

Travail 6

Circuit digitaux 2

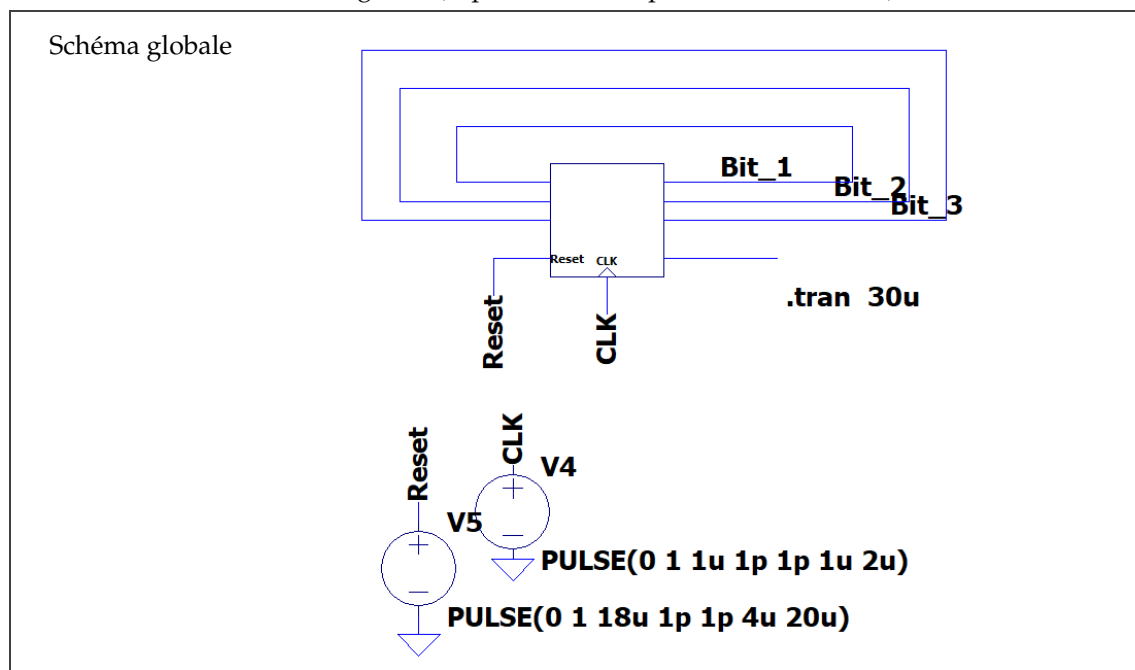
Dans ce devoir, nous allons combiner les portes logiques CMOS que vous maîtrisez maintenant à merveille afin d'obtenir un système plus complexe rythmé par un signal d'horloge.

Plus précisément, nous vous demandons d'implémenter via LTspice un compteur synchrone sur 3 bits avec un signal de reset. Lorsque le signal de reset est actif, la valeur du compteur est réinitialisée (i.e. remise à zéro). À l'inverse, lorsque ce dernier n'est pas actif, le compteur compte indéfiniment et sa valeur évolue comme suit : 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, ... Pour ce devoir, on considère un reset *active-high*, ce qui signifie que le reset est actif lorsque celui-ci a une valeur logique haute. Pour cela, utilisez les transistors N_50n et P_50p (dont les modèles sont fournis dans le fichier "cmosedu_models.txt" et fonctionnant entre 0V et 1V). Pour améliorer la représentation du circuit (et vous faciliter la tâche), nous vous demandons d'utiliser des sous-circuits pour implémenter la D-FF et la logique combinatoire (cfr [cette vidéo](#) sur [ce site](#) pour plus d'information sur l'utilisation des sous-circuits dans LTspice). Notez que le sous-circuit de la D-FF vous est donné (cfr les fichiers [D_FF.asc](#), [D_FF.asy](#), [Inverter.asc](#) et [Inverter.asy](#)). Effectuez le dimensionnement de chaque transistor (à l'exception de ceux de la D_FF). Enfin, simulez le circuit et déterminez (par simulation) la fréquence maximale de fonctionnement de votre circuit. Il vous est conseillé d'utiliser des paramètres SPICE pour définir la fréquence de votre circuit. ¹

