Liaisons série et parallèle

Les transmissions de signaux numériques peuvent être effectuées de deux manières : série ou parallèle selon qu'elles sont faites via un seul ou plusieurs conducteurs.

Très longtemps la liaison série était la plus utilisées car elle ne nécessitait qu'un seul conducteur. Puis dans les années 80, la transmission parallèle a détrôné la liaison série partout où il était possible de multiplier les conducteurs pour obtenir des voies de communication plus large. C'est efficace sur de courtes distances et c'est ce principe qui a été retenu pour les bus sur la carte mère tels que le FSB ou le bus PCI. C'est aussi le cas des nappes IDE pour échanger les données avec les disques et celui du port parallèle destiné à l'imprimante. Les communications séries étaient moins performantes pour ce genre de liaisons rapides et à courtes distances.

Actuellement, les transmissions parallèles sont à leur tour sont dénigrées à cause de problèmes d'interférences électromagnétiques entre les conducteurs disposés côte à côte. Et la solution n'est autre que le retour à la liaison série, adaptée il est vrai pour pouvoir augmenter les vitesses de transmission sans provoquer d'interférences ni y être sujettes. La liaison série se fait par deux câbles appariés qui transmettent des signaux symétriques. Nous y reviendrons, en parlant des câbles SATA, des ports USB ou des PCI express. Commençons par décrire ce que sont en principe les transmissions série et parallèle.

Liaison série

Conducteur unique ou boucle de courant

La liaison est en principe faite par un conducteur unique. C'est par exemple le cas des ports COM des PC un peu anciens. C'est aussi le cas des ports PS/2 du PC destinés au clavier et à la souris. Certaines transmissions par ondes électromagnétiques, les liaisons infrarouges ou les liaisons par fibre optique peuvent aussi être considérées comme des liaisons séries sur un seul conducteur.

Les techniques de transmissions série les plus récentes, USB, SATA et PCI express, ainsi que les câbles réseau à paires torsadées utilisent deux conducteurs (boucle de courant) pour transmettre simultanément les mêmes informations par des signaux symétriques. Cela évite les interférences électromagnétiques mais le principe de la transmission série reste le même qu'avec une liaison par un seul conducteur.

Sérialisation

Les informations à envoyer sont transmises bit par bit sur l'unique ligne de transmission. Au lieu d'être envoyés simultanément sur 8 fils parallèles, les 8 bits de l'octet à envoyer sont "sérialisés" par un registre à décalage (*shift register*) et envoyés les uns à la suite des autres sur un seul conducteur.

Le récepteur reçoit les 8 bits qui se succèdent dans un autre registre où ils sont remis côte à côte ("dé-sérialisés") pour reformer l'octet d'origine.

Vitesse de transmission

Les vitesses de transmission et de réception doivent être identiques. Ces vitesses sont exprimées en bits par secondes mais aussi parfois en bauds.

Modes simplex, half-duplex et full-duplex

Les communications sur un conducteur peuvent n'être à un moment donné qu'unidirectionnelles (mode simplex). Mais il est possible aussi de se servir du même conducteur pour alternativement émettre puis pour recevoir, c'est ce qu'on appelle le *half duplex*. On parle de mode *full duplex* quand il y a des transmissions simultanées dans les deux sens. Ce n'est possible qu'avec deux conducteurs, l'un pour l'émission, l'autre pour la réception.

Protocole

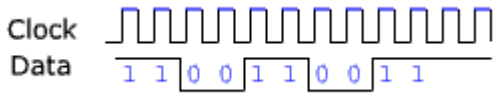
Il faut que l'émetteur et le récepteur utilisent le même protocole.

Ce protocole définit :

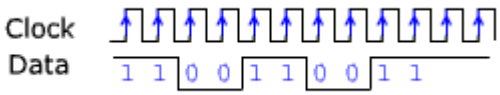
- La vitesse de transmission
- Les signaux qui annoncent le début de la communication
- Les signaux qui en marquent la fin
- Des informations de contrôle pour vérifier la validité du message reçu

Synchronisation

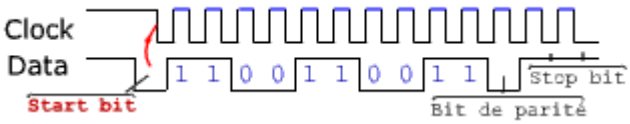
Le récepteur doit savoir à quelle fréquence l'émetteur envoie les bits sur la ligne de transmission. Sans cela, il est impossible d'interpréter les signaux reçus. La figure suivante illustre le cas de l'envoi du code 11001100. S'il ne savait à quelle cadence les bits sont émis, le récepteur pourrait très bien lire deux fois trop lentement et conclure que le code reçu est 1010. C'est une des raisons pour laquelle les lectures sont cadencées par un signal d'horloge.



