

تكملة عن ال Buses ومعايير الأداء

د. خولة العلي

محتوى مجاني غير مخصص للبيع التجاري

بنيان الحاسوب

22/05/2022

RB Informatics;

Bus Configuration

لدينا طريقتين لبناء النواقل:

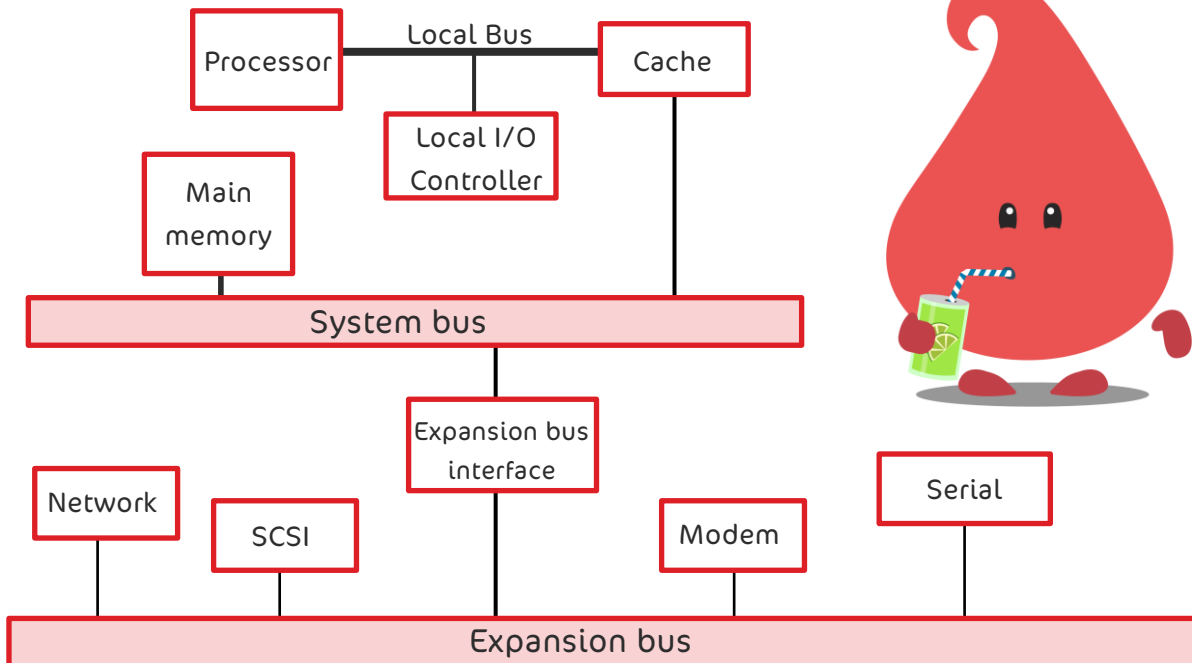
البنيان التقليدي للنقل (Traditional)

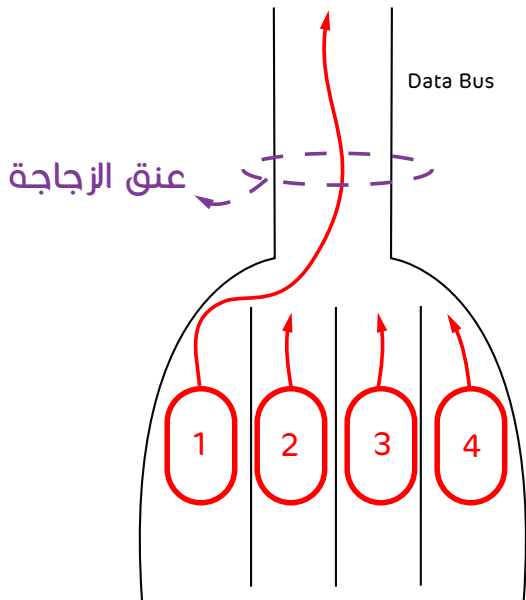
البنيان عالي الأداء (High-performance)

1. Traditional Bus Architecture

إذا كان لدينا عدد كبير من الأجهزة مشبوكة على النواقل وكل واحد من هذه الأجهزة يلزمنا باستخدام ناقل معين (مهما كان نوعه) فسيتم حرق الكثير من الوقت في الانتظار (كل جهاز سينتظر الذي قبله للانتهاء) وسنواجه في هذه الحالة مشكلة عنق الزجاجة.

