

# Architecture matérielle d'un ordinateur

## Bus de communication, carte mère, processeur et mémoire

Michel Salomon

IUT Nord Franche-Comté  
Département d'Informatique

# Concept de bus

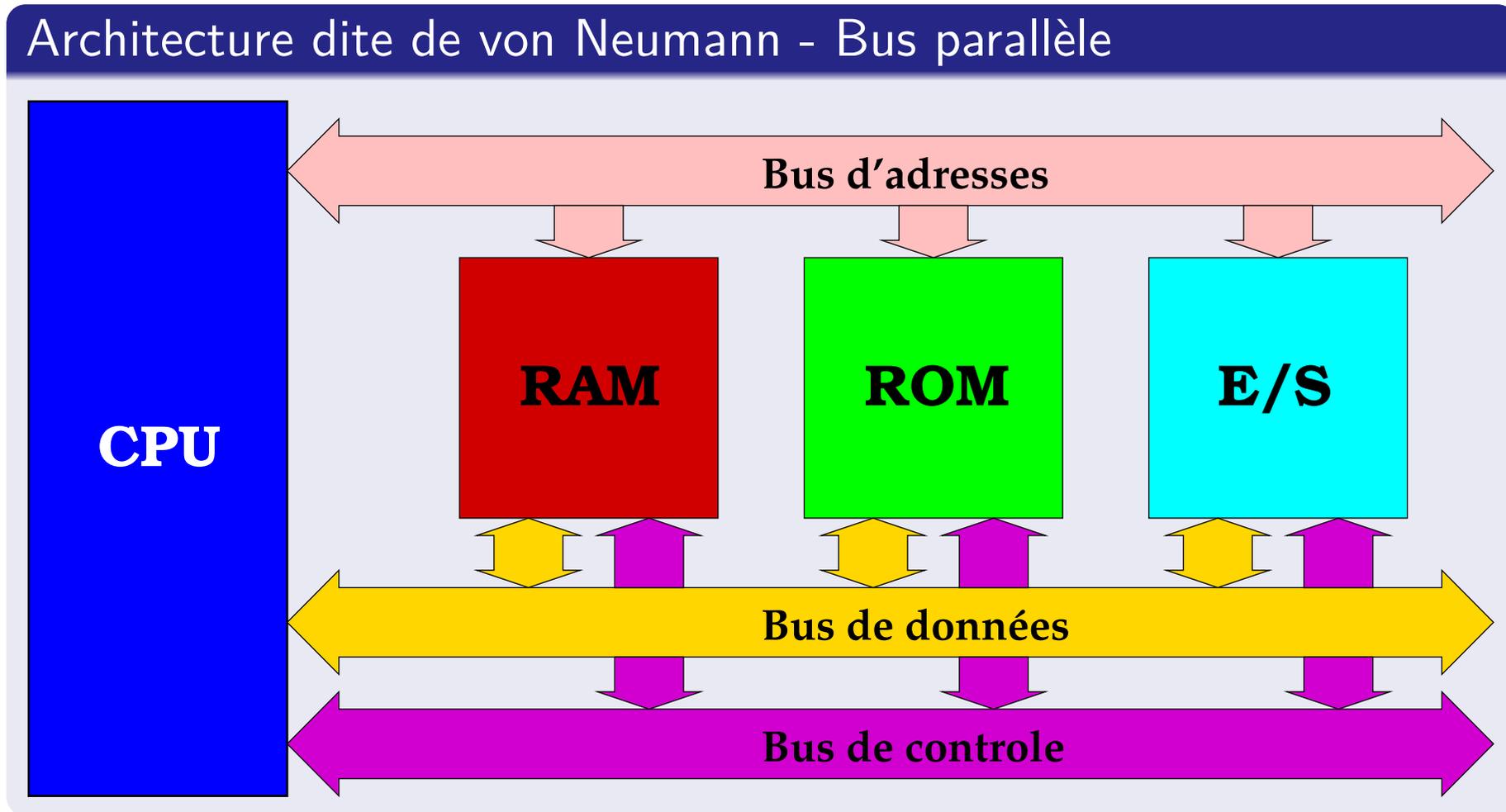
## Architecture des ordinateurs (von Neumann / Turing - 1945)

- On distingue 3 éléments :
  - ① l'**unité centrale de traitement** ou **processeur** (on utilise également les termes *microprocesseur* ou *Central Processing Unit*) ;
  - ② la **mémoire centrale** ou **principale** (contenant programmes et données - informations codées en binaire sous forme de bits 0 et 1) ;  
Mémoire vive (RAM) et mémoire morte (ROM)
  - ③ les **interfaces d'Entrées/Sorties** (liens entre le processeur et les périphériques externes/internes) ;
- reliés par des canaux de communication appelés **bus**

## Que peut transporter un bus de communication ?

- des **adresses** ;
- des **données** ;
- des **commandes** (ou instructions de contrôle)

# Concept de bus



Aujourd'hui le bus parallèle est "obsolète"

Avènement des **bus série** et **série / parallèle** (plusieurs liaisons série)

# Généralités

## Qu'est-ce qu'un bus de communication ?

- Un chemin électrique faisant communiquer ensemble
  - des composants électroniques
  - ou des périphériques électroniques
- Exemples
  - des composants de la carte mère  
→ processeur et barrettes mémoire, etc.
  - un composant et des périphériques internes ou externes  
→ carte graphique, disque, imprimante, etc.

## Citer quelques bus de communication

- *Universal Serial Bus* ;
- *Peripheral Component Interconnect Express* abrégé par **PCIe** ;
- *Serial Advanced Technology Attachment* ; *Serial-Attached SCSI* ;
- etc.

