

حسنة الأفرعشان تشتغل: inactive device (processors):

التاريخ:

(شخالة تنفيذ)

الموضوع:

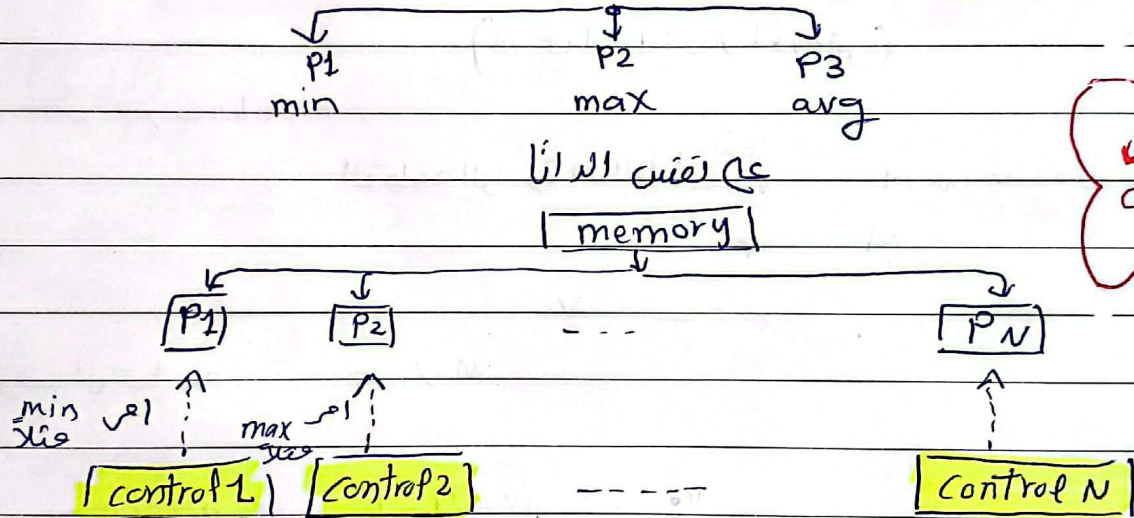
active devices ← devices ال هسقة

NOTE عىمكة الفكرة ال خاتى دى ال processors تكون على GPU
جوه جهاز واحد، وعىمكة تكون أجهزة كاملة و server
يتكلم فيها

3] MISD (Multiple Instruction single data) (pipelined processors)

وهو طبق على أكثر من أمر، و الداتا بسج فيه بعض،
وأكثر من processor

تتفع مع التطبيقات ال بتتعال مع الداتا بأكثر من شكل.



أكثر من أمر ← عمل
أكثر من Control unit

Pipeline

الداتا هتدخل فيه وكان واحد على P1 يعمل عملية

ما تم يحط الناتج في buffer 1 و لو P2 شغال على

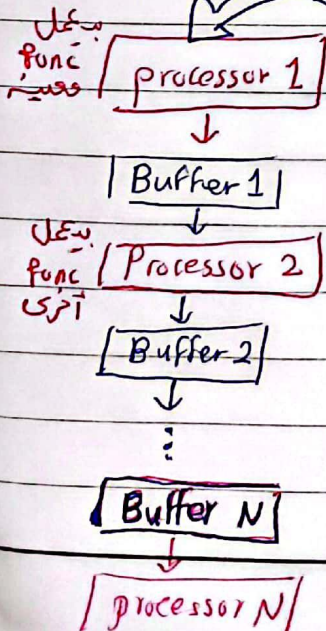
ما ج بال فعل فالناتج هسكن في ال buffer 1 لحد ما يخلص

ثم يدخل على P2 و P2 بيعتج الناتج بياخه ال buffer ال بعده و كذا

طبعاً بمجرد ما P1 بيعتج الناتج ل buffer 1 بياخذ داتا جديدة

ليستغل عليها حتى بيستنى فى اكل واحد شغال

على داتا و ختلفة.



