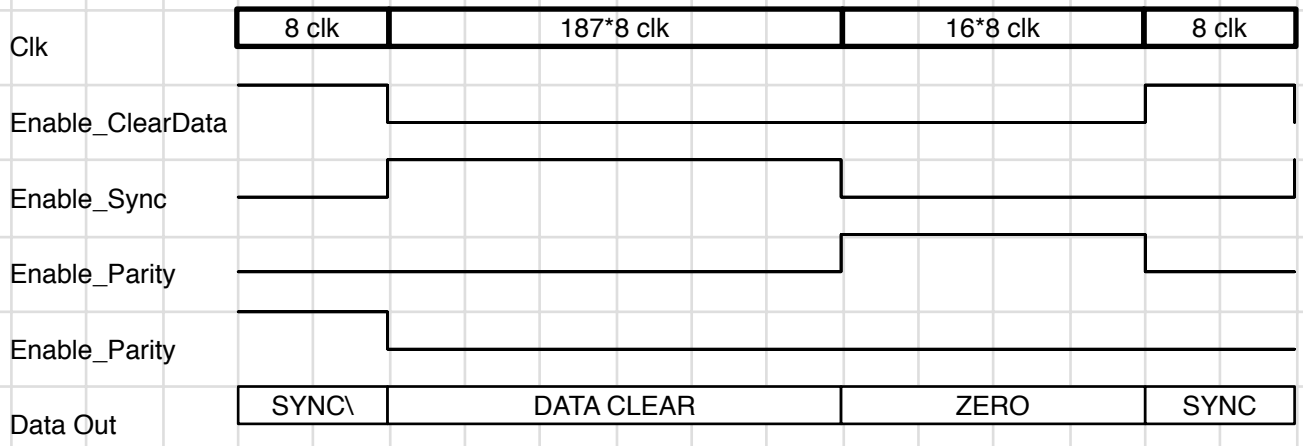
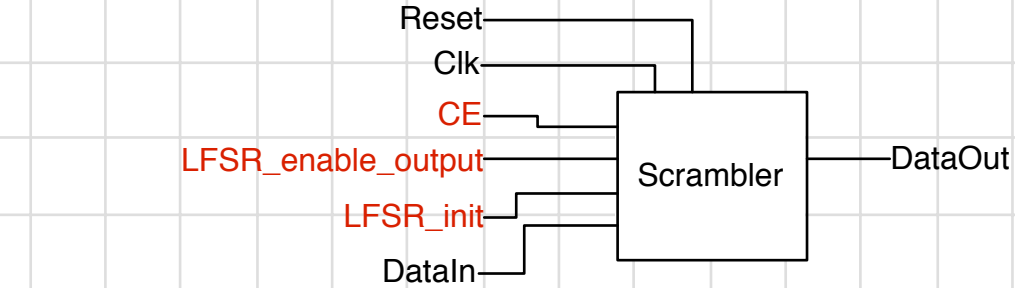
