

Circuits logiques combinatoires et séquentiels

Guy Bégin

9 novembre 2022

Circuit séquentiels : registres et compteurs

Objectifs

- Être familier avec les principaux circuits séquentiels classiques : registres, registres à décalage, compteurs
- Savoir comment sont implémentées les différentes opérations : chargement, remise à zéro, comptage vers le haut/bas, décalage
- Pouvoir utiliser des compteurs pour générer des séquences de synchronisation
- Pouvoir distinguer entre compteur synchrone et compteur asynchrone

- Un registre est un groupe de bascules activées par un même signal d'horloge.
- Chaque bascule permet de stocker un bit en mémoire.
- Différentes configurations d'interconnexion entre les bascules et éventuellement des composantes combinatoires permettent de concevoir des types de registres pouvant remplir des rôles variés.

Registre parallèle à quatre bits

- La figure 1 montre un registre parallèle de quatre bits, qui permet de stocker 4 valeurs binaires indépendantes.
- Le schéma du bas est une représentation symbolique du registre, dans laquelle on représente les entrées et sorties comme des vecteurs de quatre bits.

Registre parallèle à quatre bits

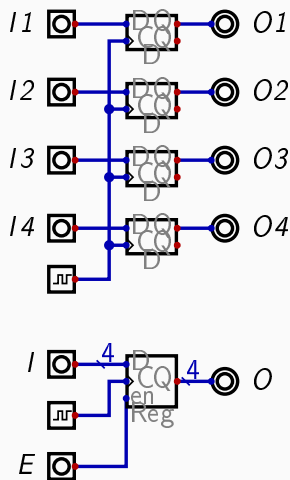


Figure 1 – Registre parallèle à quatre bits

Chargement parallèle

- Si on veut concevoir un registre parallèle polyvalent, on doit le munir de la possibilité de le charger à partir des entrées ou de maintenir les valeurs déjà mémorisées.
- On ajoutera donc une entrée *charge* au registre pour contrôler ces opérations.
- Comment mettre en oeuvre ce chargement/maintien demande un peu de réflexion.
- Il serait possible d'agir (à la façon d'un signal *enable* via une porte ET par exemple) sur l'entrée d'horloge des bascules pour empêcher leur contenu d'être affecté par les entrées.
- Mais alors, on briserait le principe de synchronisation qui veut que tous les éléments d'un système soient tous commandés en même temps par une même horloge.

La solution consiste à toujours mettre à jour le contenu des bascules :

1. Lorsque *charge* est inactif (fonction maintien), la sortie de chaque bascule, réacheminée à l'entrée, est sélectionnée pour récrire le même contenu.
2. Lorsque *charge* est actif (fonction chargement), c'est l'entrée externe qui est sélectionnée pour écrire un nouveau contenu.

Sélection de l'entrée des cellules

- La sélection se fait au moyen d'un multiplexeur deux-vers-un à l'entrée de chaque bascule.
- La figure 2 montre le schéma du registre chargeable.

Registre parallèle à quatre bits chargeable

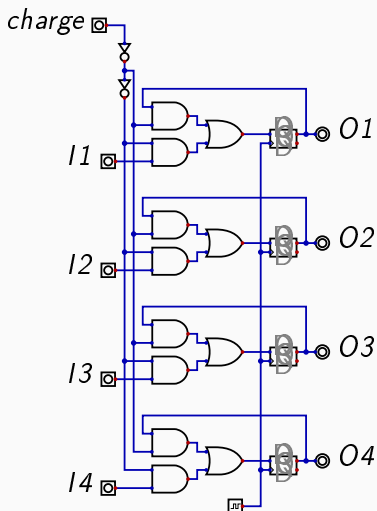


Figure 2 – Registre parallèle à quatre bits chargeable

Registres à décalage

- Un registre à décalage consiste en une chaîne de bascules, la sortie de l'une reliée à l'entrée de la suivante.
- La figure 3 montre un registre à décalage de quatre bits.
- À chaque coup d'horloge, l'entrée est insérée dans la première bascule, à droite, et le contenu du registre est décalé d'une position vers la gauche.
- La sortie provient de la dernière bascule à droite.

