

MATERIALIZATION D'UN GRAFCET
(Câblage par des bascules RS)

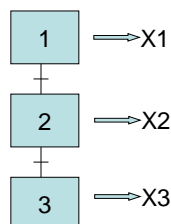
Mr KHATORY

MATERIALIZATION D'UN GRAFCET

BUT : Mettre en ouvre un grafcet à l'aide de composants Tout ou Rien ToR (portes et bascules).

Un grafcet est constitué par un ensemble d'étapes :

A chacune de ces étapes, on associe une variable **X_i**



$X_i=1$ si l'étape **i** est active

$X_i=0$ si l'étape **i** est inactive

MATERIALISATION D'UN GRAFCET

On matérialisera cette variable X_i par la variable de sortie Q_i d'une bascule.

Rappel: Bascule RS

La bascule RS possède deux entrées **S** (Set) et **R** (Reset); et deux sorties de mémorisations **Q** et \bar{Q}

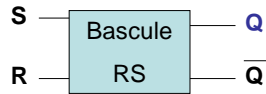


Table de vérité:

R	S	Q	
0	0	q	Mémorisation
0	1	1	Mise à 1
1	0	0	Mise à 0
1	1	-	- (interdit)

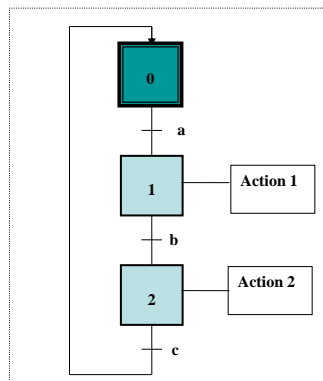
Équation:

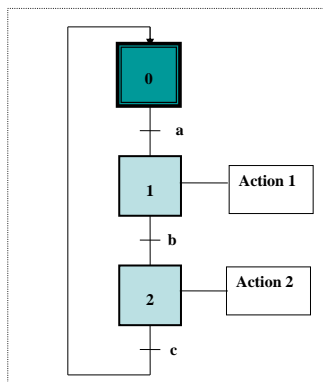
$$Q = S + \bar{R}q$$

MATERIALISATION D'UN GRAFCET

La synthèse de l'automate est alors le problème de calcul des entrées (R_i , S_i ou J_i , K_i) de ces bascules, et des sorties de l'automate en fonction des X_i .

Cas d'un grafcet linéaire :

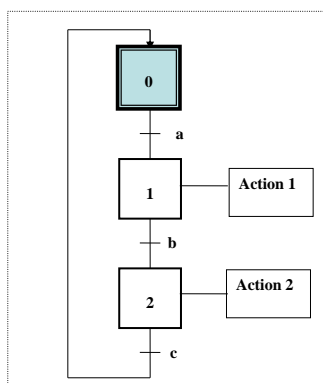




Il suffit d'utiliser **une bascule RS** par étape.

Une étape est activée si l'étape précédente est active et que la réceptivité d'entrée est vraie. **(Règle 3)**

Dans le cas d'un grafcet linéaire, on désactivera une étape quand la suivante est active **(Règle 3)**



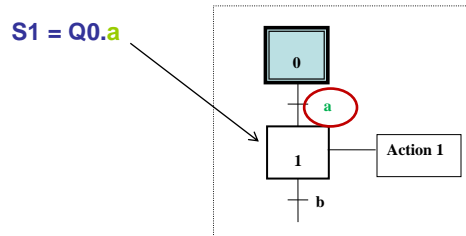
On peut gérer de différentes manières l'étape initiale.

Dans la plupart des cas, le plus simple est d'utiliser des bascules se mettant à 0 à la mise sous tension, et d'initialiser l'automatisme à l'aide d'un bouton qu'on notera "**Init**", qui peut également servir à réinitialiser le grafcet en cours de fonctionnement sans éteindre le système.

Notons, pour l'étape numéro i , son entrée Set (activation) S_i , son entrée Reset (désactivation) par R_i et sa sortie par Q_i .

