lec 2

Bus is a shared resource

بيتنقل ع ال Bus ٣ حاجات
- * Control signals ←
- * Data ←
- * Address ←

وحدة العمليات اللي بتحصل ع ال Bus اسمها Transaction
يعني وحده كامله من أي حاجه بتحصل ع ال Bus.

Transaction .. الكلمه دي بنسمعها في البنوك .. بتروح البنك
تاخد رقم وتستنى لحد ما يتنادى رقمك وتروح تقف ع الشباك
بتديه رقم حسابك وتديه فلوس للإيداع أو بتسحب فلوس.

نفس الكلام ده بيحصل في ال cpu-Architecture

* Steps of Transaction

① **Arbitration** → هي ال Process اللي بتحكي انهي Master
بيتحكم في ال Bus يعني يقول مين هيـ Read ومين هيـ Write
→ It is only applicable to multi-master buses
← يعني لو عندي Master واحد بس في ال Bus كله مش هحتاج
ال Arbitration في حاجه ... ممكن نعمل Arbitration بالدور أو بال Priority ...
وبكده أكون خلصت ال Arbitration وفضيت الاشتباك بين ال Masters الكتير
زي بالظبط لما بنادي في البنك ع رقم العميل وباقي العملاء بيستنوا دورهم

② **Addressing** → ال Master بيبعت لل slave اللي عايز يكلمه وهو
بيبعت ال Address بيبعت معاه control signal ← for enable info. و ال slave
بيرد ع ال request سواء read او write ..

③ **Actual data transfer** → عملية نقل الداتا بين الماستر و ال slave
زي العميل في البنك بعد ما بيدي للموظف رقم حسابه بيبدأ
عملية نقل الفلوس

* Broadcasting ⟹ talking to all devices like * initialization * reset * shut down
* Multy casting ⟹ Master talks to many slaves at a time.
* Uni casting ⟹ talking with one Particular device

لو جيه سؤال في الامتحان كم Phase للـTransaction

← الإجابة الصح حسب نوع

(2) ← single Master

(3) ← MultyMaster

* حيث ان الـ Arbitration دي مترتبة ع تعدد الـMasters

* * * * *

Synchronization ← طريقة من الطرق عشان اتأكد وقت ما هقرأ أو أكتب اللي موجود ع الـ Bus ده Valid مش garbage

في ٣ طرق

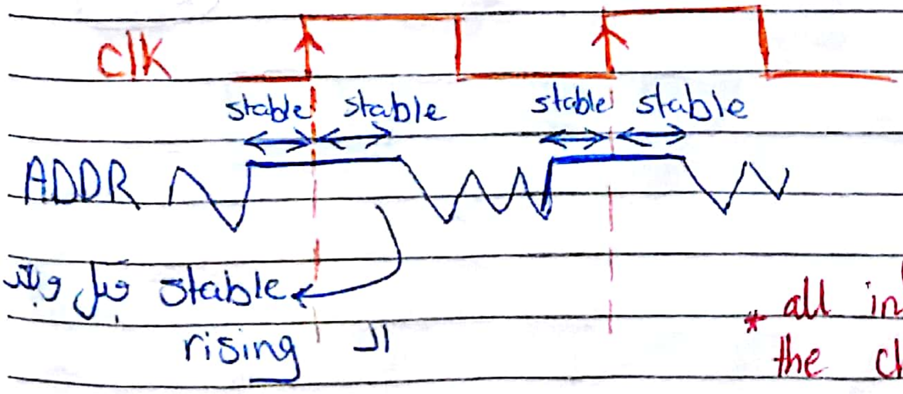
① Synchronous

② Semi-Synchronous

③ Asynchronous

① Synchronous ← بيكون عندي refrence clock الـMaster والـ Slave شغالين عليها "Perfectly" سواء (rising edge أو falling edge)

* وبتأكد قبل لحظة الـ rising مثلاً ان الداتا stable قبلها بفترة و stable بعدها بفترة ← عشان اتأكد إنه قراها صح



