illustre le fonctionnement naturel du système. À l'état 4, le vérin V est en train de sortir. À l'état 5, le vérin V est complètement sorti et le capteur « b » passe de 0 à 1. Comme le vérin est maintenu sorti à l'état 5, il est impossible que le capteur « b » passe de 1 à 0, donc de retourner à l'état 4 où b = 0.

Certaines flèches sont bidirectionnelles (comme celle reliant les états 5 et 6). Cela se produit souvent lorsque l'opérateur peut agir sur des boutons dont il ne faut plus tenir compte à certains moments. Ainsi, aux états 5 et 7, on maintient le vérin V sorti. Les capteurs « a » et « b » sont actionnés. La différence entre ces deux états, c'est que l'opérateur a la main sur le bouton « m » (à l'état 7) ou non (à l'état 5). Donc, si l'opérateur pianote sur le bouton « m », le système évoluera entre les états 5 et 7. Cela durera jusqu'à ce que le verrouillage soit complété (capteur « a » passe de 1 à 0), ce qui fait passer le système à l'état 6 ou à l'état 8.

Dans l'exemple du plateau tournant, la description du fonctionnement tirée du diagramme des transitions est exactement la même que celle donnée en figure 3-2 avec le diagramme des phases. Notons simplement l'ajout des états 7 et 8. Pour tenir compte du fait que l'on ne considère plus le bouton « m » une fois l'automatisme en marche, il fallait ajouter des états similaires aux états 5 et 6, mais ayant m = 1.

Le diagramme des transitions est souvent utilisé avec le diagramme des phases. Le diagramme des phases montre le fonctionnement nominal souhaité, tandis que le diagramme des transitions complète l'analyse en montrant les états et évolutions particulières du système.

1.4 Construction de la table de vérité du système

Une fois les diagrammes des transitions et des phases obtenus, il faut élaborer la table de vérité du système. Cela est possible, si pour chacune des combinaisons des entrées, nous avons toujours la même combinaison des sorties. Si pour une des combinaisons des entrées, il existe plus d'une combinaison de sorties, l'approche de la logique combinatoire doit être abandonnée. Dans le cas où l'approche de la logique combinatoire reste possible, il suffit simplement de remplir la table de vérité en utilisant les informations des diagrammes des transitions et des phases. Pour l'exemple du plateau tournant, la table de vérité suivante fait la synthèse de l'analyse de la séquence à partir du diagramme des transitions :

État	*m*	*a*	*b*	*W*	*V*
1	0	0	0	0	0
6	0	0	1	0	0
4	0	1	0	1	1
5	0	1	1	0	1
2	1	0	0	1	0
8	1	0	1	0	0
3	1	1	0	1	1
7	1	1	1	0	1

1.5 Trouver les équations logiques des sorties

Il suffit de traiter la table de vérité comme normalement en logique combinatoire.

Ainsi, dans cet exemple, les équations des sorties « W » et « V » sont trouvées en fonction des entrées « a », « b » et « m » en utilisant des tables de Karnaugh ou de Mahoney. Ce qui donne $W = a\overline{b} + \overline{b}m = \overline{b}(a + m)$ et V = a.

Ces équations sont suffisantes pour faire fonctionner ce système tel que requis. La figure 3-5 montre le schéma de commande du plateau tournant.