Étudions l'étape 1:

L'étape s'active si l'étape 0 est active et la réceptivité a est vraie

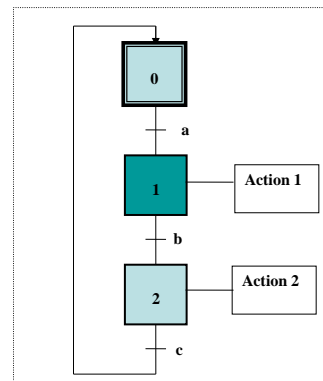
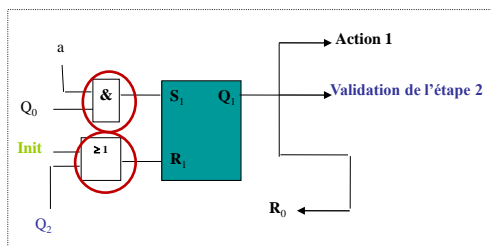


Tout le temps quelle est active, la sortie **Action1** est active (égale à 1). Elle est désactivée quand la réceptivité de sortie (b) est vraie, mais il faut attendre que l'étape 2 soit active. Elle peut être également désactivée par **Init**.

Équations:

$$S_1 = Q_0.a$$

$$R_1 = Q_2 + \text{Init}$$

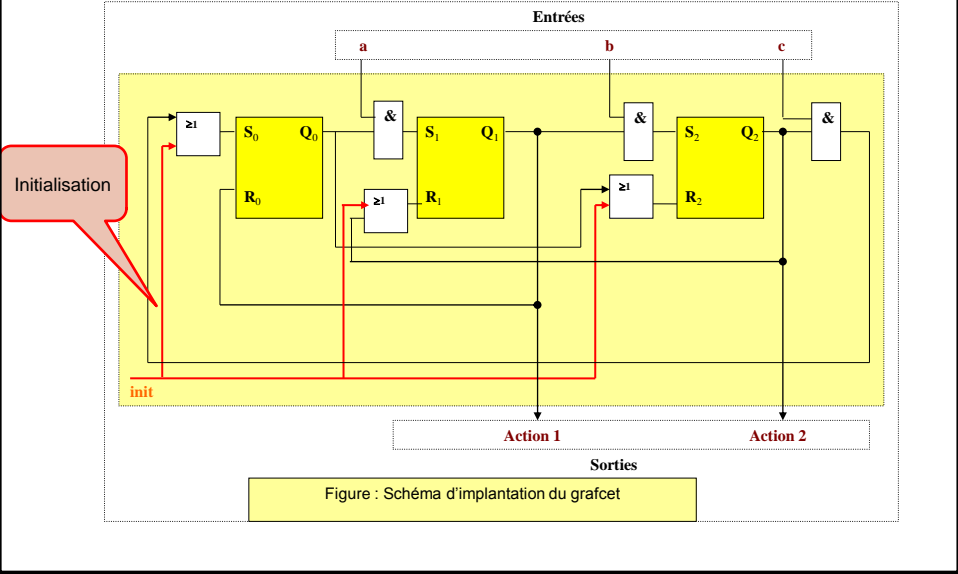


Il suffit de répéter cela pour chaque étape et relier le tout.

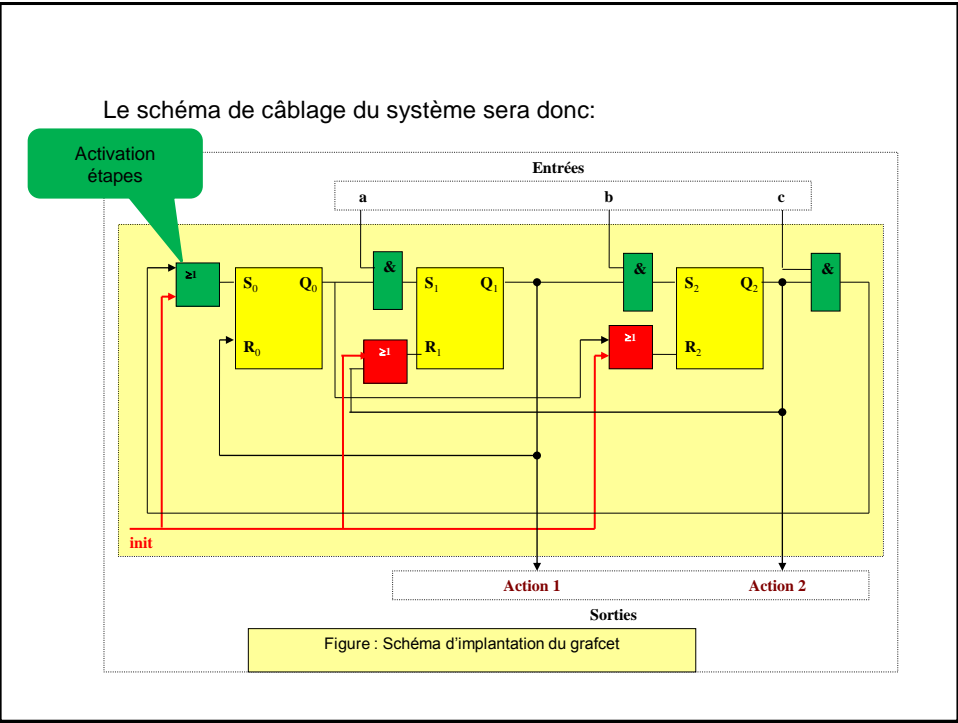
Les équations logiques des commandes des bascules sont :

