

Architecture des systèmes à processeur

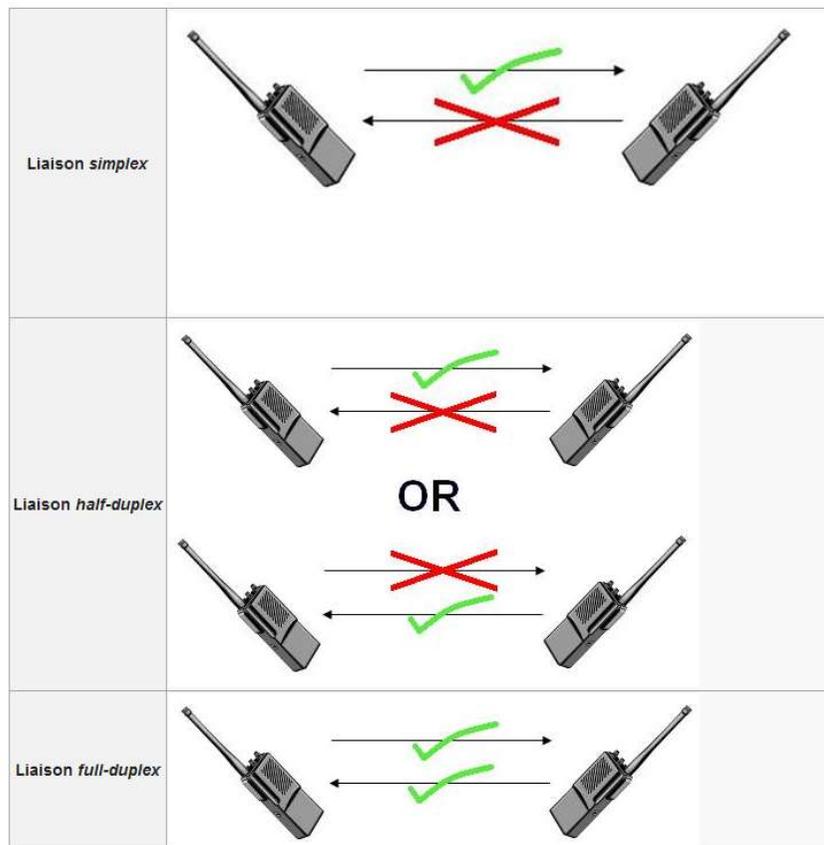
Prof. Géraldine Conti

Basé sur les cours des Profs. Sanchez, Starkier, Mosqueron et Dassatti

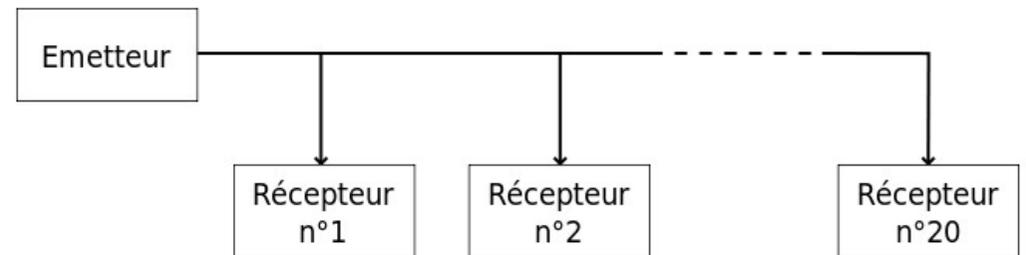
Bus et transfert de données

Transfert de données : introduction

Liaisons point-à-point : connecte
2 composants

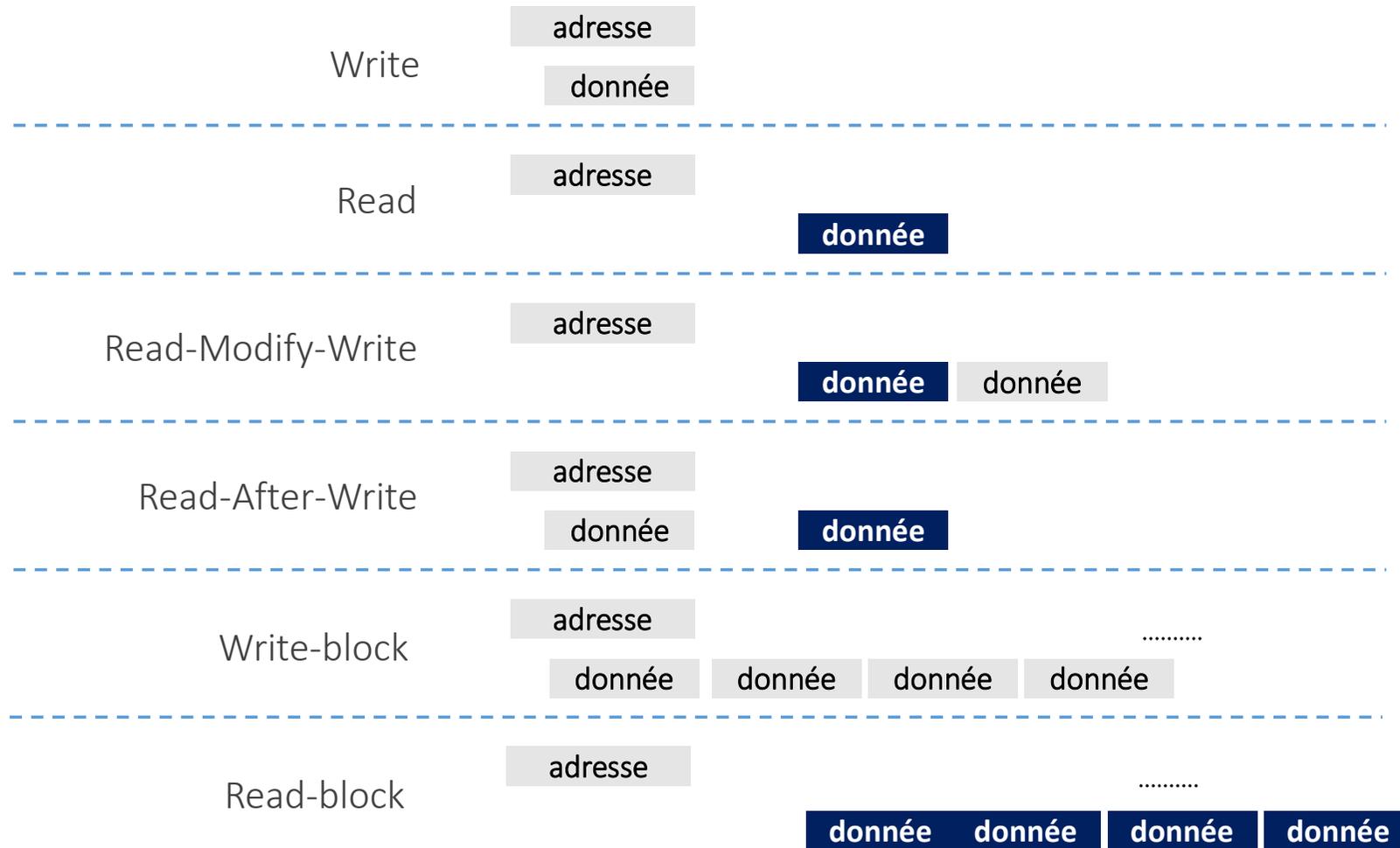


Bus de communication :



Plusieurs composants connectés (envoi de données à plusieurs composants en une seule fois)

Types de transfert

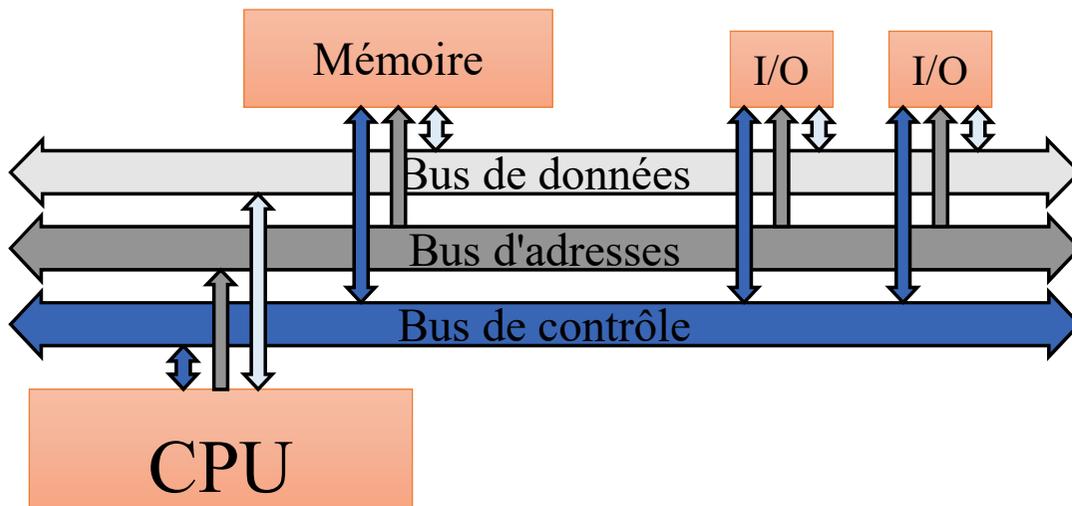


Bus : définition

- *Support de transfert de données entre les différents sous-ensembles d'un système informatique*
- Physiquement : **ensemble de fils conducteurs**
- **Données** : numériques binaires
- **Sous-ensembles** : les parties (unités) d'un processeur, les mémoires, les périphériques

Types de bus (1)

- Bus **parallèle**
 - Signaux adresse/data/contrôle transmis parallèlement
 - Multiplexage possible
 - **Exemple : bus mémoire, PCI,...**



- Bus **série**
 - Signaux adresse/données non-différenciés
 - Transmission bit par bit
 - Forme préférée (plus compacte, moins chère)
 - **Exemple : PICe, USB, I2C**

Dans les deux cas, transmissions **synchrones** et **asynchrones**.