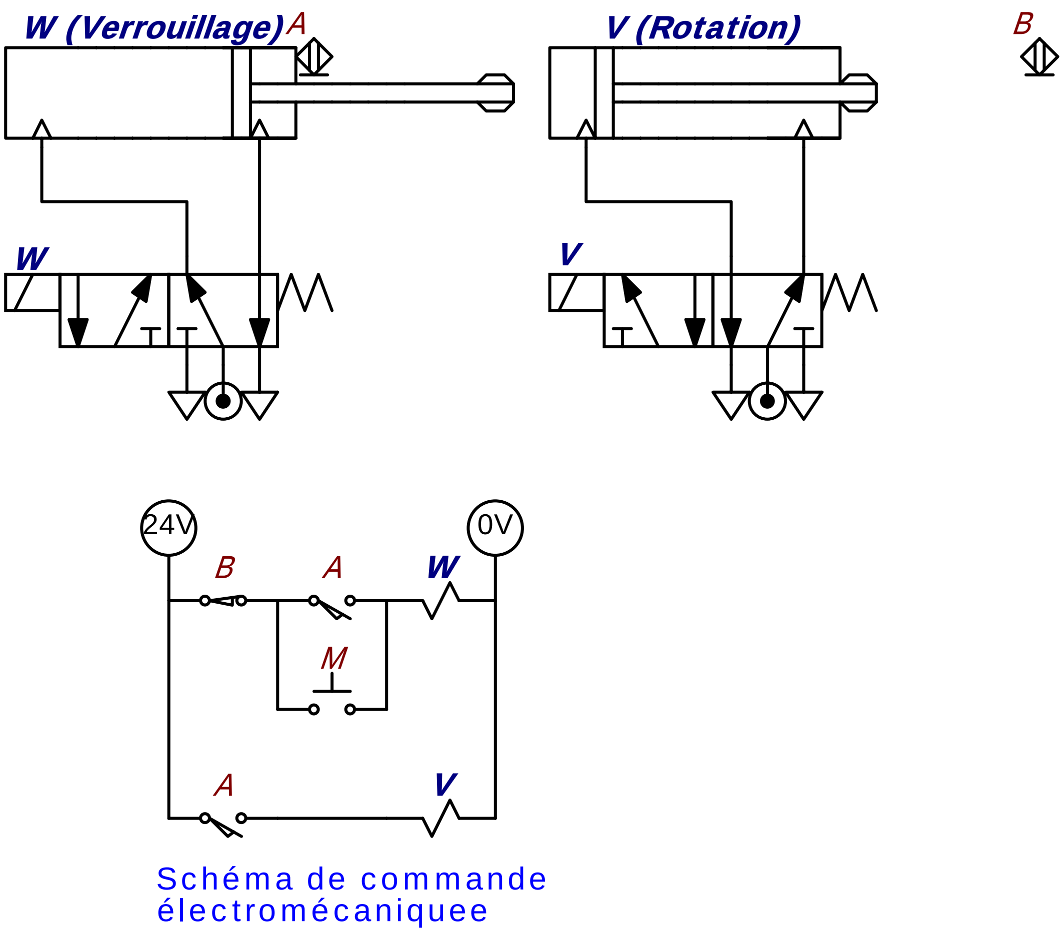


Figure 3-5 : Schéma de commande de plateau tournant.

2 La méthode de Huffman

L'approche par la logique combinatoire ne peut résoudre un système séquentiel pour lequel il y a pour une même combinaison d'entrées plusieurs combinaisons de sorties possibles. Pour bien comprendre, prenons l'exemple d'un moteur dont le contacteur « C » est commandé par deux boutons, l'un pour mettre le moteur en marche (bouton « m »), l'autre pour l'arrêter (bouton « a »). On désire mettre le moteur en marche dès que l'on appui sur le bouton de mise en marche et qu'il reste en marche jusqu'à ce que l'on appui sur le bouton d'arrêt.

Si l'on regarde la séquence détaillée, on retrouve :

- Moteur à l'arrêt : entrées a = 0, m = 0 / sortie C = 0 ;
- Mise en marche : entrées a = 0, m = 1 / sortie C = 1 ;
- Moteur en marche : entrée a = 0, m = 0 / sortie C = 1.

Un gros problème vient d'apparaître. Pour la combinaison a = 0 et m = 0, le moteur est soit en marche (C = 1) soit arrêté (C = 0). Ceci fait en sorte qu'il est impossible de résoudre ce système, pourtant très simple, avec l'approche par la logique combinatoire. Cet exemple sera utilisé au cours des paragraphes suivants pour montrer l'application de la méthode de Huffman.

Pour résoudre un tel système selon la méthode de Huffman, on utilise la démarche suivante :

- Dénombrer tous les états possibles ;
    - Établir un diagramme des phases ;
    - Établir un diagramme des transitions.
- Construire la table primitive des états ;
- Construire la table réduite des états ;
    - Définir des variables secondaires.
- Trouver les équations logiques des actionneurs et des variables secondaires.

2.1 Dénombrement de tous les états possibles

La méthode de Huffman utilise une approche fort semblable à l'approche par la logique combinatoire. Il faut dénombrer tous les états possibles du système en utilisant le diagramme des phases et le diagramme des transitions. Le lecteur est prié de lire les sections 1.1, 1.2, et 1.3, puisque la technique est la même.

Ainsi, pour notre moteur, le diagramme des phases serait tel que montré à la figure 3.6.

