

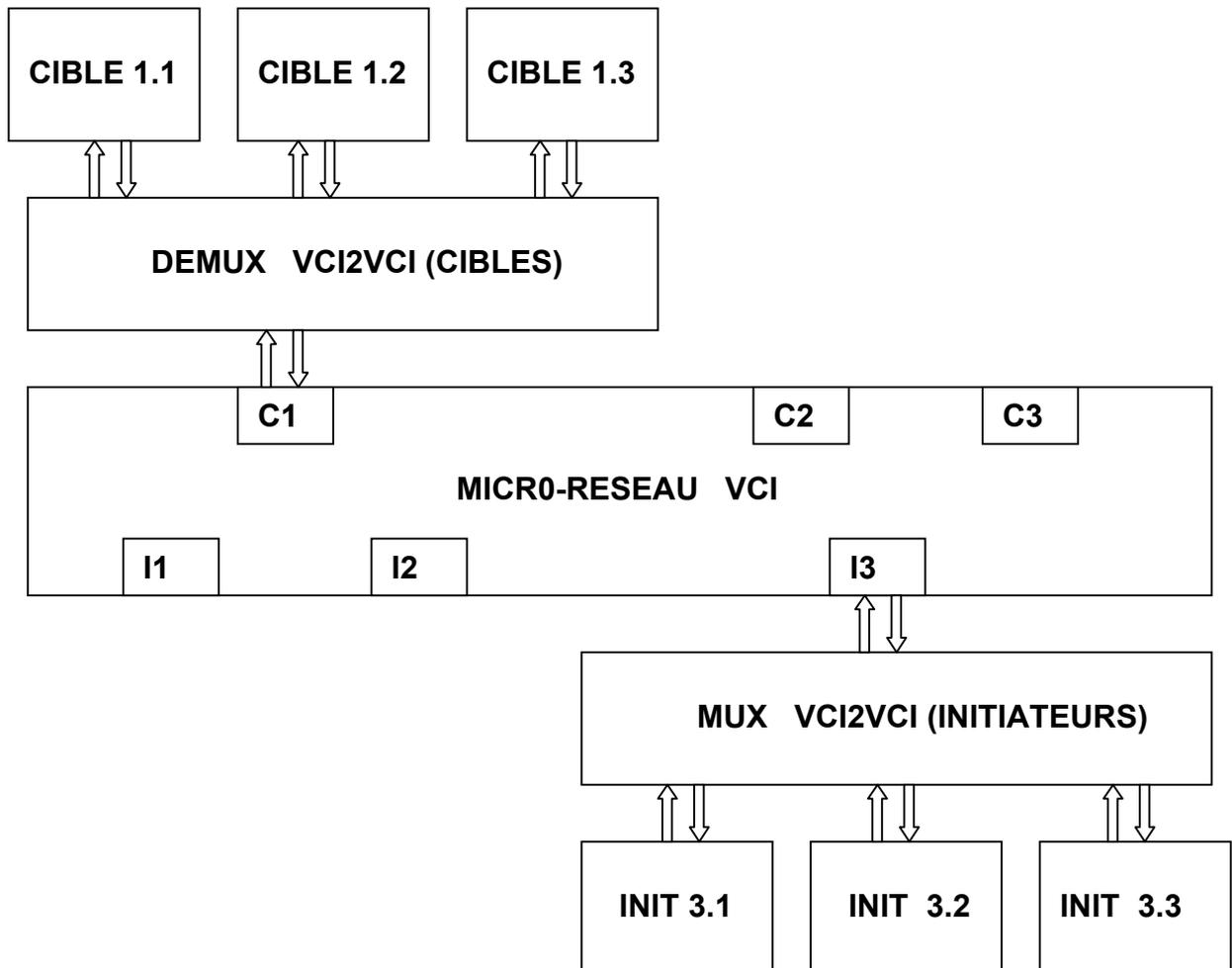
DEA ASIME / DESS CISAN

EXAMEN "SYSTEMES INTEGRES"