Généralement chaque bus existe en plusieurs versions / générations

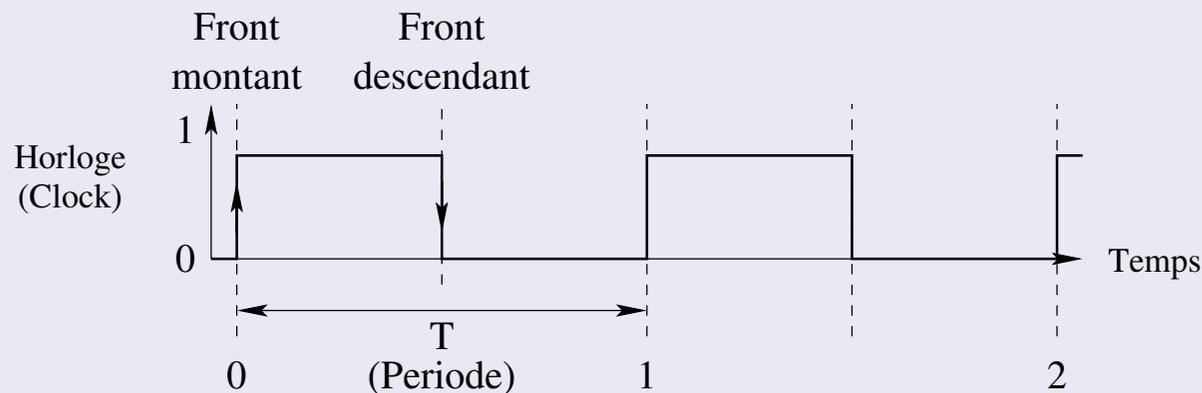
# Généralités

Un bus achemine des bits d'information entre composants / périphériques

- Un bus est composé d'un certain nombre de lignes
  - Chaque ligne transporte un bit à la fois via un signal numérique
  - Signal numérique  $\rightarrow$  grandeur discrète appartenant à  $\{0, 1\}$
- Les bits se suivent sur une même ligne

Synchronisation des éléments d'un ordinateur via un signal d'horloge

- Un quartz génère un signal d'horloge commun (*Clock* / CLK)
- Signal périodique comportant une alternance de 1 et de 0
- Chronogramme  $\rightarrow$  évolution du signal dans le temps



# Généralités

## Synchronisation des éléments d'un ordinateur via un signal d'horloge

- Cycle (ou battement) d'horloge → une période de durée  $T$
- Fréquence d'horloge → nbre de cycles par unité de temps
- Durée  $T$  d'un cycle → définie par la fréquence d'horloge  $f$

$$f \text{ (en Hz)} = \frac{1}{T \text{ (en secondes)}}$$

Exemple :  $f = 500 \text{ MHz} \rightarrow T = 2 \text{ ns}$  (nanosecondes)

- Signal utilisé pour obtenir de nombreuses fréquences réelles
  - Processeur, puces mémoire, etc.
  - Bus de communication

(souvent via un circuit appelé *Phase-Locked Loop*)

Un cycle d'horloge réelle permet de transporter un ou plusieurs bits

# Généralités

## Différents “types” de bits circulent dans un bus

- Un bus peut transporter 3 types de bits :
  - des bits de donnée  
→ l'information à acheminer ;
  - des bits d'adresse  
→ identifient l'origine / le destinataire de l'information ;
  - des bits de contrôle  
→ signaux de synchronisation, sens du transfert, etc.

## Types de bus de communication

- Parallèle (“obsolète”)  
→ transport séparé des bits (bus de donnée, adresse, contrôle)
- Série  
→ transport “unifié” (sérialisé) de tous les bits
- Série / parallèle → plusieurs transports sérialisés en parallèle

# Caractéristiques d'un bus / 1. Type et largeur

## Parallèle

- Unidirectionnel à l'alternance (*half-duplex*)  
→ transmission dans les 2 sens, alternativement
- Plusieurs lignes pour transporter les bits
- Cadencement sur une fréquence d'horloge commune  
→ à l'émetteur et au récepteur

## Série ⇒ 1 voie

- Une voie (*lane*) est composée de 2 liaisons séries  
→ 1 émission et 1 réception (ex. : 7 fils dans un câble SATA)
- Bidirectionnel simultané (*full-duplex*) (**pas tous les bus série**)  
→ transmission simultanée dans les 2 sens
- Transport d'un bit à la fois dans un sens  
→ une donnée sur plusieurs bits est sérialisée

# Caractéristiques d'un bus / 1. Type et largeur

## Série (suite)

- Pas de distinction entre bit de donnée, d'adresse, de contrôle  
→ bits regroupés dans des “paquets” qui sont transportés
- Fonctionne comme le transport de paquets dans les réseaux
- PCIe Gen 1.0 et 2.0 (PCI Express)
  - Utilisent un encodage dit 8b/10b
  - Pour envoyer 8 bits (1 octet) → émission de 10 bits
- PCIe Gen 3.0, 4.0 et 5.0
  - Utilisent un encodage dit 128b/130b
  - Pour envoyer 128 bits (16 octets) → émission de 130 bits
- PCIe Gen 6.0
  - Fin du 128b/130b utilisant le code NRZ (cf. Semestre 2)
  - Remplacé par un encodage avec code PAM4 et mode FLIT

## Série / parallèle ⇒ plusieurs voies

- Plusieurs liaisons séries unidirectionnelles en parallèle

# Caractéristiques d'un bus / 1. Type et largeur

## Série / parallèle (suite)

- Quelques exemples (il en existe d'autres...)
  - Intel
    - **Direct Media Interface** 3.0  $\approx$  PCIe  $\times$ 4 Gen 3.0
    - **Direct Media Interface** 4.0  $\approx$  PCIe  $\times$ 8 Gen 4.0
    - **Quick Path Interconnect** (20 voies)
  - AMD
    - **Unified Media Interface**  $\approx$  PCIe  $\times$ 4 Gen 2.0
    - **Infinity Fabric**  $\approx$  PCIe  $\times$ 4 Gen 3.0 puis Gen 4.0
      - Diffère du PCIe depuis la version Infinity Fabric 3.0
- Bus DMI et Infinity Fabric
  - Utilisés par les dernières générations de proc. Intel et AMD
  - Processeurs intégrant les contrôleurs mémoire, graphique, ...

Largeur  $\Rightarrow$  nombre de bits transférables simultanément dans un sens

- Bus parallèle  $\rightarrow$  largeur  $> 1$ ; Bus série  $\rightarrow$  largeur = 1
- Bus série / parallèle  $\rightarrow$  largeur = nbre de liaisons séries  $> 1$

# Caractéristiques d'un bus / 2. Nombre de Transferts par seconde

Basé sur la fréquence d'horloge réelle du bus ( $F_R$ )

Dépend du nombre de cycles d'horloge réelle du bus par seconde

- Au départ  
→ 1 transfert par cycle (front montant)
- **Double Data Rate (Dual pumped)**  
→ 2 transferts par cycle (fronts montant et descendant)
- **Quad Data Rate (Quad pumped)**  
→ 4 transferts par cycle (fronts montant et descendant, plus 2 fronts obtenus par déphasage de l'horloge) - “**obsolète**”

S'exprime de façon équivalente en Hertz

- Utilisation du terme fréquence d'horloge effective ( $F_E$ )
- d'où  $1 \text{ T/s} = 1 \text{ Hz}$

# Calcul du débit (unidirectionnel)

## Que représente le débit binaire ?

- Quantité de bits pouvant y transiter par unité de temps  
→ reflète l'efficacité du bus
- S'exprime en octets ou unité basée sur les octets par seconde  
→ Mo/s, Mio/s, etc.

