

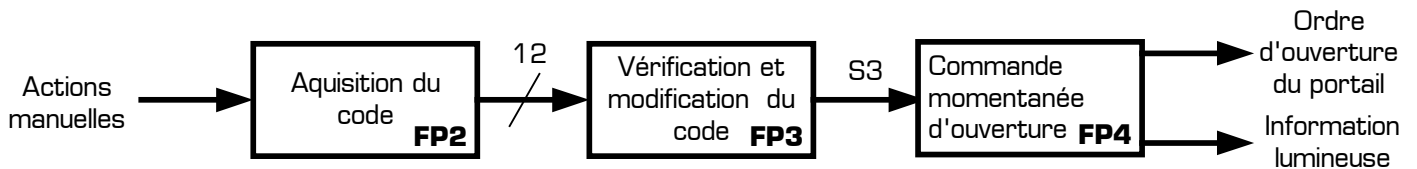
**🌀 Mise en situation et objectifs du TP 🌀**

Parmi les différents mode d'ouverture, l'ouvre portail possède un code d'accès à taper sur un clavier numérique. Le clavier à code est installé à l'entrée du parking à côté du portail. Il permet de limiter l'accès au parking aux seules personnes munies du code d'entrée. Celui-ci est constitué de 4 caractères alphanumériques. Contrairement à une clé, le code en usage peut être modifié facilement s'il est trop divulgué. Ce TP a pour but de valider une solution permettant de détecter l'unique code d'accès du portail. Après analyse d'une solution permettant de décoder un code à 3 chiffres, vous devrez proposer puis valider une solution permettant de décoder un code à 4 chiffres, comme sur l'ouvre portail Faac.

**🌀 Travail demandé 🌀**

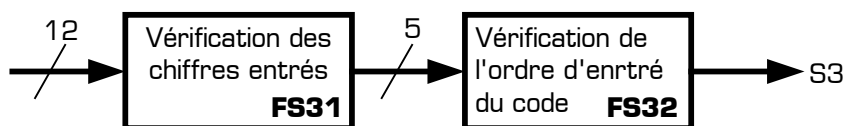
**I - Décomposition fonctionnelle du système**

Dans l'ouvre portail Faac, 3 fonctions principales participent à l'acquisition du code et à l'ouverture du portail :



- \* **FP2 Acquisition du code** : elle génère un signal logique correspondant à la touche appuyée
  - Entrée : action manuelle sur la touche
  - Sortie : niveau logique haut sur le fil correspondant à la touche appuyée
- \* **FP3 Vérification / Modification du code** : elle vérifie l'exactitude du code entré et permet de modifier le code en usage
  - Entrée : niveau logique sur le fil correspondant à la touche enfoncée
  - Sortie : niveau logique haut lorsque le code entré est exact
- \* **FP4 Commande d'ouverture** : elle génère un ordre d'ouverture du portail
  - Entrée : niveau logique haut lorsque le code entré est exact
  - Sorties : ordre d'ouverture du portail et information lumineuse indiquant que le code est valide

La fonction FP3 du système a pour rôle de vérifier si le code tapé par l'utilisateur est exact. Cette fonction principale se divise en 2 fonctions secondaires FS31 et FS32 :



**I - 1** - Encadrez sur le schéma de la *Figure 1* [page 2] la fonction FP2.

**II - Analyse de la solution décodant un code à 3 chiffres**

Le montage de la *Figure 1* est une solution permettant de décoder un code à 3 chiffres. Les questions suivantes sont relatives au schéma de la *Figure 1* :

**II - 1** - Sur la *Figure 1*, quelles sont les seules touches du clavier qui sont utilisées ?

.....

**II - 2** - Quel est le niveau logique actif des entrées asynchrones **R** et **S** des bascules d'après leur symbole ?

.....

**II - 3** - Quel est le front actif de l'entrée d'horloge **C1** des bascules ?

.....