* all info. is read only on the clk edge

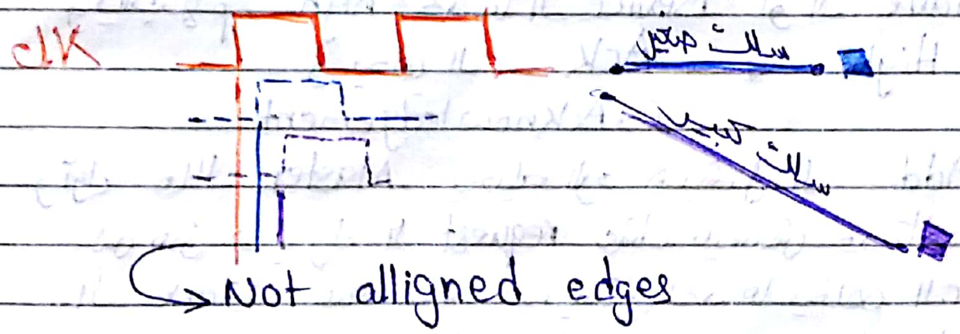
Subject موضوع الدرس

Date التاريخ

إيه عيوب الـ Synchronous ؟؟

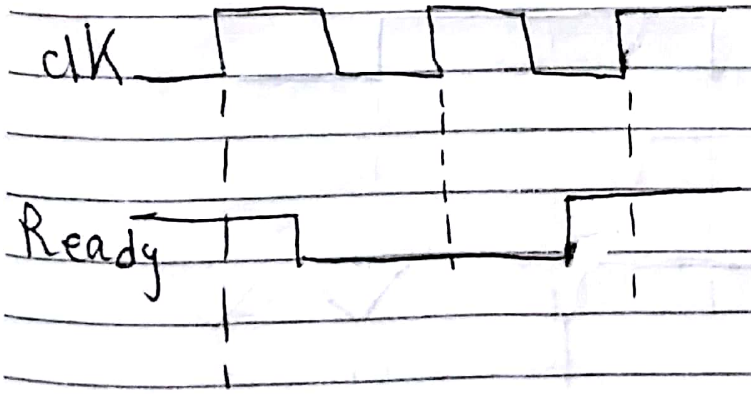
* المشكلة بتظهر في إني إزاي أخلي كل الـ Components فالـ system اللي عندي تشتغل بـ clk واحدة؟ .. ممكن أظبط الـ clk ع كل Comp. أو جزء وأعمل clk division وده ممكن يسبب shift في الـ clk وشوية مني

* كمان في الـ Propagation time .. لما أبعت لحاجة قريبة وحاجة بعيدة ع ما توصل الـ edges بتاعت الـ clk will not be alligned



* * *

② Semi-Synchro. ← هي طريقة بتسمح لـ الـ devices اللي بتتكلم بسرعات مختلفة إنها تتعامل بـ clk واحدة عن طريق Ready أو Wait سجنال يعني لو الـ slave اتأخر عن الـ Master لازم الـ Master هيستنى الرد من الـ slave بأنه يخلي Ready signal ← High