## Attention !! Formule valable si 1 octet sur le bus = 8 bits

débit = (nbre de transferts par seconde  $\times$  largeur) / 8 octets/s

## Attention !!

- Usage abusif du terme bande passante à la place de débit
- Les constructeurs donnent souvent un débit bidirectionnel

# Calcul du débit (unidirectionnel) (suite)

## Exemple : débit du bus AGP 4× - ancêtre lointain du PCIe

- Caractéristiques
  - Type parallèle
  - Largeur de 32 bits
  - Bus QDR
  - Fréquence réelle  $F_R = 66,66$  MHz
- Calcul du débit (1 Mo = 1000 ko = 1000 × 1000 octets)
  - débit = bits/s
  - débit = octets/s
  - débit = Mo/s
  - débit = Mio/s

Souvent en Gigaoctets ou Mégaoctets par sec. (Go/s ou Mo/s)

# Calcul du débit (unidirectionnel) (suite)

## Exemple : débit du bus AGP 4× - ancêtre lointain du PCIe

- Caractéristiques
  - Type parallèle
  - Largeur de 32 bits
  - Bus QDR
  - Fréquence réelle  $F_R = 66,66$  MHz
- Calcul du débit (1 Mo = 1000 ko = 1000 × 1000 octets)
  - débit =  $((66660000 \times 4) \times 32) = 8532480000$  bits/s
  - débit =  $(66660000 \times 4 \times 32)/8 = 1066560000$  octets/s
  - débit =  $((((66660000 \times 16)/1000)/1000) = 1066,56$  Mo/s
  - débit =  $((((66660000 \times 16)/1024)/1024) = 1017,15$  Mio/s

Souvent en Gigaoctets ou Mégaoctets par sec. (Go/s ou Mo/s)

# Bus parallèle *versus* bus série

## Un avantage, mais de nombreux inconvénients

- Avantage d'un bus parallèle

En théorie, plus un bus parallèle est large,  
plus il peut transporter de données

- Défauts d'un bus parallèle

- Encombrant et coûteux → prend plus de place
- Fabrication délicate → longueur des lignes identiques, sinon désynchronisation des transmissions
- Interférences électromagnétiques susceptibles de perturber les signaux → compétitif sur de petites distances
- Pénalisant car généralement partagé → défaut majeur

## Bus série ⇒ corrige les défauts du bus parallèle

Requiert des composants supplémentaires car les données sont acheminées sous forme de “paquets” (comme dans un réseau)

# Exemples de bus actuels

## Bus PCI Express (PCI-E ou PCIe) - Périphériques internes

- Gen. 3.0 (2010), 4.0 (2017), 5.0 (2019), 6.0 (2022), 7.0 (2025), 8.0 ?
- Caractéristiques
  - Type série (série/parallèle si plusieurs voies)
  - Connecteur de taille variable (1, 2, 4, 8, 16 ou 32 voies)
  - Voie composée de 2 liaisons : émission et réception
  - *Bidirectionnel simultané, avec un contrôle de l'intégrité*
- Débits
  - PCIe Gen. 3.0 - 8 GT/s
    - $1\times \Rightarrow 984,6 \text{ Mo/s}$ ; ...;  $16\times \Rightarrow 15,754 \text{ Go/s}$
  - PCIe Gen. 4.0 - 16 GT/s
    - $1\times \Rightarrow 1969,23 \text{ Mo/s}$ ; ...;  $16\times \Rightarrow 31,504 \text{ Go/s}$
  - PCIe Gen. 5.0 - 32 GT/s
    - $1\times \Rightarrow 3938,46 \text{ Mo/s}$ ; ...;  $16\times \Rightarrow 63,015 \text{ Go/s}$
  - PCIe Gen. 6.0 - 64 GT/s

# Exemples de bus actuels

## Bus USB (*Universal Serial Bus*) - Unifie les connexions bas débit

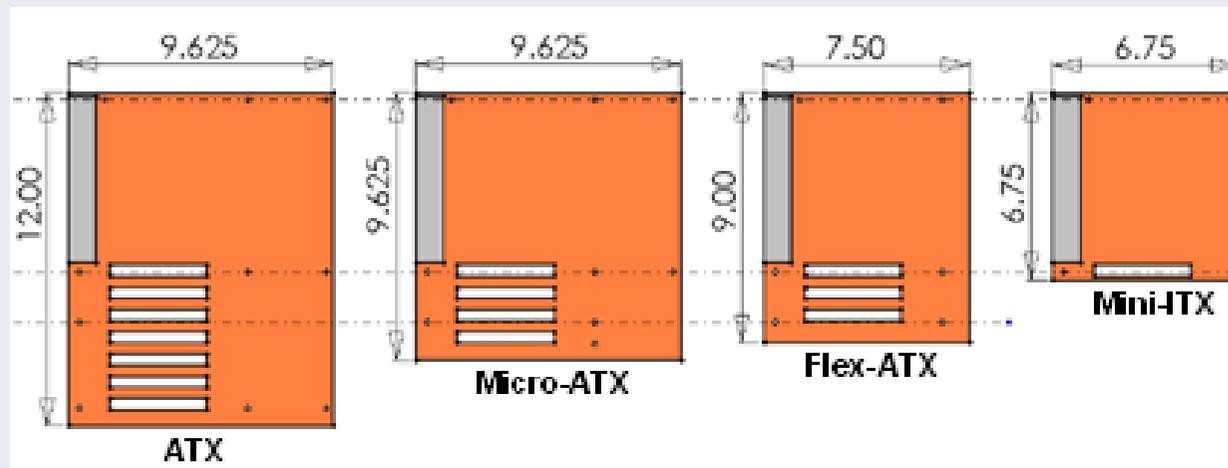
- Compaq, Intel, IBM, Microsoft, NEC & al. 1996
- Débits
  - Norme 1.1 / FULL-SPEED → débit de 1,5 Mo/s  $\Leftrightarrow$  12 Mbit/s
  - Norme 2.0 / HI-SPEED → débit de 60 Mo/s  $\Leftrightarrow$  480 Mbit/s
  - Norme 3.0 / 3.1 Gen. 1 / SUPER-SPEED (2008)  
→ débit de 5 Gbit/s (8b/10b)
  - Norme 3.1 / 3.1 Gen. 2 / SUPER-SPEED+ (2013)  
→ débit de 10 Gbit/s (128b/132b) - Type-A,B,C et Micro-A,B,AB
  - Norme 3.2 Gen. 2x2 / SUPER-SPEED++ (2017)  
→ débit de 20 Gbit/s (128b/132b) - Type-C uniquement  
(Gen. 1x1 = 3.1 Gen. 1, Gen. 1x2, Gen. 2x1 = 3.1 Gen. 2)
  - Norme 4 Gen. 2x2 / 4 Gen. 3x2 (Thunderbolt 3 / 4) (2019)  
→ débit de 20 Gbit/s (64b/66b) pour Gen. 2x2 - Type-C  
→ débit de 40 Gbit/s (128b/132b) pour Gen. 3x2
- Thunderbolt 5 (2024 - 120 Gbit/s)  $\neq$  USB Norme 5