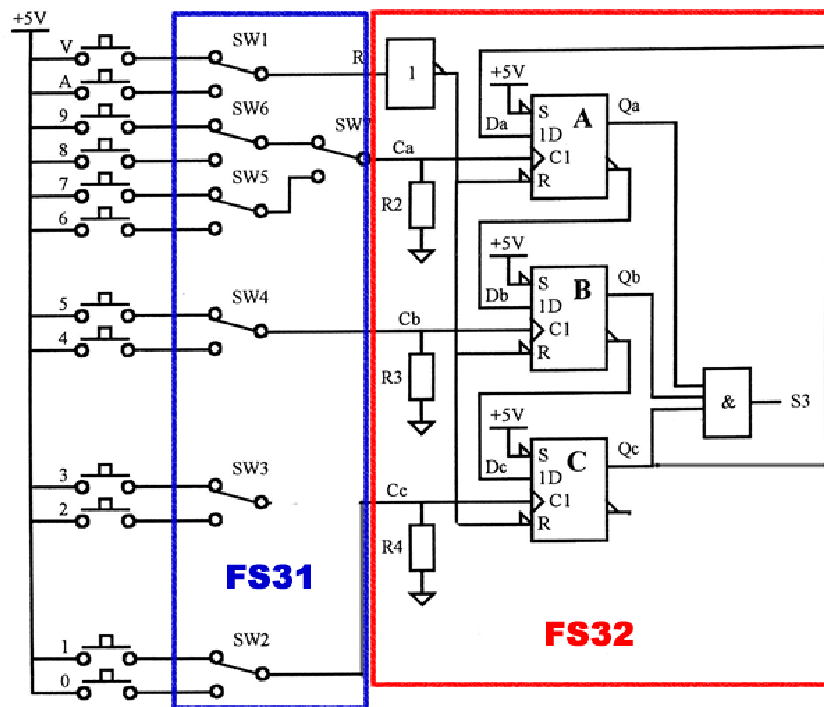


Figure 1 : Schéma des fonctions FP2 et FP3

II - 4 - Quel doit être l'état logique en sortie des bascules pour que la sortie **S3** du montage passe à 1 ?

.....

II - 5 - Quel état logique constant est appliquée sur les entrées S : **0** ou **1** ?

.....

II - 6 - Un appuie sur la touche **V** du clavier provoque quelle action sur les bascules D ?

.....

II - 7 - Après un appuie sur la touche **V** du clavier, quel est l'état logique de chacune des entrées D les bascules ?

.....

II - 8 - Déduisez des questions II - 7 et II - 4 le chiffre par lequel le code ne peut pas commencer :

.....

II - 9 - Sachant que le portail s'ouvre si **S3 = 1** et que la touche **V** est enfoncée avant de taper un code, quel est le code d'accès qui permet d'ouvrir le portail sur le montage de la Figure 1 ?

- 951     
  915     
  519     
  591     
  159     
  195

**III - Proposition et validation d'un montage à bascules D décodant un code à 4 chiffres**

Vous allez dans cette partie **III** proposer une solution permettant de décoder un code à 4 chiffres, puis valider votre proposition sur le logiciel de simulation Proteus. Pour réaliser votre solution vous disposez de 4 bascules D **4013** dont les symboles sont rappelés sur la Figure 2 et d'une porte logique ET à 4 entrée **4082**.

III - 1 - Quel est le niveau logique actif des entrées asynchrones **R** et **S** des bascules 4013 ?

.....