مثال: لنفترض أنه لدينا شارع عريض فيه أربع أقسام (وكل قسم فيه سيارة وهذه السيارات تمثل الأجهزة ولها نفس الوجهة، في نهاية هذا الشارع لدينا محطة (مثلاً DataBus).





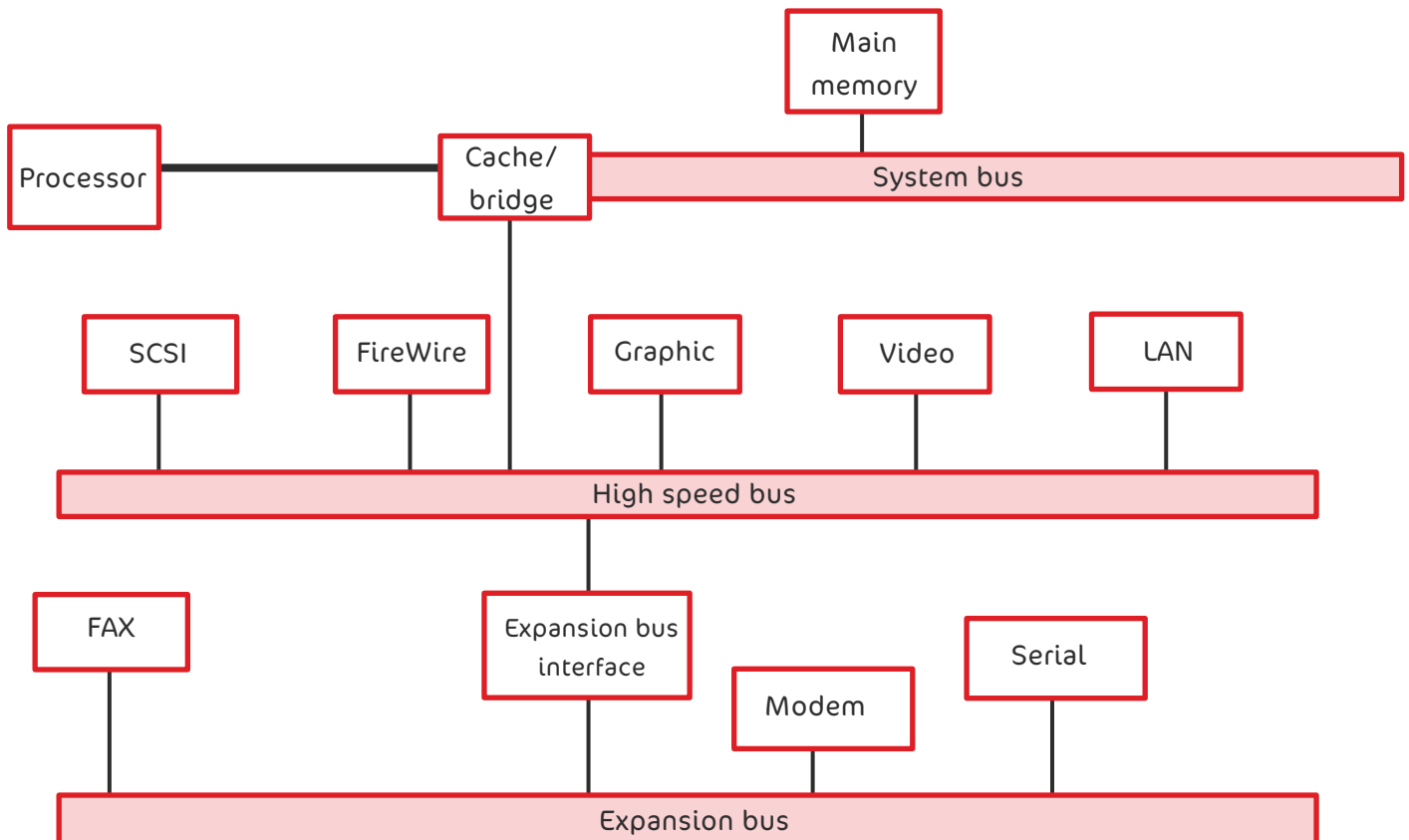
ولكن بعد قطع مسافة معينة ضاق الشارع حتى أصبح يتسع لسيارة واحدة فقط، فهنا كل سيارة ستنتظر سابقتها للانتهاء من المحطة (Data Bus) وسيصبح لدينا تأخير. ولكن تم حل هذه المشكلة باستخدام البنية الثاني.