1. Donnez TOUTES les équations logiques du compteur.

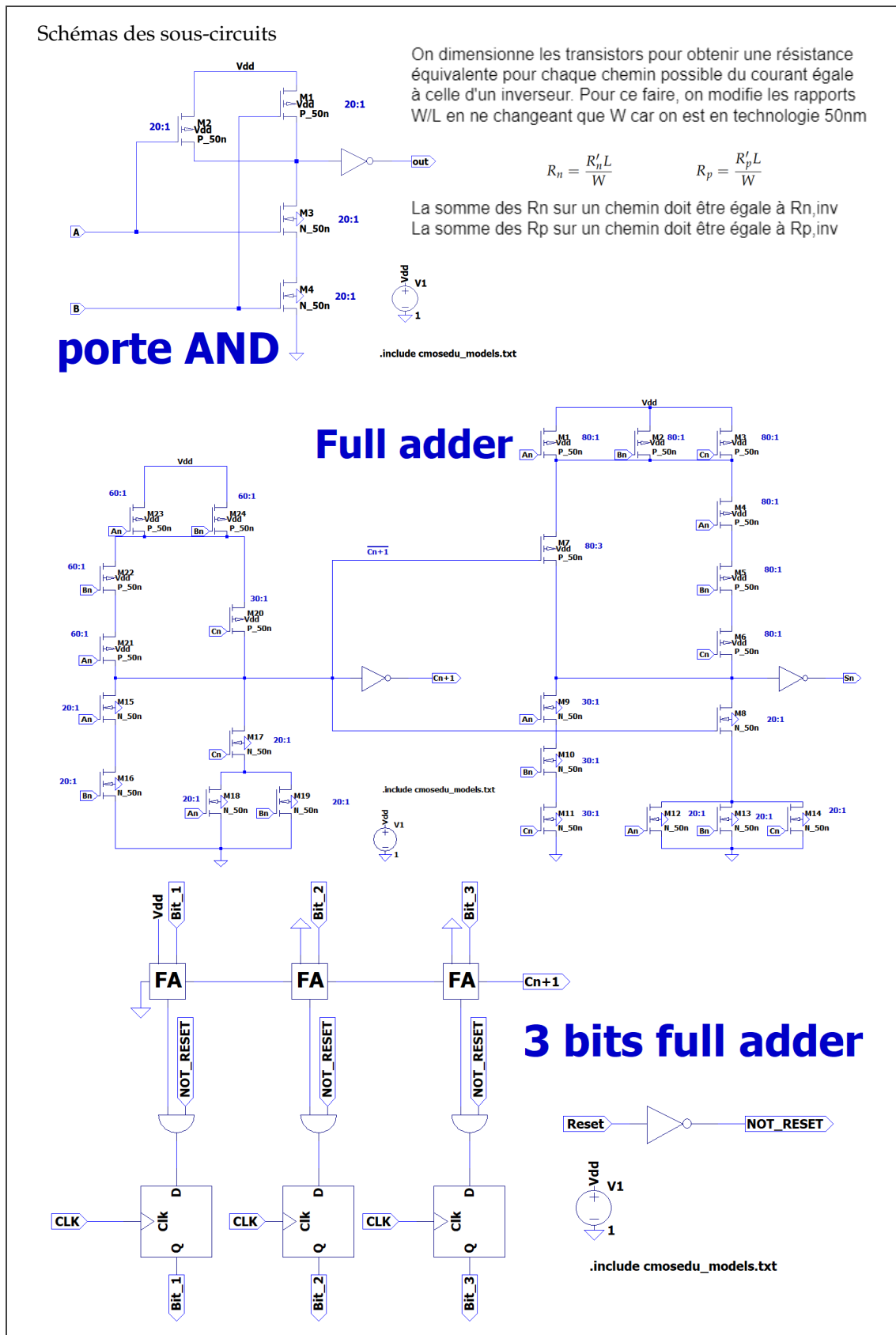
Equations logiques		
$C_{n+1} = A_n \cdot B_n + C_n(A_n + B_n)$ $S_n = \bar{A}_n \bar{B}_n C_n + \bar{A}_n B_n \bar{C}_n + A_n \bar{B}_n \bar{C}_n + A_n B_n C_n$ $Q_i = S_i \bar{R} \quad \text{où } R = \text{Reset}$		
Pour Q1 : $A_n = V_{dd} = 1$, $B_n = B_1$, $C_n = 0$	Pour Q2 : $A_n = 0$, $B_n = B_2$, $C_n = C_{n+1,1}$	Pour Q3 : $A_n = 0$, $B_n = B_3$, $C_n = C_{n+1,2}$
$S_1 = \bar{B}_1$	$S_2 = \bar{B}_2 B_1 + B_2 \bar{B}_1$	$S_3 = \bar{B}_3 B_1 B_2 + \bar{B}_1 \bar{B}_2 B_3$
$C_{n+1,1} = B_1$	$C_{n+1,2} = B_1 B_2$	$C_{n+1,3} = B_1 B_2 B_3$
$Q_1 = \bar{B}_1 \bar{R}$	$Q_2 = (\bar{B}_2 B_1 + B_2 \bar{B}_1) \bar{R}$	$Q_3 = (\bar{B}_3 B_1 B_2 + \bar{B}_1 \bar{B}_2 B_3) \bar{R}$