Registre à décalage 4 bits

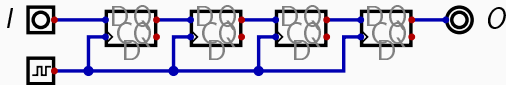


Figure 3 – Registre à décalage 4 bits

Sélection de l'entrée des cellules

- En utilisant un multiplexeur quatre-vers-un pour sélectionner ce qui sera inséré dans une bascule, il est possible de concevoir un registre à décalage universel.
- Les différents opérations sont le *maintien*, le *décalage à droite* avec entrée G , le *décalage à gauche* avec entrée D et le *chargement parallèle*, avec les entrées $I_i, i = 1, \dots, 4$.

- Les différentes opérations sont commandées par les deux signaux de sélection, tel qu'indiqué dans le tableau 1.

Table 1 – Codes de sélection et opérations

Sél.	Action
00	Maintien
01	Décalage droite
10	Décalage gauche
11	Chargement parallèle

Registre à décalage universel

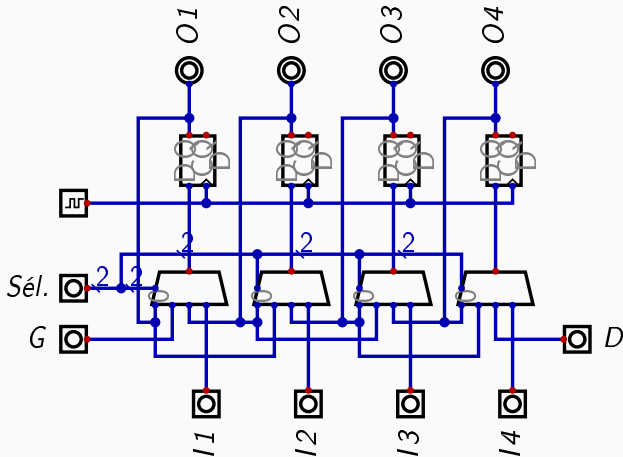


Figure 4 – Registre à décalage universel

- Les registres à décalage sont notamment utilisés pour convertir des données parallèles en données sérielles et vice versa, des opérations très utiles dans le contexte d'interfaces de communication.
- On peut également s'en servir pour faire des multiplications ou divisions par deux, comme on l'a vu précédemment.

Compteurs

- Un registre dont la séquence d'états est systématique est appelé un **compteur**.
- Le comptage peut être contrôlé par une entrée spécifique ou par l'entrée d'horloge.
- La séquence d'états est toujours la même pour un type de compteur donné.
- Par exemple, les états d'un compteur binaire à deux bits suivent la séquence $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00 \dots$. La sortie correspond directement aux bits d'état.
- On distingue les compteurs **asynchrones** et les compteurs **synchrones**.
- Quel que soit le type de compteur, il y a toujours un retour vers l'état initial, car la séquence d'états est un cycle.

Compteur asynchrone

- Dans un compteur binaire asynchrone, la sortie d'une bascule de poids moins significatif est acheminée à l'entrée d'horloge de la bascule suivante.
- C'est la transition de la sortie de la bascule de poids moins significatif qui déclenche la bascule suivante.
- La figure 5 montre un compteur asynchrone construit à partir de bascules T.
- La séquence de sortie est donnée dans le tableau 2.
- On peut voir qu'après huit étapes, la séquence se répète.

Compteur asynchrone

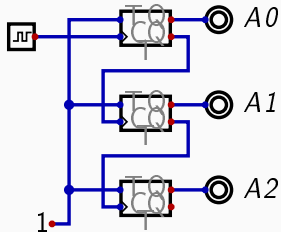


Figure 5 – Compteur asynchrone

Table 2 – Séquence du compteur

A_2	A_1	A_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

Avantages et inconvénients

- Les compteurs asynchrones sont très simples, mais l'inconvénient est que les transitions d'états ne sont pas synchrones.
- En particulier, les bits d'état ne changent pas tous en même temps.
- Par exemple, si le compteur passe de 0111 à 1000, la sortie peut passer par des états intermédiaires parasites : $0111 \rightarrow 0110 \rightarrow 1100 \rightarrow 1000$.

Compteur synchrone

- Dans un compteur synchrone, toutes les bascules sont commandées par un même signal d'horloge, et les changements d'états sont synchronisés.
- Les changements d'états sont contrôlés par les signaux d'entrée appliqués aux bascules, comme dans le fonctionnement normal d'un circuit séquentiel synchrone.
- Le diagramme d'état d'un compteur trois bits (huit états) est un cycle, comme on peut le voir sur la figure 6.
- Le tableau d'états correspondant est donné dans le tableau 3.

