

Examen de Conception de Circuits

E. Mesnard
26 avril 2012

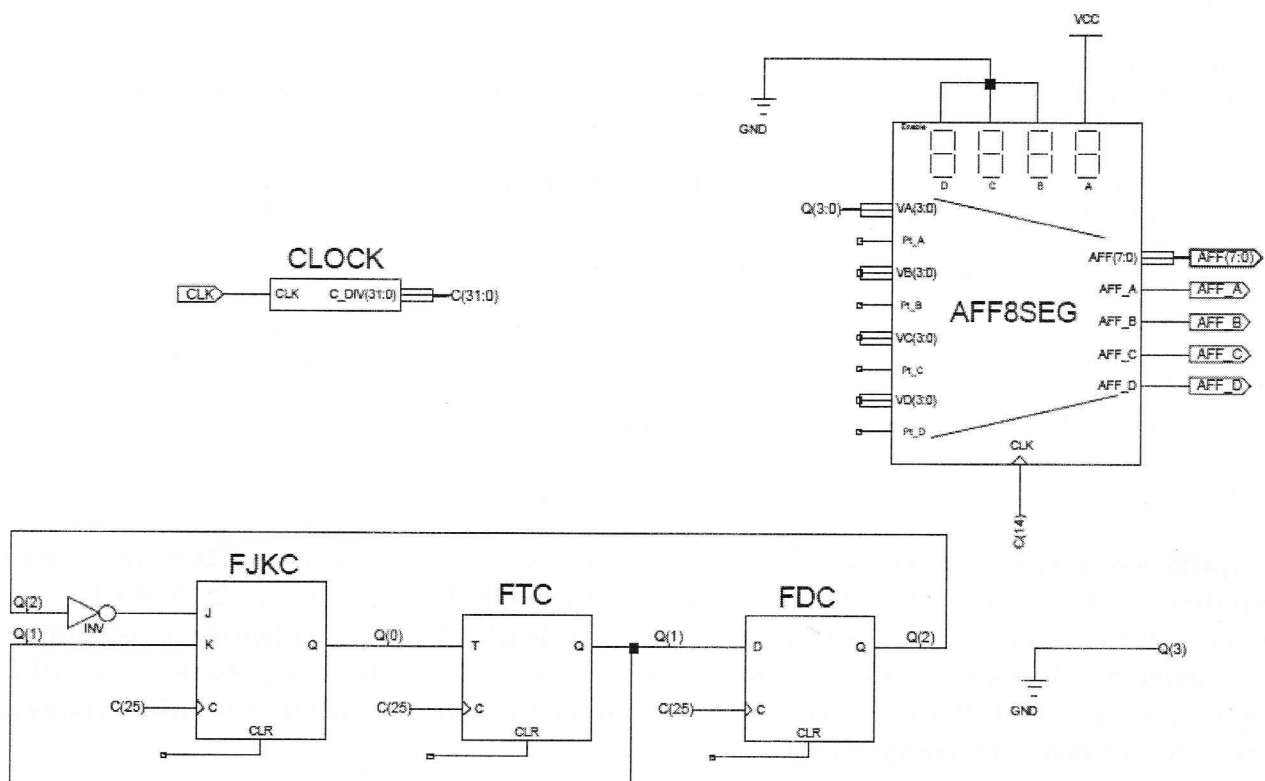
Documents autorisés : feuille A4 manuscrite Recto/Verso

Durée : 2 heures

Exercice 1 (4 points) Analyse d'un circuit séquentiel

Soit le circuit séquentiel suivant, constitué principalement de 3 bascules (voir schéma ci-dessous) :
 une bascule FJK d'entrées respectives $\text{NOT}(Q(2))$ et $Q(1)$, et de sortie $Q(0)$,
 une bascule FT d'entrée $Q(0)$ et de sortie $Q(1)$,
 une bascule FD d'entrée $Q(1)$ et de sortie $Q(2)$.

Indiquer alors ce que l'afficheur 8 segments, qui exploite en entrée ces valeurs $Q(2:0)$, va afficher lorsque le circuit sera en fonctionnement. **Justifier** en détaillant toute l'analyse.

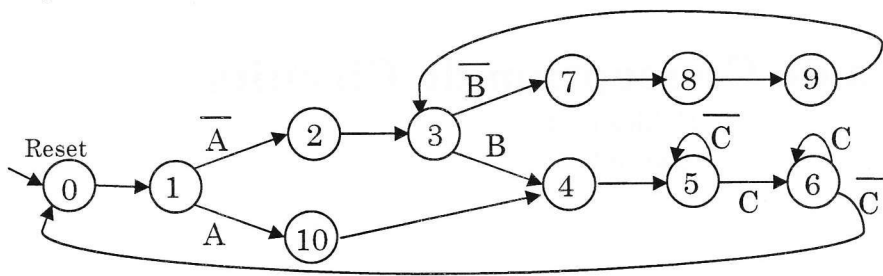


Noter que $C(25)$ est un signal d'horloge qui présente un front montant environ toutes les secondes.

Exercice 2 (5 points)

Synthèse d'automate

Soit le circuit dont le fonctionnement est décrit par le diagramme d'états (logique de transition τ) et par le tableau de sortie (logique de sortie σ) fournis ci-dessous.



Etat	Sortie
0,1,4,9	S1
2,4,5,7	S2
1,2,8	S3

La méthode de synthèse imposée est la **synthèse par compteur chargeable**.

- 1) Donner tout d'abord le « pseudo-programme » décrivant cette logique de transition.
- 2) Dessiner le schéma combinatoire destiné à être relié à l'entrée CE du compteur.
- 3) Dessiner le schéma combinatoire destiné à être relié à l'entrée L du compteur.

$$CE = NOP = E5.C + E6.C$$
$$L = SAUT = E1.A + E3.B + E6.C + E9 + E10$$

Problème (11 points)

Réalisation d'un multiplieur 4 bits

Le but de ce problème est de concevoir **intégralement** (jusqu'au dessin de l'ensemble des schémas logiques) un circuit qui réalise la multiplication de 2 nombres binaires sur 4 bits – A(3:0) et B(3:0) – et qui affiche les 8 bits du résultat R(7:0) sur des LEDs. L'algorithme qu'il faut implémenter est le suivant (écrit ici en langage C) :

```
void main(void) {
    unsigned char A,B,R;           // Entiers non signés d'au maximum 8 bits
    while(1) {                     // Traitement en boucle infinie
        Lire_Switch(&A,&B);        // Acquisition des nombres A et B
        R = 0;                      // Initialisation du résultat R
        while (A != 0) {
            if ((A & 0x01) != 0) {
                R += B;           // Accumulation de B sur R
            }
            B = B << 1;          // Décalage de B à gauche (pour faire B*2)
            A = A >> 1;          // Décalage de A à droite (pour faire A div 2)
        }
        printf("\nValeur du produit : %d",R);
    }
}
```

L'interface avec l'utilisateur doit être très minimaliste : un unique bouton BTN pour valider et acquitter, les 8 switches SW(7:0) pour la saisie **simultanée** des valeurs A et B (A sur les poids faibles et B sur les poids forts des switches), et les 8 LEDs LED(7:0) pour l'affichage du résultat.

A noter que la bibliothèque de composants possède, entres autres, un additionneur ADD8. D'autre part, aucune méthode de synthèse n'est imposée ici. Enfin, **détailler et justifier** toutes les étapes des méthodes de conception choisies.