Multiplexage

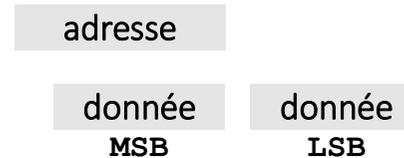
- **Multiplexage adresse /donnée**

- Suppression du bus d'adresse



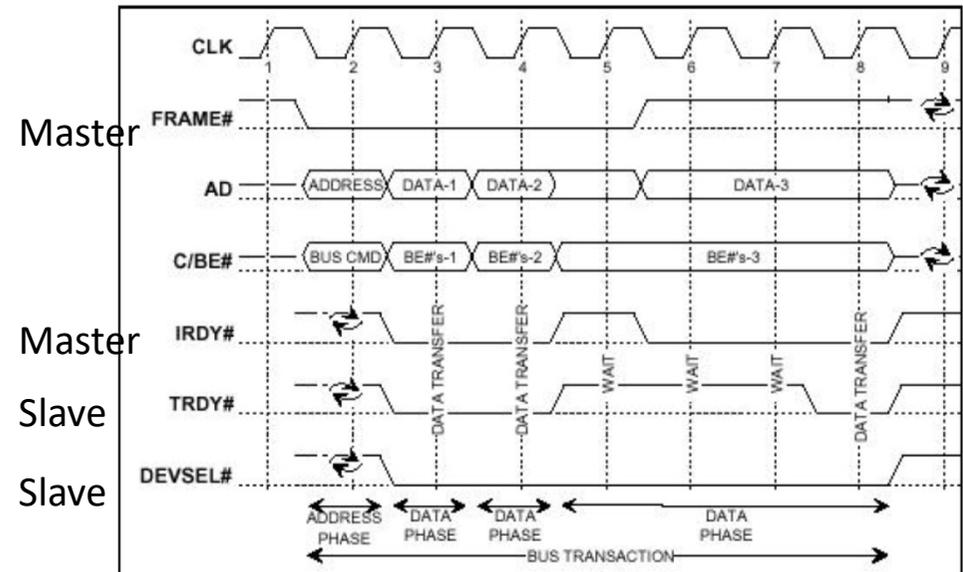
- **Multiplexage données**

- Réduction de taille du bus de données
- Ex: data 32 bits sur bus 16 bits



- **Multiplexage adresses**

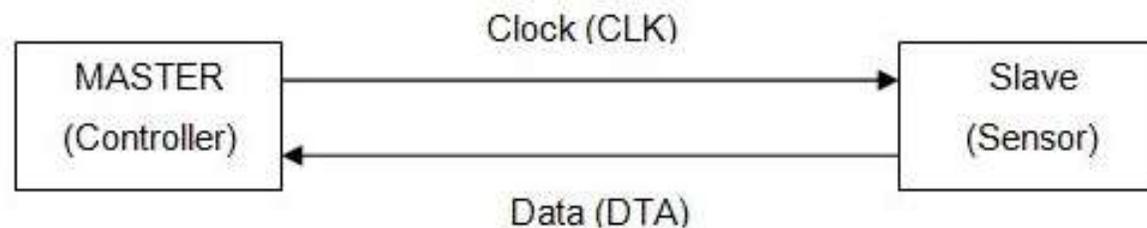
- Réduction de taille du bus d'adresse
- **Ex: adresses 24 bits sur bus 16 bits**



Types de bus (2)

- **Maître / esclave**

- Transferts initiés et contrôlés par le maître
- **Ex: bus local processeur**

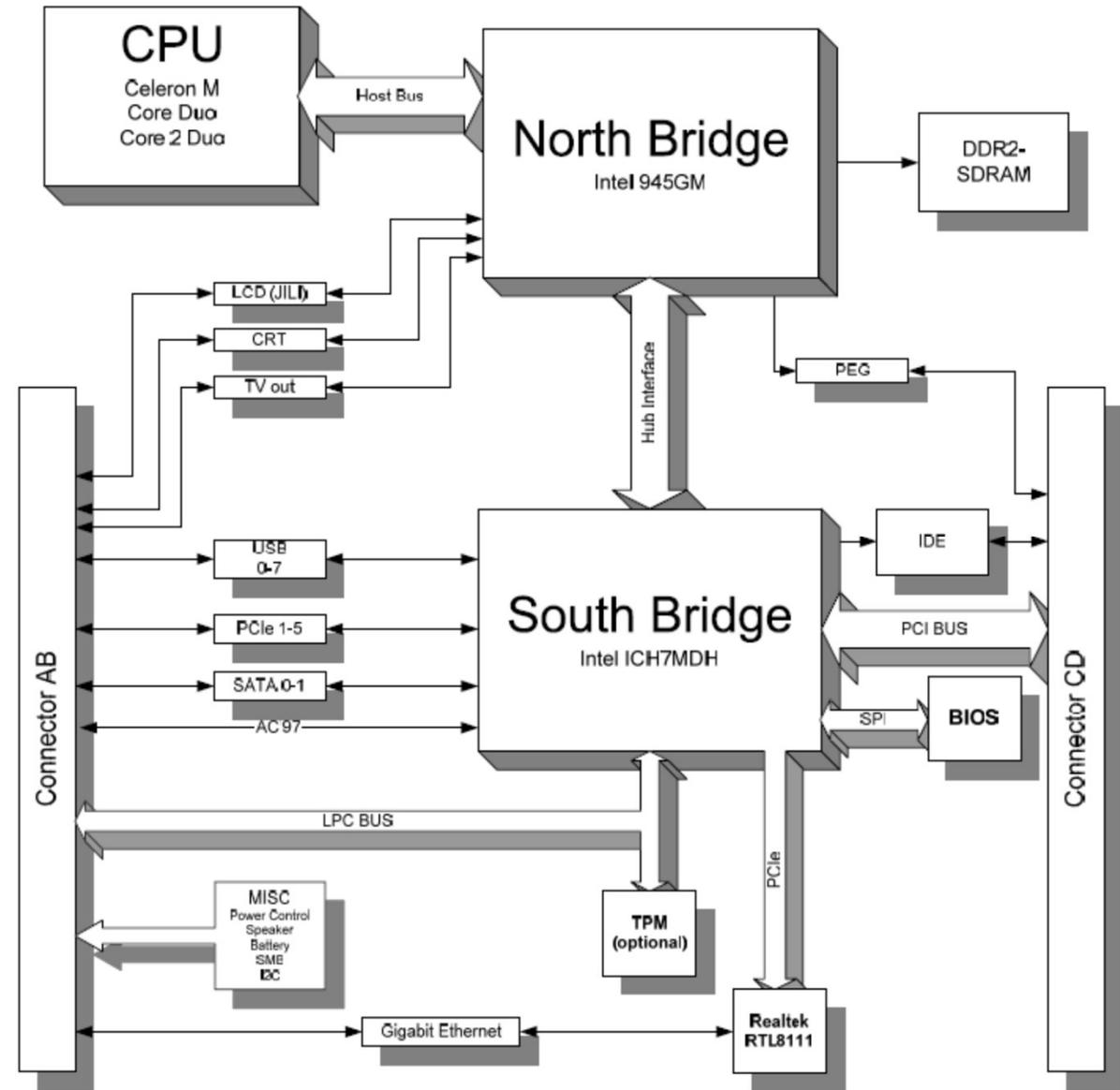


- **Multi-maître**

- Contrôle partagé
- Nécessite un arbitrage
- **Ex: réseaux, CPU-DMA**

Types de bus (3)