Cours de A. Greiner

Janvier 2004

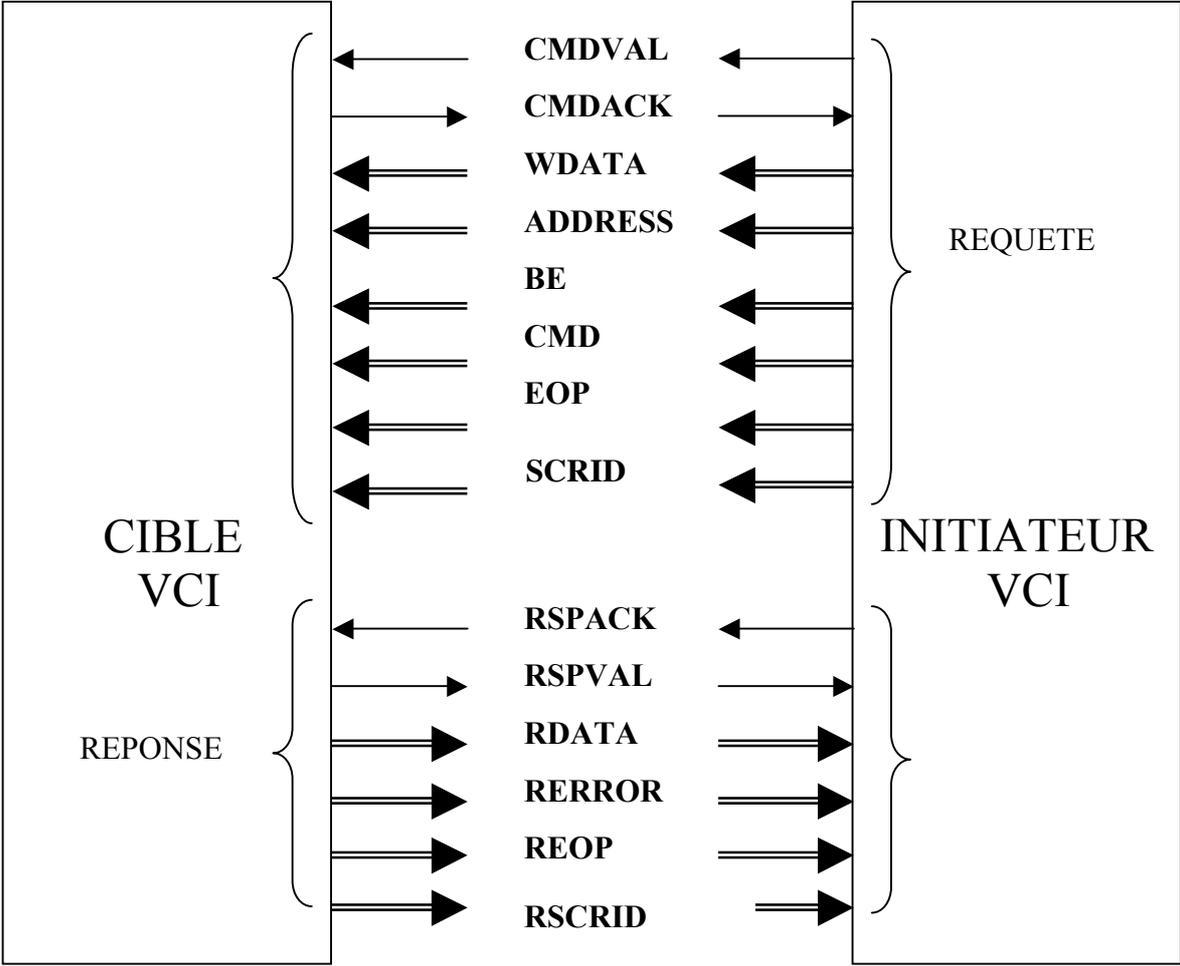
On se propose d'étudier l'architecture d'un "démultiplexeur" VCI2VCI permettant de connecter plusieurs cibles VCI à un micro-réseau intégré sur puce, en utilisant un seul port VCI cible. On étudiera également un "multiplexeur" VCI2VCI permettant à plusieurs initiateurs VCI de partager le même port VCI initiateur pour accéder au réseau. (Voir schéma ci-dessous)



Dans ce schéma, les flèches montantes sont les commandes, les flèches descendantes sont les réponses.

L'intérêt de ces composants matériels MUX et DEMUX est de permettre la réalisation de systèmes d'interconnexion respectant la norme VCI, mais possédant une structure hiérarchique à deux niveaux : Un système intégré sur puce est souvent construit en interconnectant plusieurs sous-systèmes. Chaque sous-système contient un ou plusieurs initiateurs et une ou plusieurs cibles. Grâce aux composants matériels MUX et DEMUX, tous les composants matériels d'un même sous-système sont connectés au micro-réseau par un seul port initiateur, et un seul port cible. Dans la suite on utilisera le mot cluster pour désigner un tel sous-système. Dans la figure ci-dessus le port I3 du micro-réseau est partagé par les trois initiateurs I3.1, I3.2, I3.3, appartenant au cluster 3, et le port C1 du micro-réseau est partagé par les trois cibles C1.1, C1.2, C1.3, appartenant au cluster 1.

Pour simplifier l'exercice, on utilise une interface VCI "allégée", dont on a supprimé certains signaux VCI (tels que les signaux CFIXED, CLEN, CONTIG, CONST, WRAP, PLEN, PKTID et TRDID).



La commande VCI comporte donc 6 champs (en plus des 2 signaux de contrôle de flux)

- ADDRESS (32 bits)
- WDATA (32 bits) : donnée à écrire, en cas d'écriture
- BE (4 bits) : indique quels octets doivent être écrits, en cas d'écriture
- CMD (2 bits) : définit le type de la requête : READ/WRITE/LL/SC
- EOP (1 bit) : marqueur de fin de paquet requête VCI.
- SCRID (8 bits) : numéro de l'initiateur source de la requête

La valeur des champs CMD, SCRID et BE, ainsi que les bits de poids fort de l'adresse sont constants pour toutes les commandes d'un même paquet.