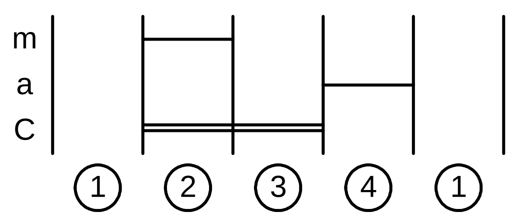


Figure 3-6: Diagramme des phases d'un moteur.

Ce diagramme des phases montre que le système séquentiel comporte quatre états distincts. À l'état 1, aucun bouton n'est actionné et le moteur est à l'arrêt. À l'état 2, l'opérateur appui sur le bouton « m » et le moteur démarre. À l'état 3, aucun bouton n'est actionné, mais le moteur reste en mouvement. Enfin, à l'état 4, l'opérateur appui sur le bouton « a » et le moteur s'arrête, puis une fois le bouton d'arrêt relâché, le moteur reste à l'arrêt.

En fait, il existe un cinquième état qui n'a pas été envisagé, mais qu'il faut prendre en compte. Cet état 5 correspond à un appui simultané des deux boutons. Par sécurité, l'arrêt sera prioritaire sur la mise en marche, et la machine restera immobile (ou s'arrêtera) si les deux boutons sont actionnés.

Le diagramme des transitions de notre moteur est montré à la figure 3-7.

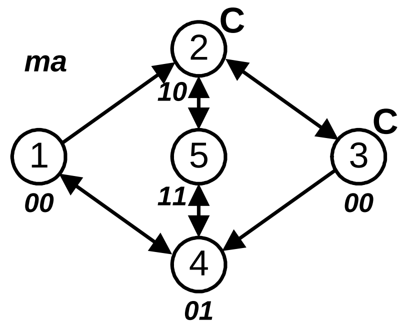


Figure 3-7 : Diagramme des transition d'un moteur.

C'est à partir d'ici que la méthode de Huffman diffère de l'approche par la logique combinatoire. Puisqu'il existe des combinaisons d'entrées identiques qui donnent différentes combinaisons de sorties, il n'est plus possible de tirer une table de vérité. Il faut plutôt construire la matrice primitive des états.

2.2 Construction de la matrice primitive des états

La matrice primitive des états est une transcription du diagramme des transitions. Elle permet de représenter sous forme matricielle l'évolution du système. Cette matrice se divise en deux parties. La partie de droite montre l'état des sorties (dans notre exemple, la sortie « C »), alors que la partie de gauche montre les états stables et transitoires du système en utilisant les entrées que l'on appelle ici les variables primaires (dans notre exemple, les entrées « m » et « a »). Le nombre de colonnes de la partie de gauche dépend du nombre d'entrées, car il faut prendre en compte toutes les combinaisons possibles. Les combinaisons doivent êtres disposées en respectant le *code binaire réfléchit* pour faire en sorte qu'une seule variable d'entrée change lorsque l'on passe d'une colonne à une colonne adjacente.

Les états stables sont représentés par les numéros d'états encerclés. Le nombre de lignes de la matrice primitive des états dépend du nombre d'états de la machine, car on ne peut avoir qu'un état stable par ligne. Sur chaque ligne, il est indiqué le niveau logique des sorties pour l'état stable étudié et l'évolution du système vers d'autres états. Ces états appelés états transitoires servent à indiquer vers quels états un état stable peut évoluer. Par exemple, pour notre moteur, il est possible d'évoluer de l'état 1, vers l'état 2 en appuyant sur le bouton « m ». Comme il faut un certain temps à l'électronique ou aux relais pour réagir, on passe par un état transitoire que l'on identifie par le même nombre que l'état stable qui suivra (ici, 2). L'état transitoire est représenté par un nombre non-encerclé.

Pour le système séquentiel du moteur, la matrice primitive des états est la suivante :

m
a

				C
1	4	X	2	0
3	X	5	2	1
3	4	X	2	1
1	4	5	X	0
X	4	5	2	0

On remarquera sur chaque ligne la présence d'un X qui représente un état indifférent. Cela vient du fait qu'une des hypothèses émises par Huffman, c'est qu'une seule variable primaire (ou entrée) change à la fois. Dans certains cas, le X s'explique par le fait que l'évolution correspondante n'est pas prévue par le diagramme des transitions.

