

# Examen de Conception de Circuits

E. Mesnard  
24 mai 2016

Documents autorisés : feuille A4 manuscrite Recto/Verso  
Durée : 2 heures

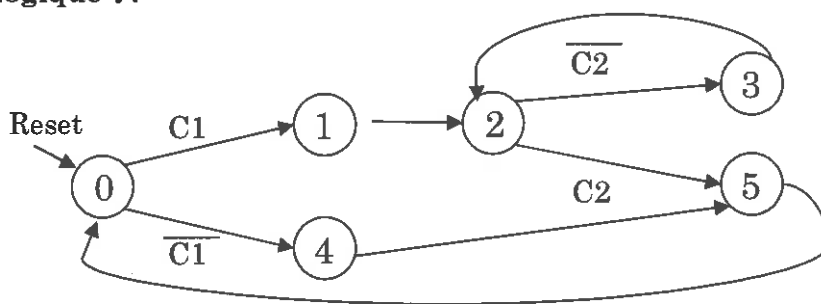
## Exercice 1 (5 points) Questions de cours, indépendantes

- 1) Donner le nom d'une synthèse (de circuits !) permettant facilement d'obtenir un schéma logique à partir d'un **tableau de Karnaugh** (ce tableau lie les entrées à la sortie du circuit considéré).
- 2) Expliquer pourquoi la synthèse **séquentielle à jeton** porte ce nom.
- 3) Représenter les deux circuits a) et b) ci-dessous. Donner uniquement la vue externe du schéma, en précisant toutefois l'utilité de tous les signaux.
  - 3-a) circuit nommé « **démultiplexeur 4 vers 1** ».
  - 3-b) circuit permettant tout d'abord de **charger** un nombre binaire sur 16 bits, puis ensuite, permettant de le **multiplier par 2**.

## Exercice 2 (6 points) Synthèse séquentielle micro-programmée

Faire la synthèse micro-programmée de l'automate dont les logiques de transition  $\tau$  et de sortie  $\sigma$  sont les suivantes :

Logique  $\tau$  :



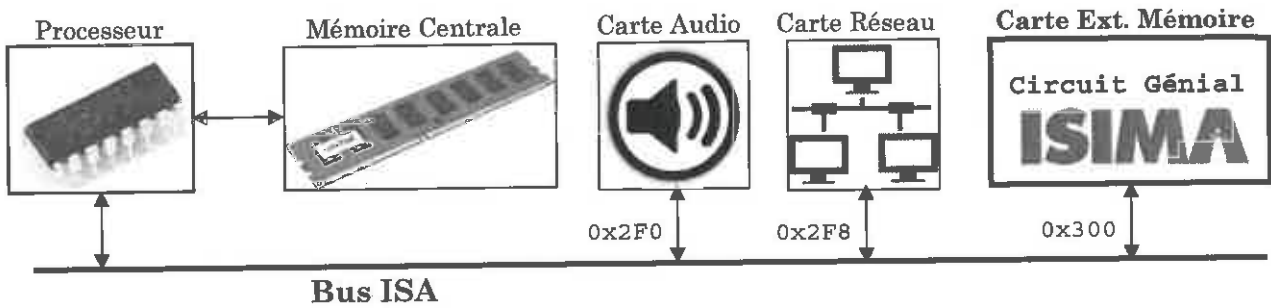
Logique  $\sigma$  :

Etat	Sortie
0	S1
1	S1, S2
2	S3
3	S4
4	S2
5	-

## Problème (9 points) Synthèse UC/UT et synthèse à jeton d'une carte PC

L'objectif de ce problème est de concevoir en synthèse UC/UT une carte électronique numérique d'**Extension Mémoire** pour ordinateur de type PC. Le rôle de cette carte est de mémoriser un unique octet (c'est le principe qui compte !). Le PC doit donc pouvoir fournir un octet à la carte d'extension, puis quelques temps plus tard, retrouver ce nombre en interrogeant la carte.