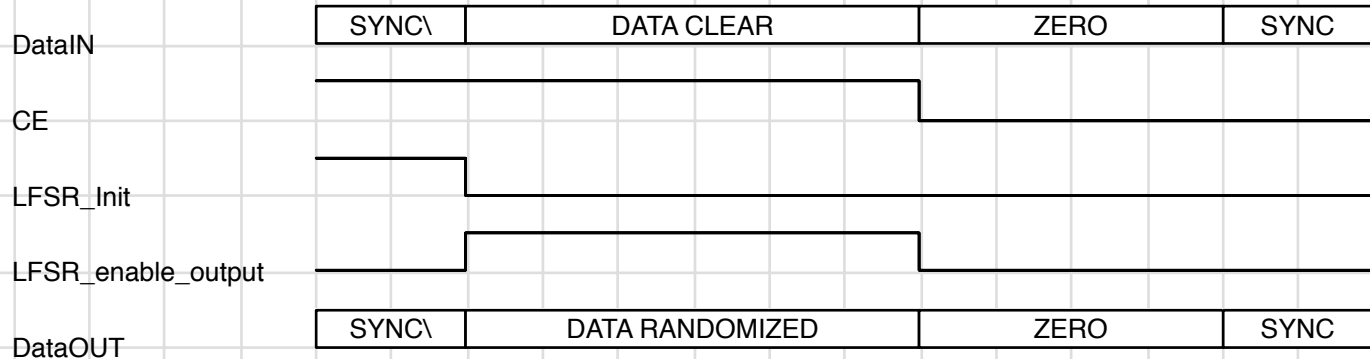
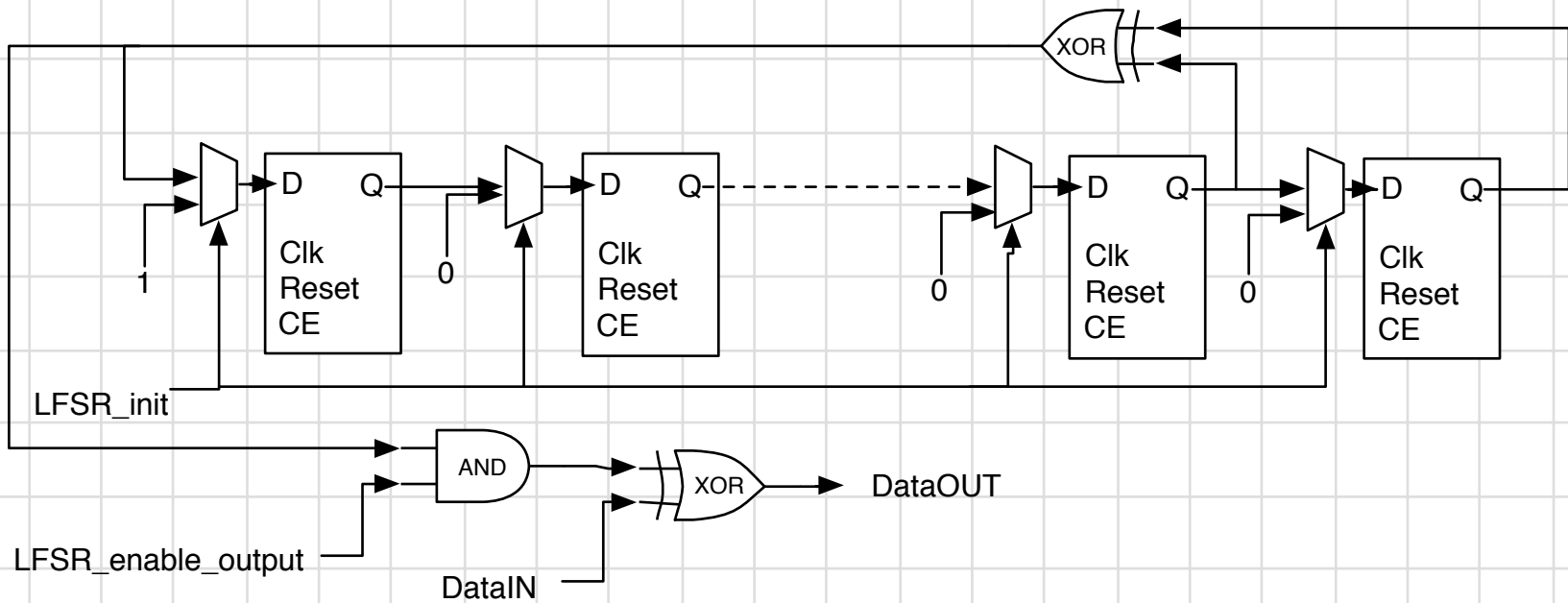