La matrice primitive de notre exemple comporte cinq lignes, car il y a cinq états possibles pour le système. Avant de sortir les équations logiques de notre système, il est nécessaire de simplifier la matrice primitive des états pour obtenir la matrice contractée des états. Cette dernière matrice nous permettra de trouver les équations logiques de notre système.

2.3 La matrice contractée des états

Pour construire la matrice contactée des états, il faut se baser sur la matrice primitive des états. Pour ce faire, il faut regrouper des lignes de la matrice primitive des états en une ligne dans la matrice contractée.

Le regroupement de lignes de la matrice primitive doit obéir aux règles suivantes :

- Les états sur chacune des lignes à regrouper doivent être les mêmes ou correspondre à un X ;
- Les niveaux logiques de la ou des sorties doivent être les mêmes sur les lignes à regrouper.

Ainsi, pour la matrice primitive des états montrée à la page précédente, la première ligne ne peut se regrouper ni avec la deuxième, ni avec la troisième (violation des deux règles). Par contre la première ligne peut se regrouper avec la quatrième et la cinquième ligne. Les deuxième et troisième lignes peuvent se regrouper ensemble.

La matrice contractée résultante est montrée ci-bas :

m a				C
1	4	5	2	0
3	4	5	2	1

Donc, les lignes correspondant aux états stables 1, 4 et 5 sont fusionnées. De même pour les lignes correspondant aux états 2 et 3.

La matrice contractée des états doit inclure un système de codification indiquant sur quelle ligne le système est situé. Pourquoi faut-il inclure ce système de codification ? C'est parce que l'information sur les états des entrées est insuffisante pour savoir dans quel état est la machine. Par exemple, si les seules informations disponibles sont que « m » et « a » sont tous deux à 0, il est impossible de savoir si la machine est dans l'état 1 ou dans l'état 3.

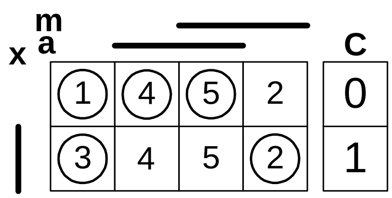
Les variables que l'on ajoute pour faire cette codification sont dites « *variables secondaires* ». C'est en sachant les états des variables secondaires que nous aurons une certitude quant à l'état de la machine, car nous saurons maintenant sur quelle ligne est l'état de la machine. Ainsi, en sachant le numéro de ligne et les états des entrées on connaîtra l'état exact de la machine.

L'automatisme devra gérer la valeur des variables secondaires en fonction de l'évolution de la machine entre ses divers états. Ainsi, dans l'exemple du moteur, lorsque la machine passera de l'état 1 à l'état 2, le code devra changer, car nous passons de la première ligne à la deuxième de la matrice réduite des états.

L'ajout de ces variables permet de comprendre pourquoi on réduit le nombre de lignes de la matrice primaire des états pour obtenir la matrice contractée des états. Un code composé d'une seule variable secondaire (dit aussi « d'un bit ») permet de sélectionner une ligne parmi deux (2^1). Un code de deux bits (deux variables secondaires) permet de sélectionner une ligne parmi quatre (2^2). Un code de trois bits, une ligne parmi huit (2^3).

Ainsi, si la matrice n'avait pas été réduite, on aurait du prévoir un codage du numéro de ligne sur trois bits, car il faut pouvoir identifier une ligne parmi cinq. Avec la matrice réduite, le codage est simplifié, car un seul bit est nécessaire.

On ajoute donc le système de codage sur la matrice réduite des états, tel que montré ci-dessous :



Donc, si la variable secondaire x est égale à 0, la machine est dans un des états de la première ligne. Si la variable secondaire x est égale à 1, la machine est dans un des états de la seconde ligne. Dans l'exemple du moteur, lorsque la machine passera de l'état 1 à l'état 2, la valeur de la variable secondaire x passera de 0 à 1.

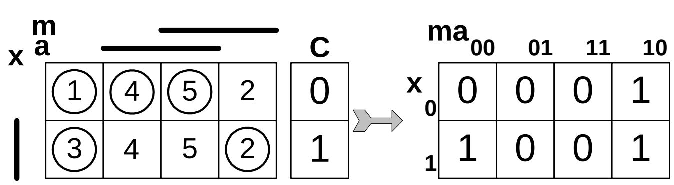
Pour solutionner le système, il nous reste à trouver l'équation logique des sorties (ici, de la sortie C) et l'équation logique des variables secondaires (ici, de la variable x). Il faut comprendre qu'il est de la responsabilité du concepteur de bien gérer les variables secondaires, car ces dernières nous permettrent de mémoriser l'état exact de la machine.

