Examen électronique logique mai 1999 **Licence EEA et LPA**

CORRECTION

Barème : il est donné en face de chaque titre d'exercice. Le total donne 26, + 2 points accordés en fonction de la présentation. La note est ensuite ramenée sur 20, arrondie au demi-point.

I - Questions de cours [3]

a) A=B=1 ; les deux diodes BE du transistor d'entrée sont en inverse ; Ii est positif (entrant), de l'ordre de 20 µA. T1 est bloqué, T2 et T3 sont passants, S=0, Io est positif, de l'ordre de plusieurs mA.
b) A=1, B=0 ; le courant Ii sur A est le même que précédemment, le courant Ii sur B est négatif (sortant) et vaut quelques mA. T3 est bloqué, S=1 (le schéma est celui d'une porte NAND), Io est négatif (sortant), de l'ordre de quelques mA.
2) Si une entrée est en l'air, la diode d'entrée correspondante est bloquée (le courant ne peut pas passer !), ce qui tend à bloquer T1, de manière équivalente à une entrée à 1.

II- Compteurs [5.5]

On construit un compteur asynchrone à partir de JK en mode 'toggle', les deux entrées J et K sont cablées à 1. La sortie de chaque bascule est reliée à l'entrée d'horloge de la bascule suivante, et ainsi de suite avec autant de bascules que de bits nécessaires pour le compteur souhaité. On appelle un tel compteur asynchrone car les sorties Q ne basculent pas simultanément. Il faut 4 bits pour compter jusqu'à 10, donc 4 bascules. Pour réaliser un compteur comptant en boucle de 0 à 15, il faut cabler 4 bascules en laissant /R et /S à 1 (le compteur compte au max possible = 15 puis revient naturellement à 0). Sur le schéma proposé, les bascules fonctionnent au front ***montant***, il faut donc relier les sorties **/Q** aux entrées d'horloges des bascules suivantes, sinon le compteur décompte.
Pour que le compteur compte de 0 à 9, il faut prévoir une remise à zéro de chaque bascule lorsque le nombre Q4Q3Q2Q1 vaut 1010 (10) ; on peut utiliser un NAND dont les 4 entrées sont Q4/Q3Q2/Q1 et dont on relie la sortie à toutes les entrées /R des bascules.
Dans un compteur asynchrone, les sorties Q1, Q2, etc; basculent à tour de rôle, et le nombre de sortie présente des états temporaires indésirables (de durée quelques nanosecondes, le temps de transfert des bascules). Ainsi, après 1 (0001), le compteur affiche fugitivement 0 (0000) lorsque Q1 bascule avant Q2, puis le compteur affiche 2 (0010), puis 3 (0011) sans transitoire, puis 2 (0010) et 0 (0000) de manière indésirable avant de passer à 4 (0100), etc. La différence essentielle dans un compteur synchrone est la simultanéité des horloges sur toutes les bascules du compteur, donc la simultanéité de basculement des sorties, donc a priori l'absence de transitoires indésirables.

III- Gestion de caméra électronique 32x32 = 1024 pixels [10]

ROM=Read Only Memory : circuit mémoire qu'on ne peut que lire. Une ROM est le nom générique de ce type de mémoire, et plus particulièrement un circuit mémoire programmé sur lequel on n'a aucune action possible à part la lecture (CDROM, programme BIOS, etc). Une PROM peut être programmée une fois par l'utilisateur et seulement lue ensuite. Une EPROM peut être effacée et reprogrammée plusieurs fois.
Cycle de lecture de caméra : T=100ns ; un pixel = 4xT = 400 ns ; temps de lecture d'une ligne = 32 x lecture pixel = 12.8µs ; temps de lecture image = 32 lignes = 1024 x 4 T = 0.4096ms. La mémoire doit contenir 1024 x 4 cases mémoires d'un octet (7 bits $\Rightarrow$ 1 octet) : capacité totale = 4Koctet. Pour repérer 4096 cases, il faut 12 bits d'adresse.

