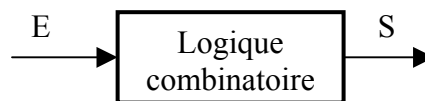


La fonction mémoire

1. Logique combinatoire – logique séquentielle

1.1. Logique combinatoire

En logique combinatoire, pour une des combinaisons d'entrées, il existe **UNE ET UNE SEULE** combinaison de la sortie (0 ou 1) qui est toujours la même. Une même action sur les entrées entraîne le même résultat sur la sortie.

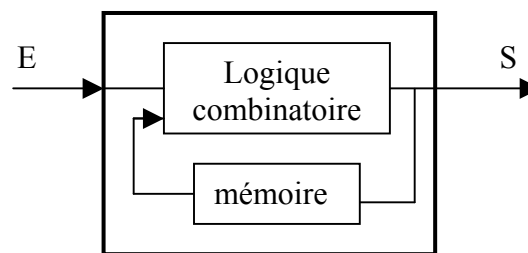


Le nombre de combinaisons de sorties est directement lié au nombre de variable d'entrées:

- 2 variables d'entrée: $2^2 = 4$ combinaisons,
- 3 variables d'entrée: $2^3 = 8$ combinaisons,
- n variables d'entrée: 2^n combinaisons...

1.2. logique séquentielle

Dans un **système séquentiel**, l'état des sorties dépend en plus de **l'histoire** (de l'état précédent): la correspondance entre l'ensemble des entrées et l'état des sorties est **variable**.



Logique séquentielle

Soient

- $S(t)$ la valeur de la sortie du système à l'instant t
- $S(t+1)$ la valeur de la sortie du système à l'instant t+1

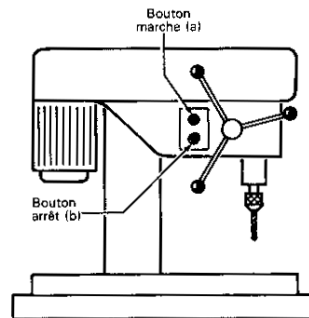
Alors

$S(t+1) = f[e, S(t)]$: la sortie à l'instant t+1 dépend de l'entrée e et de la sortie à l'instant t

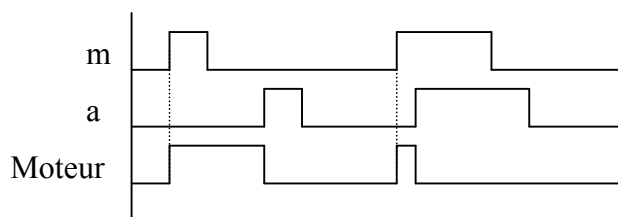
1.3. Exemple

On veut commander la mise en marche et l'arrêt d'une perceuse. On dispose de deux boutons poussoirs :

“m” : marche et “a” : arrêt



- Chronogramme de fonctionnement :



- Table de vérité

Actions chronologiques	"a"	"m"	Etat Moteur	
1. On branche l'appareil	0	0	0	Le moteur ne tourne pas
2. Appui sur « marche »	0	1	1	Le moteur démarre et tourne
3. Relâche « marche »	0	0	1	Le moteur tourne
4. Appui sur « arrêt »	1	0	0	Le moteur s'arrête
5. Relâche « arrêt »	0	0	0	Le moteur est toujours arrêté.

- Remarques :

Pour deux états identiques des variables « m » et « a », on a Moteur = 0 ou = 1.

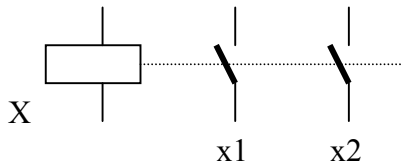
Dans la logique séquentielle, on tient compte de la combinaison des variables d'entrées et de la succession de celles-ci. Dans cet exemple, on a réalisé la fonction mémoire.

2. Conception technologique de la fonction mémoire

2.1. Mémoire à relais électromagnétique

On réalise la commande du moteur M de la perceuse par un relais X qui possède 2 contacts à fermeture x1 et x2.