Physiquement, la carte est connectée sur le bus « ISA » du PC (bus « Industry Standard Architecture », célèbre dans les années 1980/1990, et ancêtre du bus PCI...), à côté des cartes audio et de communication réseau. Son adresse (de « port ») est fixée à 0x300.

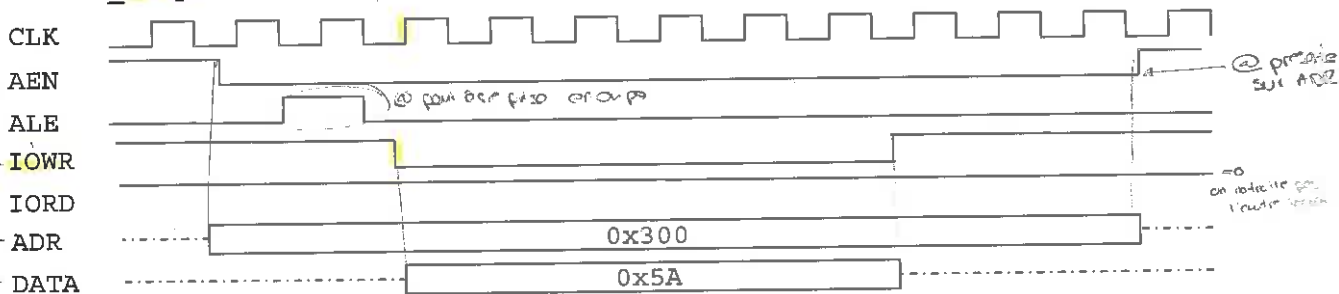


Pour exploiter la carte mémoire externe, l'utilisateur dispose alors de deux instructions C :

- écriture sur un port ISA : « `_outp(0x300, mon_octet)` ». Cette instruction permet donc de mettre en mémoire un octet, car l'adresse cible est bien celle de la carte (0x300).
- lecture d'un port ISA : « `octet_lu = _inp(0x300)` ». Cette instruction permet donc de retrouver l'octet précédemment mémorisé, si l'adresse est celle de la carte.

Les signaux présent sur le bus ISA et qui intéressent la carte sont **CLK** (l'horloge), **ADR** (le bus d'adresse sur 12 bits), **AEN** (« Adress Enable » qui vaut 0 lorsqu'une adresse est présente sur ADR), **ALE** (« Adress Latch », qui signale que l'adresse peut être prise en compte), **IOWR** et **IORD** (« Input/Output WRite or ReaD », signaux de contrôle, actifs au niveau bas, d'écriture ou de lecture des données présentes sur DATA : IOWR pour l'instruction `_outp`, et IORD pour `_inp`), **DATA** (le bus de données bi-directionnel sur 8 bits, avec valeurs maintenues durant le signal de contrôle).

Le chronogramme du bus ISA ci-dessous est un exemple d'écriture du PC vers la carte, avec l'instruction « `_outp(0x300, 0x5A)` » demandant la mémorisation du nombre 0x5A.



L'algorithme (très simplifié !) qu'il faut implémenter dans cette carte est donc le suivant :

**Tantque vrai Faire**

```

Attendre Carte Selectionnee // AEN vaut 0, ALE vaut 1 et ADR est la bonne.
Attendre 1 cycle CLK // Par securite, pour etre sur d'avoir un IOWR ou IORD
Si (IOWR = 0) Alors // le PC ecrit sur son bus en direction de la carte, donc
    phase ecriture // la donnee presente peut etre utilisee durant IOWR
Sinon Si (IORD = 0) Alors // le PC veut lire son bus ISA, donc la carte doit
    phase lecture // fournir le nombre sur le bus durant toute cette phase
Sinon ne rien faire FinSi // cas anormal provenant d'un parasite

```

**FinTantque**

Utiliser la synthèse à jeton pour l'UC, et noter que le générateur de jeton est disponible dans la bibliothèque de composants.