خلاصة الصفحة

الموضوع:

عدد الداتا \rightarrow N
 عدد stages \rightarrow M
 $speedup = \frac{NM}{N+M-1}$
 ← جزء ال pipeline التاريخ

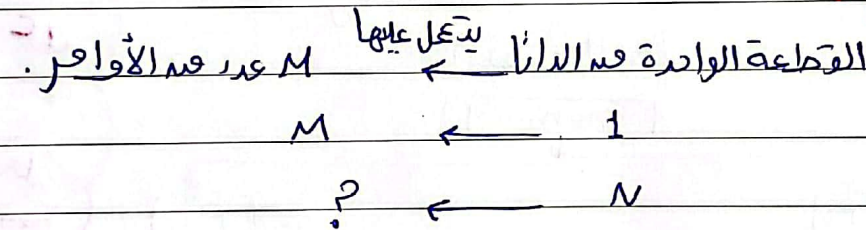
لو عملنا نظراً ال pipeline 6 وعملنا فيه غير pipeline
 فسرعة ال pipeline كم مرة فيه ال مشن pipeline ؟

الوقت ال هيا فيه لو مش pipeline
 $Speedup = \frac{\text{الوقت ال هيا فيه لو مش pipeline}}{\text{الوقت ال هيا فيه لو مش pipeline}}$

فمثلاً لو ب pipe حد 2sec 6 وعملنا فيه غير حد 4sec
 $Speedup = \frac{4}{2} = 2$

$N \rightarrow$ عدد قطع الداتا
 $M \rightarrow$ عدد ال stages ال هتجبل على كل قطعة داتا
 (ال هي العميات / الأوامر)

في غير pipeline :



الوقت ال هيا فيه $MN =$

في pipeline : $N+M-1$

$Speedup = \frac{NM}{N+M-1}$

عشان ال pipeline يكون > لو الوقت ال الأوامر هيا فيه يكون
 يساوي نفس الوقت لباقي الأوامر
 (time should be the same for all stages)

MIMD (Multiple Instruction Multiple Data) (Multi processor or multicomputers)

Asynchronous
كل P يعمل أمر مختلف
على داتا مختلفة
"عالم مفتوح دعوة بيوتها"

هو دال Distributed sys (لوا الأجهزة تتشارك
سنوا)

فدئرتين

Multi computers

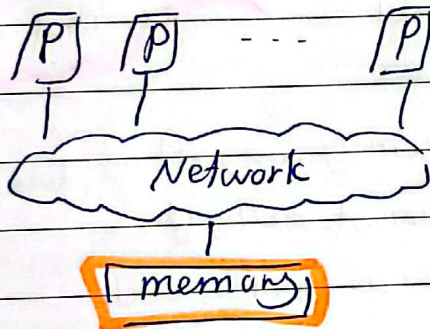
كل جهاز مستقل بذاته وله
construction مختلف

Multi processors

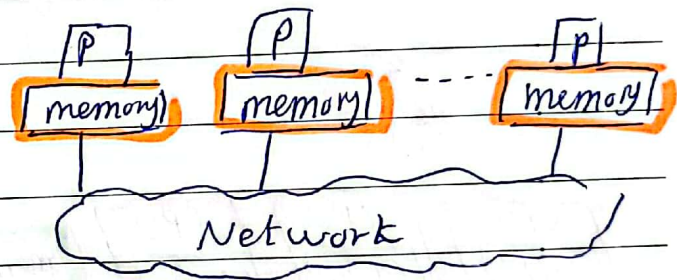
اجهزة متصلة على
single board

Distributed HW

Physical shared memory



Physical distributed memory



كل P له cache و لكنه memory
واحدة كله شاي فظ

كل P له ال memory بذاته
ولو هيتواصلوا يتواصلوا ويديعوا
اللائاع طريق ال network
عن طريق ال messages

tightly coupled → بذغل / نتشارك نفس الحاجة

loosely coupled → فشن بذغل / نتشارك نفس الحاجب

Physically shared → (فنتشارك نفس المكان) في نفس المكان

logically shared → (في اماكن بعيدة عن بعض)

physically shared

tightly coupled

اضاف نفس المكان و بذغل
نفس الحاجب

physically shared

loosely coupled

اضاف نفس المكان
وكل واحد بذغل حاجب
مختلفة

logically shared

tightly coupled

فشن في نفس المكان و كل
واحد في مكان مختلف
و بذغل نفس الحاجب
في الحافزة الاونلايه

logically shared,

loosely coupled

كل واحد في مكان مختلف
وكل واحد بذغل حاجب
مختلف

NOTES ← الأجهزة لوبعية بعد بعض وأثابيرها قد كان يوجد

عشان تعمل task وحدة

logically shared

tightly coupled

← multi processors ← مبيعة ال multi processors tightly coupled, physically shared