Relais :

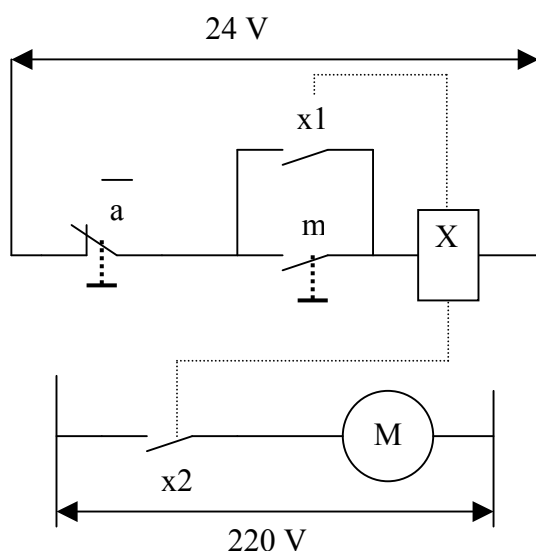


Lorsque la bobine du relais X est alimentée, le relais X s'enclenche.

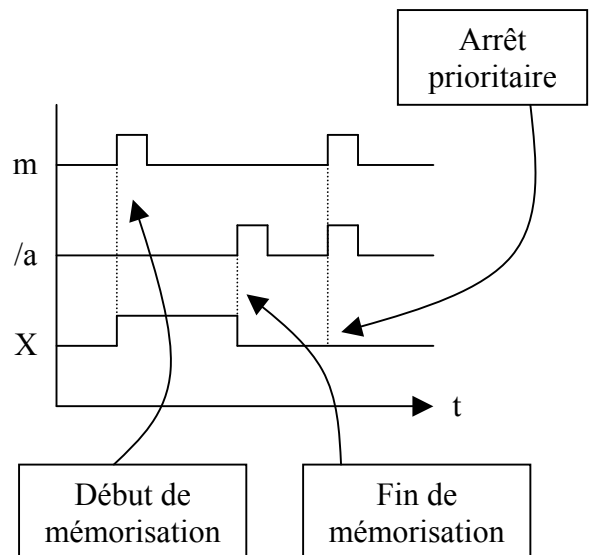
Lorsque le relais est enclenché, il ferme simultanément les contacts x1 et x2.

Inversement, si la bobine du relais X n'est pas alimentée, les contacts x1 et x2 s'ouvrent

2.1.1. Mémoire à arrêt prioritaire



Schéma



Chronogramme de fonctionnement

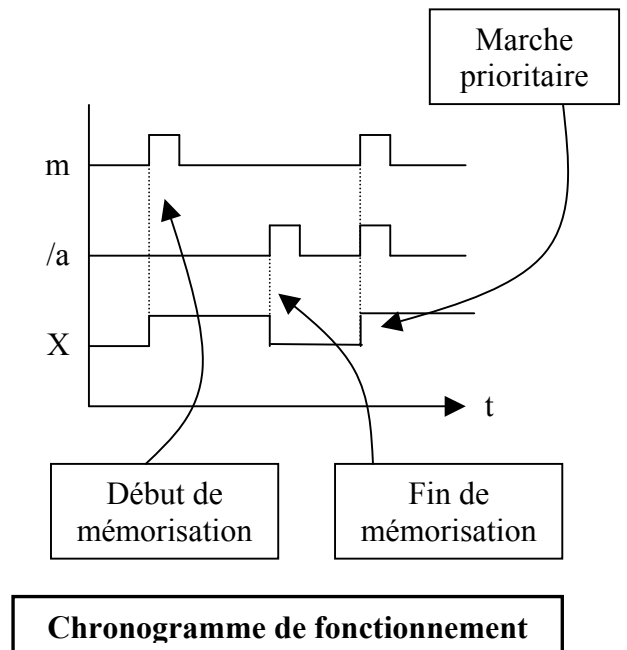
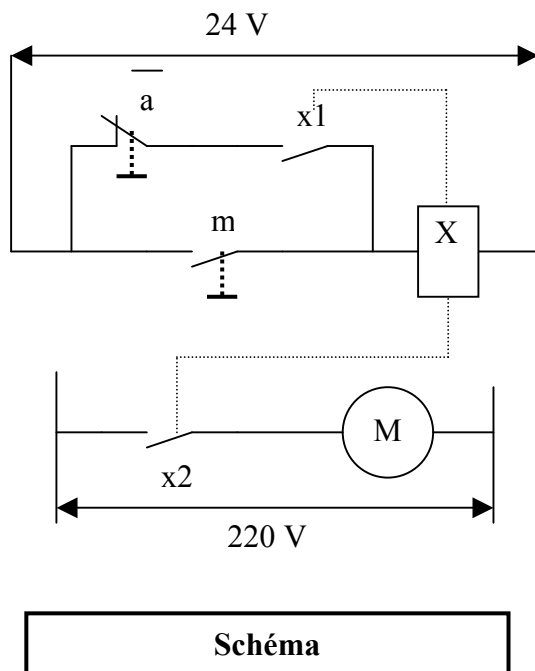
- Equation du circuit :

$$X = (m + x1) \cdot \overline{a} \quad \text{et} \quad M = x2$$

- Remarque :

Le montage est appelé auto-alimentation : Il se souvient du dernier contact appuyé.