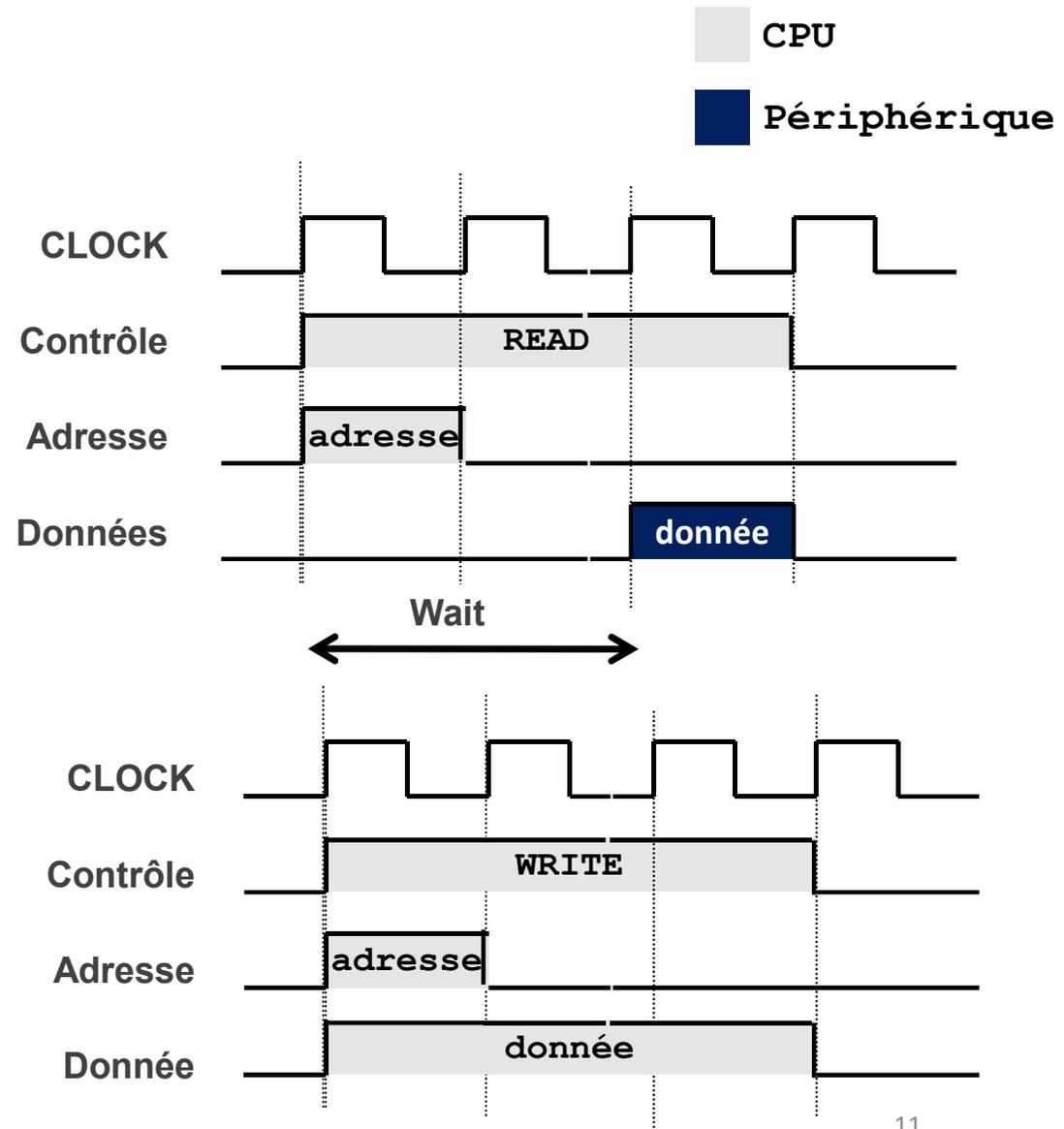
- Interne
- Externe

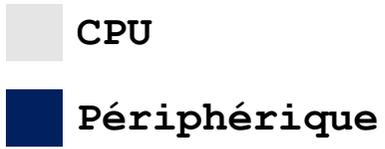


Bus parallèle

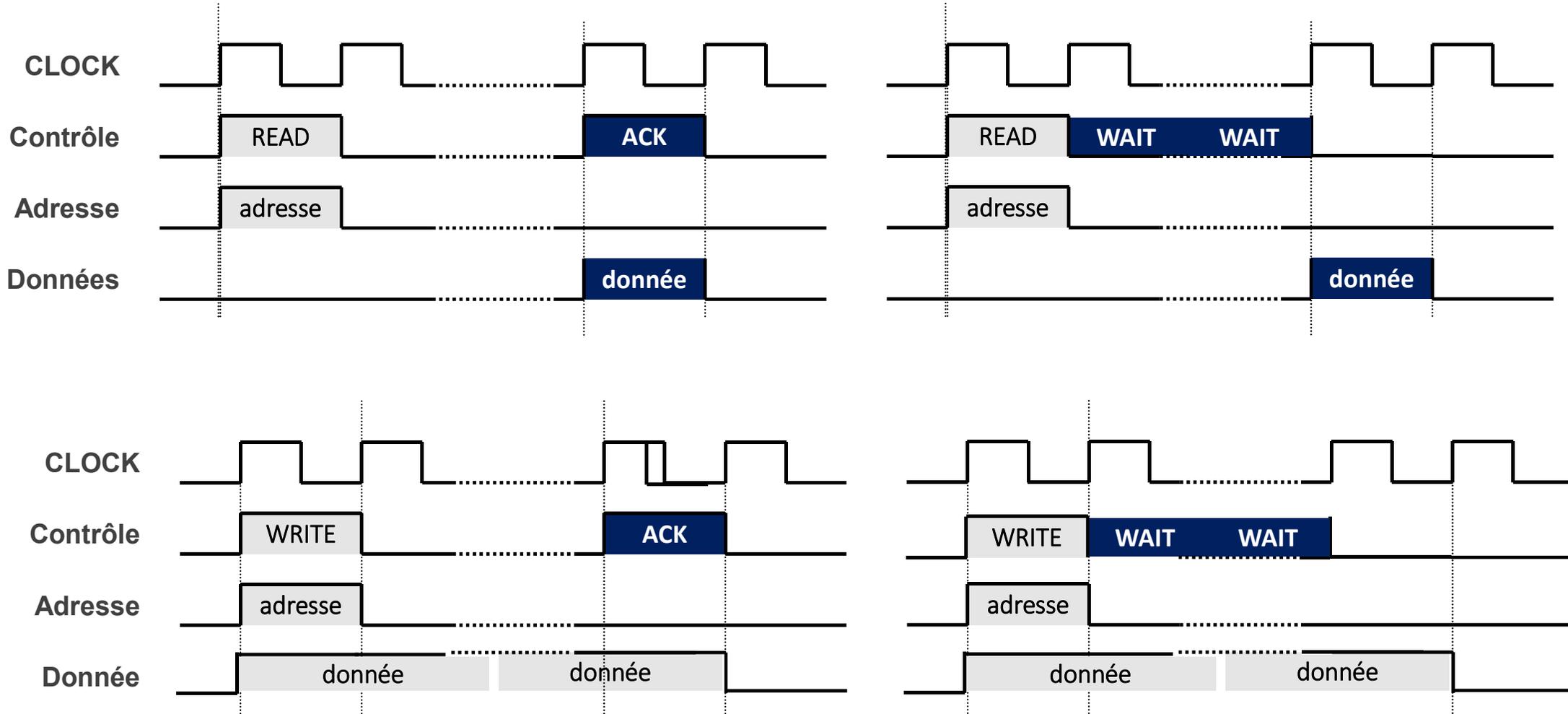
Liaison synchrone

- Transferts CPU – périphérique :
 - Le CPU est **maître**
 - Le CPU **initie les transferts** (lectures ou écritures)
 - Le CPU place l'adresse du périphérique sur le bus d'adresse
 - Le CPU peut attendre une réponse du périphérique
 - **acquiescement (ACK)**
 - **ou demande d'attente (WAIT)**



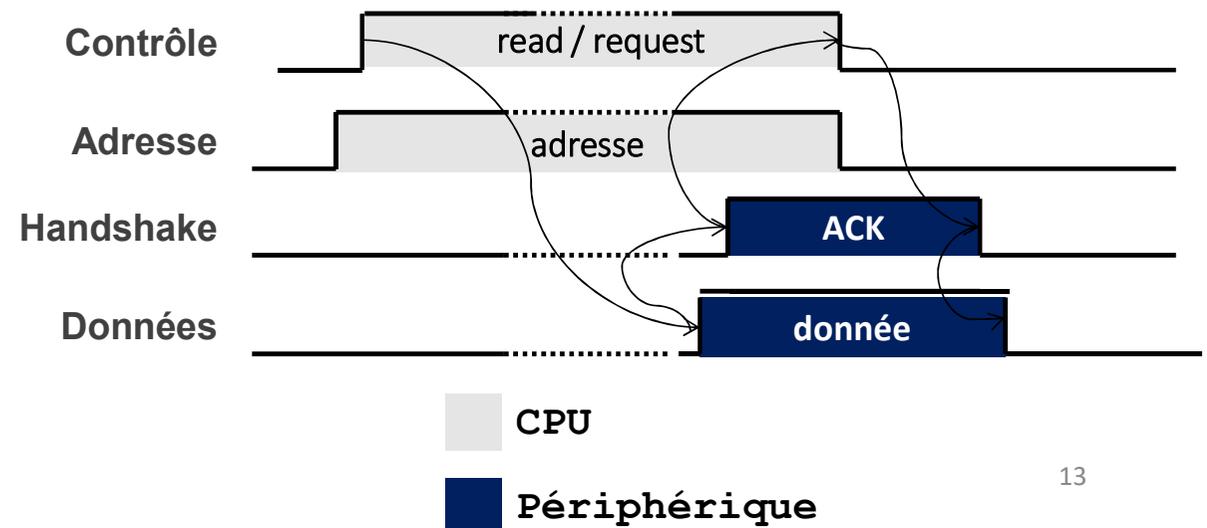
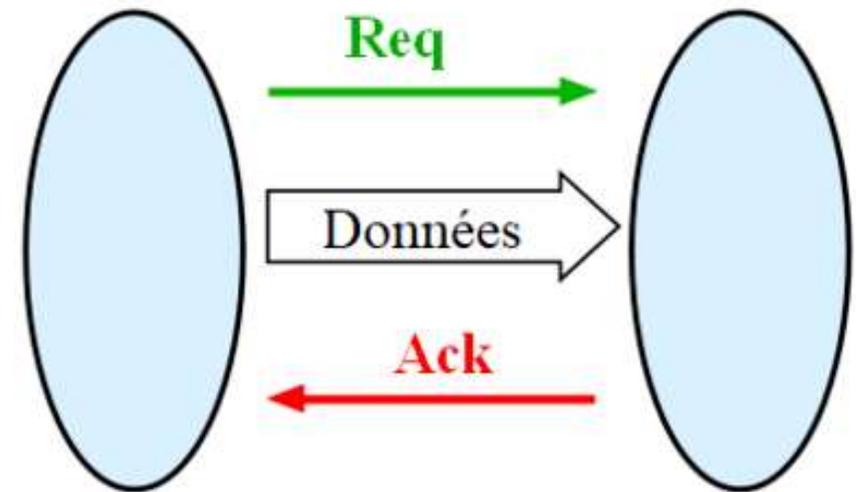


Liaison synchrone avec *handshake*



Liaison asynchrone

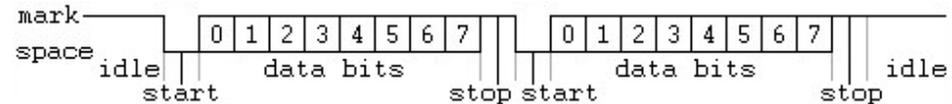
- **Pas d'utilisation d'horloge**
- Processus :
 - Premier nœud indique au second qu'il lui envoie une donnée
 - récepteur reçoit la donnée et indique qu'il l'a prise en compte
- **Synchronisation** faite grâce à des fils spécialisés qui transmettent des **bits particuliers**.
 - **Fil REQ** : indique l'envoi d'une donnée
 - **Fil ACK** : accepte la donnée indiquée par REQ