2.4 Obtention des équations des sorties et des variables secondaires

Pour obtenir les équations logiques des sorties et des variables secondaires, il suffit de se baser sur la matrice réduite des états. Cette matrice permet de construire les tables de Karnaugh ou de Mahoney des variables secondaires et des sorties.

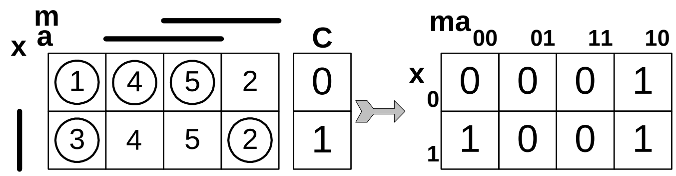
Pour trouver l'équation des variables secondaires, il faut construire une table de Karnaugh (ou de Mahoney) ayant comme variables d'entrées, toutes les entrées et toutes les variables secondaires.

Pour remplir la table d'une variable secondaire, il faut mettre dans chaque case la valeur de la variable secondaire pour l'état stable correspondant au numéro d'état de la case correspondante de la matrice contractée. Ainsi, la table de Karnaugh pour la variable secondaire « x » ressemblera à :



Donc, on trouverait que l'équation logique de x est : $x = m\bar{a} + x\bar{a} = (m + x)\bar{a}$. Remarquez que l'état futur de la variable secondaire x dépend de l'état présent de x, ce qui est caractéristique d'un système séquentiel.

Pour remplir la table d'une sortie, il faut mettre dans chaque case la valeur de la sortie pour l'état stable correspondant au numéro d'état de la case correspondante de la matrice contractée. Ainsi, la table de vérité pour la sortie C ressemblera à :



Donc, on trouverait que l'équation logique de C est : $C = m\bar{a} + x\bar{a} = (m + x)\bar{a}$. Donc on trouve que C = x.

Il est à noter que l'équation obtenue pour l'exemple du moteur est celle d'une bascule. Le signal « m » met la bascule à 1 et elle conserve cet état tant que le signal « a » est à 0. Le signal « a » remet la bascule à 0.

Comme l'exemple le montre, l'analyse est quand même longue même quand l'automatisme est petit. Cela fait en sorte que la méthode de Huffman n'est que très rarement utilisée. On préfère utiliser le GRAFCET.

2.5 Exemple avec le plateau tournant

On reprend ici l'exemple du « plateau tournant » abordé en section 1.1, en permettant à l'opérateur d'appuyer sur le bouton « m » sans avoir à attendre que le vérin « W » recule jusqu'au capteur « a ». Le diagramme des phases de ce système est donné à la figure 3-8 et le diagramme des transitions est donné à la figure 3-9.

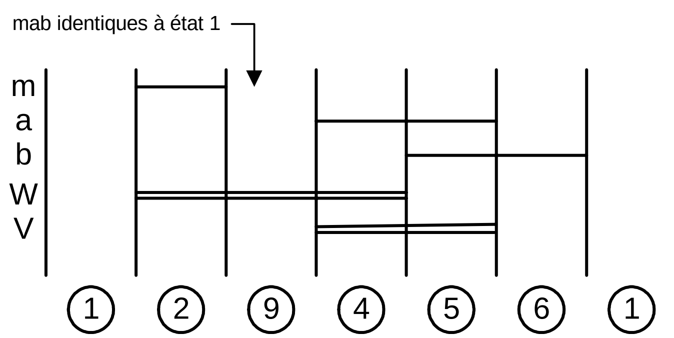


Figure 3-8: Diagramme des phases modifié du plateau tournant.

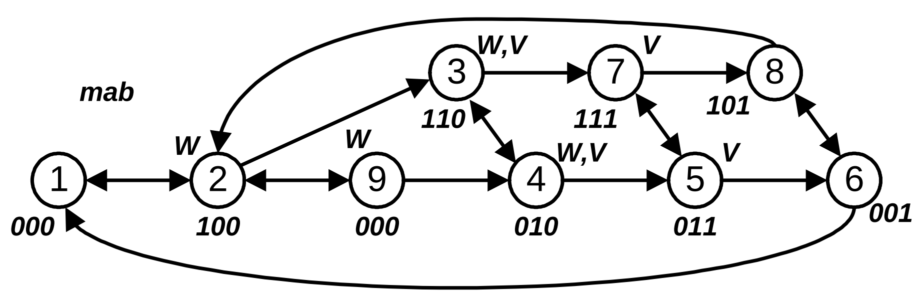


Figure 3-9: Diagramme des transitions modifié du plateau tournant.

