Le G-Chip

Cédric BAIL
Yann GUIDON

December 13, 2000

Le G-Chip sera une puce centrale de l'architecture d'un F-PC, pour la simple raison que c'est sur elle que viendront se connecter toutes les cartes d'extention. Ceci a deux avantages, rendre aussi simple l'ajout d'un F-CPU que l'ajout d'une carte graphique, par exemple, et de pemettre à chaque périphérique de bénéficier d'un maximum de débit (c'est le même principe qu'un switch qui répartit de manière efficace le trajet des requêtes entre tous les périphériques).

Mais cela veut aussi dire que la complexité de l'OS gérant le système augmente en conséquence, car il faudra gérer cette multitude de possibilités de connexion. Enfin, n'importe quel périphérique peut théoriquement accéder à n'importe quelle ressource de la machine, donc le G-Chip devra gérer les sémaphores.

1 G0

Le nom de G-Chip vient du mot "Glue" dont la réduction correspond de manière anecdotique à la lettre qui succède à la lettre F (mais il n'y aura pas de "H-Chip" ;-D). Le mot "Glue" est une référence aux puces qui controlaient les Amiga ou les Atari, par exemple, elle assure ici une fonction similaire. Aujourd'hui, dans un PC, cette fonction est remplie par ce que l'on appelle communément le "chipset". Nous essayons toutefois de ne pas rencontrer les mêmes problèmes, liés principalement à la bande passante.

Le G0, devra être le plus simple possible tout en permettant de donner une vision de ce que seront les capacités d'un F-PC. Pour cela on aura besoin d'au moins deux F-BUS, l'un pour un module F-CPU et l'autre pour la carte graphique ou une extention. De plus une connexion à une FLASH pour le BIOS sera nécessaire, ainsi qu'une interface SCSI pour les mémoires de masse (l'interface IDE est plus simple, mais demande plus de ressources CPU). Enfin une connexion à un microcontroleur sera nécessire pour les périphériques lents (clavier, souris, port I2C, ...), cette interface reste d'ailleurs à définir.

Enfin le G0 sera probablement encapsulé (dans un premier temps) dans un boitier 350 broches qui seront réparties entre les 2 F-BUS (environ 70 broches chacun), le SCSI (environ 70 broches aussi), la FLASH et l'interface microcontroleur. L'interface USB n'est pas prévue.

Dans le cas où le G-Chip communique avec une carte graphique, celle-ci nécessitera une adaptation d'interface F-BUS vers une puce graphique dédiée. Dans le cas contraire, le port F-BUS libre servirait à connecter le système à d'autres extententions : interface F-BUS <-> PCI, puce multi-port F-BUS...

2 F-BUS

Le F-BUS est divisé en deux parties unidirectionnelles et indifférenciées. Dans chaque direction, le bus a besoin de 5 bits de contrôle, 4 bits "d'ordonnancement", 4 pour le numéro de transaction (tId), 16 pour les données et 5 pour la correction d'erreur (optionnel). Le bus est complétement synchrone et contrôlé par la même horloge (qui peut être externe) dans les deux sens. Il n'y a pas de signal d'ACK, tout le protocole est concentré dans les bits de contrôle. Leur fonction n'est pas encore complétement définie. Au total, le F-BUS nécessite 68 broches, 69 avec l'horloge. Le protocole comme le brochage sont sujets à des modifications futures (extension du nombre de tId, nouveaux messages de contrôle...).

3 Evolution

Le G-Chip, pour permettre les interconnexions de type diamant et footballène, devra évoluer rapidement vers une version huit ports où le rapport nombre de F-CPU sur nombre de G-Chip connecter est le plus optimal ($4/4 = 1$). Peut-être qu'un interfacage IDE serait aussi le bien venu pour faire baisser le prix des F-PC. Enfin, la gestion d'une mémoire centrale publique est envisagée.

4 Ce qui reste à définir

Le protocol du F-BUS ne faisant pas de différence avec la puce à laquelle il est connecté, on peut imaginer réaliser des "réseaux" de F-CPU, relié entre eux par un ou plusieur G-Chip. Dans cette optique, apparait le problème de routage des paquets. Or pour cela il faut des adresses de destination pour chaques puces. Donc il faut tout d'abord, définir la manière de numéroter les puces connéctées au F-BUS, puis définir la technique de routage des messages.

Il y a deux méthodes de numérotations déjà imaginé :

- Attribuer un numéro à chaques puces en fonction de leur connexion avec un G-chip de référence
- Attribuer un numéro lors du 'parcours' du réseau de connexion ou à l'aide d'un jeton (méthode quasi similaires, sauf que l'une est en soft et l'autre en hard)

Pour la méthode de routage, il y a aussi deux possibilités :

- Avec la méthode du G-Chip de références, il suffit de faire une comparaison pour savoir où doit être envoyée la requête (Avantage : méthode simple avec une initialisation rapide; Désavantage : il n'est pas possible d'interconnecter deux branches et tous les paquets qui vont d'une branche à l'autre passent par le G-Chip de référence)
- Avec la méthode de la table de routage, on regarde dans une table créée par le soft, ou par le hard, où doit être envoyé le paquet (Avantage : avec un bon algo de plus court chemin, le paquet aura toujours un trajet optimisé, on peut imaginer faire du hot-swap facilement; Désavantage : complexité plus grande du G-Chip)

De plus il reste à définir la méthode permettant de savoir à quel type de puce on a affaire ("Plug & Really Play") ainsi que les ressources dont il dispose, savoir si le protocole peut être Hot-swappable,...

5 Corrections

Si certain ont des idées ou des corrections à apporter à ce document, venez en discuter sur la "mailing-list". (f-cpu_france@egroups.com)