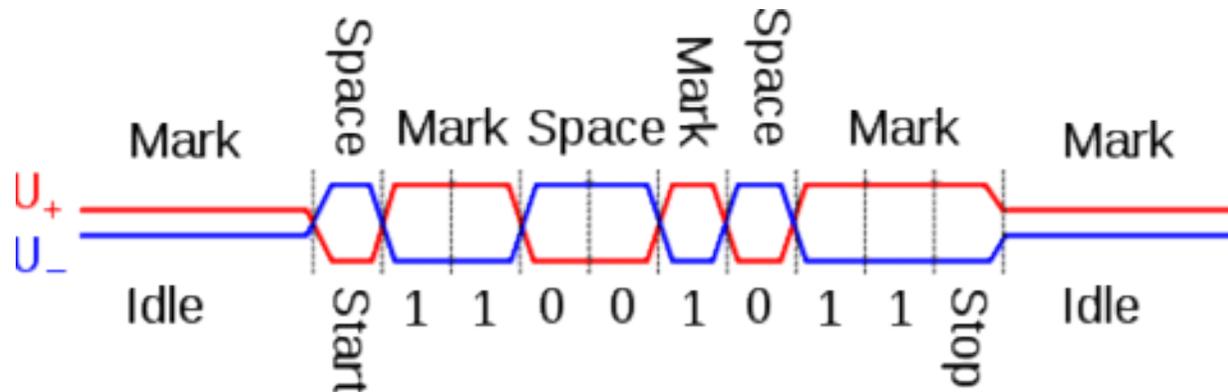
Bus série

Transmission asynchrone

- Pas d'horloge transmise, **start et stop bit**

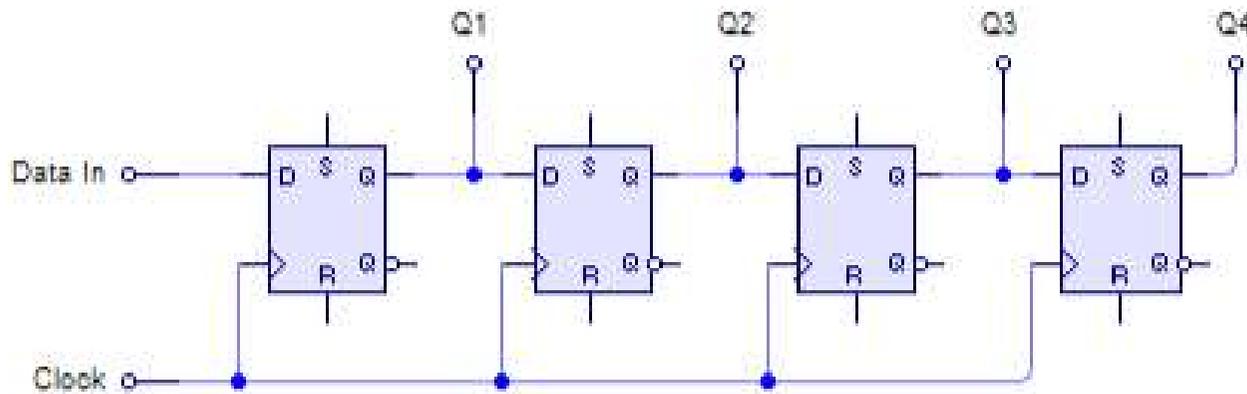


- Horloge de réception dont la fréquence est un multiple (x4 à x16) de la fréquence des données.
- Echantillonnage de l'état des bits de donnée synchronisé par un compteur
- Remise à zéro du compteur par le start bit
- **Exemple : UART (universal asynchronous receiver transmitter)**



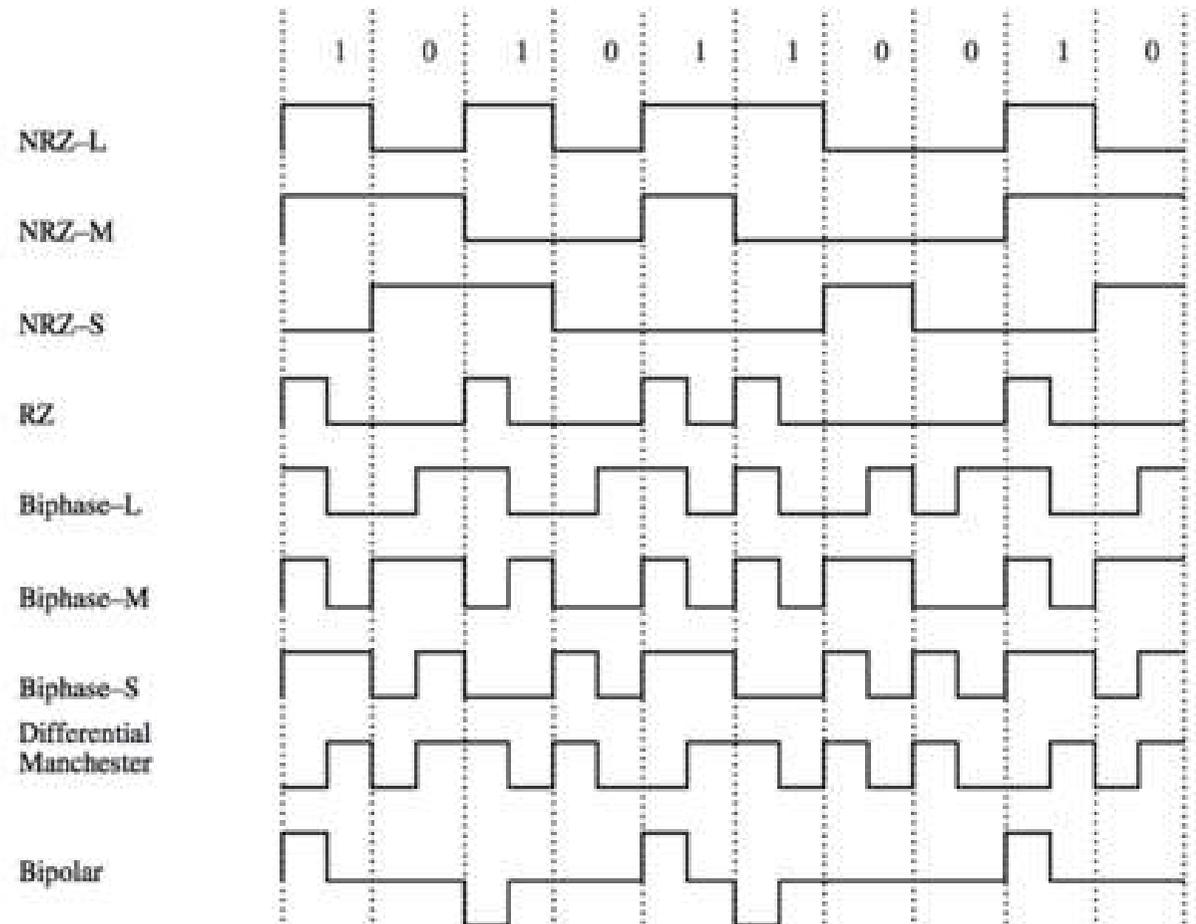
Transmission synchrone

- **Horloge** transmise séparément (bus SPI ou I2C) ou extraite des données (PCIe, USB, SATA).
 - Extraction de l'horloge par PLL (Phase Locked Loop)
- **Extraction des données** par bascules (shift register)



Encodage des données

- Bit codé avec une **tension électrique** (niveaux)
- **NRZ** ⇔ Non Return to Zero
- **Codage Manchester**
 - Transition 0 → 1 ⇒ 0
 - Transition 1 → 0 ⇒ 1
 - Composante continu = 0
- **Differential Manchester**
 - Deux transitions ⇒ 0
 - Une transition ⇒ 1
 - Composante continu = 0
 - Insensible à l'inversion de polarité



Encodage des données : 8b/10b et scrambling

8b/10b

- Codage de symboles 8 bits en symboles 10 bits
- Au minimum quatre transitions d'un état logique dans 10 bits (extraction de l'horloge)
- Composante continu = 0 (même nombre de 0 et de 1)
- 12 symboles spéciaux pour la délimitation des trames

Scrambling

- Contenu transcodé par une séquence pseudo-random
- Composante continu = 0 (même nombre de 0 et de 1)
- Délimitation des trames en rajoutant quelque bits (par exemple 64b/66b)

Exemple : Encodage 8b/10b

- mots 3 bits => 1 ou 2 équivalences en 4 bits

001 => 1001 $2x1 + 2x0$

100 => 1101, 0010 $3x1 + 1x0, 1x1+3x0$

- mots 5 bits => 1 ou 2 équivalences en 6 bits

00011 => 110001 $3x1 + 3x0$

00010 101101, 010010 $4x1 + 2x0, 2x1 + 4x0$

- Exemple

(choix de codes pour maintenir l'égalité bit a 0 et à 1)

001 00011 100 00010 001 00010 001 00010

1001 110001 1101 010010 1001 101101 1001 010010

Exercice

A faire seul

Objectif : réaliser un slide par bus/interface avec toutes les informations demandées dans le slide suivant

Liste des bus à traiter : *PCI, PCIe, SATA, I2C, USB2, USB3, USBC, SPI, JTAG*

Supports :

Notes de cours : annoter personnellement le document suivant :

Créer un compte à : <https://app.perusall.com/>

Clé du cours : CONTI-MMTQT

Internet pour les informations manquantes

Exercice : Bus/Interface ~~XXXX~~

- Acronyme
- Année d'apparition, dernière version disponible
- Type (parallèle/série, ME/MM, interne/externe)
- Caractéristiques (nombre de fils, débit, taille des mots, codage)
- Architecture (logique)
- Protocole (structure de paquet)
- Autres détails
- Exemples d'utilisation

2^e séance classe inversée : déroulement

- Chacun va travailler sur 2 bus, selon les préférences données
- Groupes de 2, travail de 45 minutes par bus
- A la fin de la séance, les personnes responsables (en couleur) envoient le(s) slide(s) correspondant au bus en question