A partir de circuits 2Koctet (27C32), il faut 2 boîtiers pour contenir tout le programme de lecture nécessaire. En plaçant les deux boîtiers en parallèle sur le même bus (8 fils de data) et en contrôlant les sorties /OE par le 12^{e} bit d'adresse avec un inverseur entre /OE1 et /OE2, on passe alternativement du circuit "bas" au circuit "haut".
Les 10 premiers mots mémoire (adresses 0 à 9, le MSB D7 vaut toujours 0) sont : $41, $62, $28, $14, $40, $60, $28, $14, $40, $60. Le motif 28/14/40/60 se répète. Pour ces 10 mots, le 8^{e} bit de parité (paire) vaut : 0,1,0,0,1,0,0,0,10. Les mots s'écrivent alors : $41, $E2, $28, $14, $C0, $60, $28, $14, $C0, $60.
Le premier top de ligne intervient à l'adresse 01 ; le suivant interviendra 32 pixels plus loin, soit à l'adresse 01+32 x 4 = 129 (=$81). Sa valeur est $62, puisque le top de ligne est présent.
Le circuit de conversion de tension utilise un transistor en montage inverseur. Le transistor est bloqué ou passant, ce qui met la sortie à la tension max (Vcc) ou à 0, et la sortie reproduit l'entrée avec des niveaux de tensions différents. L'inverseur est là pour assurer que le transistor soit passant (tension de sortie à 0) quand la tension d'entrée vaut 0 (sinon D' = /D).

IV Train d'impulsion [7.5]

Dans la suite, on repère par le No 1 la porte OU en haut sur la figure 4 (sortie D), et par le No 2 la porte OU en bas sur la figure (sortie D').
1) Lorsqu'aucun poussoir n'est appuyé, A=B=1 par l'intermédiaire des résistances de 1kΩ. Ces deux résistances servent à limiter le courant lorsque le poussoir est actionné. 2) Lorsque A=B=1, si D=0, le OU 2 a une entrée à 1 (/D), qui donne D'=1, et on retrouve bien la sortie du OU du haut à 0. De même si D=1, les deux entrées du OU 2 (/B et /D) sont à 0, et on retrouve D'=0 en sortie, qui donne bien 1 en sortie du OU 1. Les deux états D=0 ou D=1 sont donc possibles (on a toujours D'=/D).
3) Si on actionne *start* uniquement, A passe à 0, ce qui force la sortie du OU 1 à 1. Si D était à 1, cela ne change rien ; si D était à 0, son passage à 1 amène deux '0' sur le OU 2, dont la sortie D' passe à 0, ce qui amène un 1 de plus à l'entrée du OU 1 (après l'inverseur) : ce '1' confirme en quelque sorte le passage de D à 1, et assure ensuite que D reste à 1 lorsque A repasse à 0 quand on relâche le poussoir. Si on appuie sur *stop*, le même raisonnement s'applique de manière symétrique, mis à part le fait que D' attaque le OU du haut à travers un inverseur. Le passage B->0 entraîne D'=1 en sortie du OU 2, donc deux '0' (/A et /1) sur le OU 1 et le passage de D à 0 (qui vient 'bloquer' le OU 2 et qui maintient le système même lorsqu'on relâche le poussoir *stop*.
4) Si on appuie d'abord sur *start*, D passe à 1. On a alors deux '0' sur le OU 2 (à travers les inverseurs). Si on appuie sur *stop* sans relâcher *start*, B passe à 0, ce qui force D' à 1, mais tant que A reste à 0, D' ne peut rien changer (c'est /D' qui attaque le OU du haut). L'effet de stop ne peut se faire sentir qu'après avoir lâché *start*.
5) L'état initial est D=0, donc Q=0 (plusieurs fronts ont eu lieu), donc S=1 (le NAND de sortie ne laisse pas "passer" l'horloge). On appuie sur start, le front descendant de A fait passer D à 1 "immédiatement" aux temps de transfert des portes près. Q recopie D au front d'horloge descendant (voir schéma) immédiatement suivant, passant alors à 1, ce qui laisse "passer" l'horloge à travers le NAND : S recopie l'horloge en l'inversant. Lorsque stop est actionné, D repasse à 0 au front descendant de B, ET Q recopie cette valeur au front d'horloge descendant suivant, S remonte alors à 1 et se bloque : même commandé par des ordres 'asynchrones' (l'action sur les poussoirs n'est pas synchronisée sur l'horloge), le montage laisse passer un nombre entier de demi périodes d'horloge.