# Carte mère et chipset

## Carte mère

- Regroupe plusieurs éléments
  - Processeur
  - Mémoire
  - Ports
  - etc..
- Caractérisée par son chipset (lié au type de processeur)
- **Format** ATX, Mini-ATX, Micro-ATX, Flex-ATX, Mini-ITX, etc.

## Les différents formats de carte mère (unité = *inch* / pouce)



# Carte mère et chipset

## Chipset ou *jeu de composants*

Contrôleur(s) chargé(s) d'aiguiller les données entre les différents bus de communication

## Rôle et évolution du chipset

- Organise les transferts de données au sein de l'ordinateur
- Historiquement architecture à deux puces
  - 1 Pont Nord - *Northbridge* (**GMCH**)
  - 2 Pont Sud - *Southbridge* (**ICH**)
- Passage à une architecture à une seule puce (**PCH**)

Disparition du pont nord suite à l'intégration des contrôleurs mémoire et graphique dans le processeur
- Intégration "complète" dans le processeur (**System on Chip**)
  - Plateforme mobile / embarquée (*Desktop*) au départ, mais...

# Chipset (ou *jeu de composants*)

## Pont Nord ou **Graphic Memory Controller Hub**

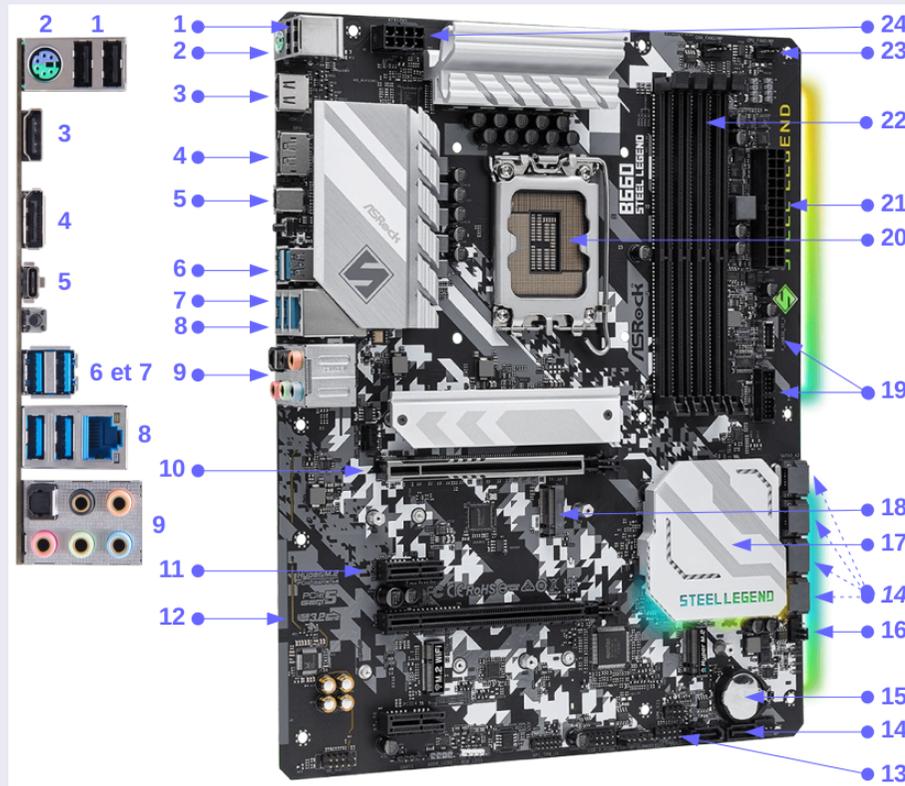
- Communications entre éléments requérant un grand débit
  - Processeur
  - Mémoire
    - Contrôleur mémoire intégré par les processeurs actuels
  - Carte(s) graphique(s) ou encore puce graphique intégrée
    - Contrôleur graphique intégré par certains processeurs actuels chez AMD et Intel Core iX (depuis 2<sup>e</sup> gen.), mais pas tous
  - Autres périphériques PCI Express

## Pont Sud ou **I/O Controller Hub**

- Com. avec les éléments qui ont un débit plus réduit
  - Anciens ports PATA (ou IDE), parallèle, série
  - PCI Express, SATA, USB, etc.