لاهم في نفس الجهاز و يعملوا نفس الناس غالباً

← multi computers ← loosely coupled, logically shared

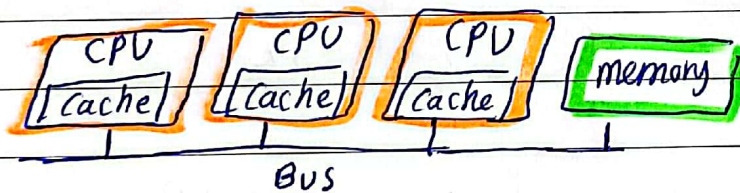
عادي اجهزة كل واحد في مكان وكل واحد يعمل الناس

ال صاحب محتاجه . (في الغالب)

Tightly coupled system (multiprocessor)

دلوقة عشي أكثر من CPU و [memory أو أكثر من memory] ايه طرق تركيبهم مع بعض

memory وحدة

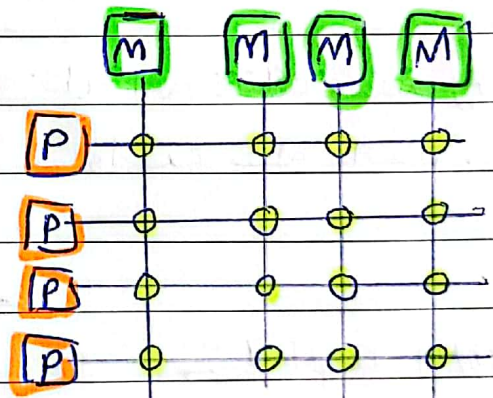


أكثر من memory

[أو القسمان نفس ال memory]

المعالج

i cross bar switch



"Bus-based multiprocessor"

ملحوظة: هناك CPU يمكن تكلم ال memory

عادي، لكن عشي بتكلم CPU أخرى

لوعايزه وهي هتكلمها عن طريق ال

memory [وسيط]

* كل P متصل بكل ال M

كل نقطة فيها switch فيها .

فمحتاجين control بحيث أقفله أو

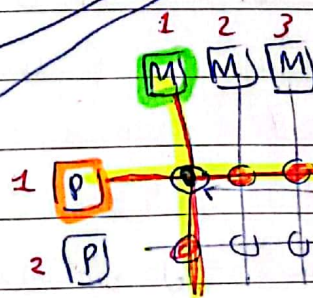
أفقه .

* لو P1 عايز يقرأ من M1 معين فلا

لازم أقفل ال switch بتلعبهم

وامنع أي شيء أو العكس أيها تنقل (اكن الخطيرة

switch



* ال control بلاعه صعب

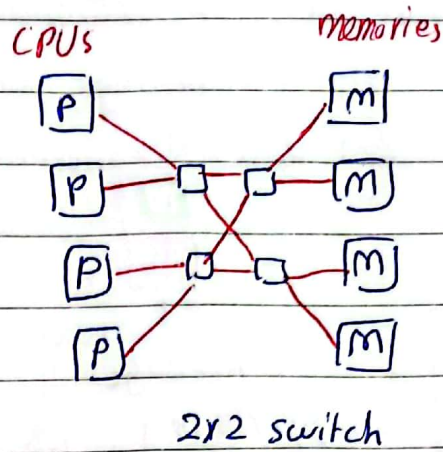
* مشكلة إيها بتسخن

برنو

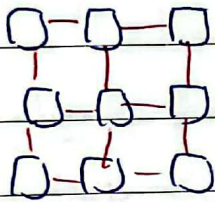
مشغولة

2] Omega switching network

قللت التوصيلات عدداً قبلها
 في لوقت ال control بقي أصعب
 لأن لو عايزيه نقفل P الأولى مع M
 الأولى كدا هقفل مثلاً ال
 switch بين ال فوق و طب لو حاج
 تانيه؟ هتتعقد أكثر
 ودول 2x2 switch فقط لو أكثر!