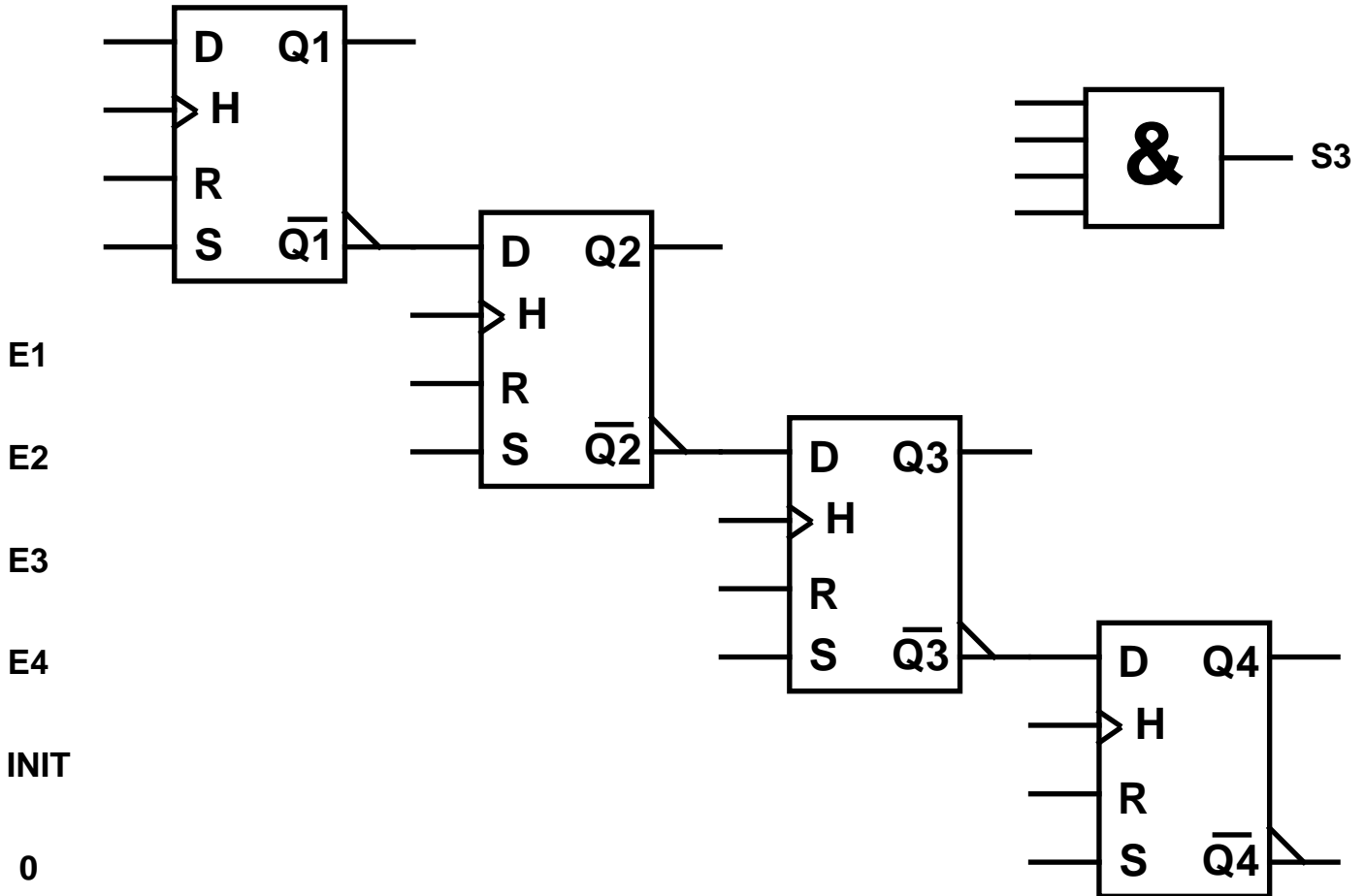
III - 2 - Quel est le front actif de l'entrée d'horloge **H** des bascules 4013 ?

.....

On désire réaliser un montage permettant de décoder le code à 4 chiffres **1234**. Sur la *Figure 2* :

- \* L'entrée logique **E1** vaut 1 si la touche **1** est enfoncée
- \* L'entrée logique **E2** vaut 1 si la touche **2** est enfoncée
- \* L'entrée logique **E3** vaut 1 si la touche **3** est enfoncée
- \* L'entrée logique **E4** vaut 1 si la touche **4** est enfoncée
- \* L'entrée logique **INIT** permet d'initialiser les bascules *avant de taper un nouveau code*
- \* La sortie logique **S3** doit passer à 1 seulement si le bon code **1234** est tapé dans le bon ordre
- \* Les entrées asynchrones non utilisées seront reliées au **0** logique

**III - 3** - En prenant exemple sur le schéma de la *Figure 1*, complétez le montage de la *Figure 2* afin qu'il décode le code **1234** en répondant au cahier des charges ci-dessus :