# Architecture d'une carte mère

Carte pour proc. Intel - Socket 1700 / V - à une puce ou **Platform CH**

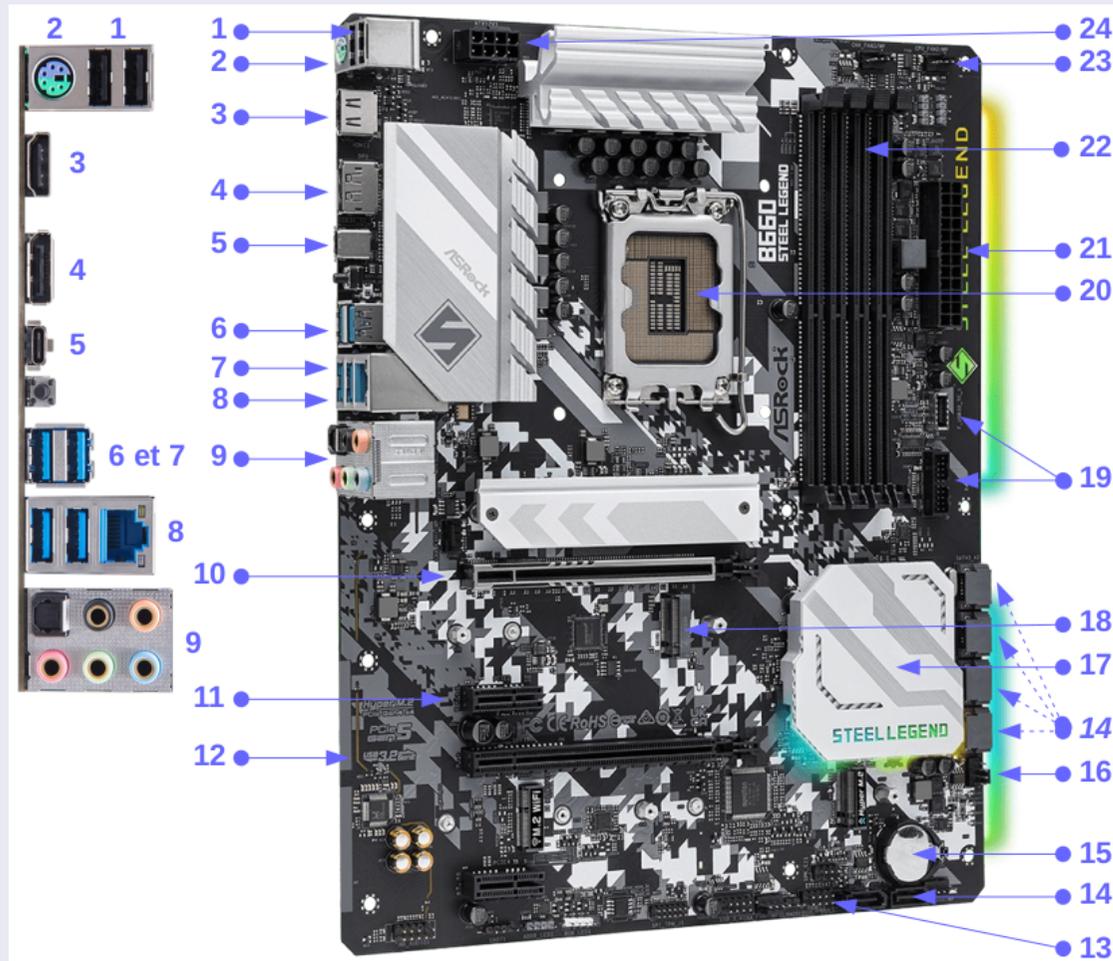


Bus de communication entre processeur et chipset

- Intel → bus série/parallèle Ultra Path Interconnect, Quick PI, *DMI Flexible Interface Display* pour lier cœur graphique et chipset
- AMD → bus série/parallèle *UMI* et *Infinity Fabric*

# Architecture d'une carte mère pour processeurs Intel

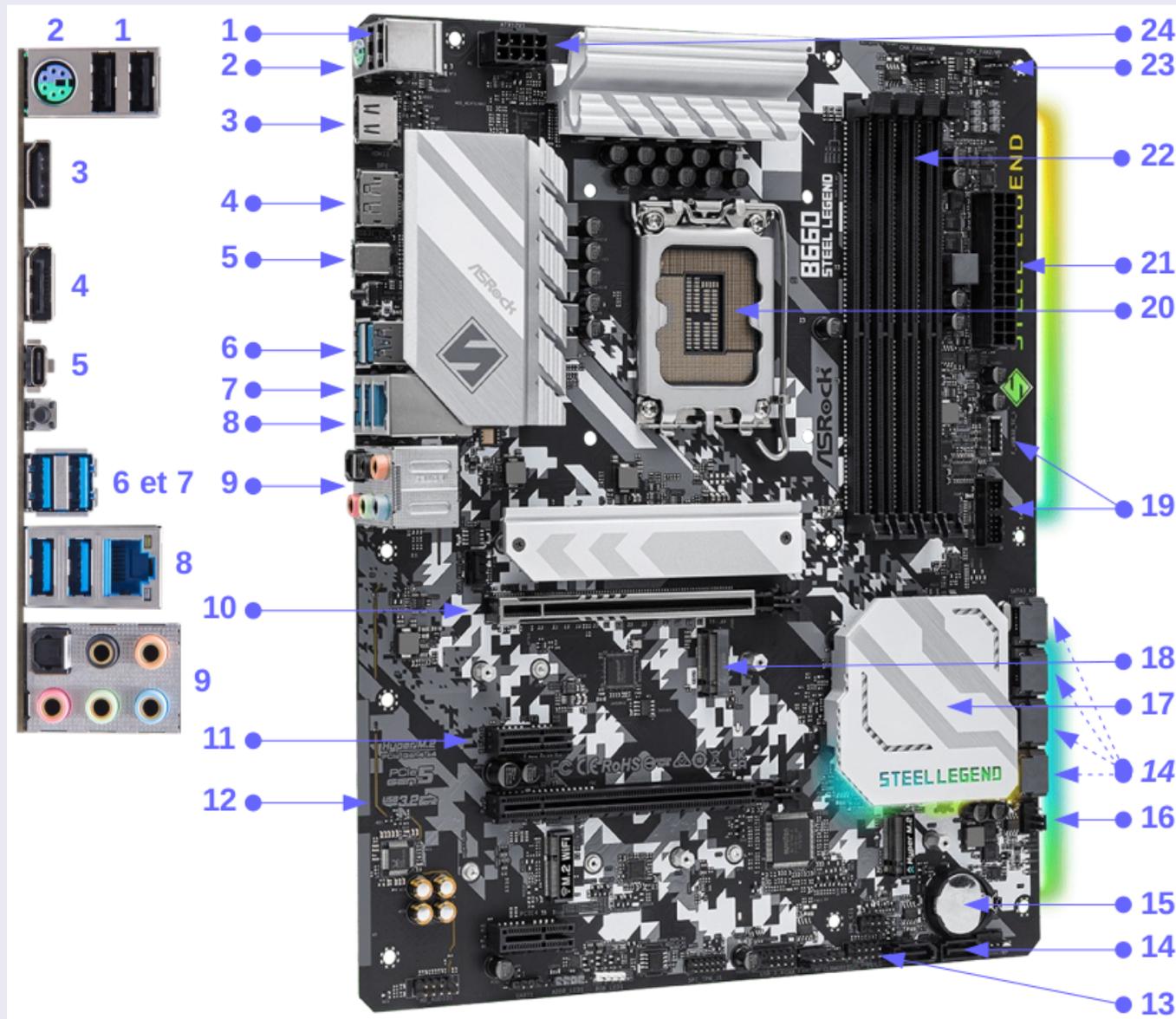
Carte pour proc. Intel - Socket 1700 / V - à une puce ou Platform CH



- Lien fabricant de la carte mère - CPU socket (LGA / PGA)
- Lien manuel de la carte à base du chipset B660