$$\begin{array}{l|l} S_0 = Q_2.c + \text{Init} & R_0 = Q_1 \\ S_1 = Q_0.a & R_1 = Q_2 + \text{Init} \\ S_2 = Q_1.b & R_2 = Q_0 + \text{Init} \end{array}$$

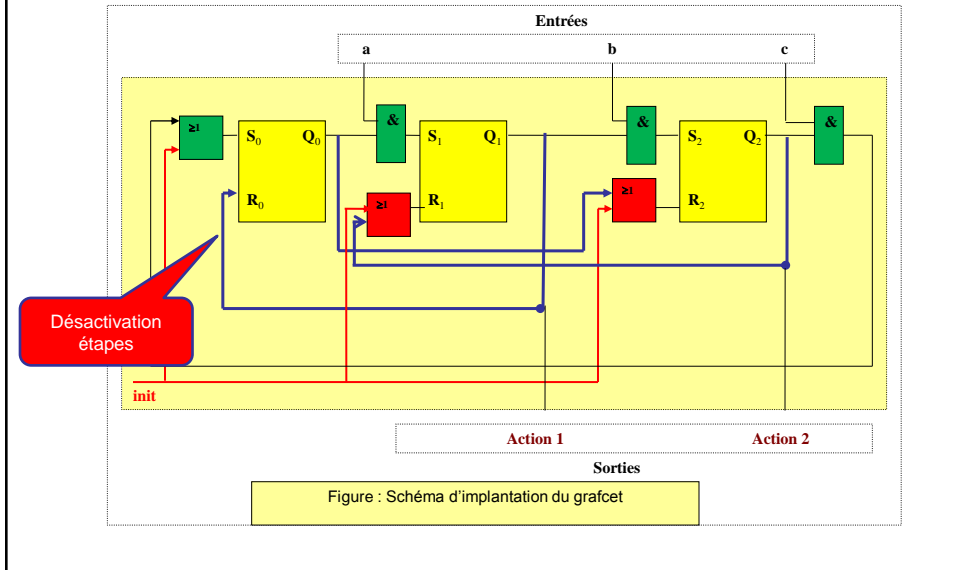
Le schéma de câblage du système sera donc:



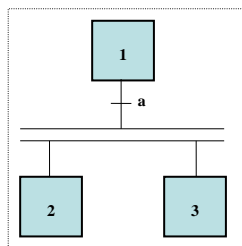
Le schéma de câblage du système sera donc:



Le schéma de câblage du système sera donc:

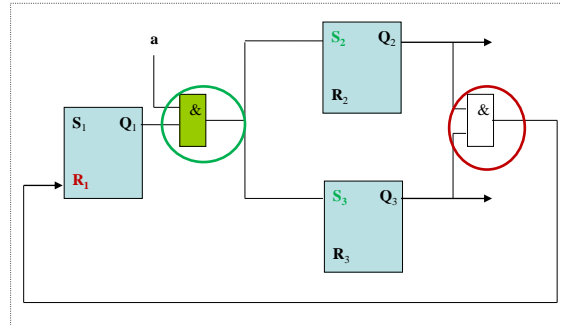
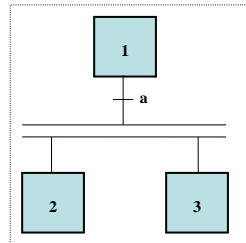


Divergence simple en ET

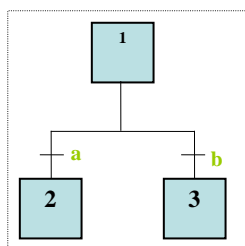


Quand la transition est franchissable, il suffit d'activer deux étapes au lieu d'une.

