



DIRO
IFT 1215

INTRODUCTION AUX SYSTÈMES INFORMATIQUES

DEVOIR N° 2

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle
[http : //www.iro.umontreal.ca/~mignotte/ift1215/](http://www.iro.umontreal.ca/~mignotte/ift1215/)
E-mail : mignotte@iro.umontreal.ca

I. Circuits Logiques Combinatoires

L'afficheur lumineux à 7 segments TIL 302 est un circuit intégré formé de 7 diodes lumineuses en formes de bâtonnets permettant de représenter tout chiffre de 0 à 9 (Cf. Fig. 1).

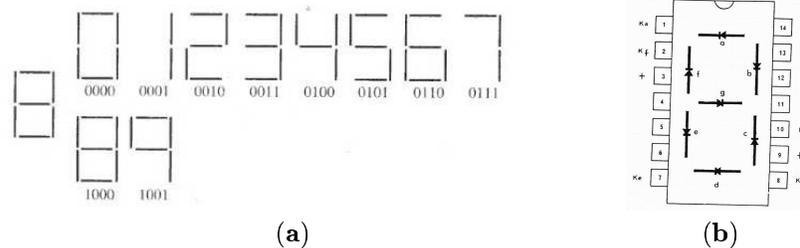


FIG. 1 – (a) Les 7 segments, (b) le circuit TIL 302.

On appelle K_a, K_b, \dots, K_g les fonctions logiques associées aux segments a, b, \dots, g . Lorsqu'on alimente le circuit TIL 302 (en mettant 5 Volts sur la patte 3, 9, 14 du circuit de la Fig. 1b) tous les segments sont allumés faisant apparaître un 8. Pour éteindre un segment, par exemple le segment "g" (pour faire apparaître un "0" sur l'afficheur 7-segment) on doit avoir $K_g = 1$ (i.e., 5 Volts sur la patte 11 du circuit TIL 302) et toute les autres fonctions à zéro ($K_a = 0, K_b = 0, K_c = 0, K_d = 0, K_e = 0, K_f = 0$).

Le but de cet exercice est de concevoir un circuit logique qui prend en entrée un nombre binaire de 4 bits (variables binaires ABCD avec A symbolisant le LSB et D symbolisant le MSB) et qui produit en sortie le signal de contrôle pour les fonctions binaires K_a, K_b, \dots, K_g , associées aux 7 segments du circuit TIL 302.

1. Faire une table indiquant suivant les lignes le chiffre affiché par l'afficheur 7-segment, et suivant les colonnes, les segments que l'on doit éteindre pour réaliser ce chiffre.

2. Utiliser cette table pour remplir les 7 tableaux de Karnaugh (en utilisant les états indifférents (i.e., “don’t care”) pour représenter les configurations d’entrées qui ne se présente pas) et trouver les 7 fonctions logiques (les plus simplifiés possibles) K_a, K_b, \dots, K_g associée aux 7 segments
 3. Dessiner le logigramme du circuit en utilisant seulement des portes NON-ET [*NAND*].
-

II. Circuits Logiques Séquentiels

On veut concevoir un circuit séquentiel réalisant la fonction majorité pour trois entrées binaires se présentant en série à l’entrée d’une FSM (machine à état finie). Le circuit doit afficher un “0” en sortie jusqu’à ce que la troisième entrée soit vue. À ce point, un “0” ou un “1” est produit à la sortie selon qu’il y a une majorité de “0” ou de “1” à l’entrée. Par exemple, une entrée de 011 100 101 doit produire une sortie de 001 000 001. Dans cet exemple le quatrième bit en sortie est bien “0” car on considère que l’on a vue qu’une entrée (i.e., un “1”) et le circuit doit afficher un “0” en sortie jusqu’à ce que la troisième entrée du deuxième triplet de bits d’entrée soit vue.

1. Faire un diagramme de transition d’états pour ce circuit.
 2. Faire la table de transition d’état, la table d’état et la table de vérité.
-

Remise & Rapport

Ce rapport devra être fait par groupe de deux étudiants. Vous devez rendre ce travail au démonstrateur le jour de la démonstration, i.e., le vendredi à 10h30, à la date de remise spécifiée dans le fichier *planning* situé sur la page web du cours dans la rubrique *Introduction & Programme*. Vous pouvez écrire votre rapport à la main ou de préférence par traitement de texte. Dans les deux cas, le rapport devra être clair, présentable et concis (il devra toutefois contenir les résultats intermédiaires nécessaires permettant de montrer sans ambiguïté que vous êtes arrivés au résultat demandé). Il y aura une pénalité de 20% par jour de retard.

¹On pourra s’inspirer du diagramme de transition d’états du détecteur de séquence vue en cours (acétate 59 du chapitre CIRCUITS LOGIQUES) dans lequel un état (l’état *A*) codait l’état initial pour lequel aucun bit en entrée n’était vue, 2 états étaient utilisés pour coder le fait que le circuit avait vu 1 bit en entrée et 4 états codaient le fait que l’on ait vue 2 bits en entrée. Dans le diagramme demandé dans cet exercice, des flèches reliront ensuite ces 4 derniers états à l’état initial *A* car le système se remettra à attendre 3 bits en entrée après avoir reçu le premier triplet de bit.