



Processeurs et Architectures Numériques
Contrôle de connaissance des étudiants de 1^{re} année

Jeudi, 7 juin 2012

Sans calculatrice.

Document autorisé : une feuille recto-verso

Durée: 45 mn

Partie théorique

N'oubliez pas d'inscrire nom, prénom, et numéro de casier sur votre copie et sur les éventuels schémas à rendre.

Bon courage!

1 Arithmétique

Question 1 Donner la représentation en binaire naturel des nombres suivants : 256_{10} , 1025_{10} et 64.4_{10} .

Question 2 Dire quel est l'intervalle des entiers représentables en binaire naturel sur n bits.

Question 3 Donner l'équation permettant d'établir la représentation en complément à 2 (CA2) d'un nombre négatif N , à partir de la représentation en binaire naturel de son opposé.

Question 4 Donner la représentation en CA2 des nombres suivants : 1101_{10} , -1_{10} et -7_{10} .

Question 5 Donner la représentation en CA2 sur 6 bits de 7_{10} et -7_{10} .

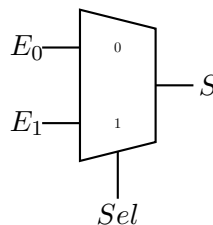
Question 6 Dire quel est l'intervalle des entiers représentables en CA2 sur n bits.

Question 7 Dire sur combien de bits doit être codée la somme de deux nombres non signés codés sur n bits.

Question 8 Dire sur combien de bits doit être codée la somme de deux nombres en CA2 codés sur n bits.

2 Logique combinatoire

Soit la porte logique complexe ci-dessous :



Question 9 Donner son nom et expliquer la fonction qu'elle réalise.

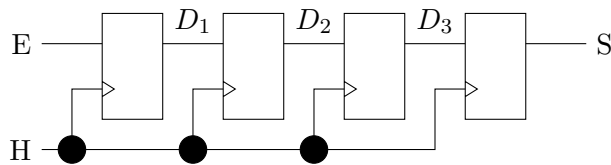
Question 10 Proposer un schéma équivalent utilisant des portes logiques élémentaires, puis donner l'équation logique de la sortie S en fonction des entrées.

3 Logique synchrone

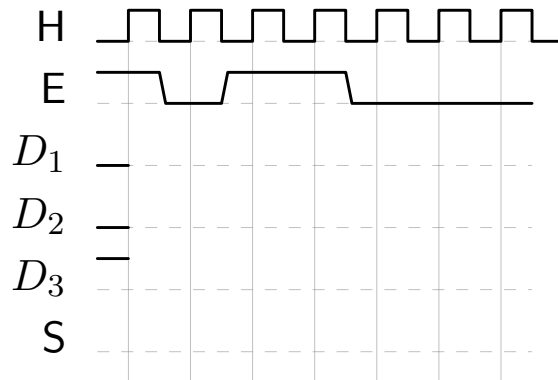
Question 11 Pour une bascule D, donner la définition :

- du temps de pré-positionnement t_{setup} ;
- du temps de maintien t_{hold} ;
- du temps de propagation t_{co} .

Soit la structure à base de bascules D ci-dessous :



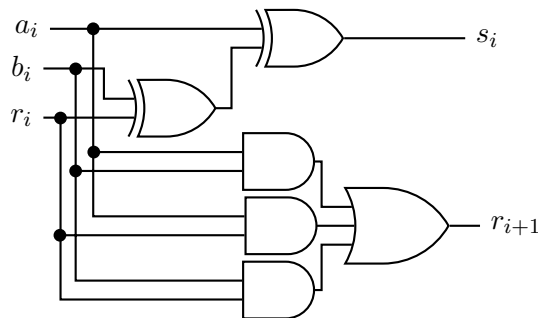
Question 12 Compléter les chronogrammes ci-dessous.



Question 13 En déduire la fonctionnalité réalisée par cette structure.

4 Temps de propagation

La structure suivante est un additionneur 1 bit à propagation de retenue. Les temps de propagation sont de 1 ns pour les portes AND et OR, et de 2 ns pour la porte XOR.



Question 14 Calculer la valeur numérique des temps de propagation des entrées vers les sorties s_i et r_i , respectivement Δs_i et Δr_i .

Question 15 Proposer une solution permettant de réaliser un additionneur 4 bits à partir de l'additionneur 1 bit précédent.

Question 16 En justifiant votre réponse, donner le temps de calcul T_{ADD4} de l'additionneur proposé.

Question 17 En déduire la formule générale permettant de calculer le temps de calcul T_{ADDn} d'un additionneur n bits.

On considère maintenant que les opérandes et le résultat fourni par cet additionneur sont stockés dans des registres à bascules D, de temps de pré-positionnement t_{setup} , de temps de maintien t_{hold} et de temps de propagation t_{co} .

Question 18 Exprimer la fréquence de fonctionnement maximale F_{MAX} de la structure complète.

5 Problème : contrôleur de bus

Soit la description comportementale suivante de l'automate fini d'un contrôleur de bus :

```
module automate_busCtrl (input logic RST, CLK,
                        input logic TOK, REQ, ACK,
                        output logic PSS, GNT);

enum logic[1:0] {ATTENTE, ATTENTE_DE_TRANSACTION, OCCUPE, PASSAGE_DU_JETON} ETAT_COURANT, ETAT_FUTUR;

// Mémorisation de l'état courant
always @(posedge CLK or negedge RST)
  if (~RST) ETAT_COURANT <= ATTENTE;
  else      ETAT_COURANT <= ETAT_FUTUR;

// Calcul de l'état suivant en fonction de l'état courant et des entrées
always@ ( * )
  case(ETAT_COURANT)
    ATTENTE:
      if (TOK)
        if (REQ) ETAT_FUTUR <= ATTENTE_DE_TRANSACTION;
        else    ETAT_FUTUR <= PASSAGE_DU_JETON;
      else      ETAT_FUTUR <= ETAT_COURANT;
    ATTENTE_DE_TRANSACTION:
      if (ACK)
        if (REQ) ETAT_FUTUR <= OCCUPE;
        else    ETAT_FUTUR <= PASSAGE_DU_JETON;
      else      ETAT_FUTUR <= ETAT_COURANT;
    OCCUPE:
      if (~REQ) ETAT_FUTUR <= PASSAGE_DU_JETON;
      else      ETAT_FUTUR <= ETAT_COURANT;
    default:
      ETAT_FUTUR <= ATTENTE;
  endcase

// Calcul des sorties en fonction de l'état courant
always@ ( * )
  case(ETAT_COURANT)
    ATTENTE:           {PSS,GNT} <= {1'b0,1'b0};
    ATTENTE_DE_TRANSACTION: {PSS,GNT} <= {1'b0,1'b1};
    OCCUPE:            {PSS,GNT} <= {1'b0,1'b0};
    PASSAGE_DU_JETON:  {PSS,GNT} <= {1'b1,1'b0};
    default:           {PSS,GNT} <= {1'b0,1'b0};
  endcase
```

Question 19 Etablir le diagramme d'état de l'automate.

Question 20 Expliquer la différence entre un automate de Moore et un automate de Mealy. Dire à quel type appartient l'automate précédent.