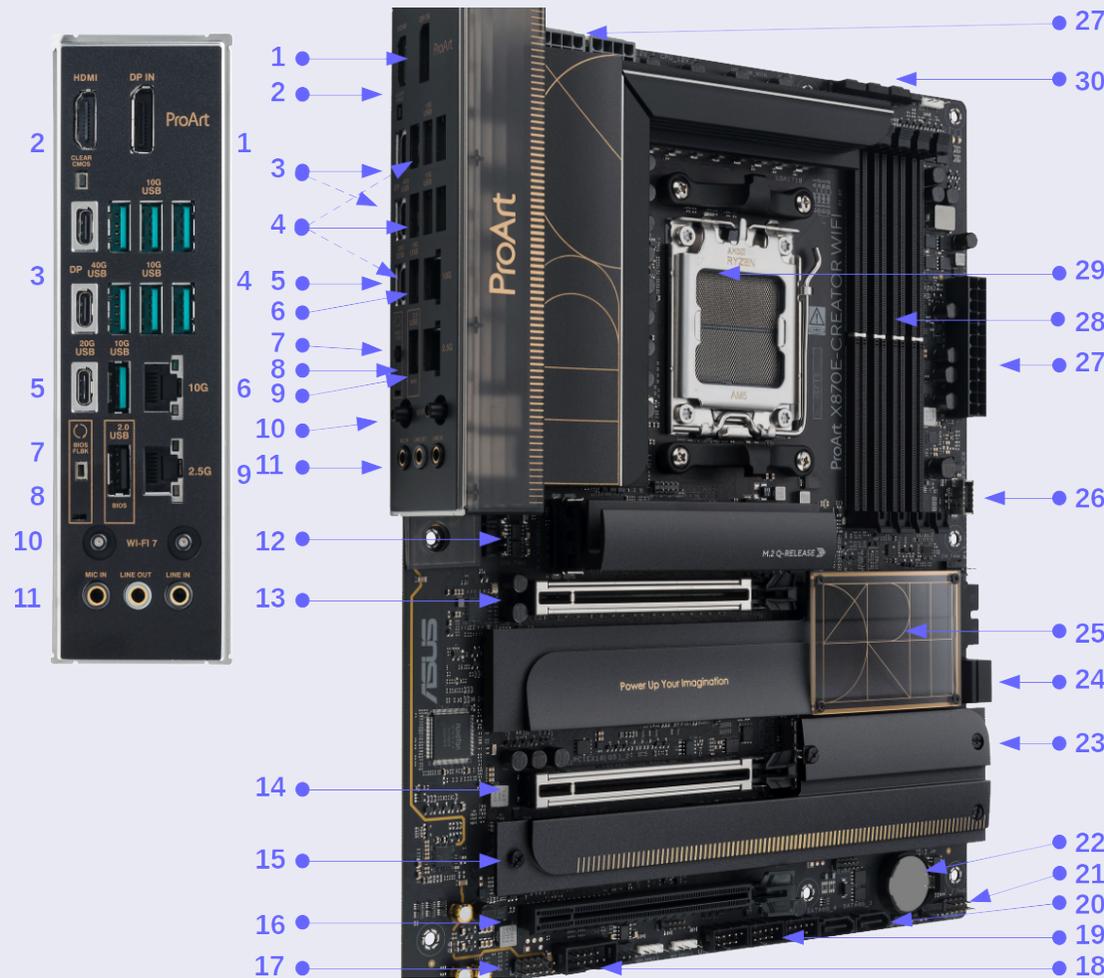
# Architecture d'une carte mère pour processeurs Intel

## Quizz - Carte mère pour proc. Intel - Socket 1700 / V



# Architecture d'une carte mère pour processeurs AMD

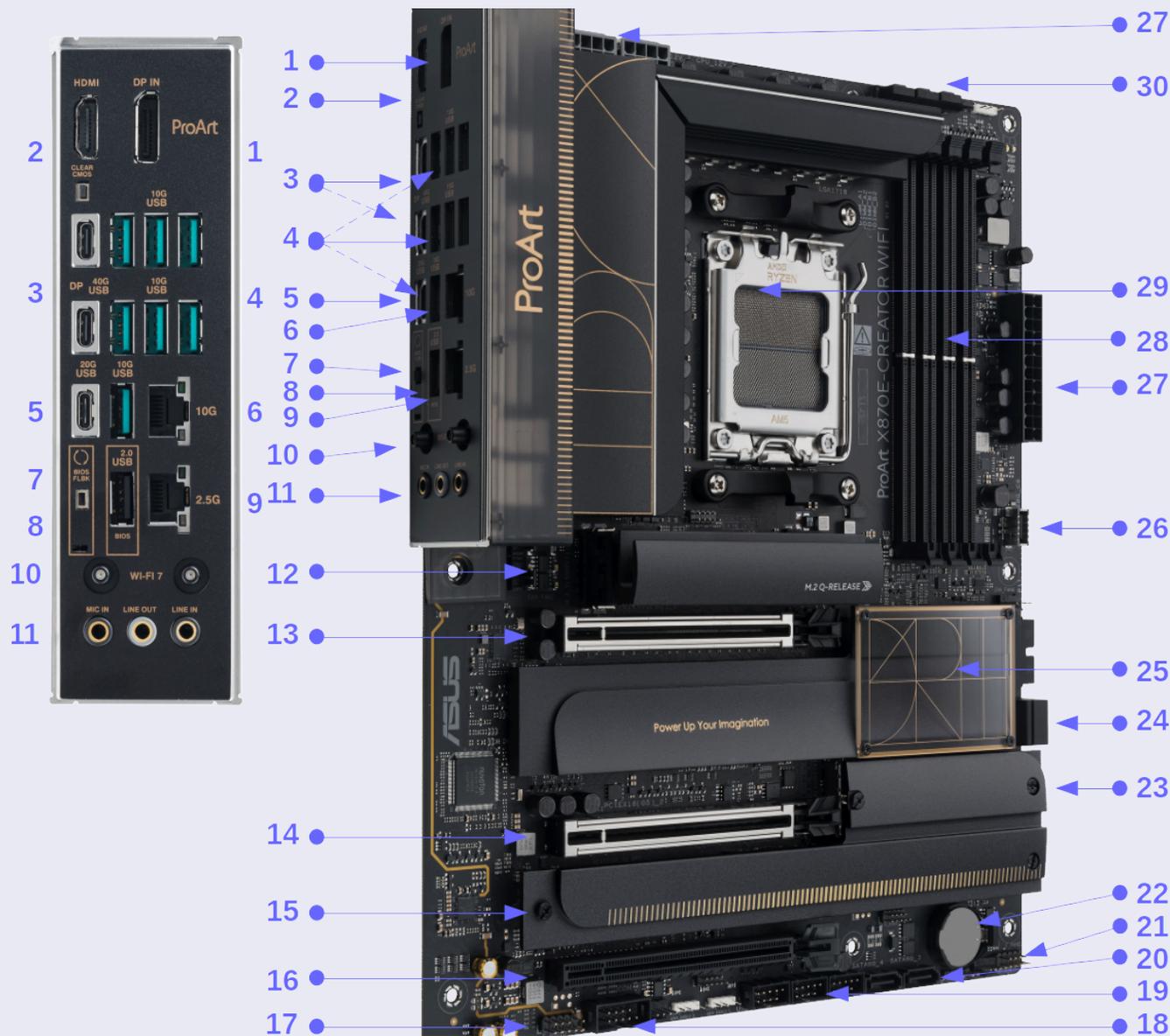
Carte pour proc. AMD - Socket AM5 - à une puce ou *Platform CH*