2. Montrez le schéma du circuit global (copie d'écran LTspice avec fond blanc).

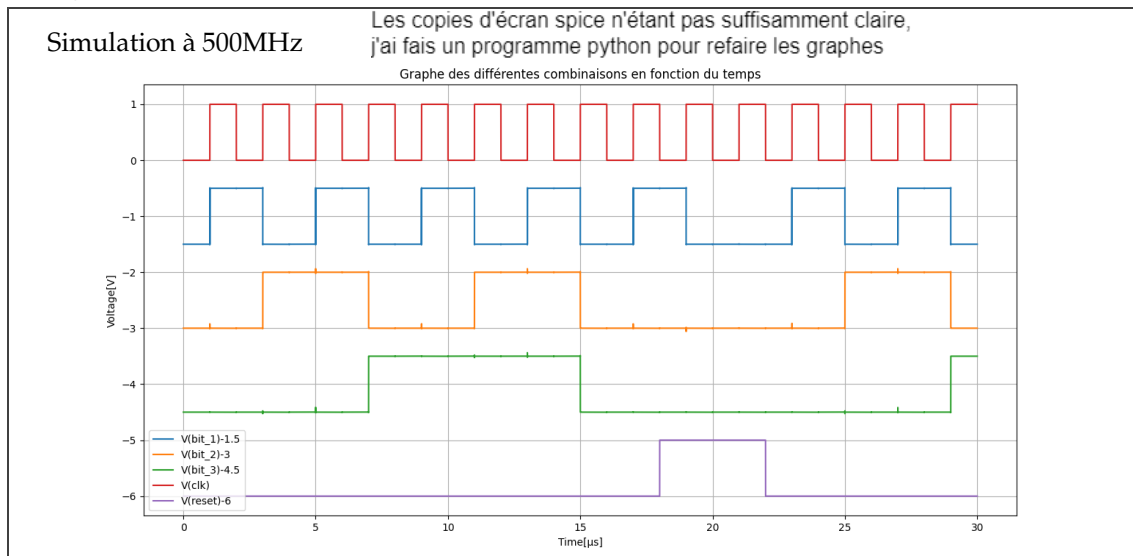


1. Le manuel LTspice et [cette vidéo](#) peuvent vous guider.

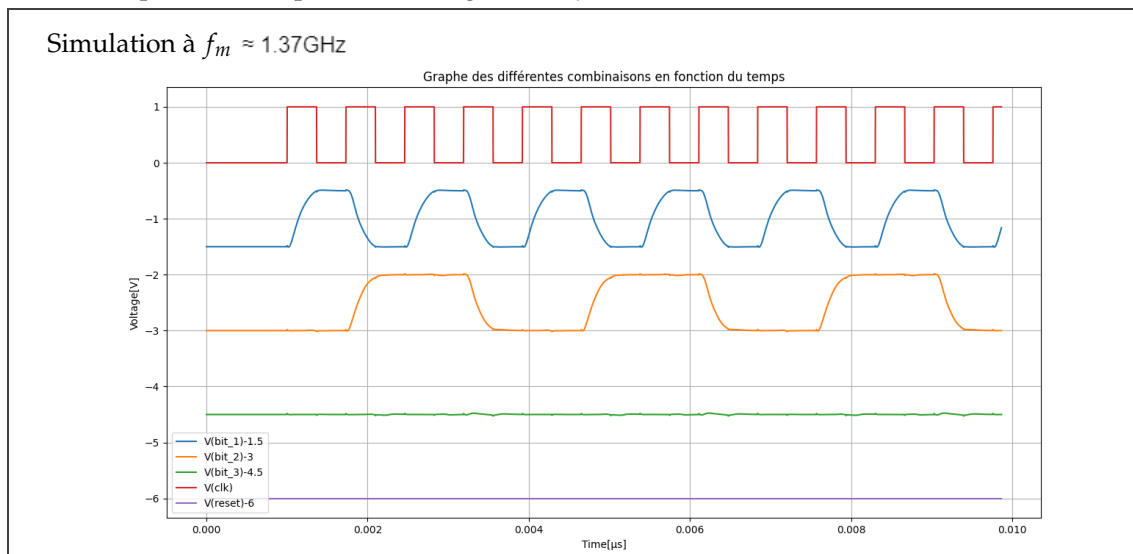
3. Montrez les schémas des sous-circuits (copies d'écrans LTspice avec fond blanc) avec les dimensionnements des transistors en expliquant brièvement votre raisonnement.



4. Affichez les résultats de la simulation qui montre toutes les fonctionnalités (copie d'écran spice avec fond blanc). Montrer tous les signaux utiles pour prouver le bon fonctionnement du circuit. Veuillez montrer une séquence complète de comptage (i.e. 0 1 ... 7 0 1). **Veillez à ne pas superposer les signaux!**



5. Identifiez la fréquence maximale de fonctionnement f_m (i.e. la fréquence de fonctionnement pour laquelle le circuit ne réalise plus correctement la fonction attendue) et affichez les résultats de la simulation pour une fréquence d'horloge valant f_m .



6. Décrivez ce que vous feriez au niveau du système de reset pour diminuer le nombre de transistors dans votre circuit.

Optimisation possible

Le circuit du reset avec la porte AND n'est pas optimal. Il comporte un grand nombre de transistor qui prend de la place. Un meilleur moyen de le faire serait de n'utiliser qu'un transistor NMOS pour décharger le noeud de la D-FF et ramener la tension à 0V.