2. High-performance architecture

هنا تتم عملية إحضار المعلومات بشكل أكبر ← أسرع يكون الـ System Bus فقط مع الذاكرة الرئيسية Main Memory وتكون المعلومات قريبة جداً من المعالج باستخدام الـ Bridge/Cache الموجود لدينا، وأيضا High-speed Bus للوحدات الطرفية التي تحتاج سرعات عالية ومعدل نقل بيانات عالي.

سؤال هام: ما هو أهم مسرى في الحاسوب؟

■ هو المسرى System Bus والذي يعمل على ربط وحدة المعالجة والذاكر ووحدات الإدخال والإخراج، كما يقوم بوظيفة الـ Data Bus لنقل المعطيات و الـ Address Bus لتحديد العناوين و الـ Control Bus لإصدار إشارات التحكم، و قد يعد عرضة لحدوث اختناق لمرور البيانات او ما يعرف بـ "عنق الزجاجة" بسبب كثافة كمية البيانات المتدفقة عبره.



ملاحظة: ذاكرة cache تخفف ضغط عن الذاكرة الرئيسية لأنها تقنية عتادية لتسريع الحاسب.

- والآن لتتعرف على العناصر اللازمة لتصميم أي ناقل.

Elements of Bus Design

1. Type

- المخصص Dedicated: يكون مخصص لوظيفة معينة أو لمجموعة فرعية من الأجهزة يسرف في الحجم والكلفة.
- مضمم Multiplexed: يعمل أكثر من وظيفة (يستطيع حمل عنوان وبيانات).
- فوائده: عدد خطوط أقل (مساري أقل) كونها تستخدم لأكثر من وظيفة والذي يوفر في الحجم والكلفة.
- سيئاته: الحاجة إلى دارات معقدة، وانخفاض الأداء بسبب مشكلة التزاحم.

2. طريقة التحكم Method of arbitration

- مركزية Centralized: (جزء واحد يتحكم في الناقل).
- توزيعية Distributed: (أكثر من جزء يتحكم في الناقل).

3. Timing

- synchronous متزامن
- Asynchronous غير متزامن

4. Bus width

- Address الحد الأقصى الذي نستطيع استخدامه من الذاكرة.
- Data مقدار البيانات التي نستطيع نقلها.

5. نوع البيانات المنقولة Data Transfer Type

- Read
- Write
- Read-modify-write
- Read-after-write
- Block



ملاحظة: استخدامية الحاسوب هي

الزمن الذي يكون فيه المعالج في حالة عمل, كلما كانت كبيرة كلما كانت أفضل (لأن زمن خمول المعالج في هذه الحالة يكون أقل).

التوقيت المتزامن للعمليات التي سيقوم بها الناقل

هنا أي عملية سوف تتم على الناقل تتحدد بواسطة clock (هي عبارة عن 1,0) وتقسم إلى clock cycles.

■ ماذا يحدث مع كل دورة؟

■ مع بداية الدورة تصدر الـ status signal

■ ومع بداية T_1 يصبح العنوان valid على Address Bus أي بما أن العنوان جاهز فسيمر إلى الـ Bus فيتم تفعيل الـ Address enabler.

■ وأي واحد من الـ Components يمكنه قراءة العنوان ومع نهاية T_1 ينتهي تفعيل العنوان.

■ ولدينا عملية Read Cycle أي الـ Microprocessor سوف يقرأ فيها معلومة من الذاكرة.

■ في بداية T_2 (Clock Cycle 2) يتم إطلاق إشارة القراءة لإعلام جميع المكونات (components) أنه سيتم القراءة من الذاكرة وذلك بناءً على العنوان الذي يحمله الـ Bus (Address) كان فعالاً طوال هذه المدة.

