

Master ACSI M2

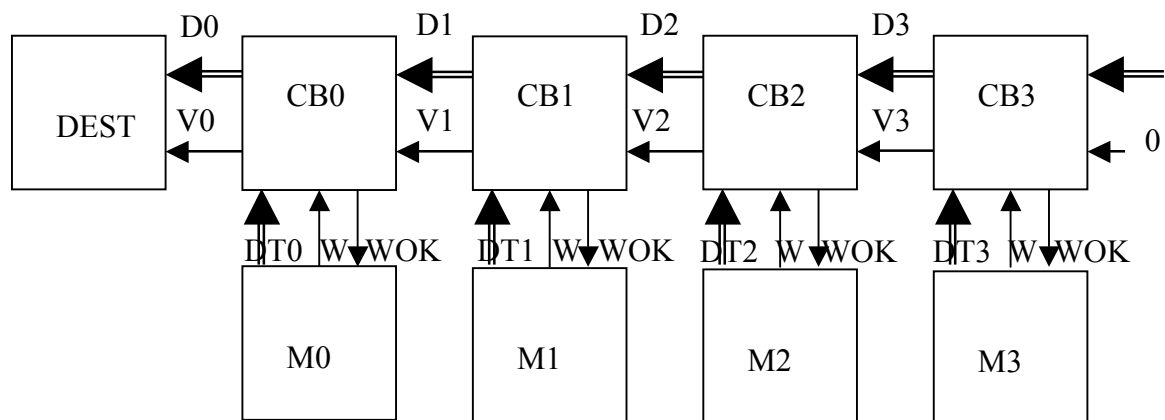
EXAMEN METHODOLOGIE

mars 2007

A. GREINER

Une "daisy-chain" est un type de bus permettant à plusieurs maîtres M_n , qui fonctionnent en parallèle et de façon logiquement asynchrone les uns par rapport aux autres, d'envoyer des données à un même destinataire DEST. Cette "daisy-chain" est constituée d'un nombre quelconque de modules contrôleurs de bus CB_n branchés en série suivant le principe ci-dessous. C'est ce module contrôleur de bus qui nous intéresse.

Il existe plusieurs protocoles possibles pour ce type de bus. Dans le schéma ci-dessous, un mot de 32 bits peut être transféré - à chaque cycle - entre un maître M_n et le module DEST. Les maîtres les plus éloignés de la cible sont les plus prioritaires, c'est à dire que pour un contrôleur CB_n , la donnée D_{n+1} en provenance du contrôleur CB_{n+1} est prioritaire par rapport à la donnée D_n en provenance du maître M_n . Dans le cas où il y a une donnée valide en provenance de CB_{n+1} , le contrôleur CB_n se contente de transmettre au contrôleur CB_{n-1} la donnée qu'il reçoit du contrôleur CB_{n+1} . Le signal V_n est à 1 lorsque le contrôleur CB_n souhaite transmettre une donnée valide au contrôleur CB_{n-1} .



Les données étant des mots de 32 bits, chaque contrôleur de bus CB_n contient deux registres de 32 bits, BUF1 et BUF2, qui permettent au contrôleur CB_n de stocker deux mots de données en provenance du maître M_n . Tous les composants du système sont synchronisés par un même signal d'horloge CK, et l'écriture dans les registres est déclenchée par le front montant de CK.