NB: Le seul problème est la désactivation de l'étape précédente: il faut être sûr que les deux étapes suivantes ont eu le temps de prendre l'information d'activation avant de désactiver la précédente.

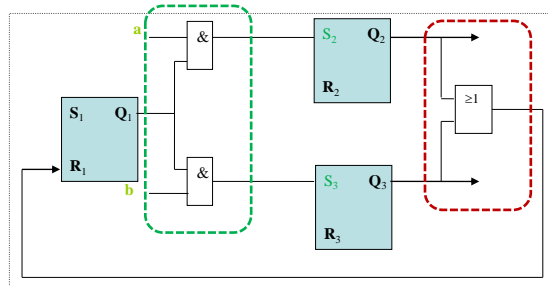
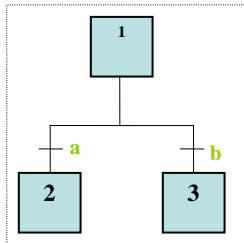
Divergence simple en ET

$$\begin{aligned} S_2 &= Q_1.a \\ S_3 &= Q_1.a \end{aligned} \quad \begin{aligned} R_1 &= Q_2.Q_3 \end{aligned}$$

Divergence simple en OU

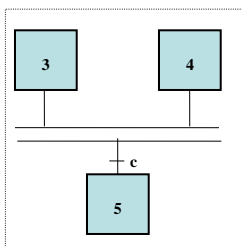
Quand l'étape 1 est active **Et** la réceptivité a (ou b) est vraie, l'étape 2 (ou 3) devient active et l'étape 1 désactive .

NB: Il est possible que l'évolution devienne simultanée si les deux réceptivités sont vraies. (voir cours de grafcet !)

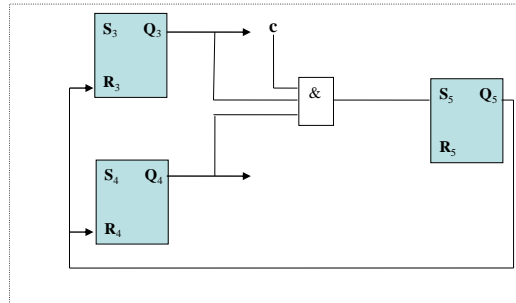
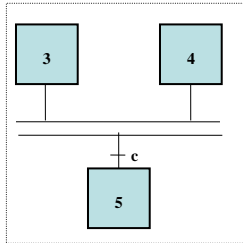
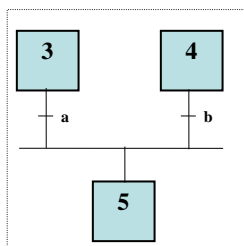
Divergence simple en OU

$$\begin{aligned} S2 &= Q1.a \\ S3 &= Q1.b \end{aligned}$$

$$R1 = Q2 + Q3$$

convergence en ET

Quand les deux étapes (3 et 4) sont actives ET la réceptivité **c** est vraie alors l'étape 5 devient active et les deux étapes (3 et 4) désactives

convergence en ET**convergence en OU**

Quand l'étape 3 (ou 4) est active ET la réceptivité a (ou b) est vraie alors l'étape 5 devient active et l'étape 3 (ou 4) désactive.