Diagramme d'état d'un compteur

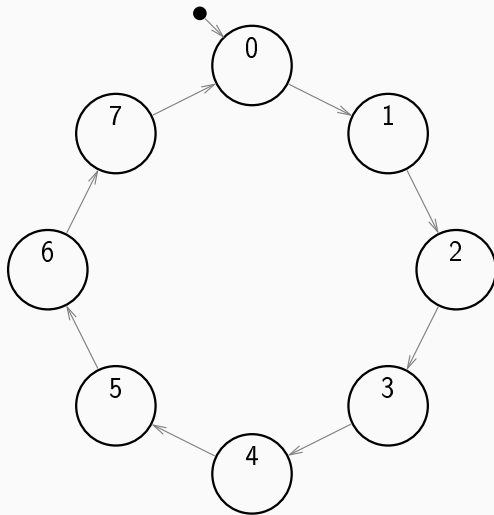


Figure 6 – Diagramme d'état d'un compteur

Tableau d'état du compteur

Table 3 – Tableau d'état du compteur

Z_2^n	Z_1^n	Z_0^n	Z_2^{n+1}	Z_1^{n+1}	Z_0^{n+1}
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Les expressions pour le décodeur de prochain état sont :

$$Z_2^{n+1} = Z_0^n \cdot Z_1^n \cdot (Z_2^n)' + (Z_0^n)' \cdot Z_2^n + (Z_1^n)' \cdot Z_2^n$$

$$Z_1^{n+1} = Z_0^n \cdot (Z_1^n)' + (Z_0^n)' \cdot Z_1^n$$

$$Z_0^{n+1} = (Z_0^n)'$$

Le schéma correspondant est donné à la figure 7.

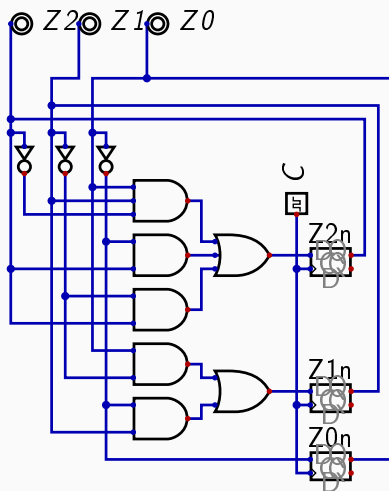


Figure 7 – Schéma logique du compteur à 3 bits

On peut ajouter aux compteurs des fonctions diverses :

- comptage vers le haut,
- comptage vers le bas,
- préchargement parallèle,
- remise à zéro, etc.

Conception de compteurs

- Le compteur de la figure 7 a été conçu comme un circuit séquentiel général, avec un décodeur de prochain état en forme *somme de produits*.
- Il est également possible de concevoir un compteur synchrone directement, sans passer par la méthodologie classique, en suivant un raisonnement tout simple.
- La bascule du bit le moins significatif Z_0 doit changer d'état à tous les coups d'horloge.
- La bascule du bit suivant Z_1 doit changer d'état seulement lorsque le bit précédent Z_0 vaut 1.
- La bascule du bit suivant Z_2 doit changer d'état seulement lorsque les bits précédents Z_1, Z_0 valent tous deux 1.

Et on peut pousser le raisonnement pour un compteur quelconque :
*la bascule d'un bit Z_i doit changer d'état seulement lorsque
les bits précédents $Z_{i-1}, Z_{i-2}, \dots, Z_0$ valent tous 1.*

Exemple pour compteur 4 bits

- Le compteur à quatre bits de la figure 8 a été conçu selon cette approche, à partir de bascules JK.
- L'utilisation d'une porte ET par bascule permet de mettre en oeuvre les conditions.
- L'entrée E est un contrôle *enable* pour activer le comptage.
- On a aussi prévu une sortie Prochain pour pouvoir connecter en cascade d'autres compteurs.