- Lien fabricant de la carte mère - CPU socket (LGA / PGA)
- Lien manuel de la carte à base du chipset X870E

# Architecture d'une carte mère pour processeurs AMD

## Quizz - Carte mère pour proc. AMD - Socket AM5



# Principaux fondateurs de processeurs : AMD et Intel

## Distinction ordinateur personnel / station de travail et serveur

- Ordinateur de type PC
  - Gamme AMD - Sockets
    - AM4 : Ryzen 3, 5, 7, 9 (Zen+, Zen 2 et Zen 3)
    - sWRX8 : Ryzen Threadripper Pro (Zen 3)
    - **AM5** : Ryzen 5, 7, 9, Pro (Zen 4 et 5) - PCIe Gen. 5 / DDR5
    - **sTR5** : Ryzen Threadripper (Zen 4 et 5)
  - Gamme Intel - Sockets
    - ~~1151 / H4~~ : Celeron, Pentium Gxxx0, Core i3, i5, i7, i9  
(Kaby Lake, Coffee Lake *Refresh* → avec contrôleur graphique)
    - 1200 / H5 : Celeron Gxxx0, Core i3, i5, i7, i9  
(Comet Lake, Rocket Lake → avec contrôleur graphique)
    - 2066 / R4 : Core i7 et i9 (grand nombre de cœurs)  
(Kaby Lake-X, Cascade Lake-X → sans contrôleur graphique)
    - **1700 / V** : Celeron, Pentium Gxxx0, Core i3, i5, i7, i9  
(Alder Lake, Raptor Lake)
    - **1851 / V1** : Core Ultra 5, 7 et 9

# Principaux fondateurs de processeurs : AMD et Intel

## Distinction ordinateur personnel / station de travail et serveur

- Station de travail - Serveur

Multi-processeur, grand nombre de cœurs,  
mémoire cache plus importante, ...

- Gamme AMD → Epyc (Genoa/Bergamo/Turin/Turin Dense → Zen 4/4c/5/5c)
  - Processeurs ayant de 16 à 128 / 192 cœurs
  - Configuration à 1 ou 2 processeur(s)
  - Socket SP3, SP5 / SP6
- Gamme Intel → Xeon
  - Déclinaison de chaque génération de processeur Intel Core (Skylake, Kaby Lake, Coffee Lake, Comet Lake, Cascade Lake, Cooper Lake, Ice Lake, Rocket Lake, Sapphir et Emerald Rapids)
  - Sockets 1151, 1200, 2066, 3647, 4189, 4677, 4710 et 7529

## Quels processeurs dans les smartphones et tablettes ?

- Architectures ARM SoC (CPU+GPU+DSP+...) vendues sous licence
- Chaque fabricant personnalise, voire mixe, des composants

# Exercices sur les bus de communication

## ① Bus Intel DMI 4.0

### ① Caractéristiques du Core Ultra 7 265K - Socket 1851 - Q4'24

- Fréquence d'horloge réelle du proc. = 100 MHz ; 20 cœurs ;
- spécification des modes Turbo (coefficients)
  - Turbo Boost Max Technology 3.0 (TBMT3) = 16 ;
  - P-core / E-core Max Turbo Freq. (TBT2) = 15 / 13
- bus processeur DMI 4.0 (8 voies sur le proc. considéré)
  - fréquence d'horloge réelle du bus = 8 GHz ;
  - opérant en DDR, 8 voies et encodage 128b/130b ;
- coefficient Xteur du proc. (P-core) = 39 (E-core = 33) ;
- largeur du bus d'adresses = 46 bits (mémoire physique) et 48 bits (mémoire virtuelle).

### ② Calculer

- ① Le nombre de transferts par seconde du bus pour 1 voie ;
- ② le débit unidirec. du bus proc. (8 voies) en Mo/s, Mio/s, Go/s ;
- ③ les fréquences de fonctionnement du processeur (Base Freq.) ;
- ④ les fréquences max. grâce aux Turbo TBT2 et TBMT3 ;
- ⑤ la taille de la mémoire virtuelle adressable.

# Exercices sur les bus de communication

## ② Bus mémoire d'une barrette de type DDR5

### ① Comment calculer la fréquence réelle du bus

- À partir de la fréquence d'horloge réelle des puces DRAM
  - DDR  $\rightarrow F_R = F_{DRAM}$  ;
  - DDR2  $\rightarrow F_R = 2 \times F_{DRAM}$  ;
  - DDR3  $\rightarrow F_R = 4 \times F_{DRAM}$  ;
  - DDR4  $\rightarrow F_R = 8 \times F_{DRAM}$  ;
  - DDR5  $\rightarrow F_R = 16 \times F_{DRAM}$ .
- obtenue à partir de la fréquence d'horloge réelle du processeur.  
 Dans le cas de la DDR5, on a  $F_{DRAM}$  qui est un multiple de  $\frac{1}{15}$  GHz,  
 soit :  $F_{DRAM} = \frac{\lambda}{15}$  GHz =  $\left(\frac{\lambda}{15} \times 10^9\right)$  Hz où  $\lambda \geq 1$  définit le débit.

### ② Caractéristiques de la barrette étudiée

- Coefficient  $\lambda = 3$  ;
- largeur du bus de données = 64 bits

### ③ Calculer

- ① Les fréquences d'horloge réelle et effective du bus mémoire ;
- ② le débit de la barrette mémoire en Go/s et Gio/s ;
- ③ en déduire la désignation de la barrette.