Remarquez l'ajout de l'état 9 qui est l'état suivant l'état 2 si l'opérateur relâche rapidement le bouton « m ». Pour cet état, on retrouve la même combinaison des entrées qu'à l'état 1, mais des combinaisons différentes pour les sorties. Donc, on doit recourir à Huffman pour trouver les équations de cet automatisme.

Donc, une fois le diagramme des phases et le diagramme des transitions sont faits, il faut construire la table primitive des états. Cette table comprendra un total de dix colonnes : soit deux colonnes pour les sorties « V » et « W », et huit colonnes pour couvrir les huit combinaisons possibles des entrées logiques « m », « a » et « b ». Le nombre de lignes sera de neuf, puisque l'évolution à partir de neuf états différents sera analysée. La table primitive des états ressemble à

m a b

								W	V
1	X	X	X	X	X	X	2	0	0
9	X	X	X	3	X	X	2	1	0
X	X	X	4	3	7	X	X	1	1
X	X	5	4	3	X	X	X	1	1
X	6	5	X	X	7	X	X	0	1
1	6	X	X	X	X	8	X	0	0
X	X	5	X	X	7	8	X	0	1
X	6	X	X	X	X	8	2	0	0
9	X	X	4	X	X	X	2	1	0

On remarque dès le départ que la table comporte beaucoup de X. Cela vient de l'évolution requise du système. Ainsi, le diagramme des transitions révèle que l'état 1 ne peut être suivi que par l'état 2, pour peu que l'opérateur appui sur le bouton « m ». Les 6 autres colonnes seront des X, car l'évolution du système pour ces cas n'est pas prévue.

La simplification de cette table primitive des états mène à une table réduite montrée ci-dessous :

m a b

x y									W	V
	1	6	X	X	X	X	8	2	0	0
	9	X	X	4	3	X	X	2	1	0
	X	X	5	4	3	7	X	X	1	1
	X	6	5	X	X	7	8	X	0	1

Les lignes ayant les mêmes sorties ont été regroupées, ce qui était possible parce dans chaque cas, chacune des colonnes avait les mêmes états. La table réduite ayant quatre lignes, nous devons utiliser deux variables secondaires x et y pour coder le numéro de ligne, tel que montré sur la table.

Il faut ensuite trouver les équations des variables secondaires. Pour ce faire, il faut construire des tables de Mahoney étant donné le nombre de variables en jeu. Ce qui donne pour x :

m a b / x y

0	0	X	X	X	X	0	0
0	X	X	1	1	X	X	0
X	X	1	1	1	1	X	X
X	0	1	X	X	1	0	X

Il faut noter que l'on inscrit dans chaque case la valeur de la variable x lorsque le système est à son état stable. Par exemple, dans la deuxième colonne, on retrouve sur la quatrième ligne l'état transitoire 6. Bien que sur cette ligne, x = 1, on inscrit quand même 0, car pour l'état stable 6 on a x = 0. Les cas impossibles restent identifiés par des X dans les tables de Karnaugh ou de Mahoney, ce qui permet de simplifier les équations.

À partir de la table de Mahoney, on obtient comme équation x = a.

Pour la variable y nous aurons la table de Mahoney suivante :

m a b / x y

0	0	X	X	X	X	0	1
1	X	X	1	1	X	X	1
X	X	0	1	1	X	0	X
X	0	0	X	X	0	0	X

Ce qui donne comme équation $y = m\overline{b} + a\overline{b} + y\overline{x}$.

Pour les sorties, on tire W = y et V = x. Cela s'explique par le fait que le codage des variables secondaires fait en sorte que la variable y a le même comportement que W et x a le même comportement que V (voir matrice réduite des états). C'est une astuce souvent utilisée par les automaticiens pour simplifier les équations logiques du système.

La figure 3-10 montre le schéma de commande du plateau en version modifiée. On peut y voir les équations réellement utilisées. Puisque x = A, alors on peut écrire que : $y = m\overline{b} + a\overline{b} + y\overline{a}$, W = y et V = a.