Principe : Générer les données pré-formatées pour le reste de la chaine
 Les « Enable » permettent de choisir quel type de donnée est générée
 Fonctionne au bit

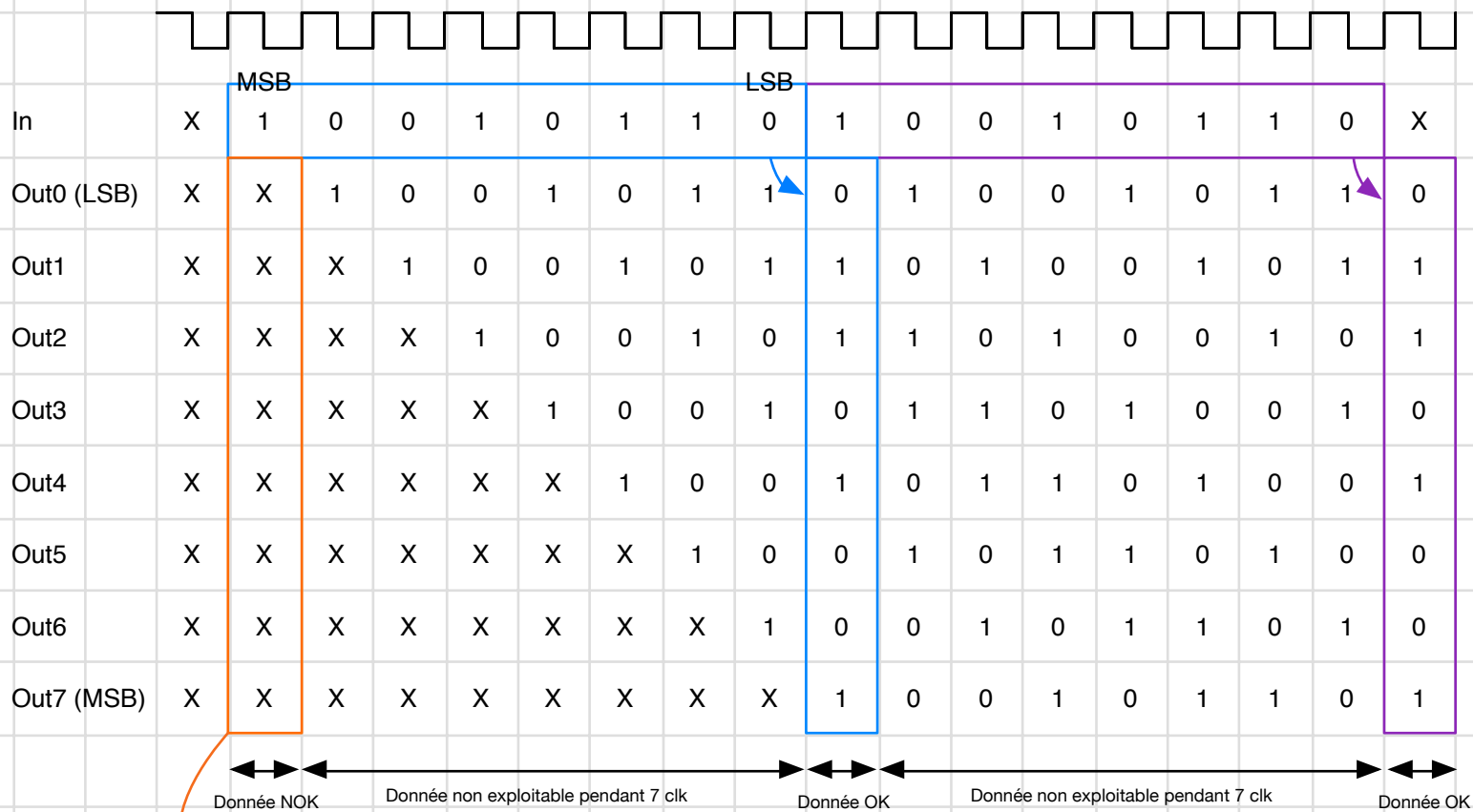
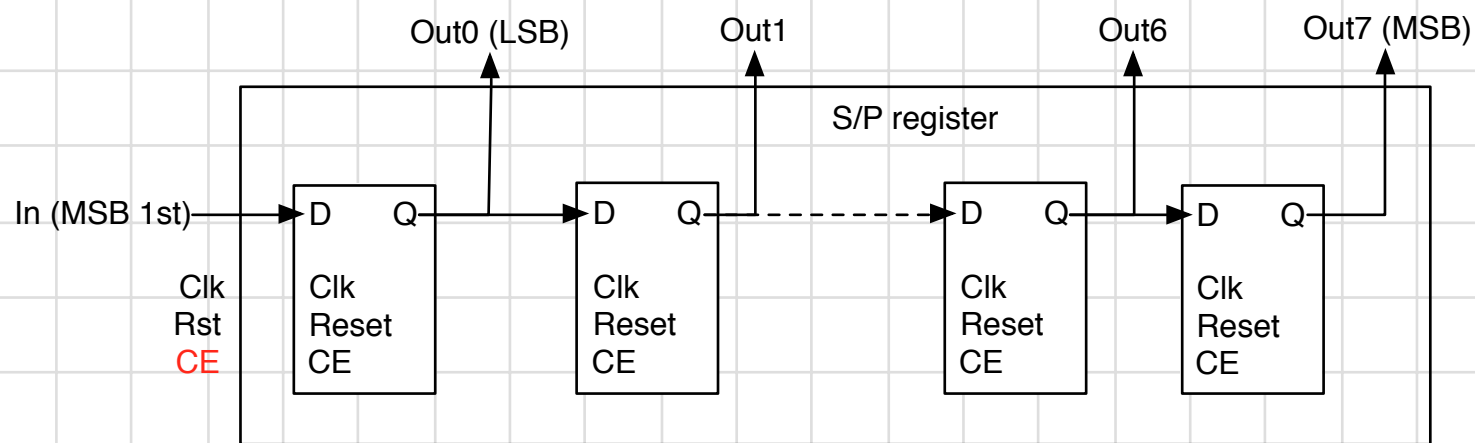