Schéma logique du compteur à 4 bits

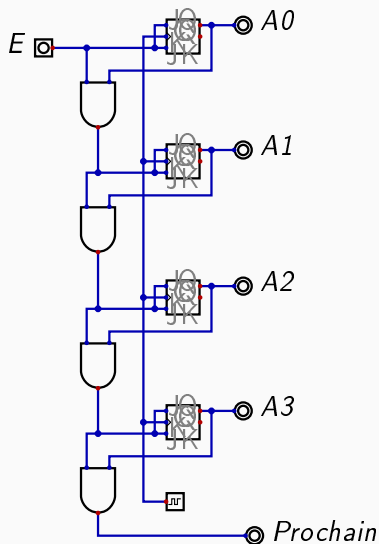


Figure 8 – Schéma logique du compteur à 4 bits

Si on réfléchit de la même façon au comptage vers le bas, on constate que la règle devient :

la bascule d'un bit Z_i doit changer d'état seulement lorsque les bits précédents $Z_{i-1}, Z_{i-2}, \dots, Z_0$ valent tous 0.

Cette fois-ci, les conditions se baseront sur les sorties complémentées des bascules précédentes.

- En combinant les deux conditions au moyen d'un multiplexeur deux-vers-un, il est facile de concevoir un compteur haut/bas, tel qu'illustré sur la figure 9.

Compteur haut/bas à 4 bits

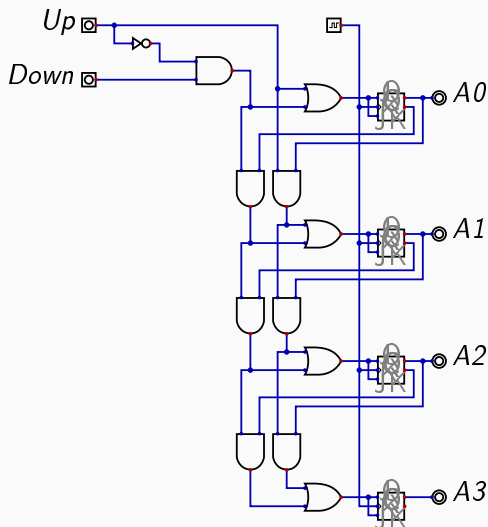


Figure 9 – Schéma logique du compteur haut/bas à 4 bits

- Un compteur en anneau est un registre à décalage connecté en boucle où une seule bascule est active à la fois.
- Il y a donc dans la sortie un seul bit 1, qui se décale de façon cyclique : $0010 \rightarrow 0001 \rightarrow 1000 \rightarrow 0100 \rightarrow 0010, \dots$
- La figure 10 illustre un compteur en anneau de quatre bits.

Compteur en anneau à 4 bits

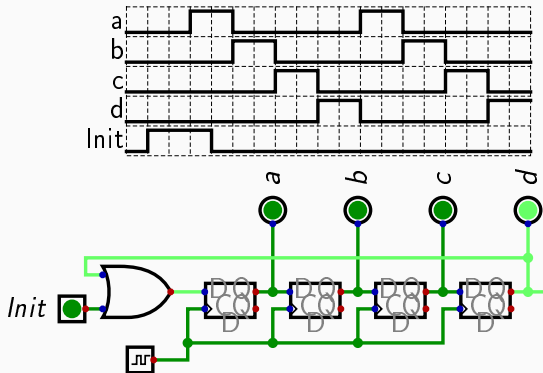


Figure 10 – Compteur en anneau à 4 bits

- Une entrée Init permet d'injecter un bit 1 dans le registre au début.
- La trace montre les formes d'onde obtenues.
- On utilise fréquemment ce type de compteur pour générer des signaux de synchronisation.
- En effet, chaque sortie devient active à son tour dans le cycle, pendant une période d'horloge.

- Un compteur Johnson permet de doubler le nombre d'états distincts par rapport au compteur en anneau en injectant le complément du dernier bit dans l'entrée du registre à décalage.
- La figure 11 illustre un compteur en anneau Johnson de quatre bits, de même que la trace de fonctionnement.
- La séquence d'états est donnée dans le tableau 4.

Compteur Johnson à 4 bits

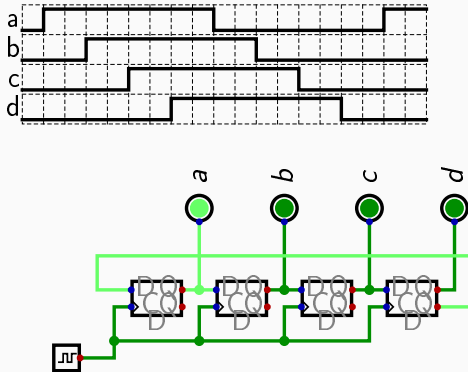


Figure 11 – Compteur Johnson à 4 bits

Séquence d'états du compteur Johnson

Table 4 – Séquence d'états du compteur Johnson

No.	a	b	c	d	ET requis
1	0	0	0	0	$a'd'$
2	1	0	0	0	ab'
3	1	1	0	0	bc'
4	1	1	1	0	$c'd'$
5	1	1	1	1	ad'
6	0	1	1	1	$a'b$
7	0	0	1	1	$b'c$
8	0	0	0	1	$c'd$