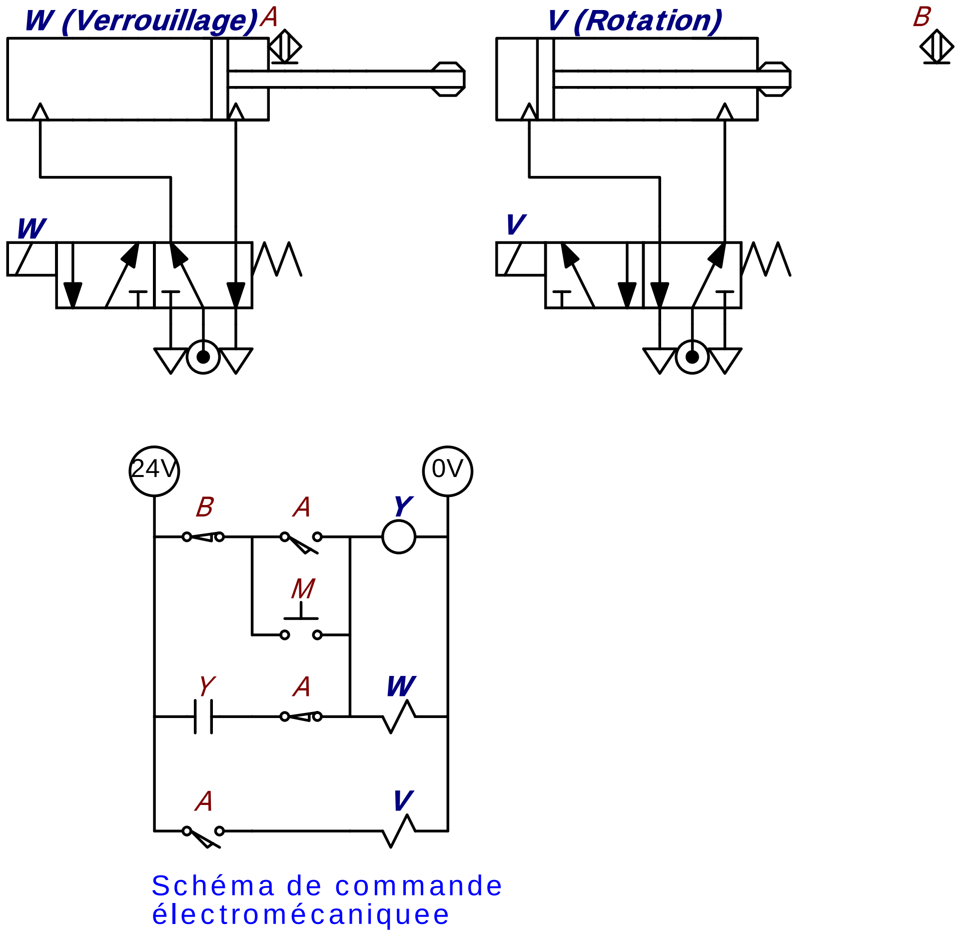


Figure 3-10 : Schéma de commande de plateau tournant.

3 Les méthode intuitives

Les méthodes intuitives sont toutes basées sur la méthode de Huffman qui a eu l'idée d'ajouter des variables secondaires. Le mot intuitif concerne simplement la façon de trouver les zones d'action des variables secondaires. Seulement une méthode intuitive sera couverte dans ce cours.

3.1 Les variables secondaires peuvent être identifiées aux sorties

Pour certains automatismes, on peut simplement faire correspondre les variables secondaires aux sorties. Pour comprendre cette approche, supposons que nous avons un moteur qui peut tourner vers la gauche (contacteur « G ») ou vers la droite (contacteur « D »). Ce moteur est commandé par trois boutons :

- « m » et « n » qui sont verrouillés mécaniquement (donc impossible à les actionner en même temps) et qui correspondent respectivement à une rotation à gauche et une rotation à droite ;
- « a » qui est le bouton d'arrêt (prioritaire si appuyé en même temps que « m » et « n »).

Pour voir si cette approche est applicable à cet automatisme, il faut tracer le diagramme des phases (Fig. 3-11) et le diagramme des transitions (Fig. 3-12).