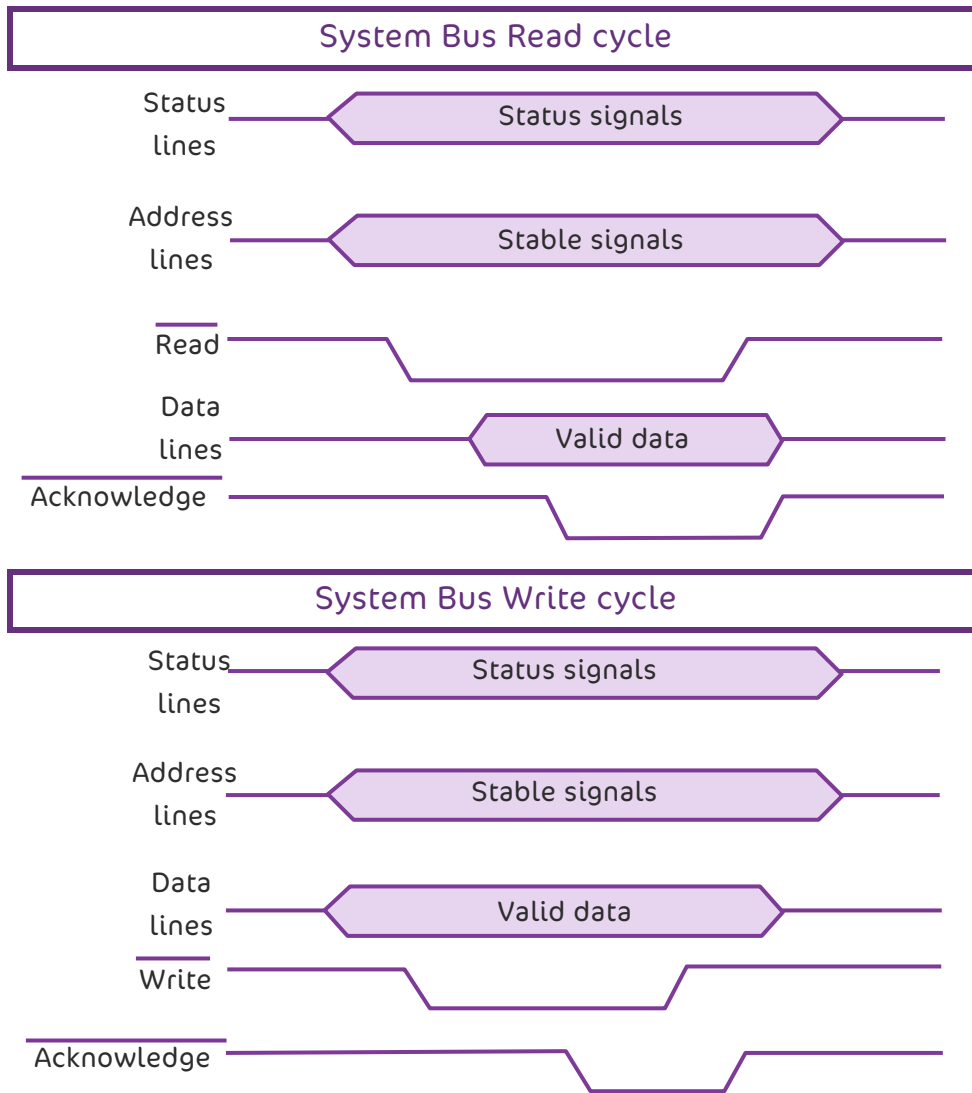
■ في بداية T_3 يأتي دور البيانات (Data) لتصبح Valid على الـ Address Bus.

■ والآن نأتي إلى Write Cycle، تتم عملية الكتابة من Microprocessor إلى الذاكرة، هنا البيانات تكون سريعة فتفعل من بداية T_2 ، أما عند القراءة تم تفعيل البيانات في بداية T_3 ويتم إطلاق إشارة الكتابة في منتصف T_2 ثم تنتهي وتصبح في فعالة في منتصف T_3 .

Read: Mem → MP Write: MP → Mem

ملاحظة:

التوقيت الغير متزامن لعمليات الناقل Timing of Asynchronous Bus operation



الفرق بين Asynchronous (غير المتزامن) و Synchronous (المتزامن)

- في المتزامن كان اعتماد العمليات على clock, أي كل حدث يتم على الناقل يعتمد بشكل مباشر على clock.
- في غير المتزامن تتم عملية Read و Write بشكل مستقل (مخطط Read + مخطط Write)

ولكن يجب الانتباه هنا إلى Active Low:

يعني أن الإشارة تكون مفعلة عند 0 وغير مفعلة عند 1 كما نرى في Read. بينما في Synchronous كانت Active high.