*Figure 2 : Solution à compléter permettant de décoder un code à 4 chiffres*

**III - 4** - Saisissez votre solution pour décoder un code à 4 chiffres dans le logiciel de simulation Proteus, en utilisant le circuit **4013** [bibliothèque **CMOS**] pour les bascules D, et en utilisant les générateurs suivant pour les entrées logiques :

- \* Les entrées **E1**, **E2**, **E3** et **E4** simulant l'appuie sur les touches seront générée par 4 générateurs **DPATTERN**
- \* L'entrée d'initialisation **INIT** sera généré par un générateur **DPULSE**
- \* Le niveau logique **0** sera généré par un générateur **DSTATE**

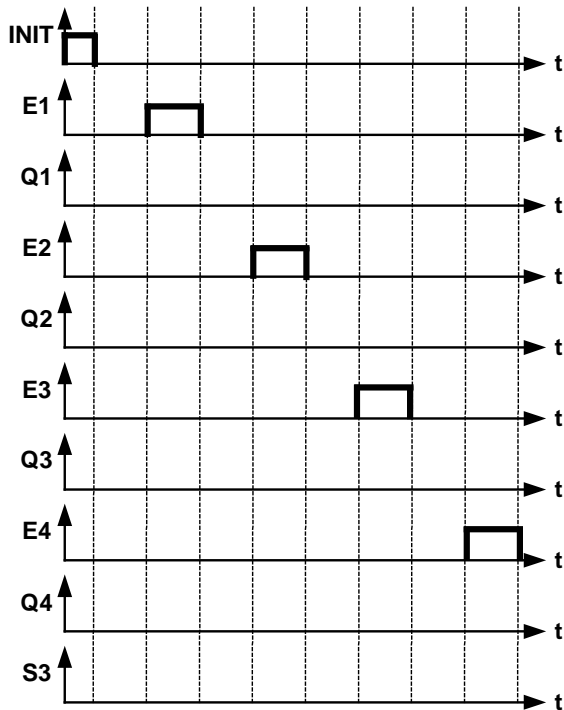
Les générateurs **DPATTERN** et **DPULSE** seront configurés afin de respecter les temps suivants :

- \* Les touches seront enfoncées durant 500 ms
- \* Un délai de 500 ms sans aucune touche enfoncée sera intercalé entre deux chiffres du code
- \* L'entrée **INIT** sera mise à 1 durant 250 ms à la mise sous tension du montage, puis passera et restera à 0

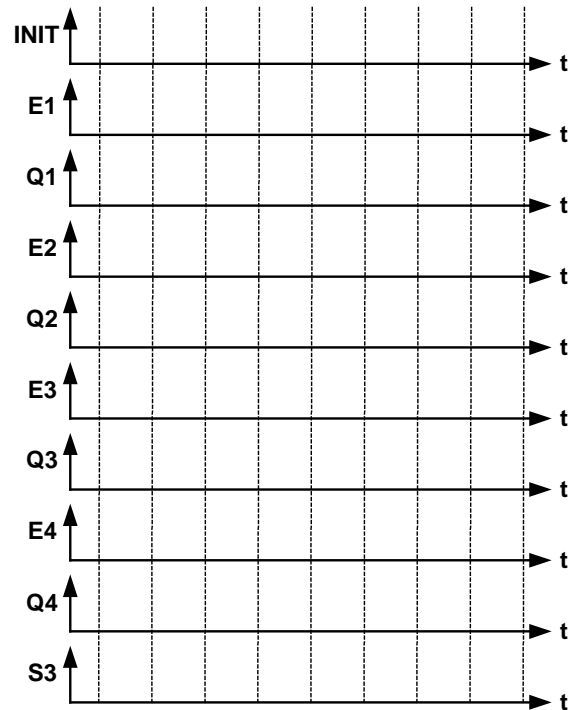
**III - 5** - Au moyen du logiciel de simulation Proteus, simulez la fonction FS32 décodant un code à 4 chiffres dans les cas suivants puis complétez les chronogrammes de la page 4 :

- \* tous les chiffres du code sont tapés dans l'ordre
- \* les chiffres du code sont tapés dans le désordre

