

## Projet : Système de gestion intelligente de parking

### Explication du module : Compteur Principal - Logique de comptage des voitures

Ce document a pour objectif d'expliquer clairement le fonctionnement du module : **Compteur Principal - Logique de comptage des voitures** afin que tous les membres du groupe puissent comprendre la logique, modifier le code si nécessaire et l'intégrer correctement au reste du projet.

#### Role du module :

Le module compteur\_places gère le **nombre de places disponibles dans le parking**.

- Il **décrémente le compteur** quand une voiture entre.
- Il **incrémente le compteur** quand une voiture sort.
- Il garantit que le compteur ne dépasse jamais la capacité maximale ni ne descend en dessous de zéro.

Ce module **ne décide pas si la voiture peut entrer ou non**. Cette décision est prise par le **module « Intégrateur Système - Connexion de tous les modules »**, qui envoie simplement les signaux d'entrée et de sortie.

#### Interface du module :

Type	Nom du signal	Description
Entrée	clk	Horloge synchronisant le compteur.
Entrée	rst	Réinitialisation. Remet le compteur à la capacité maximale.
Entrée	Voiture_entree	Impulsion envoyée par le module principal quand une voiture entre.
Entrée	Voiture_sortie	Impulsion envoyée par le module principal quand une voiture sort.
Sortie	Nb_places_dispo	Nombre de places disponibles

## Fonctionnement du module :

Le fonctionnement est très simple :

1. **Initialisation** : Au reset ( $\text{rst}=1$ ), le compteur est remis à la **capacité maximale du parking (100 places)**.
2. **Voiture entrant** : Si  $\text{voiture\_entree} = 1$  et que le compteur  $> 0$ , le compteur **diminue de 1**.
3. **Voiture sortant** : Si  $\text{voiture\_sortie} = 1$  et que le compteur  $<$  capacité maximale, le compteur **augmente de 1**.

Le module fonctionne de manière **synchrone avec l'horloge** pour assurer un comportement stable et prévisible.

## Structure interne du code :

Le **module compteur de voitures** est construit comme un **bloc unique**, mais il fait trois choses importantes :

1. **Stockage de l'état du compteur** :

Le module garde en mémoire **combien de places sont disponibles** à chaque instant ,

c'est comme un petit tableau qui se souvient toujours du nombre de places libres.

2. **Décision de comptage** :

Quand une voiture arrive ( $\text{voiture\_entree}$ ) → le compteur **diminue d'une place**.

Quand une voiture sort ( $\text{voiture\_sortie}$ ) → le compteur **augmente d'une place**.

Le compteur **ne descend jamais en dessous de zéro** et **ne dépasse jamais la capacité maximale (100 places)**.

3. **Sortie vers l'extérieur**

Le compteur transforme sa valeur en **signal lisible par les autres modules** (`std_logic_vector`).

Cette valeur peut être utilisée pour :

Afficher le nombre de places sur un afficheur 7 segments.

Permettre au module principal de savoir s'il peut autoriser une voiture à entrer.

## Intégration dans le projet :

Dans le **module principal (top-level)** :

- On instancie compteur\_places.
- On fournit clk et rst.
- On envoie les impulsions voiture\_entree et voiture\_sortie selon les actions détectées (capteurs, décisions du module principal).
- On récupère nb\_places\_dispo pour :
  - Afficher le nombre de places disponibles sur un afficheur 7 segments ou LCD.
  - Prendre des décisions comme **ouvrir ou fermer la barrière** si le parking est plein ou vide.

Le module est autonome pour **la gestion du comptage**, et le module « **Intégrateur Système - Connexion de tous les modules** » ne se préoccupe que de **l'autorisation de passage** et de l'affichage.