La communication entre le maître M_n et le contrôleur de bus CB_n utilise un mécanisme de contrôle de flux de type FIFO, qui permet au contrôleur de bus CB_n de refuser d'accepter des données lorsque ses registres BUF1 et BUF0 sont pleins :

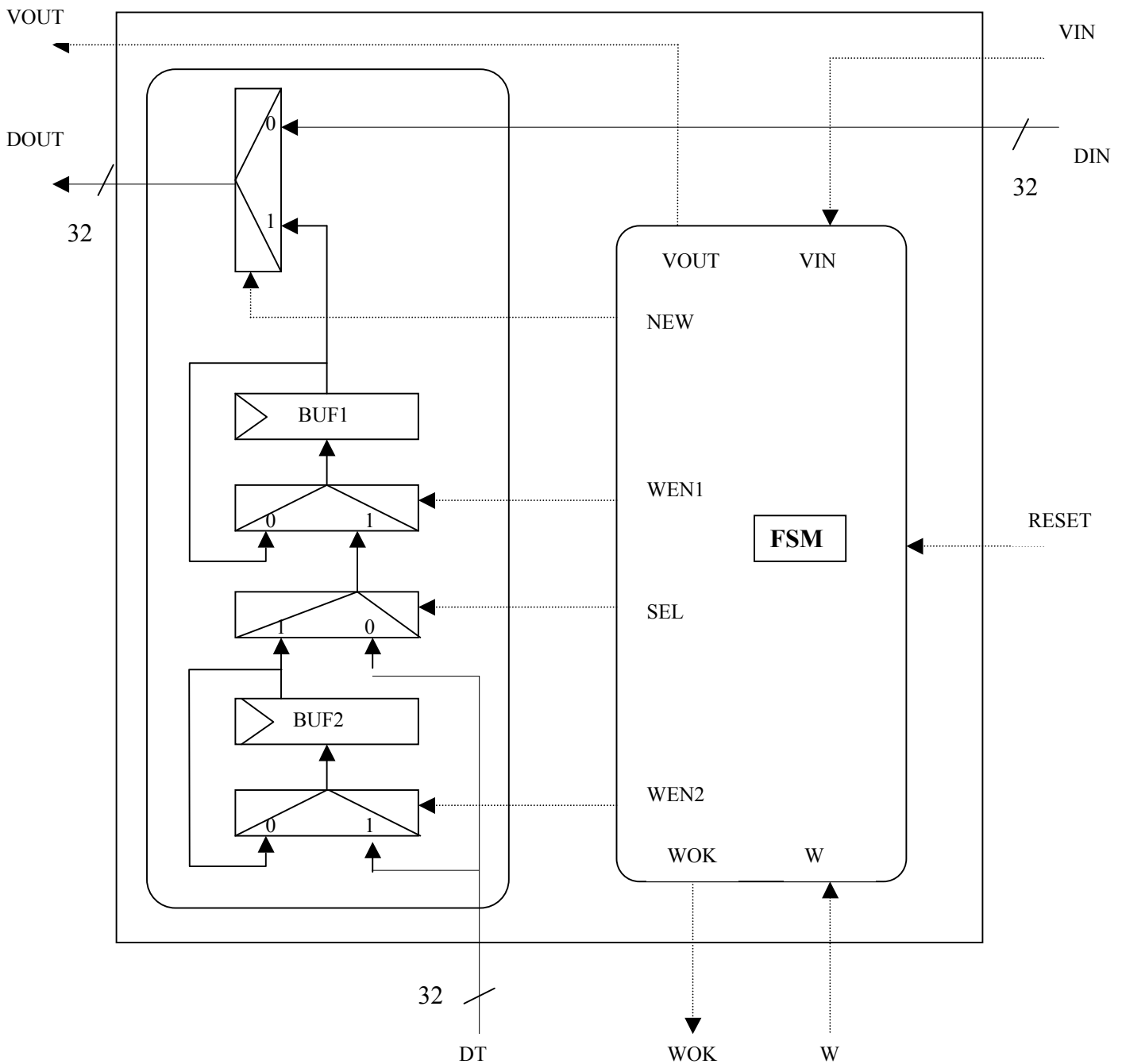
- A chaque cycle, le maître M_n doit indiquer s'il souhaite transmettre une donnée DT en donnant la valeur 1 au signal W.
- A chaque cycle, le contrôleur CB_n doit indiquer s'il est prêt à accepter une donnée DT en donnant la valeur 1 au signal WOK.

