



DIRO  
IFT 1215

## DÉMONSTRATION N° 5

*Max Mignotte*

DIRO, Département d'Informatique et de Recherche Opérationnelle, local 2377.

[http : //www.iro.umontreal.ca/~mignotte/ift1215/](http://www.iro.umontreal.ca/~mignotte/ift1215/)

*E-mail : mignotte@iro.umontreal.ca*

Exercices tirés du livre de Référence

IRV ENGLANDER.

**The Architecture of Computer Hardware  
and Systems Software : An Information  
Technology Approach.**

Wiley, ISBN : 978-1-118-32263-5, 5<sup>e</sup> édition, 2014

Annexe 1	<b>Circuits séquentiels &amp; FSM</b>
Exercice .....	Compteur Code Gray modulo 8
.....	Exercice I. (cf. page suivante)
Chapitre 6	<b>LMC</b>
Exercices .....	6.6, 6.10, 6.11, 6.13, 6.15
.....	+ Exercice II. (cf. page suivante)

## I. Gray Code Counter

Let us use the FSM design procedure to design a synchronous 3-bit Gray code counter using positive edge-triggered D flip-flops. A gray code is a code in which consecutive numbers differ in only one bit. Thus, a 3-bit Gray code consists of the following sequence : 000, 001, 011, 010, 110, 111, 101, 100. The resulting state diagram is shown in Figure 1a. The set of eight states for this state diagram can be encoded simply by using the Gray code output values as the state variables. The resulting state transition table with this encoding is shown in Figure 1b.

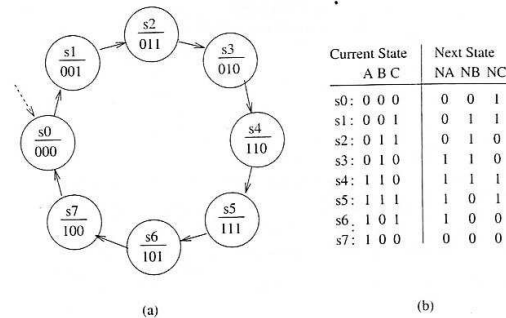


FIGURE 1 – The (a) state diagram and (b) state transition table for a 3-bit Gray code counter.

Find the logic diagram for a 3-bit Gray code counter.

## II. Assembleur/LMC

- Calculer la note finale ( $N$ ) d'un cours à partir de la note de l'examen intra ( $I$ ), final ( $F$ ) et travaux ( $T$ ). Le final, l'intra et les travaux sont entre 0 et 10. La pondération de  $F$  est 40,  $I$  et  $T$  valent 30. Toutefois, le seuil ( $S$ ), cad, la moyenne pondérée de  $I$  et  $F$  doivent être  $\geq 5$ . Sinon, alors  $T$  est remis à la valeur 5, ce qui entraîne l'échec ( $N < 5$ ). Donc le calcul est,

$$S = (F * 0.4 + I * 0.3) / 0.7$$

$$\text{si } S < 5 \text{ alors } T = 5$$

$$N = F * 0.4 + I * 0.3 + T * 0.3$$

LDA	5xx	Load
STO	3xx	Store
ADD	1xx	Add
SUB	2xx	Subtract
IN	901	Input
OUT	902	Output
COB or HLT	000	Coffee break (or Halt)
BRZ	7xx	Branch if zero
BRP	8xx	Branch if positive or zero
BR	6xx	Branch unconditional
DAT		Data storage location

On utilisera le jeux d'instruction présenté dans le chapitre LITTLE MAN COMPUTER (cf. Fig.) et auquel on ajoutera l' instruction suivante :

- MUL 4xx** M(xx) (i.e., le contenu de l'adresse xx en mémoire) est multiplié avec le contenu de l'Accumulateur
- DIV 7xx** Le contenu de l'accumulateur est divisé par M(xx) ; l'Accumulateur reçoit la partie entière du résultat.