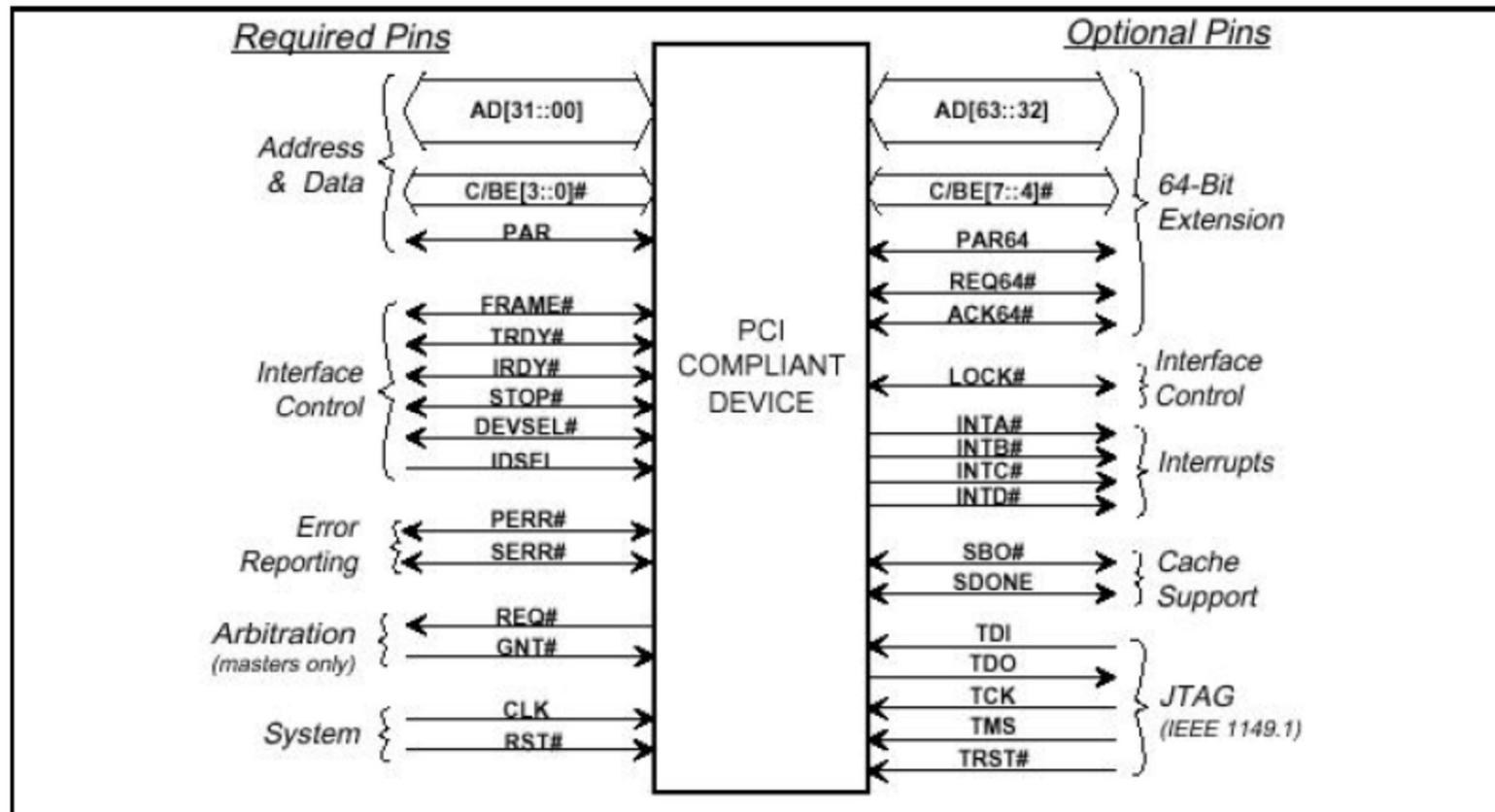
	PCIe	SATA	I2C	USB2	USB3	USBC	SPI	JTAG
14h55-15h40	Tommy	Isaia			Leo		Hayman	
	Gab	pierrick			Simon		Farouk	
15h45-16h30			Isaia	Pierrick		Leo		Gab
			Tommy	Simon		Farouk		Hayman

Test semaine prochaine : déroulement

- Test de 15min (mémoires, bus en général) + question personnalisée sur le bus que vous avez aidé à préparer
- Exposés de 8 minutes (max) de votre slide
 - Durant l'exposé, prenez des notes des questions potentielles que vous avez et remettez-les-moi à la fin de la séance.
- Note finale = Test * 0.3 + Exposé * 0.7
- Semaine suivante : Résultats et discussion sur les bus par rapport aux questions posées, puis labos

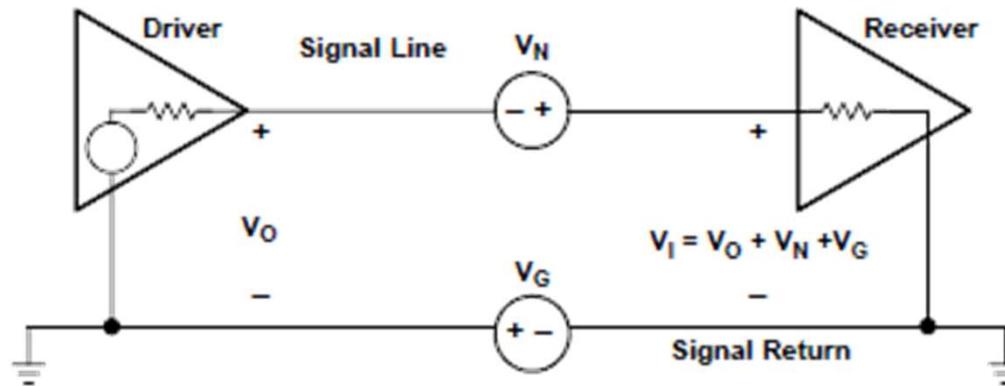
Back-Up Slides

Bus PCI



Liaison asymétrique

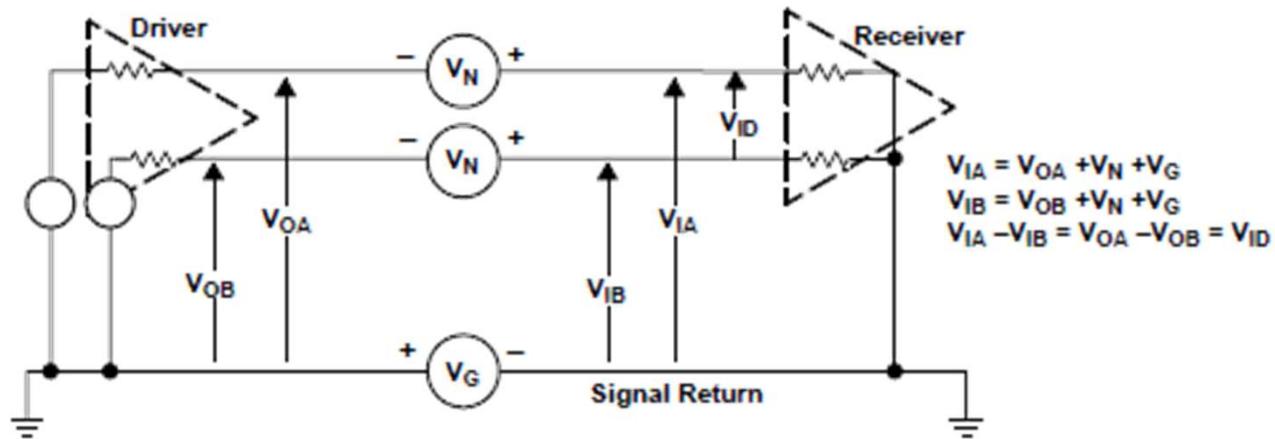
heig-vd



- Signal référencé par rapport à la masse (un fil)
- Etat logique (0/1) dépendant du niveau du signal
- Nécessite dans certains cas des terminaisons (résistances) séries ou parallèles.

Liaison différentielle (symétrique)

heig-vd



- **Signal différentiel non référencé par rapport à la masse (2 fils) => excellente immunité au bruit et aux perturbations**
- **Etat logique (0/1) dépendant de la polarité du signal**
- **Nécessite une terminaison parallèle.**

Bus I²C : Transaction

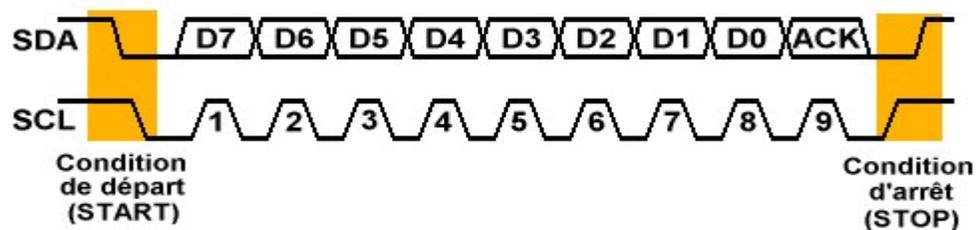
- Transaction sur bus I²C:

1. Prise de contrôle du bus
2. Start
3. Ecriture de l'adresse composant et R/W
4. Ecriture d'une commande ou d'une adresse interne (facultatif)
5. Lecture ou écriture de la donnée 1
6.
7. Lecture ou écriture de la donnée n
8. Stop

Bus I²C

Prise de contrôle du bus

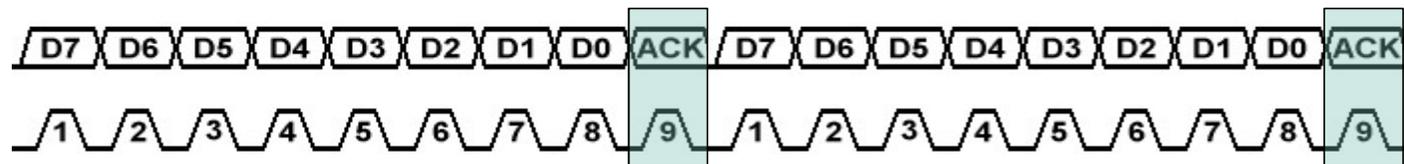
- Le bus doit être au repos avant la prise de contrôle SDA et SCL à 1
- Conditions start / stop:
 - Condition de départ: SDA passe à 0, SCL reste à 1.
 - Condition d'arrêt: SDA passe à 1, SCL reste à 1.
- Après avoir vérifié que le bus est libre, le circuit peut prendre le contrôle de celui-ci et ainsi devenir le maître.
 - C'est lui qui génère le signal d'horloge.