2.1.2. Mémoire à marche arrêt prioritaire



- Equation du circuit :

$$X = m + x1.\bar{a} \quad \text{et} \quad M = x2$$

2.1.3. Application

Problème : Commande d'un moteur à deux endroits différents avec arrêt prioritaire

Condition :

2 boîtes à boutons poussoirs marche arrêt :

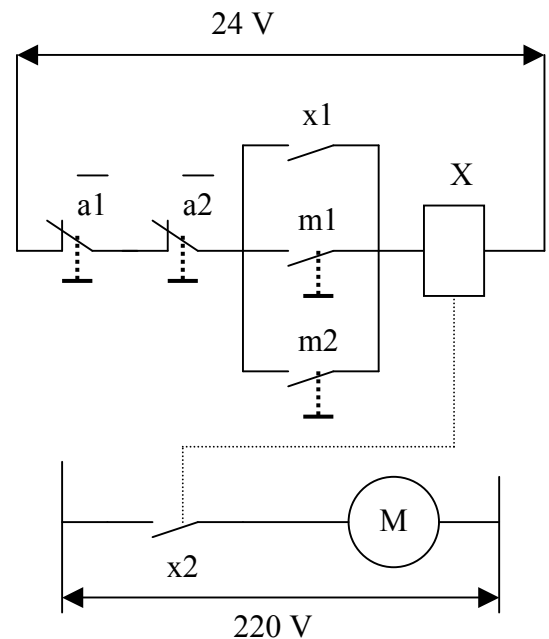
- m1, m2 : boutons « marche »
- a1, a2 : boutons « arrêt »

Solution : contacts « marche » en parallèle, contact « arrêt » en série

Equation logique:

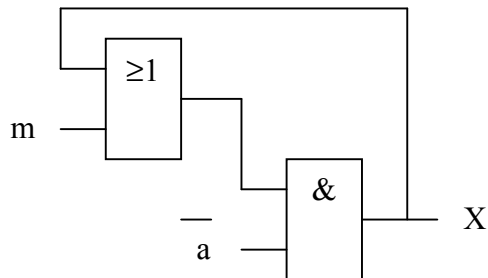
$$X = \bar{a3}.\bar{a4}.(x1 + m1 + m2)$$

$$M = x2$$



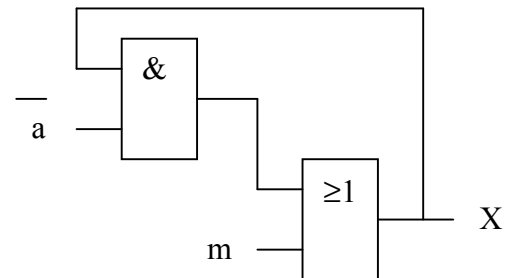
2.2. Mémoire par logigramme

Les équations logiques, définissant la fonction mémoire, peuvent se représenter sous forme de schémas à logigrammes.



Mémoire à arrêt prioritaire

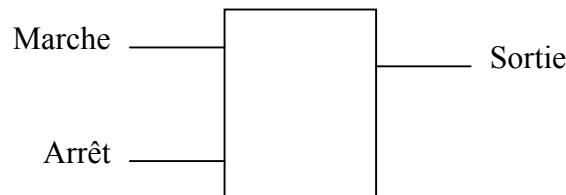
$$X = (m + X).\bar{a}$$



Mémoire à marche prioritaire

$$X = m + X.\bar{a}$$

Symbole logique :



On désigne souvent les entrées « Marche » et « Arrêt » par S et R et la sortie par Q

S pour SET (mise à 1) et R pour Reset (mise à 0)

R	S	Q(t)	Remarque
0	0	Q(t-1)	Etat précédent
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	-	Etat indéterminé