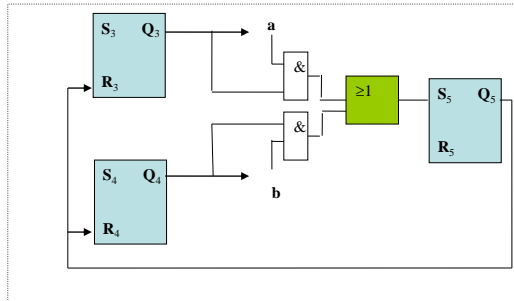
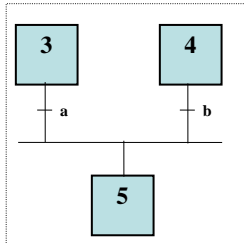
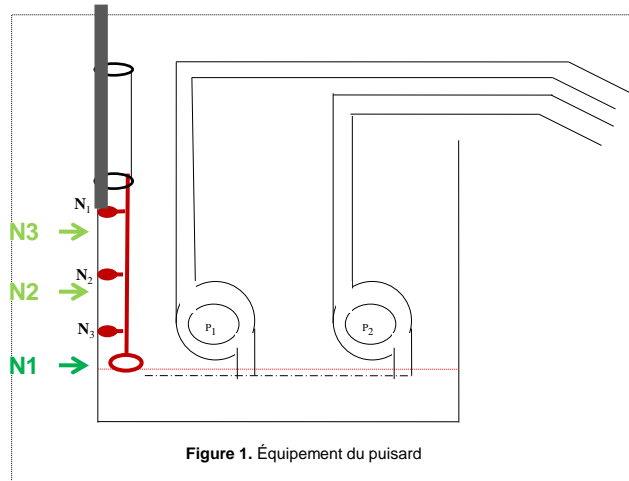
convergence en OU**Exemple: puisard**

Figure 1. Équipement du puisard

Exemple: puisard

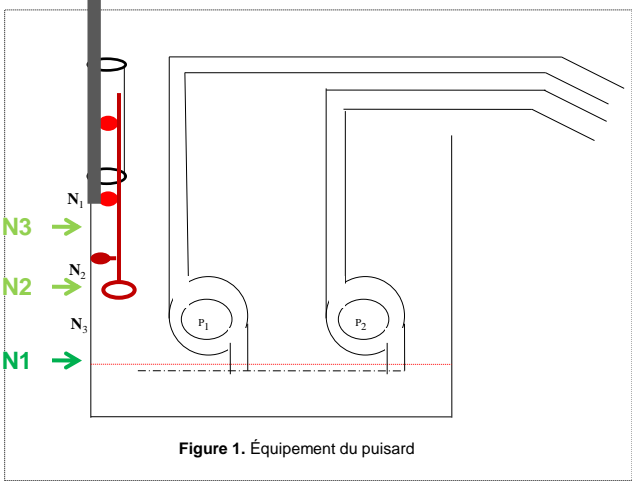


Figure 1. Équipement du puisard

Exemple: puisard

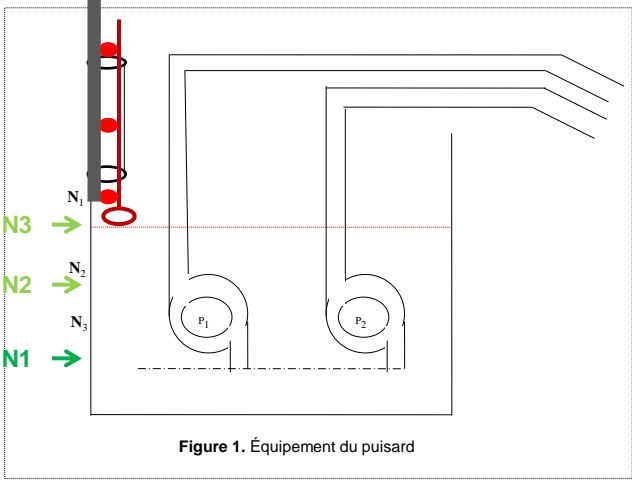


Figure 1. Équipement du puisard

Exemple de grafcet avec sélection de séquence

Un puisard sert à collecter les eaux de pluies, celles-ci s'infiltrant peu à peu dans le sol autour de la cavité du puisard. Pour éviter tout débordement d'eau en cas d'afflux trop important, on a placé deux pompes P1 et P2 et un détecteur de niveau comme indiqué sur la Figure 1. Le fonctionnement souhaité est le suivant :

- Si le niveau d'eau N est inférieur à N1 (les trois contacts N1, N2 et N3 sont relâchés). Aucune des deux pompes ne fonctionne
- Supposons que le niveau N monte ; quand N atteint N2 la pompe P1 se met en marche ;
 - Si le niveau redescend, P1 s'arrête quand N atteint N1 ;
 - Si le niveau continue de monter, P2 se met en marche lorsque N atteint N3 ;
 - Lorsque les deux pompes fonctionnent et que le niveau N atteint N2, on arrête P2, mais on laisse fonctionner P1.