- بعد تفعيل إشارة القراءة في Read Cycle يتم تفعيل Data. وبمجرد خروج Data من الـ Memory فتعطي الذاكرة Acknowledge بأن البيانات قد خرجت من الذاكرة وعند الانتهاء يتم عمل drop (أي إفراغ Bus من البيانات التي كانت عليه).

■ بعد التأكد من أن الـ Data قد خرجت من الـ Microprocessor تفعل إحدى إشارات الـ Write حتى يتم الكتابة على الذاكرة، عند الانتهاء يتم تفعيل Acknowledge لعمل drop.

والآن سنتعرف على طريقة أخرى لنقل البيانات

من نقطة إلى نقطة (Point - to- Point Interconnect (one to one)

- السبب الرئيسي للجوء لهذه الطريقة هو القيود الكهربائية التي تمت مواجهتها مع زيادة وتيرة النواقل المتزامنة الواسعة.
- وعمليات القراءة والكتابة التي قد تحدث بشكل كبير ومنتالي أدى إلى صعوبة في التزامن والتحكم في الوقت المناسب.
- إن فكرة الناقل التقليدي المشترك على نفس الشريحة، ضمنت صعوبات زيادة معدل بيانات الناقل (increasing bus data rate) وتقليل زمن استجابة الناقل (reducing Bus Latency).
- بينما لديها زمن استجابة أقل (Lower Latency) ومعدل بيانات أعلى (higher data rate) وقابلية أفضل للتوسع (better scalability).

الـ Point - to - Point لها عدة أنواع منها



Peripheral Component
Interconnect

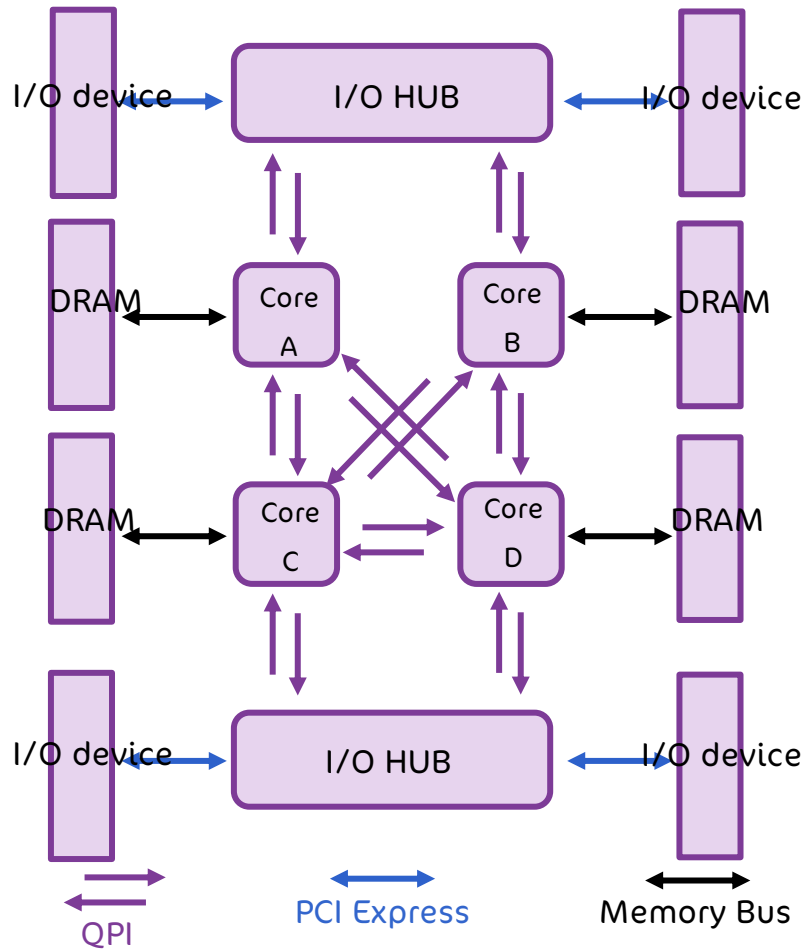
Quick Path Interconnect

سنتكلم عنهم بشكل بسيط، نوعاً ما:

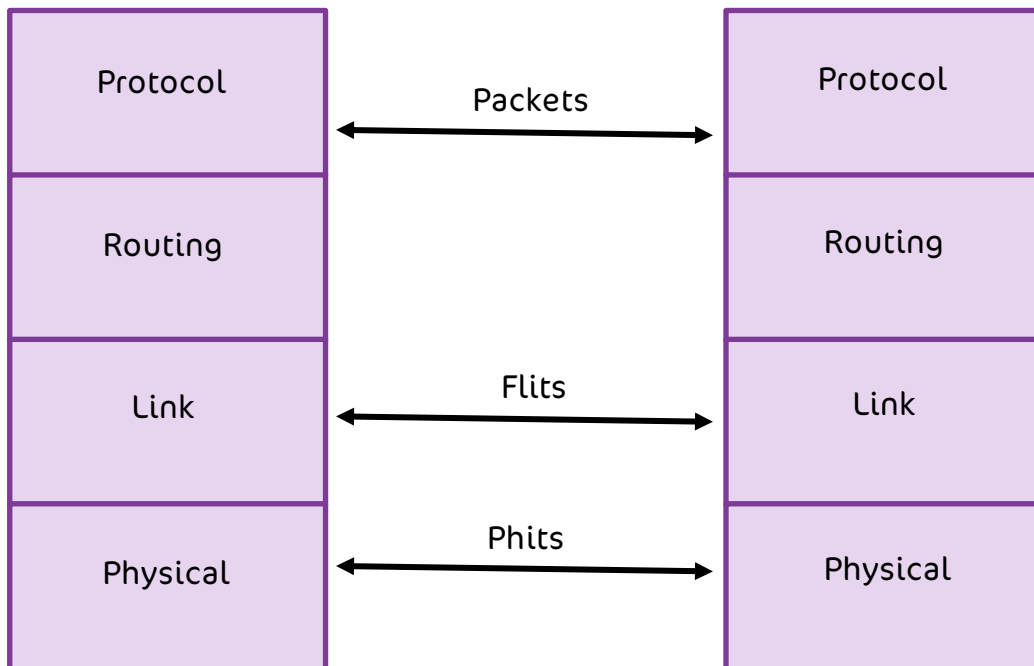
QPI

- يعتمد مبدأ Multiple direct Connections الذي هو عبارة عن Direct Pairwise Connection أي كل نقطة مشبوكة مع نقطة أخرى بـ Path (مسار) خاص.
- ومبدأ Layered Protocol Architecture أي كل طبقة مسؤولة عن مهمة معينة.
- ومبدأ Pocketized Data Transfer أي تقسيم البيانات الكبيرة إلى صغيرة لتصبح عبارة عن Packets يتم إرسالها واحدة تلو الأخرى وكل packet مؤلف من error و control headers و control codes.