Pour qu'une donnée soit transmise durant le cycle i entre le maître et le contrôleur de bus, il faut donc que les deux signaux W et WOK soient tous les deux à 1. Dans le schéma de contrôleur de bus proposé ci-dessous, une donnée est toujours écrite dans un des deux registres BUF1 ou BUF2 avant d'être émise sur le bus. Par ailleurs le signal WOK ne dépend que de l'état interne du contrôleur et pas du signal VIN : Cela signifie qu'une requête d'écriture W n'est pas prise en compte lorsque les deux registres BUF1 et BUF2 sont pleins.

L'automate d'états qui contrôle le fonctionnement d'un module CB comporte trois états :

- Dans l'état "VIDE", les deux registres BUF1 et BUF2 sont vides,
- Dans l'état "DEMI", le registre BUF1 est plein et le registre BUF2 est vide,
- Dans l'état "PLEIN", les deux registres BUF1 et BUF2 sont pleins.

Le signal RESET force l'automate dans l'état VIDE.



Architecture interne du contrôleur de bus CB
(les signaux en pointillé sont les signaux de contrôle sur 1 bit)

L'état de l'automate est codé sur les deux bits S1 et S0 :

| Etat | S1 | S0 |
|-------|----|----|
| VIDE | 0 | 0 |
| DEMI | 0 | 1 |
| PLEIN | 1 | 0 |

Les signaux NS1 et NS0 sont les signaux à l'entrée des bascules mémorisant S1 et S0 et représentent donc l'état suivant.

Q1) (3 points) Compléter sur la feuille ci-jointe le graphe de l'automate de contrôle en attachant à chaque transition l'expression Booléenne (dépendant des entrées VIN, W et RESET) qui valide cette transition.

Q2) (4 points) L'automate génère six signaux de contrôle. Attention : Cet automate n'est pas un automate de Moore, car les signaux de sortie WEN1, WEN2, SEL, VOUT et NEW dépendent directement de l'entrée VIN. (alors que le signal WOK ne dépend que de l'état de l'automate).

En analysant le schéma ci-dessous, expliquer la fonction des signaux WEN1, WEN2, NEW et SEL. Quand le signal VOUT vaut-il 1 ? Quand le signal NEW vaut-il 1 ?

Compléter la table de vérité sur la feuille ci-jointe (pour simplifier, on a délibérément omis le signal RESET qui n'intervient que sur le calcul de l'état suivant).

Pour simplifier les expressions, on considérera que le code 11 pour le registre d'état (qui est inutilisé) est équivalent au code 10 (les 4 dernières lignes de la table de vérité sont donc identiques aux quatre précédentes). Par ailleurs, on cherche à éviter les écritures inutiles dans les registres pour minimiser la consommation. On choisira donc la valeur 0 pour les signaux WEN1 et WEN2 quand cette valeur est indifférente.

Q3) (4 points) Dédurre de cette table de vérité des expressions Booléennes simplifiées pour les signaux VOUT, WEN1, WEN2, SEL, NEW, WOK, NS1, NS0, et proposer un schéma en portes pour les fonctions de transition et de génération de l'automate de contrôle.

Q4) (3 points) Ecrire en langage VHDL le comportement de l'automate qui contrôle le bloc CB. On utilisera le style FSM.

Q5) (3 points) Ecrire en langage VHDL le comportement du chemin de données du bloc CB. On utilisera le style VBE.

Q6) (3 points) On peut imaginer de remplacer un des contrôleurs de bus CB(i) décrit ci-dessus par un contrôleur de bus "simplifié" CBX(i) ne contenant qu'un seul registre BUF au lieu de deux, mais respectant les mêmes protocoles de communication vis à vis du maître M, et vis à vis des autres contrôleurs de bus CB(i-1) et CB(i+1). On économise ainsi un peu de matériel, et on simplifie l'automate (il n'y a que deux états : VIDE et PLEIN) ... Que perd-on?