La réponse comporte 4 champs (en plus des 2 signaux de contrôle de flux) :

- RDATA (32 bits) : donnée demandée, en cas de lecture
- RERROR (1 bit) : signal d'erreur
- REOP (1 bit) : marqueur de fin de paquet réponse VCI.
- RSCRID (8 bits) : numéro de l'initiateur source de la requête

On souhaite que cet interconnect à deux niveaux soit capable de router aussi bien des requêtes simples (une seule adresse), que des rafales de longueur quelconque.

Dans tout système respectant la norme VCI, l'interconnect est chargé d'acheminer les requêtes à la "bonne" cible en décodant les bits de poids fort de l'adresse VCI. Pour ce qui concerne l'acheminement des réponses, l'interconnect doit utiliser les champs SCRID pour que la réponse revienne à l'initiateur qui a envoyé la requête. Dans un interconnect hiérarchique à deux niveaux, le routage des requêtes doit être effectué à deux endroits : L'adresse doit être décodée par le micro-réseau pour router la requête vers le "bon" port de sortie cible du micro-réseau. Elle doit également être décodée par le composant DEMUX pour router la requête vers la "bonne" cible à l'intérieur du cluster. De façon symétrique, le champs SCRID doit être analysé à deux endroits pour permettre l'acheminement des réponses : Le champs SCRID doit être analysé par le micro-réseau pour router la réponse vers le "bon" port de sortie initiateur initiateur. Elle doit également être analysée par le composant MUX pour router la réponse vers le "bon" initiateur dans le cluster.

Q1) Proposer un mécanisme de décodage des adresses et un système d'indexation des cibles et des initiateurs qui supporte cette structure hiérarchique à deux niveaux.

On s'intéresse maintenant au composant DEMUX. Le composant DEMUX comporte deux blocs : Le bloc DEMUX_REQ permet d'acheminer les requêtes, le bloc DEMUX_RSP permet d'acheminer les réponses. Ces deux blocs sont totalement disjoints, car les requêtes et les réponses sont indépendantes et on ne souhaite pas introduire de couplage entre le flux des requêtes et le flux des réponses.

Q2) Représenter le chemin de donnée du bloc DEMUX_REQ dans le cas 1 vers 3, ainsi que l'automate qui contrôle ce composant.

Pour simplifier les notations, on utilisera l'indice 0 pour le port VCI permettant la communication avec le micro-réseau, et les indices 1, 2, 3 pour les trois ports VCI correspondant aux trois cibles du cluster. On fournira une représentation graphique des transitions de l'automate, et le tableau définissant la fonction de génération. Pour minimiser le coût matériel, on suppose que les commandes VCI ne sont pas stockées dans des registres, mais passent directement du port d'entrée vers l'un des ports de sortie par des chemins "combinatoires". En revanche on suppose que l'allocation du port de sortie (port 0) à un des 3 ports d'entrée, lors de la réception d'une nouvelle commande en provenance du réseau, nécessite un cycle. On définira précisément la table de vérité des signaux de contrôle de flux CMDACK0, CMDVAL1, CMDVAL2, CMDVAL3.

Q3) Représenter le chemin de donnée du bloc DEMUX_RSP dans le cas 1 vers 3, ainsi que l'automate. On fournira une représentation graphique des transitions de l'automate, et le tableau définissant la fonction de génération. Ici encore, on suppose que les réponses VCI ne sont pas stockées dans des registres, mais passent directement du port d'entrée vers l'un des ports de sortie par des chemins "combinatoires". On suppose que l'arbitrage entre les réponses provenant des 3 cibles nécessite un cycle, et que l'algorithme d'arbitrage garantit l'absence de famine.. On définira précisément la table de vérité des signaux de contrôle de flux RSPVAL0, RSPACK1, RSPACK2, RSPACK3.

Le composant MUX permettant à plusieurs initiateurs d'accéder au réseau est également décomposé en deux blocs MUX_REQ et MUX_RSP, et sa structure est donc très semblable à celle du composant DEMUX.

Q4) Préciser quelles sont les différences entre les composants MUX et DEMUX. Qu'est-ce qui limite le nombre de cibles (resp. initiateurs) qui peuvent être connectées à un composant DEMUX (resp. MUX) ?

Une autre façon - plus classique - de réaliser un mécanisme d'interconnexion à deux niveaux consiste à utiliser un micro-réseau pour interconnecter les clusters entre eux, et à utiliser un bus à l'intérieur de chaque cluster. Tous les initiateurs et toutes les cibles d'un même cluster sont connectés à un bus local, et la communication entre le cluster et le micro-réseau est assurée par un "bridge" qui réalise la conversion de protocole entre le réseau VCI et le bus.

Q5) Quels sont les avantages et quels sont les inconvénients des deux approches ?