Signaux de synchronisation à partir des sorties Johnson

- On peut construire des signaux de synchronisation distincts en combinant deux par deux au moyen d'une porte ET des signaux de sortie voisins (dans le cycle) ou leurs compléments.
- Le tableau 4 donne les paires de sorties à combiner pour ce faire avec le compteur Johnson de quatre bits.
- Nous avons appliqué ce principe à un compteur Johnson de deux bits, présenté sur la figure 12.
- La figure montre une trace d'exécution avec les signaux de sortie.
- On y voit que chacun des quatre signaux est activé à son tour.

Compteur Johnson à 2 bits et circuit de décodage

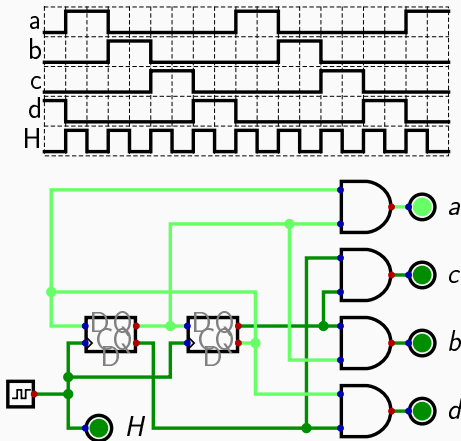


Figure 12 – Compteur Johnson à 2 bits et circuit de décodage pour signaux de synchronisation

Si on s'intéresse aux sorties des bascules de ce même compteur Johnson, on peut voir sur la trace d'exécution de la figure 13 qu'on obtient des signaux en **quadrature**, c'est-à-dire que les sorties sont déphasées de 90 degrés les unes par rapport aux autres, comme le sont des fonctions $\sin()$, $\cos()$, $-\sin()$, $-\cos()$.

Génération de signaux en quadrature

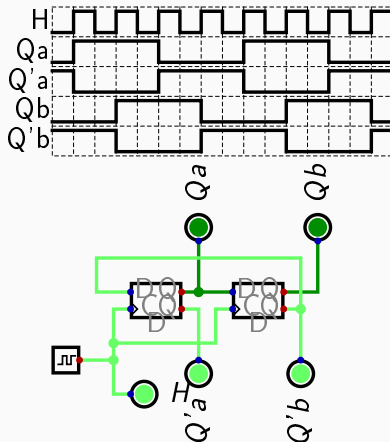


Figure 13 – Signaux en quadrature obtenus au moyen d'un compteur Johnson à 2 bits

Diviseur de fréquence

- Un compteur peut être utilisé pour diviser la fréquence d'un signal périodique.
- Par exemple, la sortie d'un compteur à un bit change d'état à tous les deux coups d'horloge, ce qui constitue une division par deux de la fréquence d'horloge.
- Cette approche permet naturellement des divisions par des puissances de deux.
- On peut aussi utiliser un compteur Johnson, qui permettra alors, selon le nombre d'étages du compteur, des divisions de fréquence par des diviseurs, comme trois ou cinq, qui ne sont pas des puissances de deux.

Compteur à chargement parallèle

- Un compteur à chargement parallèle est illustré à la figure 14.
- En activant l'entrée Compte, le comptage se fait vers le haut.
- En activant l'entrée Charge, les entrées $I_i, i = 0, \dots, 3$ sont insérées dans les bascules.
- Il y a aussi une sortie ov qui indique lorsque le compteur atteint sa valeur maximale.
- Cette sortie peut être utilisée pour activer un autre compteur pour des bits de plus haut niveau.