Principe : Une valeur pseudo aléatoire est générée par les bascules. Un XOR est effectué avec la data (et seulement elle). Puis lorsque *CE* et *LFSR_init* sont à '1' (pendant SYNC\ uniquement) une valeur d'init est chargée dans chaque bascule.



Rouge = provenance du controleur

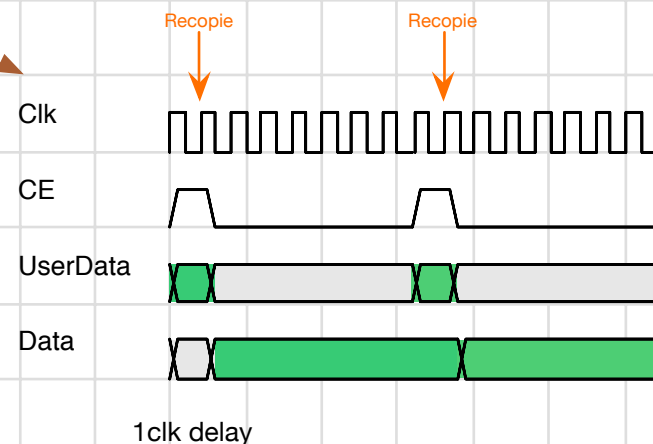
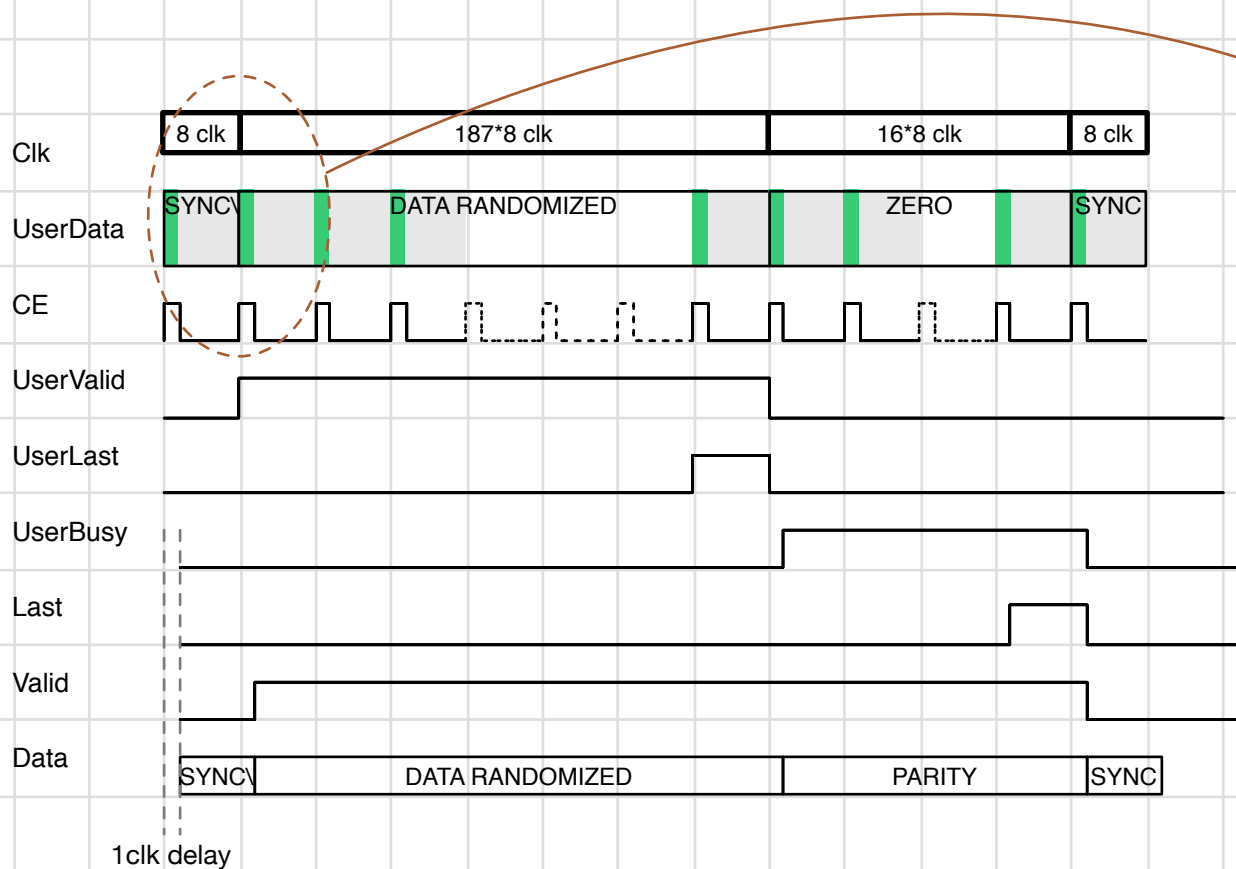