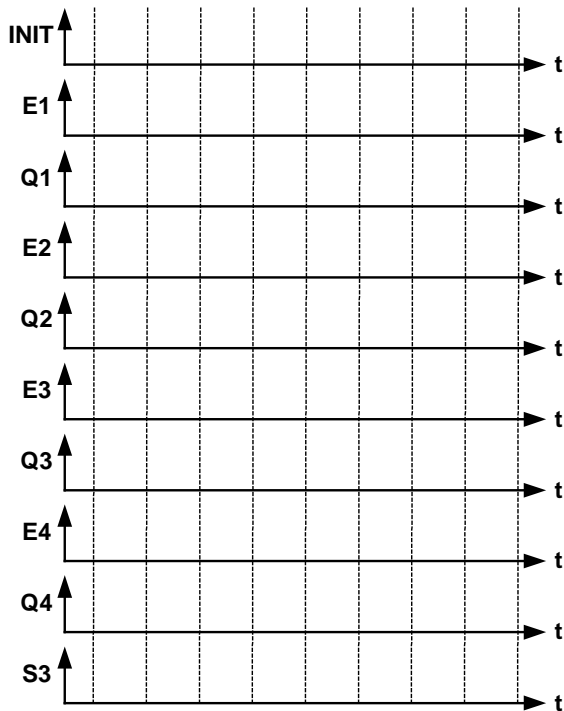
Si le code d'accès de votre montage *Figure 2* n'est pas **1234**, corrigez votre schéma, puis testez-le à nouveau dans le simulateur jusqu'à ce que le portail ne s'ouvre que si on tape le code **1234**.



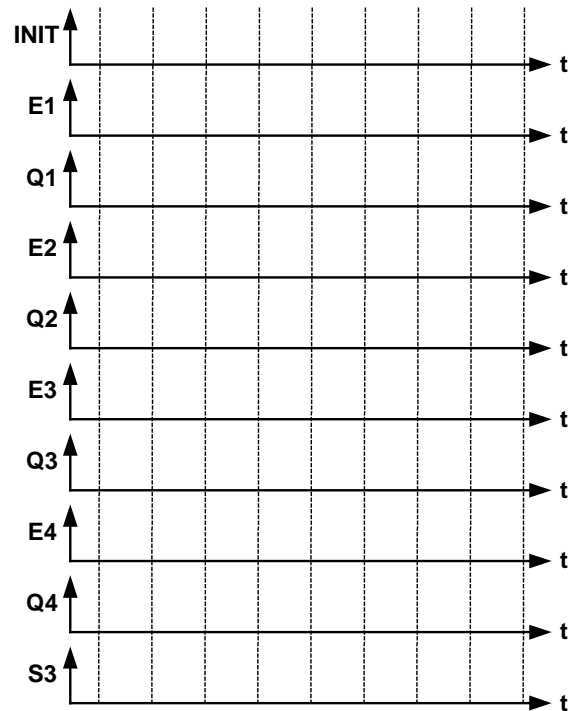
Le code tapé est **1234**



Le code tapé est **1243**



Le code tapé est **4321**



Le code tapé est **2413**

### RESSOURCE : Configuration du générateur DPATTERN sous Proteus

Un « pattern » [en anglais] signifie un « motif » [en français]. Dans la configuration du générateur **DPATTERN** il faut donc indiquer le motif du signal logique, c'est-à-dire les différents états logiques successifs [0 ou 1] sur une période. Ce motif s'appelle « **le train d'impulsion** » et se répète indéfiniment. En plus de la forme du train d'impulsion, il faudra également indiquer dans la fenêtre de configuration du générateur **DPATTERN** :

- \* la durée d'un bit [0 ou 1] dans le train d'impulsion en cochant **Temps états identiques**
- \* l'état initial du signal [état haut ou état bas] dans **Etat initial**
- \* cochez **Séquence continue d'impulsions**
- \* cochez **Train d'impulsion spécifiques** puis précisez le motif. Exemple de train d'impulsion : 0001001010011

**Rappel :** pour indiquer les grandeurs temporelles dans les générateurs **DPULSE** ou **DPATTERN** le logiciel Proteus accepte les préfixes **m** pour milli, **u** pour micro, et **n** pour nano [une valeur sans préfixe étant en seconde].