Compteur à chargement parallèle

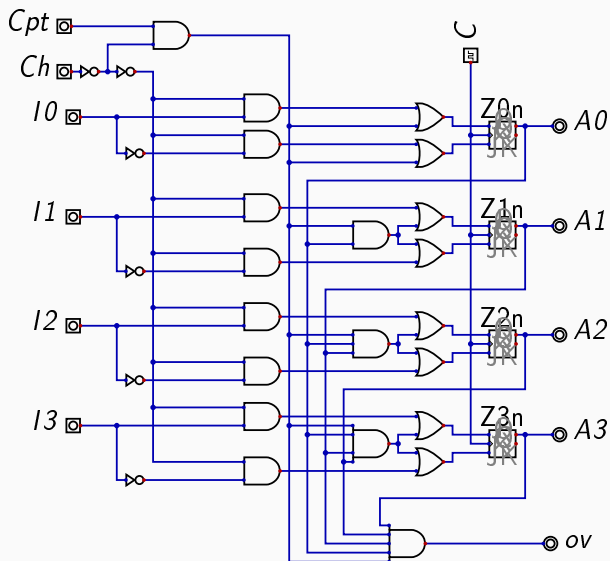


Figure 14 – Compteur à chargement parallèle

Trace d'exécution, de 0 à 15

- La trace d'exécution de la figure 15 montre le comptage de 4 jusqu'à 15 et retour à 0.
- On voit le signal ov s'activer sur 15.

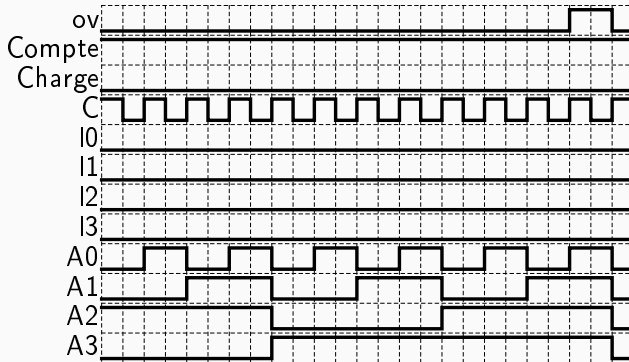


Figure 15 – Trace d'exécution, de 0 à 15

Trace d'exécution, de 0 à 5 et chargement de 12

- La trace de la figure 16 montre le compteur qui passe de 0 à 5, puis un chargement parallèle de la valeur 12.

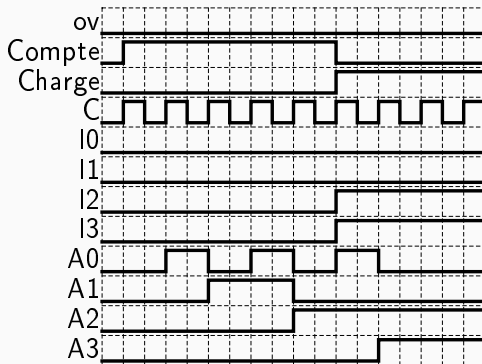


Figure 16 – Trace d'exécution, de 0 à 5 et chargement de 12

Compteur modulo

- On peut réaliser aisément un compteur modulo dont le cycle est plus court que le maximum possible, en utilisant un compteur avec chargement parallèle.
- Par exemple, pour réaliser un compteur qui compte de 0 jusqu'à 12, il suffit de décoder au moyen d'une porte ET l'état qui doit être le dernier du cycle, et d'utiliser le signal de sortie obtenu pour charger la valeur 0 dans le compteur.
- On obtient ainsi un compteur modulo-13, dont le cycle compte 13 états.
- La figure illustre le compteur modulo-13, de même qu'une trace d'exécution sur laquelle on voit le passage de 12 à 0.

Compteur modulo-13

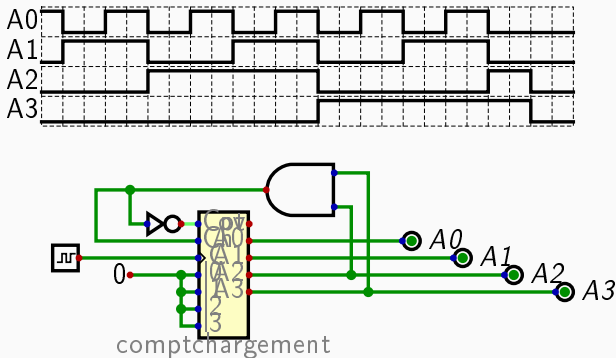


Figure 17 – Compteur modulo-13

On pourrait aussi réaliser un compteur qui compte par exemple de 4 à 15, en utilisant cette fois la sortie ov pour activer le chargement d'une valeur initiale 4.