تكون Multicore Configuration using QPI بالشكل

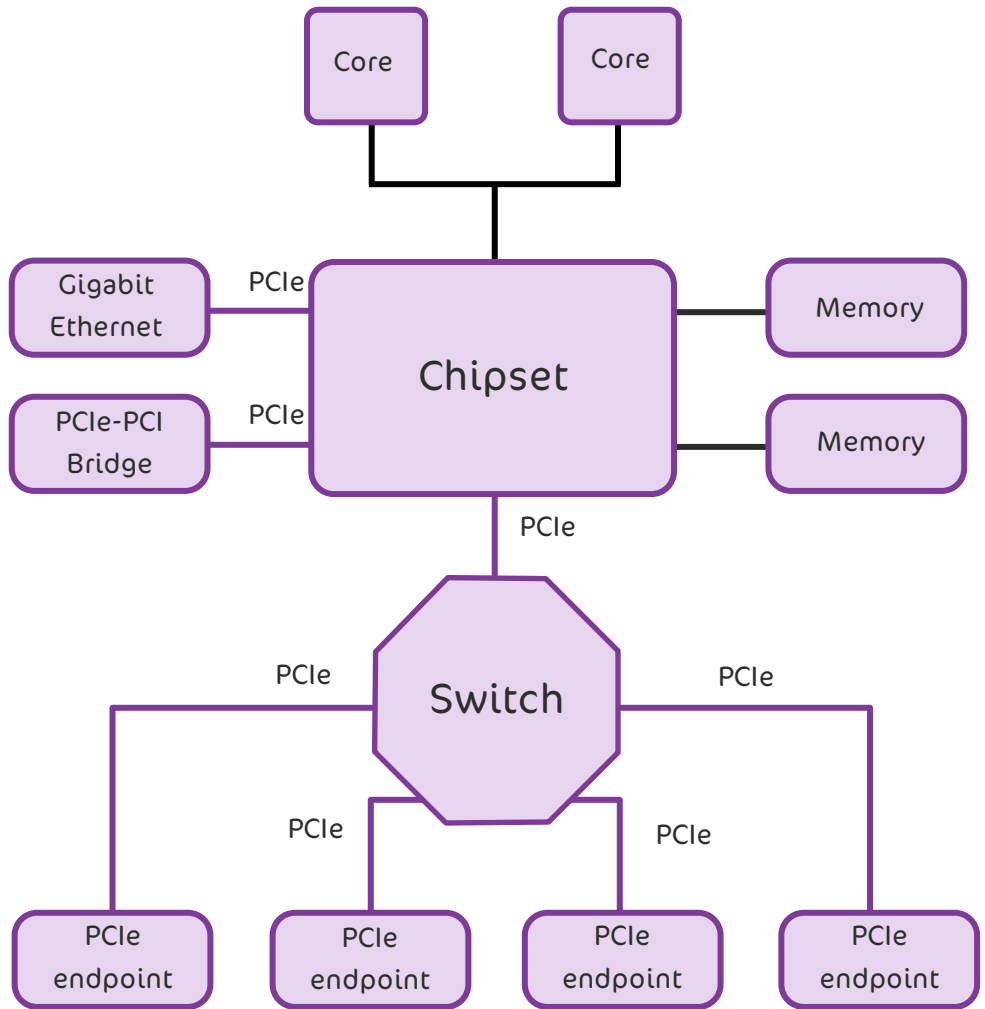


طبقاته QPI Layers:

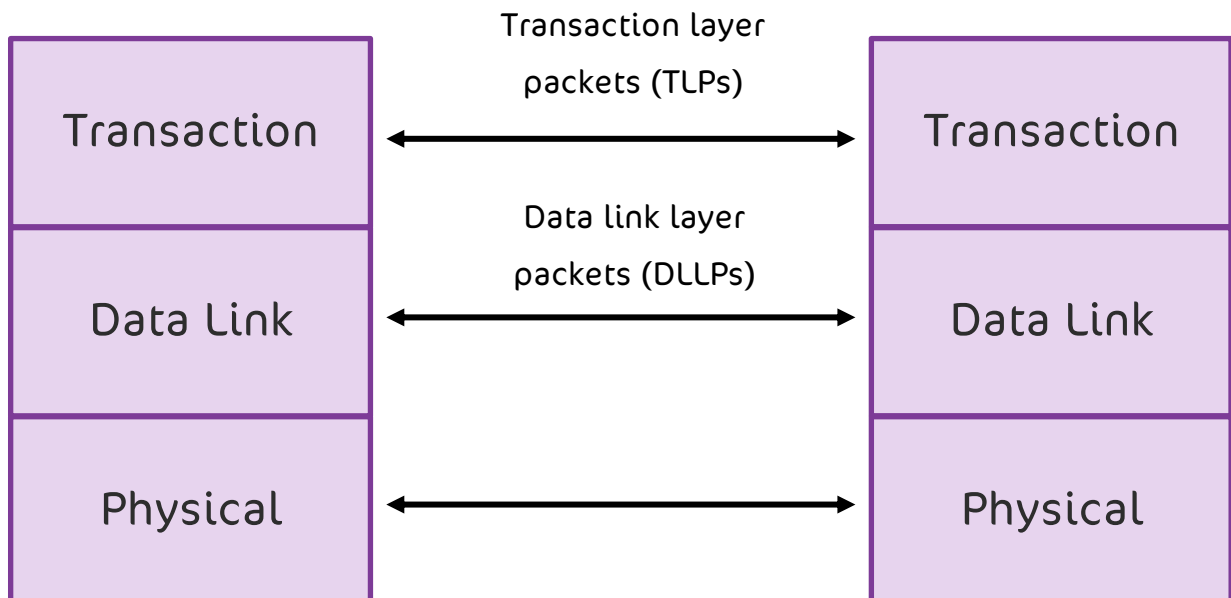


تكوين PCIe

- Chipset: شريحة
- Core: نواة معالجة
- PCIe-PCI: ربط طرفيه عن طريق جسر Bridge.
- Switch: جهاز شبكي يمرر البيانات ويربط وحدات الإدخال والإخراج ببعضها، ويرسل البيانات إلى المرسل إليه فقط (أي ليس لجميع الأجزاء).
- Gigabit Ethernet: نوع الإشارات التي تنتقل.