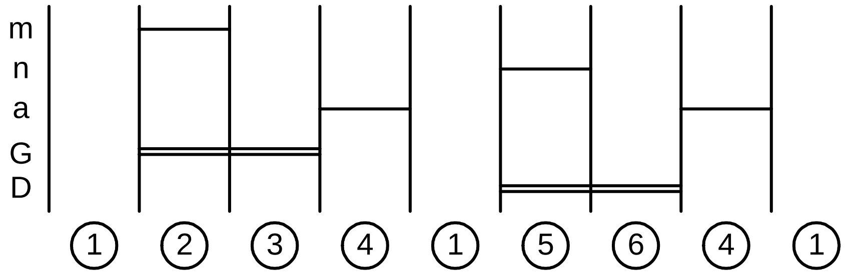


Figure 3-11 : Diagramme des phases d'un moteur bidirectionnel.

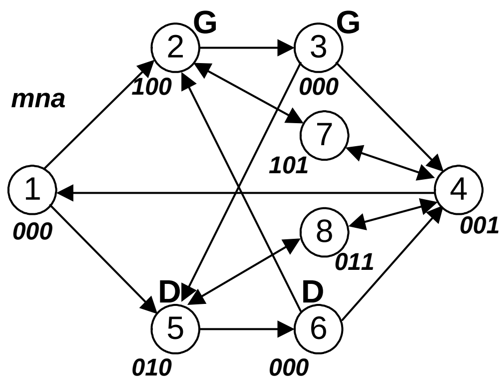


Figure 3-12 : Diagramme des transitions correspondant.

Dans le diagramme des transitions, on constate que l'on a ajouté les deux situations de mise en rotation et d'arrêt simultané (états 7 et 8). On a ajouté les inversions du sens de rotation du moteur (liens 3-5 et 6-2).

Sur ce même diagramme, on constate que pour les entrées « m », « n » et « a » toutes égales à 0 (donc, la combinaison 000), on trouve trois états différents : l'état 1, l'état 3 et l'état 6. Qu'est-ce qui distingue ces trois états ? Simplement l'état des sorties « G » et « D ». À l'état 1, « G » et « D » sont à 0. À l'état 3, on a G = 1 et D = 0. Enfin, à l'état 6 on a G = 0 et D = 1.

Dans une telle application, nous pouvons définir deux variables secondaires x et y et faire en sorte que x = G et y = D. À ce moment, pour les états 1, 4, 7 et 8, on trouve x = 0 et y = 0 ; pour les états 2 et 3, on trouve x = 1 et y = 0 ; pour les états 5 et 6, on trouve x = 0 et y = 1. La combinaison x = 1 et y = 1 n'est pas utilisée.

De là, en utilisant le diagramme des transitions, nous pouvons construire directement la matrice réduite des états :

x y \ m n a									G	D
	1	4	8	5	X	X	7	2	0	0
	6	4	8	5	X	X	X	2	0	1
	X	X	X	X	X	X	X	X	X	X
	3	4	X	5	X	X	7	2	1	0

La matrice primitive des états n'est plus nécessaire, car de façon intuitive, en associant chaque sortie à une variable secondaire, nous avons réduit la complexité du problème.

Il ne reste plus qu'à trouver les équations des variables secondaires x et y. Pour x, la table de Mahoney est :

m
n
a
x y

0	0	0	0	X	X	0	1
0	0	0	0	X	X	X	1
X	X	X	X	X	X	X	X
1	0	X	0	X	X	0	1

ce qui donne $x = \left(m\bar{a} + x\bar{n}\bar{a}\right) = \left(m + x\bar{n}\right)\bar{a}$. Cependant, puisque G = x et D = y, il faut éviter que le moteur reçoit deux ordres contradictoires (G = 1 et D = 1). Pour cette raison, nous devons ajouter un non-y à la réponse finale : $x = \left(m + x\bar{n}\right)\bar{a}\,\bar{y}$.

Pour y, la table de Mahoney est :

m
n
a
x y

0	0	0	1	X	X	0	0
1	0	0	1	X	X	X	0
X	X	X	X	X	X	X	X
0	0	X	1	X	X	0	0

ce qui donne $y = \left(n\bar{a} + y\bar{m}\bar{a}\right) = \left(n + y\bar{m}\right)\bar{a}$. Pour éviter que le moteur reçoit deux ordres contradictoires (G = 1 et D = 1), nous devons ajouter un non-x à la réponse finale pour y : $y = \left(n + y\bar{m}\right)\bar{a}\,\bar{x}$.

Enfin, pour les sorties, on a directement G = x et D = y, puisque nous avons choisi les sorties pour définir les variables secondaires.