Bus I²C

Ecriture de données

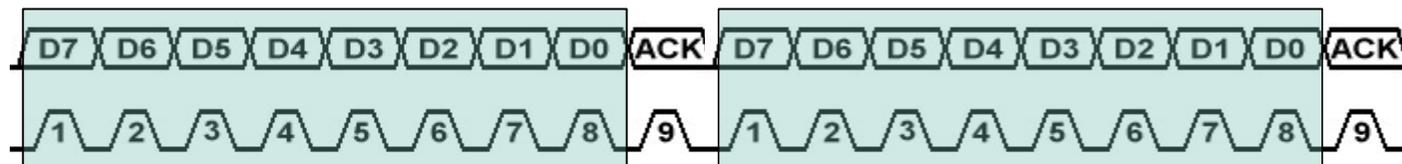
1. Le maître place le bit de poids fort D7 sur SDA
2. Il valide la donnée en appliquant un niveau '1', puis '0' sur SCL
3. L'opération est répétée avec D6, D5, etc. jusqu'à ce que l'octet complet soit transmis.
4. Il envoie le bit ACK à '1' en scrutant l'état réel de SDA.
L'esclave doit imposer un niveau '0' pour signaler que la transmission s'est déroulé correctement.



Bus I²C

Lecture de données

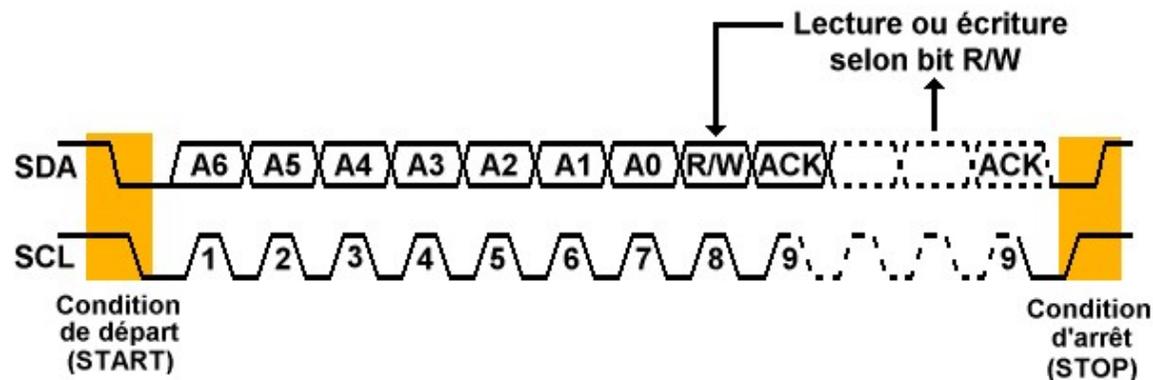
1. L'esclave place le bit de poids fort D07 sur SDA
2. Il maintient la donnée pendant que le maître applique un niveau '1', puis 0 sur SCL
3. L'esclave place un nouveau bit. L'opération est répétée avec D6, D5, etc. jusqu'à ce que l'octet complet soit transmis.



Bus I²C

Transmission d'une adresse

- Le maître envoie sur le bus l'adresse du composant avec qui il souhaite échanger des données.
 - Grâce au bit D0 (R/W), il précise également s'il souhaite lire (bit à 1) ou écrire (bit à 0).



JTAG – Contrôleur TAP

- Le contrôleur TAP (*Test Access Port*) gère la partie JTAG du circuit
 - Machine d'état (16 états)
 - Registres id, instructions et données
 - 10 instructions normalisées pour le test des circuits.
 - Fréquence horloge 10 à 40 MHz (jusqu'à 100MHz)
- Interface JTAG 4 broches (+ 1 optionnelle)

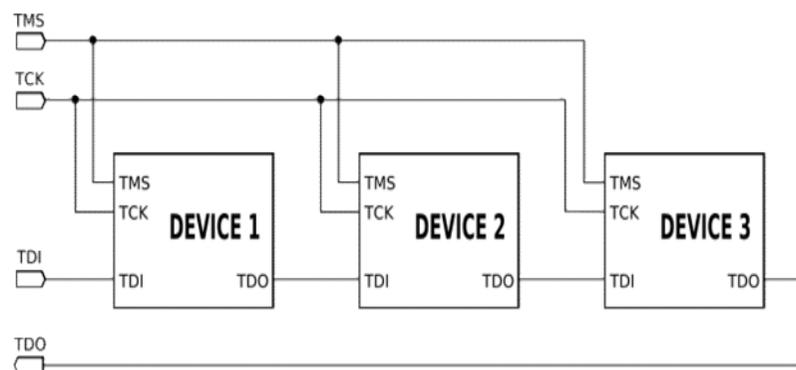
TCK: horloge (Test Clock)

TMS: activation (Test Mode Select)

TDI: entrée donnée série (Test Data In)

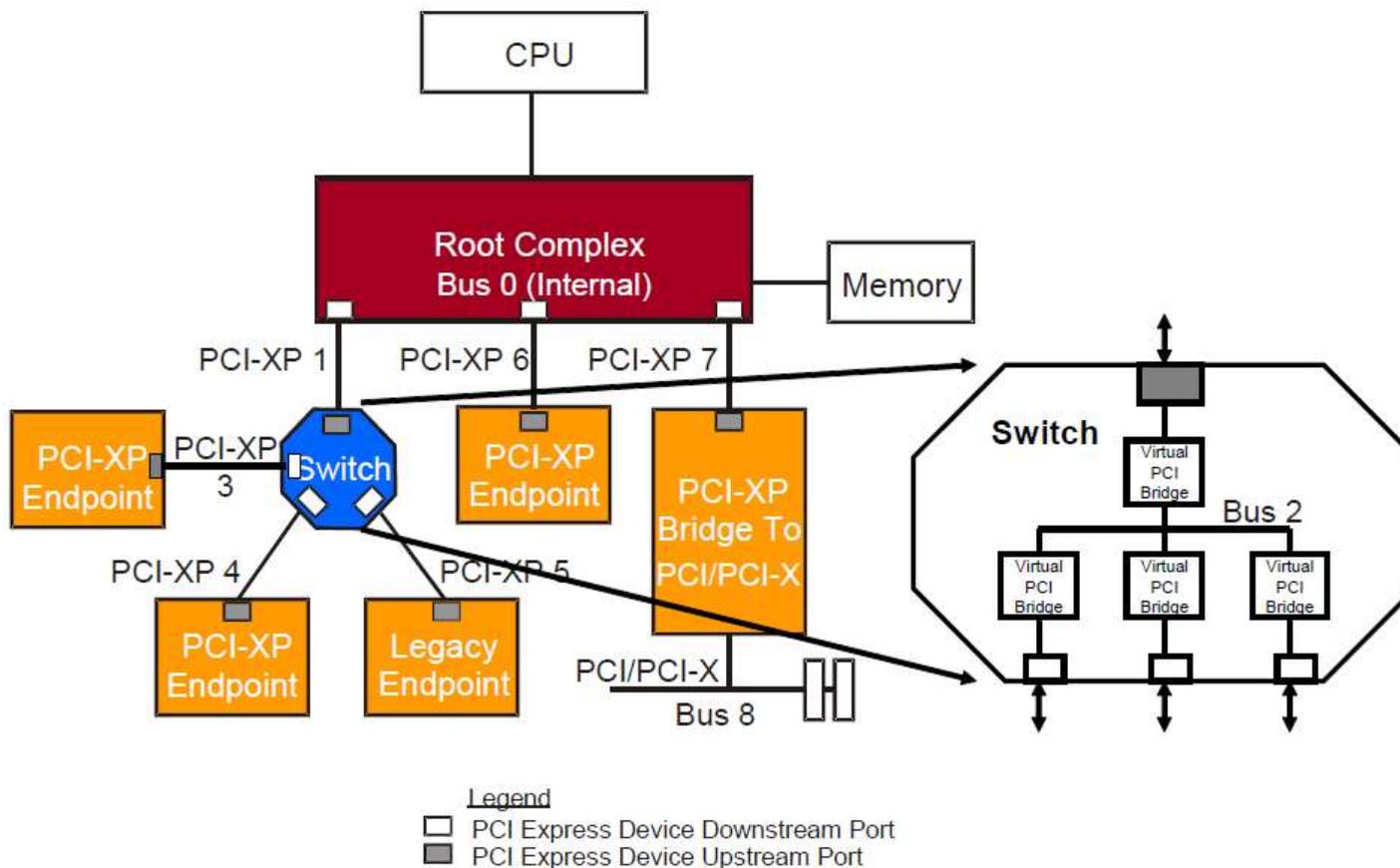
TDO: sortie donnée série (Test Data Out)

(TRST: reset du TAP)