3] Grid

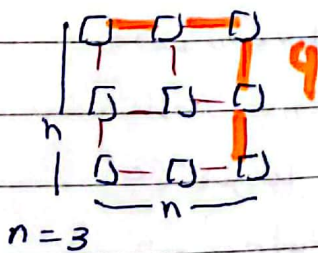


multi processor Grid

(موصلة في شكل grid)

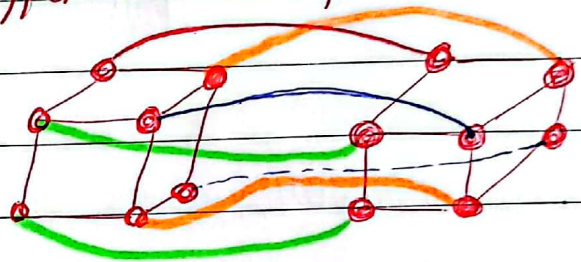
rectangular array of CPUs
 with connection only to
 neighbors "موصلة بالجيران فقط"
 Corner أبعد رسالة في

(corner آخر هحتاج)
 $(n-1) + (n-1)$ hops



$$(3-1) + (3-1) = 4$$

4] Hypercube multiprocessor



كجيبه واصل كل vertex بال vertex
 المقابلة.

ال في ال vertices دول ال processors
 وهي شكل التوصيلة بيتر

لو وكعب واحد فهو 3d
 لكنه المكعبه دول بيعتبروا أكثرهم وكعب واحد 4d

* n-dimensional cube
 with CPUs @ vertices

عدد CPUs
 vertices

$$2^n \text{ CPUs}$$

ex: $n=3 \Rightarrow 8 \text{ vertices } 8 \text{ CPUs}$
 $n=4 \Rightarrow 16 \text{ vertices } 16 \text{ CPUs}$

* لو هجبت رسالة في CPU للآني هحتاج
 $n \text{ hops } (\log_2 (\text{CPUs No}))$
 $\log_2 (2^n)$

multi computers \Rightarrow physically shared, loosely coupled
العزيب \Rightarrow أفاديكون tightly coupled، cloud أفاديكون task والدة

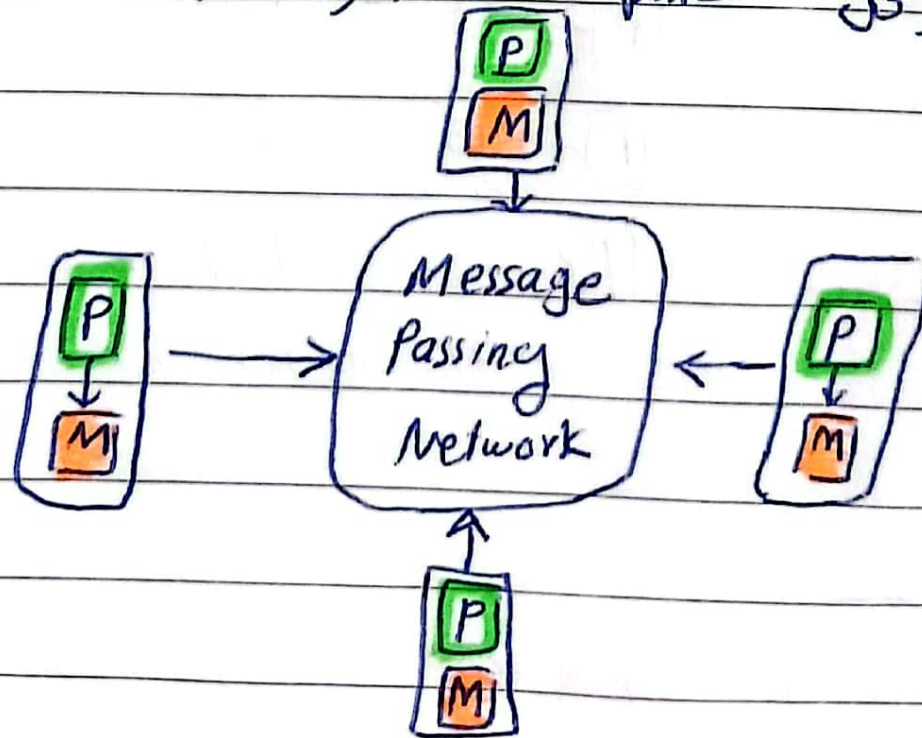
التاريخ:

الموضوع:

هناك computer عن طريقه الآخر عن طريق
message passing Network
(كل computer P و M)

Loosely
Coupled sys

الفيه
(multiple computers)



أفاديكون دا كل computer
تفادي Task فتيقة
لانه loosely
في الحالة دي