PCIe Protocol Layers



معايير قياس الأداء

Clock Speed

وتعبر عن عدد العمليات الحسابية التي يستطيع المعالج إنجازها خلال الثانية الواحدة وتقاس بـ

1. KHz : تستغرق الدورة الواحدة 1 Millisecond
2. MHz : تستغرق الدورة الواحدة 1 Microsecond
3. GHz : تستغرق الدورة الواحدة 1 nanosecond
4. THz : تستغرق الدورة الواحدة 1 picosecond

Instruction / application performance

■ أهم النقاط:

1. MIPS (millions of Instructions per second)

2. FLOPS (Floating point instruction per second)

■ على سبيل المثال:

يملك المعالج الرسومي GeForce GTX Titan حوالي 2688 نواة وبقوة (4.5) Tera Flops

3. Benchmark

■ ببساطة عبارة عن اختبار يستخدم لمقارنة المنتجات المتماثلة بحيث يقوم بتطبيق سلسلة من الاختبارات المحددة على الحاسوب لقياس ادائه , تنتج معايير (UL) درجة يمكننا استخدامها لمقارنة أنظمة الكمبيوتر بحيث تشير الدرجة الأعلى إلى الأفضل.

4. What are “CPI” and “IPC”

■ CPI : Cycles per instruction وهي عدد الدورات اللازمة لإنجاز عملية ما

■ IPC: Instruction per cycle وهي عدد العمليات التي يتم تنفيذها خلال دورة ما وتساوي مقلوب ال CPI

$IPC = \frac{1}{CPI}$ وتستخدم الـ IPC بشكل متكرر أكثر من الـ CPI.

5. Different instructions have different cycle costs

- على سبيل المثال يتم إنجاز تعليمة الجمع خلال دورة واحدة
- بينما تحتاج عملية القسمة إلى أكثر من 15 دورة حتى يتم إنجازها
- يعتمد على ترددات التعليمات المتقاربة (النسبية)

مثال: ليكن برنامج ما لديه النسب الآتية :

Integer = 1 , Memory = 2 , FP = 3

احسب ال CPI :

$$CPI = (33\% \times 1) + (33\% \times 2) + (33\% \times 3) = 2$$

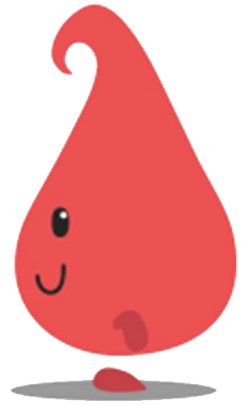
مثال: ليكن لدينا الحاسوب A,B وفق الآتي :

A : Clock = 5 GHz , CPI = 2

B : Clock = 3 GHz , CPI = 1

أيهما أفضل؟

الحاسوب B هو أفضل علماً بأن لكلا الحاسوبين نفس ال ISA وال Compiler



سنتعلم في المحاضرة القادمة طريقة تحديد اي الحاسوبين افضل

Latency

6.

يكون الحاسوب بوضع مثالي عندما يملك:

1.High throughput

2.Low Latency

الوقت اللازم لتنفيذ البرنامج يعرف أيضاً بزمان الاستجابة أو التأخير

Throughput

7.

وهي الإنتاجية وتعبر عن كمية العملي المنجز خلال واحدة الزمن

كيف نجعل الحاسوب أسرع ؟



1. التقليل من معدل التأخير

2. المسار الحرج: بحيث يحدد أطول مسار الحد الأدنى من الزمن اللازم لتنفيذ عملية

3. تحسين التأخير (معدل الاستجابة على المسار الحرج) وذلك عن طريق:

1. التوازي (parallelism)

2. التوارد (pipelining)

3. التقنيتين معاً

Single-multi cycle data path

Single cycle

جلب التعليم (Fetch) وفكها (Decode) وتنفيذها (Execute) خلال دورة واحدة

Low CPI

Long Clock period

Multi cycle

جلب التعليم (Fetch) وفكها (Decode) وتنفيذها (Execute) خلال دورات متعددة

High CPI

Short clock period

انتهت المحاضرة