Topology DCI_Fvnracc

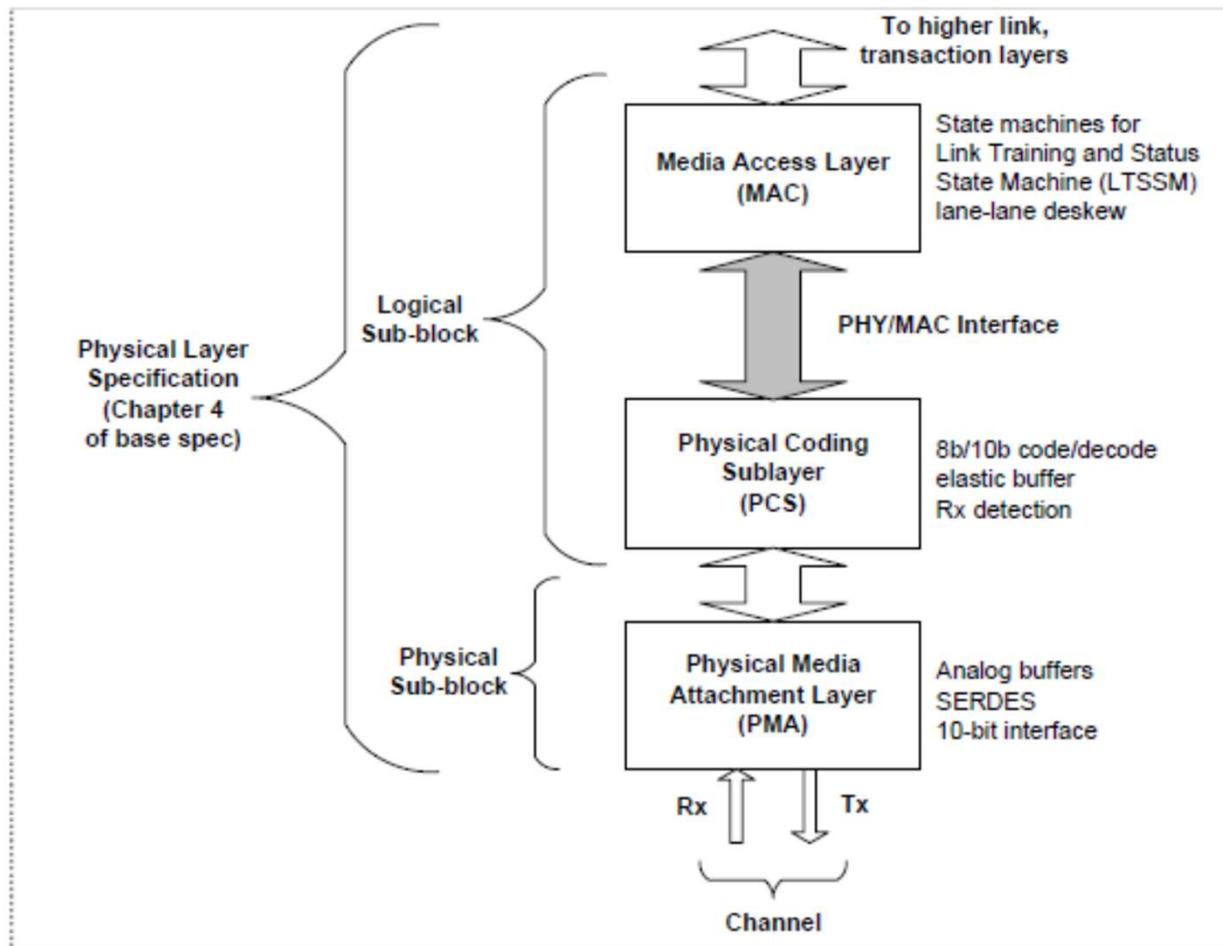
- adfaf



PCI-Express modèle en couche

- Physical Layer
 - PHY => extraction horloge et donnés
 - encodage / décodage 8b/10b ou scrambling
 - MAC => initialisation / synchronisation / erreurs
- Data Link Layer
 - gestion des paquets
- Transaction Layer
 - transferts avec séparation des demandes et des réponses

PCI-Express: couche physique

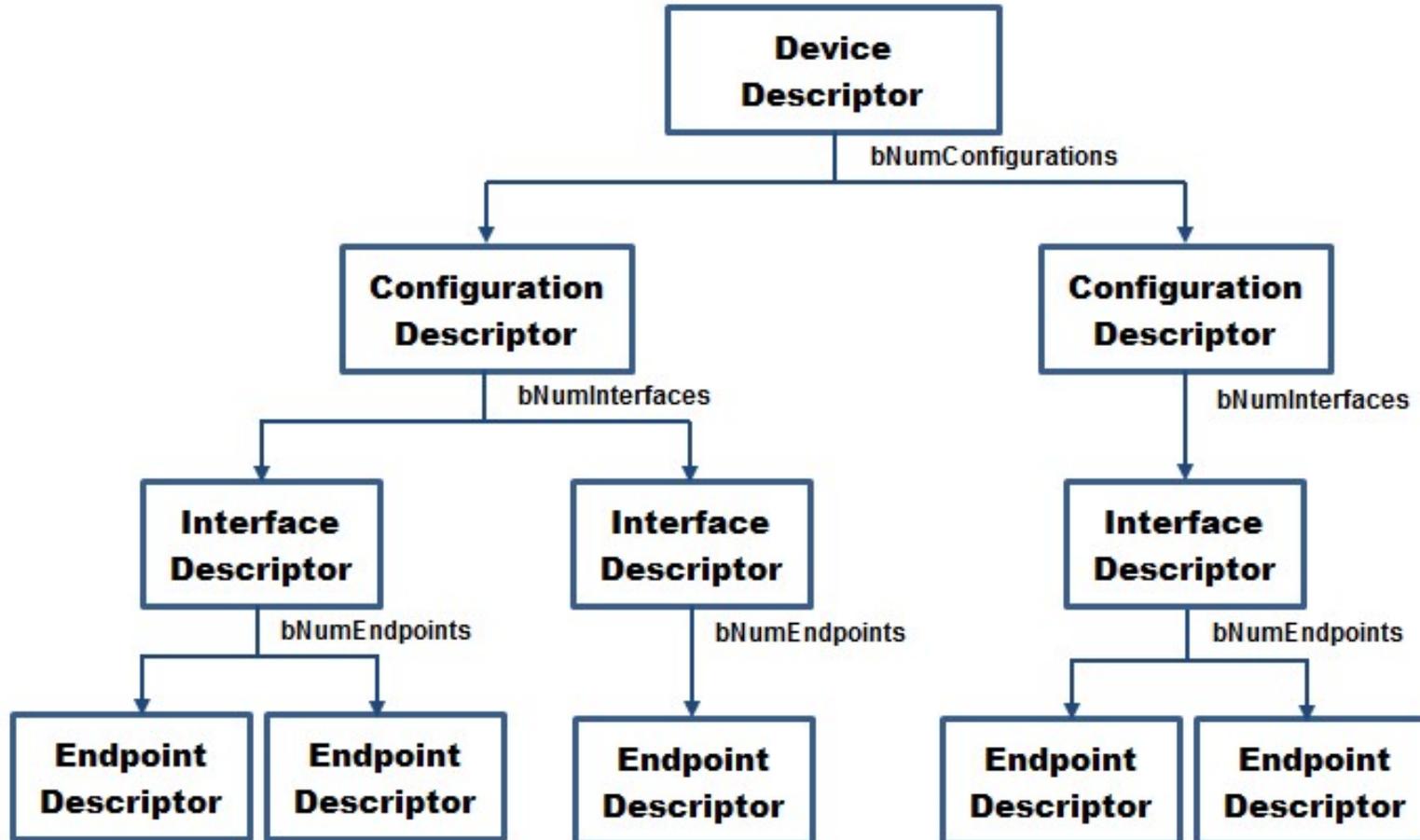


USB : transactions

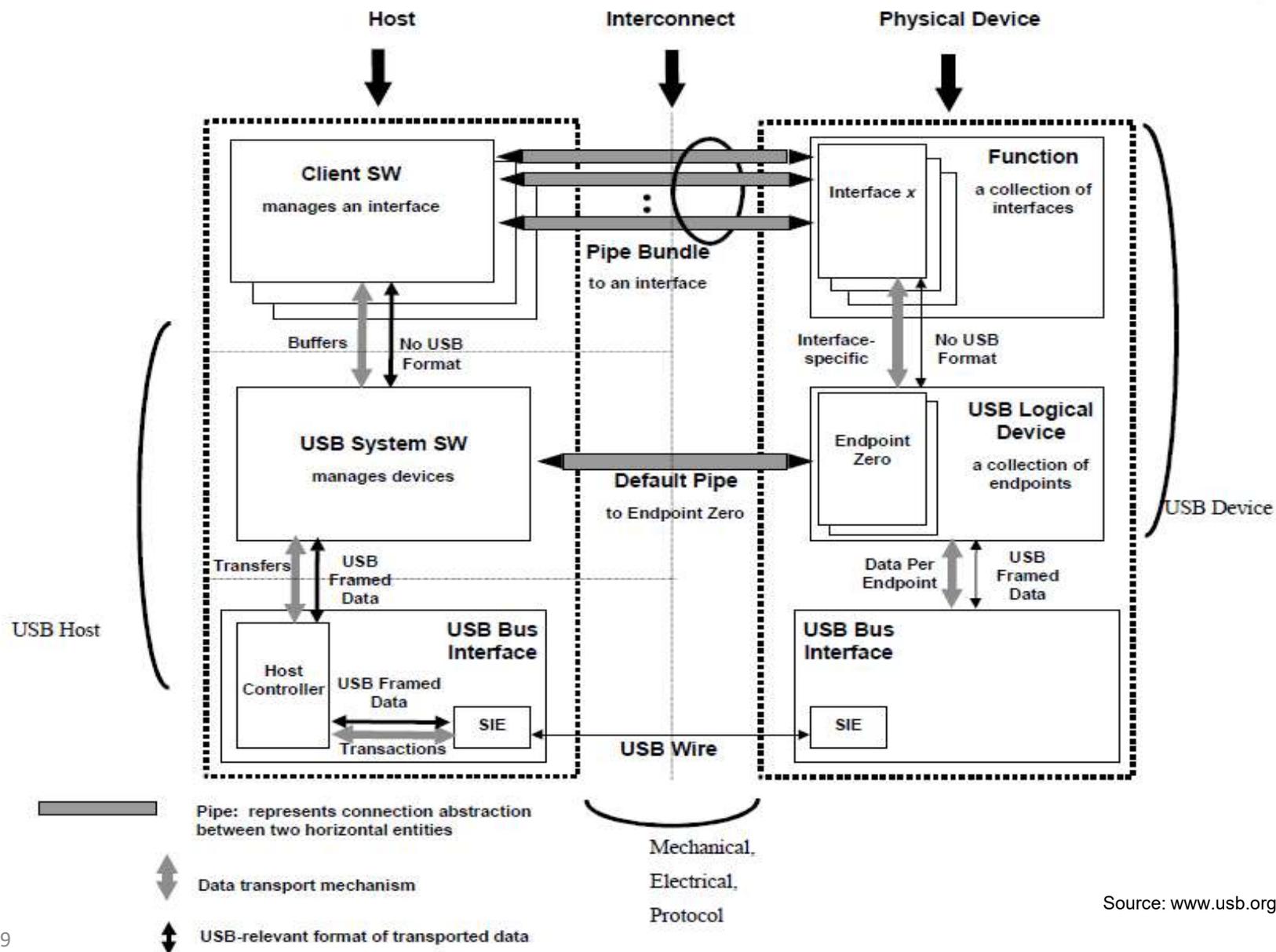
- Control transfers : Commande, status et initialisation
 - Interrupt transfers : transferts de données avec temps de latence garanti, par exemple pour souris et claviers
 - Isochronous transfers : transferts de donnée avec bande passante garantie, par caméra ou périphérique audio
 - Bulk transfers : transferts par blocs pour grande quantité de données
-
- Types de paquets : Token (commande), données et status