# Exercices sur les bus de communication

## ③ Bus PCIe Gen. 2.0

- ① Caractéristiques du bus série  $1 \times$  (à une voie)
  - Fréquence d'horloge "réelle" du bus = 2,5 GHz
  - bus pouvant être vu comme opérant en DDR ;
  - pour 8 bits de données, 10 bits envoyés (encodage 8b/10b)
- ② Calculer
  - ① Le nombre de transferts par seconde ;
  - ② le débit du bus (largeur = 1 bit) ;
  - ③ la perte de débit unidirectionnel due aux bits supplémentaires ;
  - ④ à titre de comparaison, donner le pourcentage de perte pour l'encodage 128b/130b.

## Vision "DDR" du PCI Express : simplification qui n'est pas la réalité

- Bus réel isosynchrone → pas de signal d'horloge explicite
- Aussi "équivalent" à un bus avec  $F_R = 16$  GHz, ni DDR, ni QDR

# Unités de mesure

Le bit (0 ou 1) est la plus petite unité de mesure

## Unités de mesure basées sur l'octet

- Principalement utilisées pour parler du stockage de données
- Deux types d'unités : en puissances de 2 ; en puissances de 10

<i>Puissances de 2</i>			
1 Kibioctet	Kio ou KiB	1024 octets	$2^{10}$
1 Mébioctet	Mio ou MiB	1024 Kio	$2^{20}$
1 Gibioctet	Gio ou GiB	1024 Mio	$2^{30}$
1 Tébioctet	Tio ou TiB	1024 Gio	$2^{40}$
1 Pébioctet	Pio ou PiB	1024 Tio	$2^{50}$
<i>Puissances de 10</i>			
1 kilooctet	ko ou kB	1000 octets	$10^3$
1 Mégaoctet	Mo ou MB	1000 ko	$10^6$
1 Gigaoctet	Go ou GB	1000 Mo	$10^9$
1 Téraoctet	To ou TB	1000 Go	$10^{12}$
1 Pentaoctet	Po ou PB	1000 To	$10^{15}$

- 1 Kibioctet = 1 "Kilo binaire octet"

# Exercices sur les bus de communication - Corrections

## ① Bus Intel DMI 4.0

- ① Nombre de transferts par seconde ( $N$ ) pour 1 voie égal à

$$\begin{aligned} N &= \text{fréquence d'horloge réelle du bus (Hz)} \times 2 \text{ (car DDR)} \\ &= (8 \times 10^9) \times 2 = 16 \times 10^9 \text{ T/s} = 16 \text{ GT/s} \end{aligned}$$

- ② Débit (unidirectionnel) du bus processeur

$$\begin{aligned} \text{débit} &= (\text{nombre de transferts par seconde} \times \text{largeur}) \text{ bits/s (sans encodage)} \\ &= ((16 \times 10^9) \times 8) \text{ bits/s} = 128 \text{ Gbit/s (sans encodage)} \\ &= \left( ((16 \times 10^9) \times 8) \times \frac{128}{130} \right) / 8 = 15753846153,8 \text{ octets/s} \\ &= 15753846,1538 \text{ ko/s} (= 15384615,3846 \text{ Kio/s}) \\ &\approx 15753,85 \text{ Mo/s} (\approx 15024,04 \text{ Mio/s}) \\ &\approx 15,75 \text{ Go/s} \end{aligned}$$

- Remarques (sur le calcul du débit)
  - Largeur égale à 8 car 8 voies.
  - Multiplication par  $\frac{128}{130}$  car encodage 128b/130b.

# Exercices sur les bus de communication - Corrections

## ① Bus Intel DMI 4.0 (suite)

### ③ Fréquences de fonctionnement du processeur ( $F_P$ et $F_E$ )

$$\begin{aligned}
 F_P &= \text{fréq. réelle du proc. (Hz)} \times \text{coeff. multiplicateur P-cores} \\
 &= (100 \times 10^6) \times 39 = 3900 \times 10^6 = 3900 \text{ MHz} = 3,9 \text{ GHz} \\
 F_E &= (100 \times 10^6) \times 33 = 3300 \times 10^6 = 3300 \text{ MHz} = 3,3 \text{ GHz}
 \end{aligned}$$

### ④ Fréquences max. grâce aux Turbo TBT2 et TBMT3 (en MHz)

- Max Turbo Frequency ( $F_{M2P}$  et  $F_{M2E}$ )

$$\begin{aligned}
 F_{M2P} &= 3900 + (15 \times 100) = 5400 \text{ MHz} = 5,4 \text{ GHz} \\
 F_{M2E} &= 3300 + (13 \times 100) = 4600 \text{ MHz} = 4,6 \text{ GHz}
 \end{aligned}$$

- Turbo Boost Max Technology 3.0 ( $F_{M3}$ )

$$F_{M3} = 3900 + (16 \times 100) = 5500 \text{ MHz} = 5,5 \text{ GHz}$$

### ⑤ Taille de la mémoire physique adressable

- Mémoire vive  $\rightarrow$  “bibliothèque” dont chaque emplacement, repéré par une une adresse, permet de stocker un octet ;
- Une adresse = un nombre binaire  $\rightarrow$  il y a autant d’adresses que de nombres binaires représentables

# Exercices sur les bus de communication - Corrections

## ① Bus Intel DMI 4.0 (suite)

### ⑤ Taille de la mémoire virtuelle adressable (suite)

- Nombre d'adresses défini par le nombre de bits utilisés :
  - 1 bits  $\rightarrow 2^1 = 2$  adresses possibles  $\Rightarrow 0$  et  $1$  ;
  - 2 bits  $\rightarrow 2^2 = 4$  adresses possibles  $\Rightarrow 00, 01, 10$  et  $11$  ;
  - 3 bits  $\rightarrow 2^3 = 8$  adresses possibles  $\Rightarrow 000, 001, \dots, 110, 111$  ;
  - etc. ;
  - 48 bits  $\rightarrow 2^{46}$  adresses virtuelles.
- Taille de la mémoire virtuelle adressable =  $2^{48}$  octets  
(une adresse par octet)
  - Soit 256 Tio en unités en puissances de 2 ;
  - Soit  $\approx 281,475$  To en puissances de 10.
- *Remarques* :
  - Généralement un SE ne gère pas autant de mémoire ;
  - Habituellement c'est de l'ordre des Gio ou quelques dizaines de Tio avec dans ce cas plusieurs processeurs ;
  - Les périphériques se voient également attribués des adresses.