La figure ci-dessus montre un signal où les données doivent être lues quand le signal d'horloge (*clock*) est au niveau haut. Certains systèmes considèrent que les données sont valides en fonction du niveau haut ou bas du signal d'horloge (*level-triggered*). D'autres systèmes utilisent le flanc montant ou descendant du signal d'horloge (*edge-triggered*) Dans la figure suivante le signal est validé par le flanc montant du signal d'horloge.



Ce signal d'horloge généré par l'émetteur est parfois transmis au récepteur par un conducteur qui accompagne la ligne des données. C'est le cas de la communication synchrone. Lorsqu'il est impossible d'ajouter cette ligne pour le signal d'horloge, le récepteur doit générer ce signal lui-même. Le fait de générer une fréquence identique à celle de l'horloge de l'émetteur n'est pas trop difficile à réaliser, le plus dur est d'obtenir un signal parfaitement synchronisé avec l'horloge de l'émetteur. Il faut donc que le récepteur sache exactement à quel instant ce signal d'horloge démarre. C'est la raison d'être du bit de départ (*start bit*) dans la communication asynchrone.



Liaison série asynchrone

Le signal d'horloge nécessite d'une liaison synchrone nécessite un conducteur pour véhiculer le signal d'horloge. Cela n'est pas toujours réalisable. La liaison asynchrone peut se passer de ce signal. Un oscillateur interne au récepteur peut générer son propre signal d'horloge qu'il synchronise sur l'émetteur au moment du signal de début de transmission.

- Un "*start bit*" marque le début du message et permet au récepteur de synchroniser son oscillateur interne qui sera l'horloge qui lui indiquera à quels moments les bits successifs pourront être lus.
- Des bits supplémentaires peuvent servir à détecter les erreurs éventuelles. C'est le rôle du bit de parité.
- Un, un et demi ou deux bits d'arrêt (*stop bit*) vont marquer la fin du message envoyé.

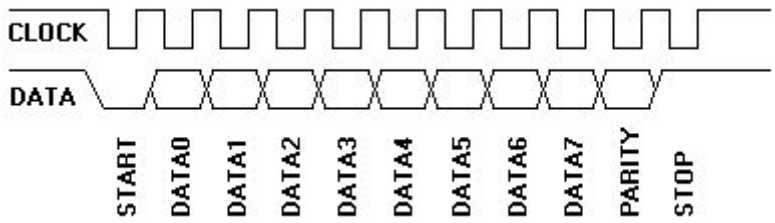
Applications :

La transmission asynchrone convient si les données ne sont pas présentes en permanence. Les bits qui encadrent le message à transmettre réduisent la capacité de transmission. Cette liaison était celui des ports COM des anciens PC (RS232) connecteurs DB9 mâles (ou DB25 mâle sur les PC encore plus anciens)

Liaison série synchrone

Le conducteur qui véhicule les données est accompagné d'un second conducteur affecté au signal de synchronisation (horloge ou *clock*) pour cadencer les transmissions.
Le flux de données peut dès lors être ininterrompu sans séparation entre les données envoyées.

On retrouve ce type de communication dans les ports PS/2 pour le raccordement du clavier et de la souris avant que les connexions USB ou sans fil ne leur vole la vedette.



Source de l'image : http://www.computer-engineering.org
Author : Adam Chapweske Last Updated 05/09/03

Liaison parallèle

Des fils mis côte à côte, généralement 8, 16, 32 ou même 64, transmettent simultanément autant de bits qu'il y a de conducteurs. L'intérêt de ce type de communication est que le débit des données est multiplié par le nombre de fils mis côte à côte. Ce type de transmission a longtemps été considéré comme plus rapide que les transmissions de type série.

Application :

- Port de l'imprimante (DB25 femelle)
- Nappe IDE pour la connexion des disques
- Bus PCI et AGP pour les cartes d'extension
- Bus système (FSB) entre le processeur et le pont nord

Limitations :

Des interférences dues à des phénomènes d'induction électromagnétique apparaissent entre les conducteurs électriques mis côte à côte. Ce problème de diaphonie (*cross-talk*) devient gênant pour les fréquences élevées ou sur les lignes trop longues.