USB Descriptors

heig-vd



reig-vo

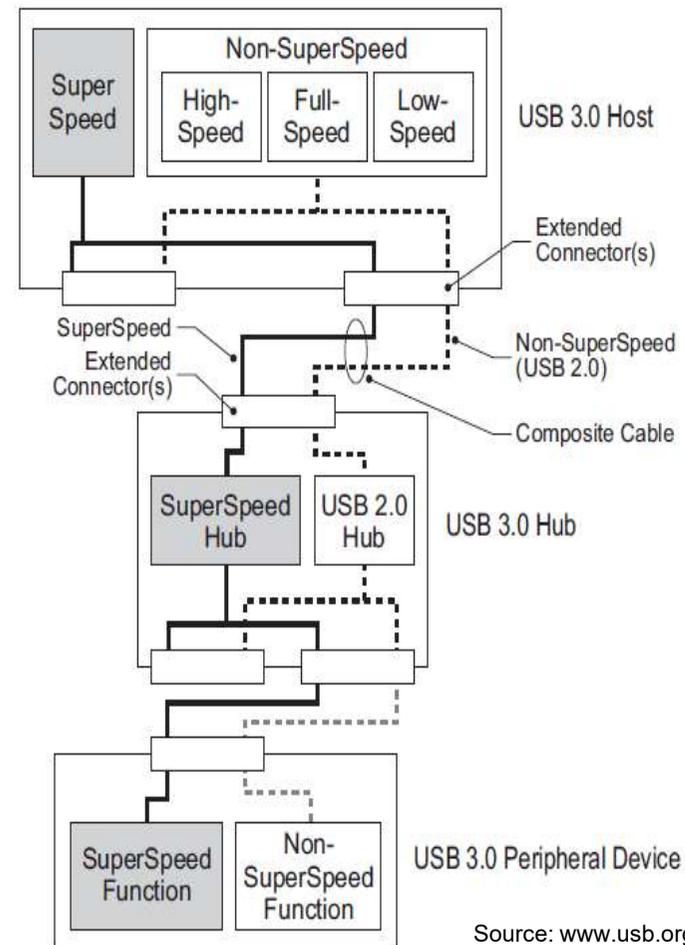


Source: www.usb.org

USB 3.0

heig-vd

- USB 3.0 comprend USB 2.0 et le bus SuperSpeed



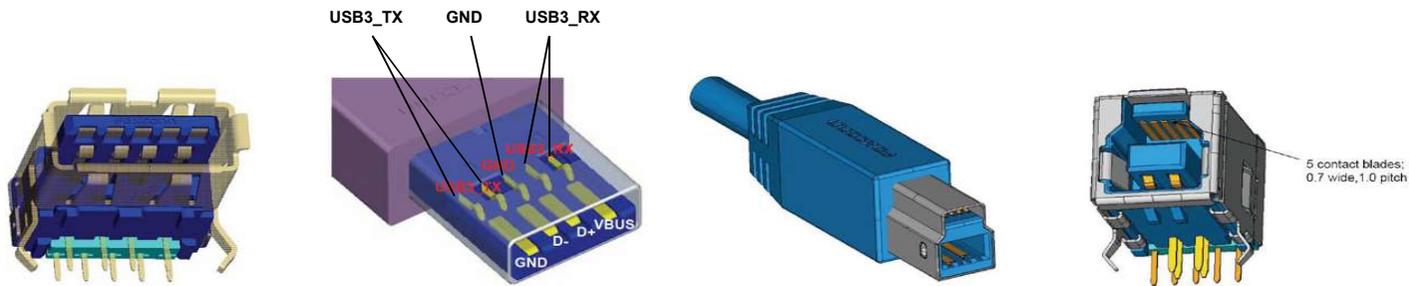
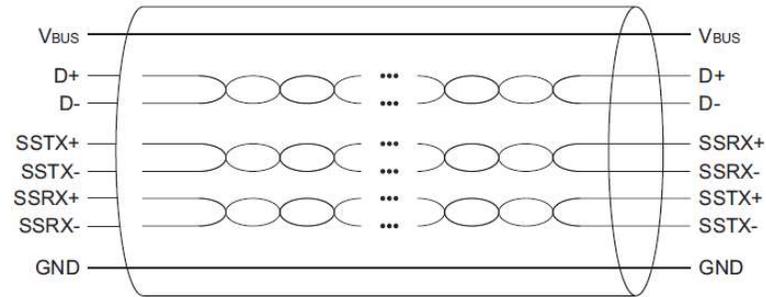
USB 3.0

heig-vd

- Electrical Interface

- 2 + 1 paires pour les données: 2 pour SuperSpeed + 1 pour USB 2.0
- 1 pair for power

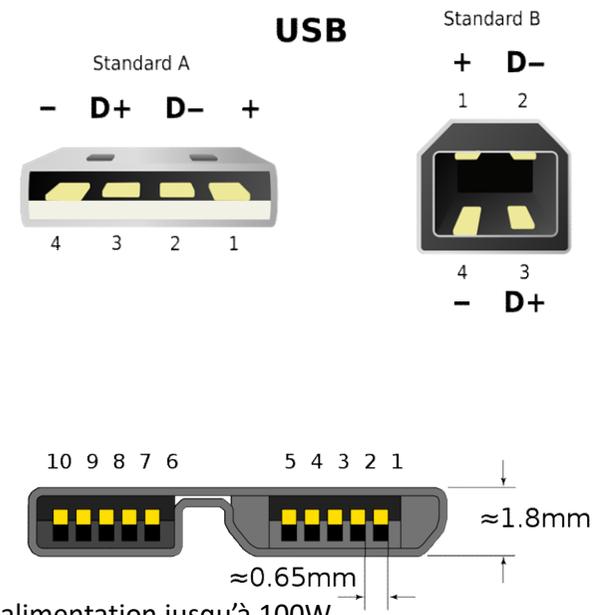
- Mechanical Interface



Source: www.usb.org

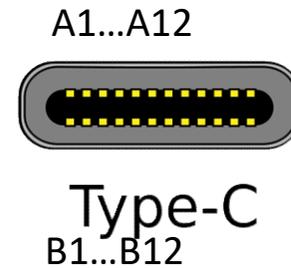
USB : couche physique

- De base USB 1.1 et USB 2.0
 - 1 paire différentielle (half-duplex) + alimentation
 - Alimentation +5V - 500 mA
 - Données (D-)
 - Données (D+)
 - Masse (GND)
- USB 3.0, USB 3.1
 - power line (VBUS)
 - USB2.0 D-
 - USB2.0 D+
 - USB OTG's ID line
 - GND
 - USB3.0 Sending Data line-
 - USB3.0 Sending Data line+
 - GND
 - USB3.0 Receiving Data line-
 - USB3.0 Receiving Data line+
- USB-C
 - Plug reversible USB 3.1 avec possibilité de video 4K et alimentation jusqu'à 100W



USB-C

L'USB Type-C s'inscrit surtout dans l'objectif de proposer un câble USB capable de transmettre des données, de la vidéo et du courant.



Pin	Nom	Description	Pin	Nom	Description
A1	GND	Ground return	B12	GND	Ground return
A2	SSTXp1	SuperSpeed differential pair #1, TX, positive	B11	SSRXp1	SuperSpeed differential pair #1, RX, positive
A3	SSTXn1	SuperSpeed differential pair #1, TX, negative	B10	SSRXn1	SuperSpeed differential pair #1, RX, negative
A4	V _{BUS}	Bus power	B9	V _{BUS}	Bus power
A5	CC1	Configuration channel	B8	SBU2	Sideband use (SBU)
A6	Dp1	USB 2.0 differential pair, position 1, positive	B7	Dn2	USB 2.0 differential pair, position 2, negative
A7	Dn1	USB 2.0 differential pair, position 1, negative	B6	Dp2	USB 2.0 differential pair, position 2, positive
A8	SBU1	Sideband use (SBU)	B5	CC2	Configuration channel
A9	V _{BUS}	Bus power	B4	V _{BUS}	Bus power
A10	SSRXn2	SuperSpeed differential pair #2, RX, negative	B3	SSTXn2	SuperSpeed differential pair #2, TX, negative
A11	SSRXp2	SuperSpeed differential pair #2, RX, positive	B2	SSTXp2	SuperSpeed differential pair #2, TX, positive
A12	GND	Ground return	B1	GND	Ground return

USB 2.0 differential pair connects only in one position; position 2 is not physically present in the plug