Au démarrage/reset, 1er octet non exploitable = 8 clk de latence

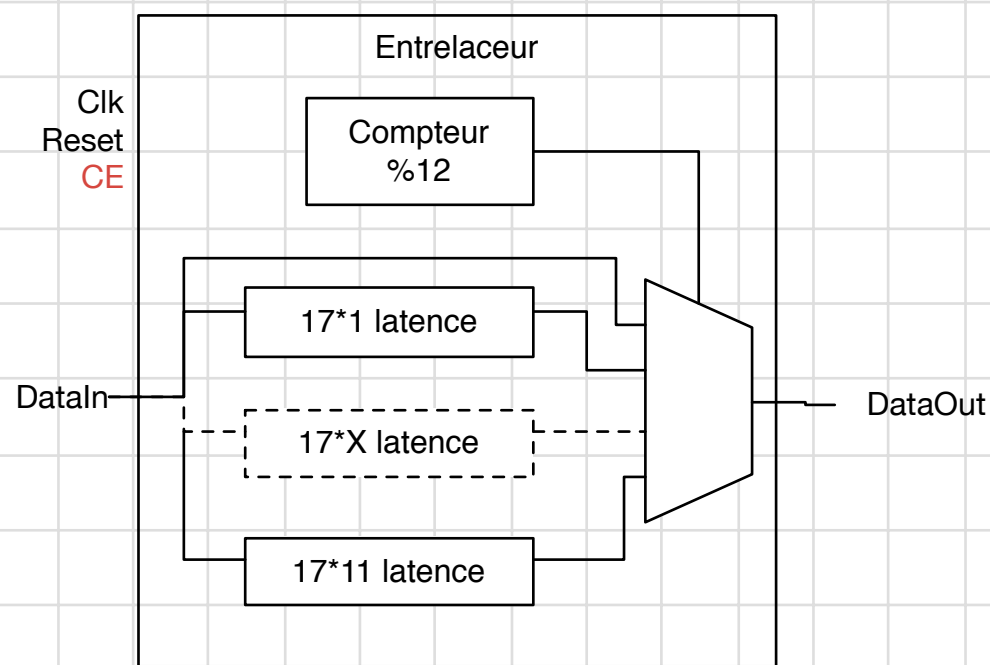
Rouge = provenance du controleur

Clk
 Rst
 CE
 UserValid
 UserLast
 UserData
 Valid
 Last
 UserBusy
 Data

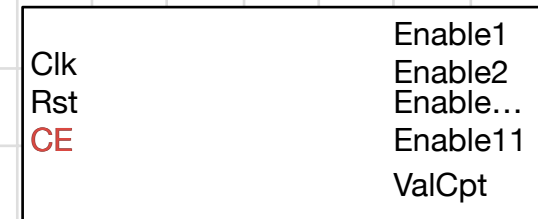
Principe : la parité est calculée sur la data (UserValid) puis est insérée dans le signal après la data (UserLast)
 Le circuit ne doit fonctionner que sur les octets valides (verts) : ajout d'un CE à '1' sur 1clk sur 8
 UserValid est positionné à '1' pendant la data pour le calcul de la parité
 UserLast est positionné à '1' pendant le dernier octet de data
 Circuit séquentiel : 1clk de retard
 Le circuit doit aussi fonctionner pendant les sync (sans UserValid ni UserLast) pour recopie entrée sur sortie
 Les signaux de sorties sont envoyés au controleur
 À cause des 8clk de latence du s/p, UserValid, UserLast et CE doivent (au reset) débiter avec 8clk de retard



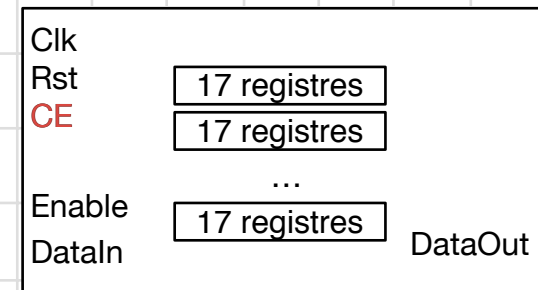
Rouge = provenance du controleur



Compteur %12



17*X latences :



Nous créons un **bloc de 17 registres**.

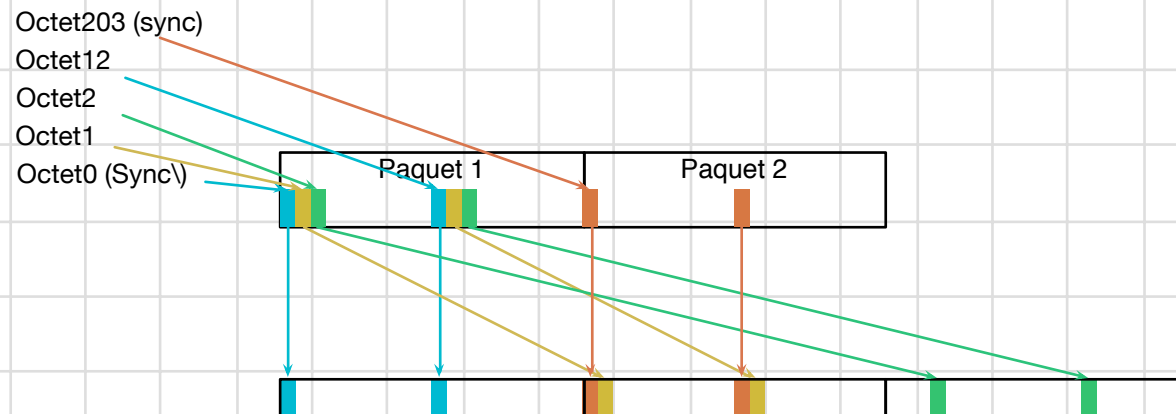
Chaque **bloc de latence** possède un nombre différent de bloc de registres.

Chacun (registre/latence) est cadencé grâce à un Enable.