La Figure 2 donne le grafcet correspondant à ce système.

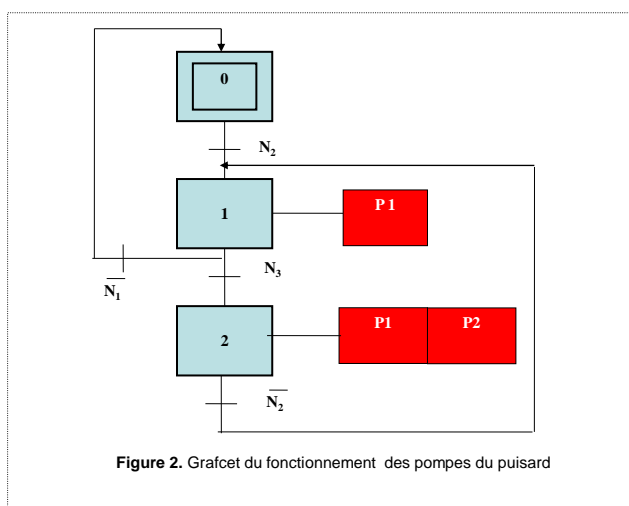


Figure 2. Grafcet du fonctionnement des pompes du puisard

Équations logiques de commande des bascules :

$$S_0 = Q_1 \cdot \overline{N_1} + \text{Init}$$

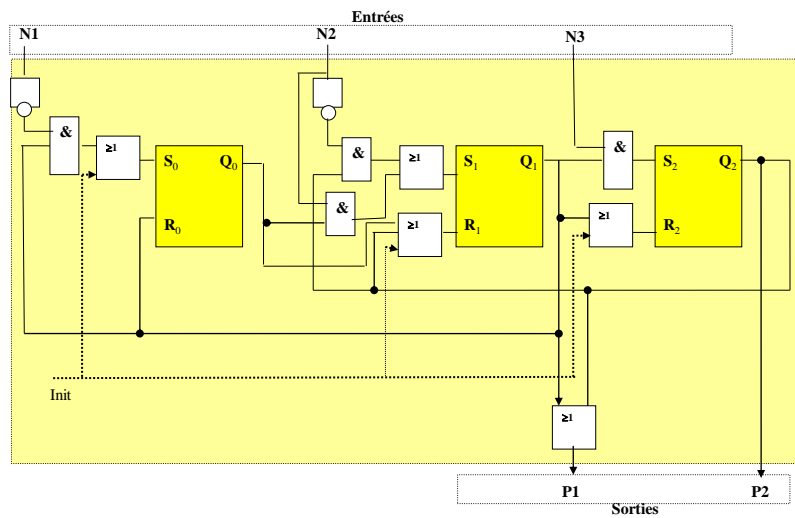
$$S_1 = Q_0 N_2 + Q_2 \overline{N_2}$$

$$S_2 = Q_2 \cdot N_3$$

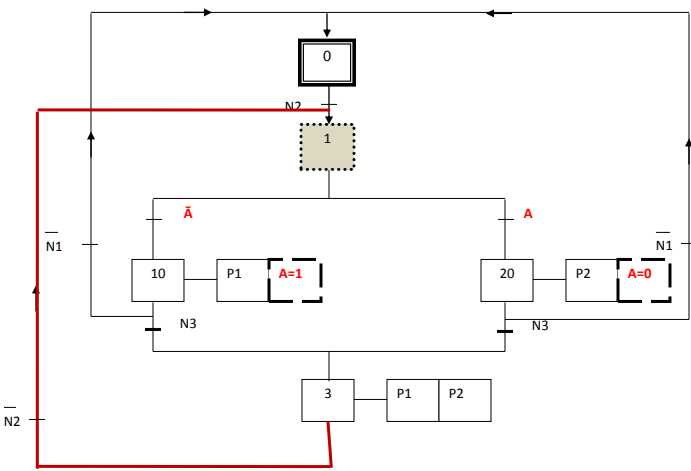
$$R_0 = Q_1$$

$$R_1 = Q_0 + Q_2 + \text{Init}$$

$$R_2 = Q_1 + \text{Init}$$



Les pompe P1 et P2 marchent à tour de rôle !



FIN

ET MERCI