Tronc commun 3A - Durée : 90 mn **Examen du** **Cours EE341** **NOM :**	1er session Année 2011 – 2012 Un seul document autorisé : « VHDL résumé de syntaxe » Calculatrice interdite **PRENOM :**
1 Les points donnés dans l'énoncé entre crochets [X] après chaque question représentent le barème et indiquent le temps à passer en minutes sur chaque question *2 La qualité de la rédaction (lisibilité, orthographe) sera prise en compte*	

I Partie cours [30/90]

1.0 *Dans toute la suite*, les entrées et les sorties des entités seront toujours de type *std_logic* (ou *std_logic_vector*). Quelles déclarations (donner le code VHDL) permet d'utiliser ce type. [2]

Conseil : dans la suite il ne sera plus nécessaire de rappeler cette déclaration

1.1 En utilisant uniquement des *std_logic* (et des *std_logic_vector)*, écrire la déclaration de l'entité *memory* d'une mémoire dont les ports d'entrée et de sortie sont : [4]

addr: *12 bit address input*
wr_n: *1-bit write-enable control signal*
oe_n: *l-bit output-enable control signal*
data: *8 bit bidirectional data bus*

1.2.1 Soit *a* et *y* deux vecteurs de type *unsigned* ; expliquer à quelle opération arithmétique l'expression suivante peut être assimilée. [4]

y <= "000" & a(7 downto 3);

1.2.2 En utilisant l'instruction *&* précédente, donner entité et architecture VHDL d'un *compteur de Johnson* modulo 16 [14].

Conseils :

- Utiliser une entrée reset synchrone (actif au niveau haut)

- Eviter l'utilisation de ports en mode inout

1.3 A quel composant correspond le code ci-dessous suivant que b et q sont de simples std_logic ou des vecteurs de std_logic de tailles n. Expliquer. [6]

```
process ( a , b)
begin
        if a='1' then
                q <= b ;
        end if ;
end process;
```

II Partie exercices : [60+2/90]

Les exercices 0, 1, 2 et 3 sont indépendants.

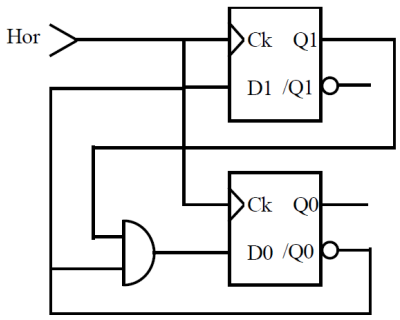
2.0 Compléter en respectant la contrainte suivante l'entité et l'architecture du décodeur combinatoire donné en annexe 1 [10] : Utiliser un signal d'entrée enable *en* pour que quand *en* vaut '1' le décodeur fonctionne normalement et quand *en* vaut '0' la sortie du décodeur reste à « 0000 ».

Conseil : la sortie du décodeur n'est pas *x* mais par exemple *x_final*

2.1 Donner la fonction booléenne du signal *even* en fonction de *a(0), a(1) et a(2)* décrite ci-dessous [4]? Expliquer son intérêt [2] Quel est le délai maximum entre les entrées et la sortie even ? [2]

```
    signal p1, p2, p3, p4 : std_logic;
begin
    even <= (p1 or p2) or (p3 or p4) after 20 ns;
    p1 <= (not a(2)) and (not a(1)) and (not a(0)) after 15 ns;
    p2 <= (not a(2)) and a(1) and a(0) after 12 ns;
    p3 <= a(2) and (not a(1)) and a(0) after 12 ns;
    p4 <= a(2) and a(1) and (not a(0)) after 12 ns;
end sop_arch ;
```

2.2.1 Décrire en VHDL RTL (« synthétisable et sans instanciation de composants ») l'entité et l'architecture correspondant au schéma ci-dessous. [10]



2.2.2 Optionnel (à faire à la fin) : En admettant que les 2 bascules sont initialement à '0', faire un chronogramme faisant apparaître l'horloge et les deux sorties Q0 et Q1. [+2]

Conseil : Pour vous aider à construire ce chronogramme faite aussi apparaître les deux entrées D0 D1 des deux bascules D.

2.3 Compteurs modulo m [32]

Attention aux types des signaux utilisés dans les opérations arithmétiques : l'annexe 2 rappelle les fonctions de conversion du paquetage numeric_std.

2.3.1 Décrire en VHDL l'entité *mod10* et l'architecture *rtl* d'un compteur synthétisable modulo 10 comptant de 0 à 9 indéfiniment (le compteur compte de 0 à 9 puis se répète indéfiniment). [16]

Contrainte : utiliser un signal reset *rst* asynchrone pour initialiser le compteur cpt à 0.

2.3.2 Combien de bascules seront générées lors de la synthèse de votre compteur mod-10? [4] Combien théoriquement au minimum en faut-il ? Expliquer l'écart éventuel. [2]

2.3.3 Modifier votre code précédent de manière à obtenir un compteur modulo m programmable (mod-m). Pour cela, ajouter une nouvelle entrée *m* à l'entité du compteur précédent et modifier le code de l'architecture [8].

Contrainte : *m* est compris entre 1 et 100

2.3.4 Combien de bascules seront générées lors de la synthèse de votre compteur mod-m ? [4] Combien théoriquement au minimum en faut-il ? [2]

Annexe 1

```
-- à completer [2]


entity decoder4 is
   port(
-- à completer [2]









   );
end decoder4;


architecture sel_arch of decoder4 is
-- à completer [2]




begin
-- à completer [4]
```

```
- à ne pas modifier :
   with s select
         x <=     "0001" when "00",
                  "0010" when "01",
                  "0100" when "10",
                  "1000" when others;

end sel_arch;
```

Annexe 2

Rappel des fonctions de conversion du paquetage numeric_std.