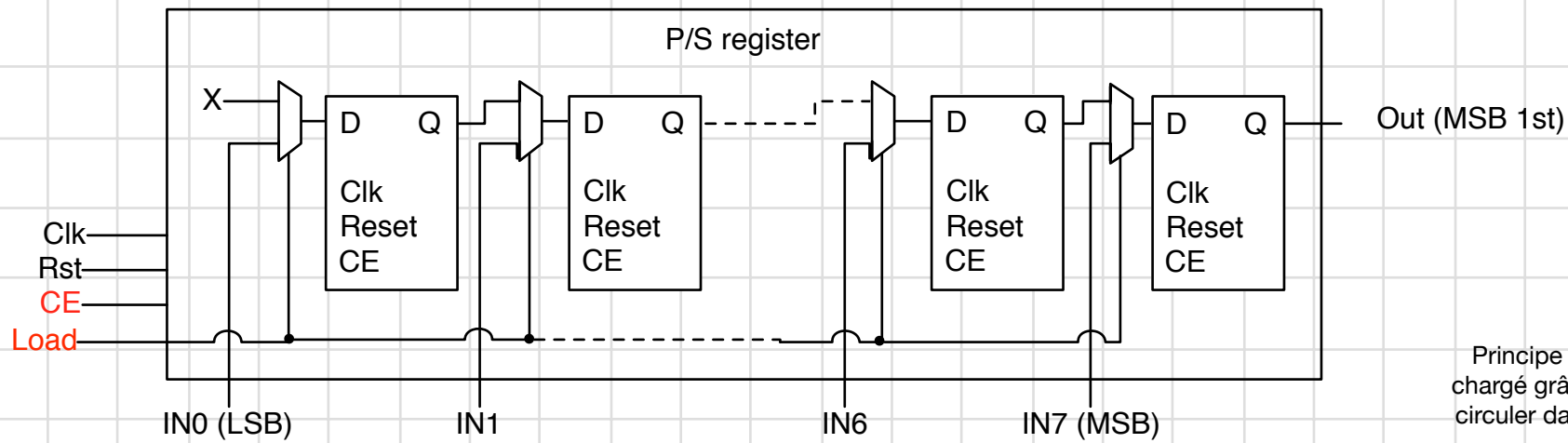
Le compteur compte de 0 à 11 et génère les 12 signaux « enable » en fonction de sa valeur.