فكأني هنا قسمت الـ clk cycle على 2

③ Asynchronous ← مفيش clk بس استبدلتها بـ 2 Wires

Ack. request

↳ control wires

* الـ Master هينده الـ slave عن طريق إنه يخلي

الـ request ← High

ADDress ← High

ويسيبهم high لحد ما الـ Device أو الـ slave يرد عالـ Master

ويخلي الـ Ack ← High

↳ Acknowledgement

وأول مالـ Master يجيله الرد هيشيل الـ Add ← Low

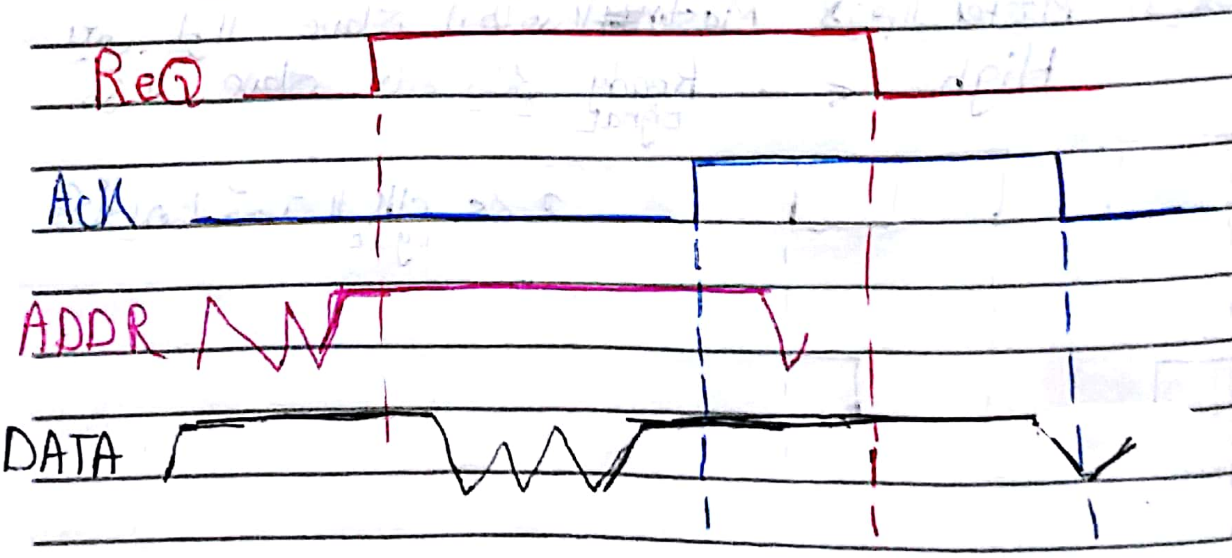
بس مش هينزل الـ request عشان ميسبش حد تاني يعمل request وياخد الـ slave منه .. هيستنى لحد ما يخلص الـ service بتاعته وينقل الداتا

وبعدين يخلي الـ request ← Low

الـ slave هيفهم أنه خلص وهيخلي الـ

data ← Low

Ack ← Low



* Asynchronous Protocols are gaining Popularity in on-chip communications as it dowes not suffer the constraints / delays imposed by the slowest signal

⇐ عشان في الـ synch. الـ clk على ما توصل لكل الـ Components
ممكن توصل متأخرة بـ 2 or 3 ← clks

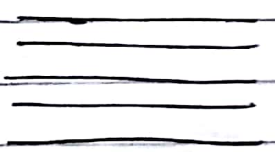
* * * *

multiple wires ← [Parallel & Serial Bus] → single wire

الـ Parallel أسرع من الـ Serial بس ليه عيوب

① clk skew

same sourced clk signal may arrive the components at different times



$t + \tau_1$
$t + \tau_2$
$t + \tau_3$
$t + \tau_4$ → different timing

→ مضطر أخلي الـ clk تمشي مع أبطأ واحدة

② cross talks between Parallel lines

نتيجة الـ mutual Inductances والـ mutual capacitances بيأدي لتداخل الـ info وبيحصل noise

③ distance & freq

كل ما زادت المسافة أو زاد الـ freq .. الـ cross talks بتزيد فالداتا بتوصل غلط

$$f_{clk} = \frac{1}{T_{Dmax} + T_{skew}}$$

Propagation time
T_D بناخد أبطأ

For On-chip we can choose → serial / → Parallel
For off-chip → use serial

Parallel Buses

ال Bus اللي استخدمته شركة ARM عندها اسمه ال AMBA Bus

AMBA
→ Advanced Microcontroller Bus Architecture
→ is a feature-rich example of Parallel on-chip bus standard
→ it defines 2 Buses → AHB: High-Performance Bus
→ APB: slow bus

← وده اتجاه ال Hierarchical Bus (متسلسل هرمي) ع حسب السرعات

في سلايد 27 :-
الحاجه الوحيده اللي ممكن تكون Master و slave في نفس الوقت ← ال Bridge
→ Slave in AHB
→ Master in APB

AHB: is fast, Parallel, multi master, Pipelined bus with support for burst & split transactions
→ هنتكلم عنهم المحاضرة الجاية

multimaster → means 3 Phases for transaction
Pipelining →

①	Arbitration	Addressing	Data transfer	
②		Arbitration	Addressing	Datatrans
③			Arbitration	Addressing

which means
throughput Now → 1 transaction through 1 clk cycle
before → 1 " " 3 " "