Fonctionne à l'octet : nécessite un CE à '1' sur 1 clk sur 8

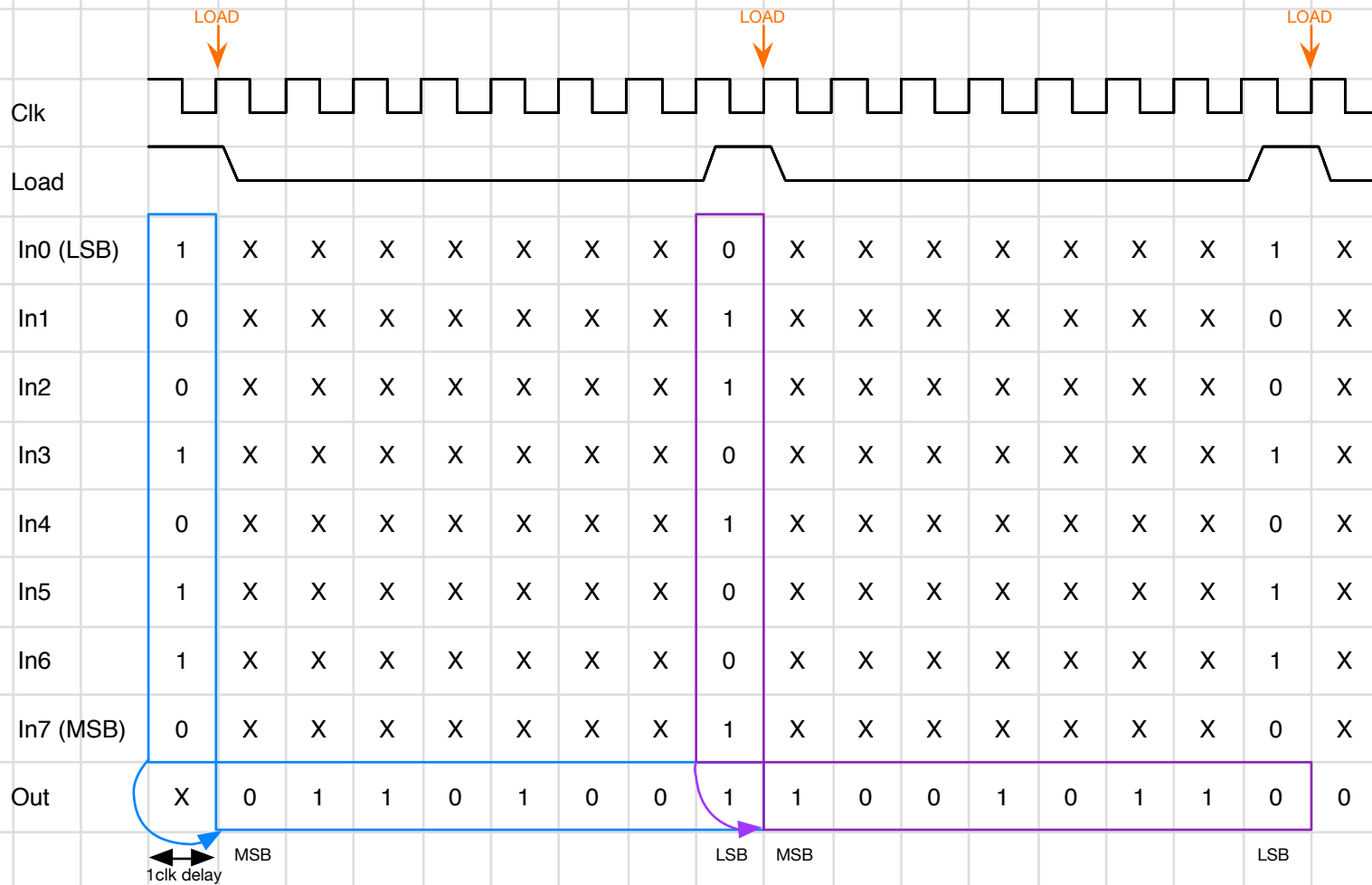
Aucune latence



Rouge = provenance du controleur

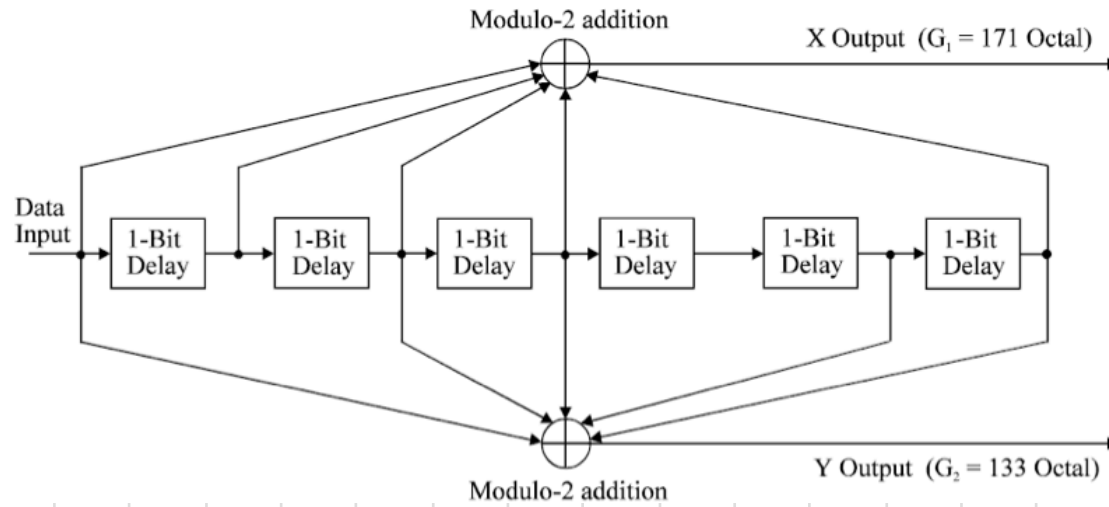


Principe : s rialise l'octet pr alablement charg  gr ce au signal « load » en le faisant circuler dans les bascules. Introduit 1 coup d'horloge de retard.



Rouge = provenance du controleur

Codeur convolutif



Nous recopions le schéma de la norme
Les additionneurs sont réalisés avec des XOR à 2 entrées en cascade
Les « 1 bit delay » sont des bascules D