Computer Architecture: Fundamentals, Tradeoffs, Challenges

Chapter 9: Input/Output (I/O)

Yale Patt

The University of Texas at Austin

Austin, Texas

Spring, 2023

Outline

- ***Characteristics of I/O***
- ***Bus Transactions***
- ***An example: asynchronous bus with central arbitration***
  - ***Arbitration***
    - ***Race Conditions***
  - ***Transfer***
- ***RAID (Redundant Array of Disks)***

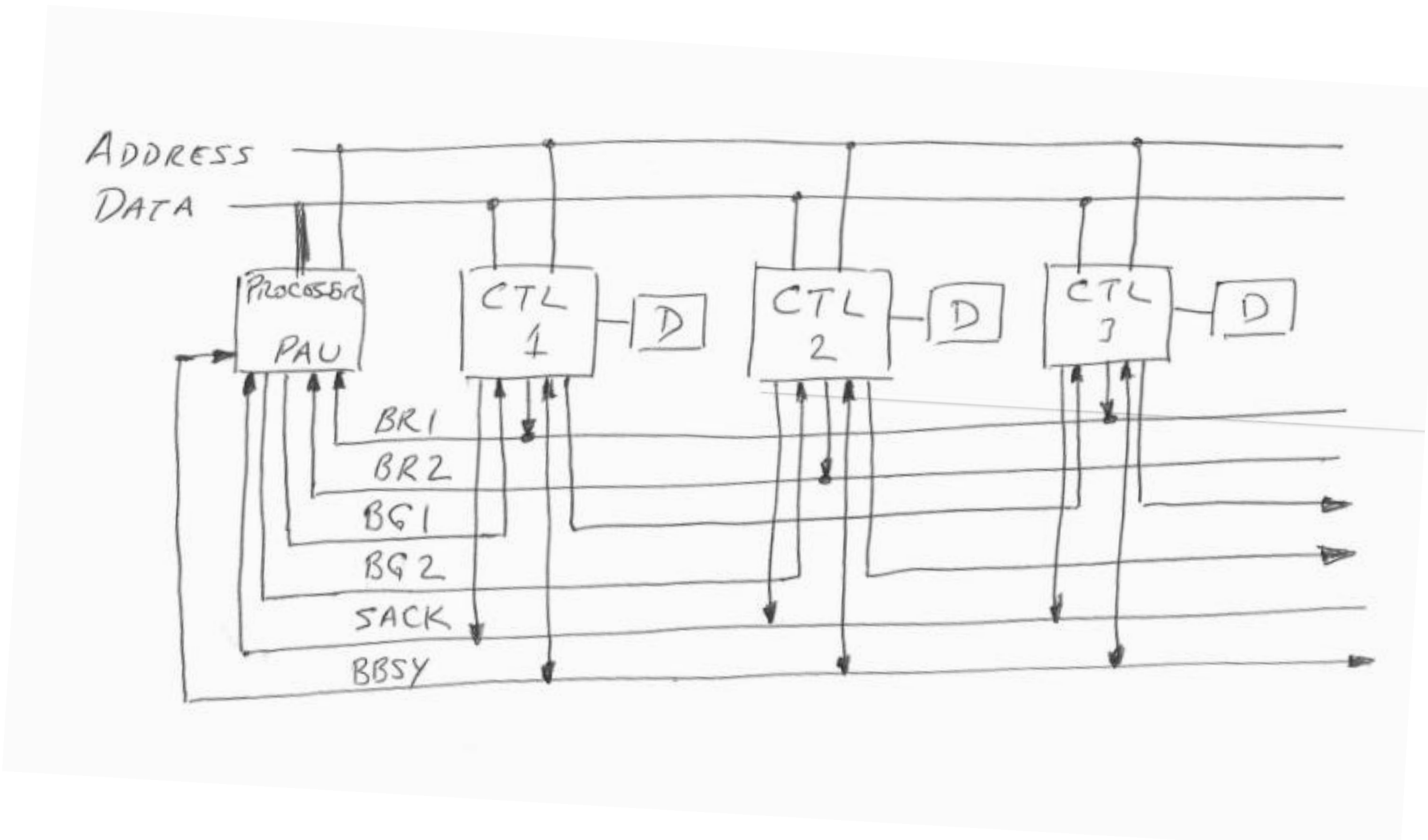
Characteristics of I/O

- ***Three parts***
  - ***The medium (e.g., the magnetic field in the track)***
  - ***The device itself (e.g., the disk)***
  - ***The controller***
- ***How***
  - ***Polling***
  - ***Interrupt driven***
  - ***DMA (the I/O control block)***
  - ***I/O processor***
- ***Instructions***
  - ***Memory-mapped***
  - ***Special I/O instructions***

Bus Transactions

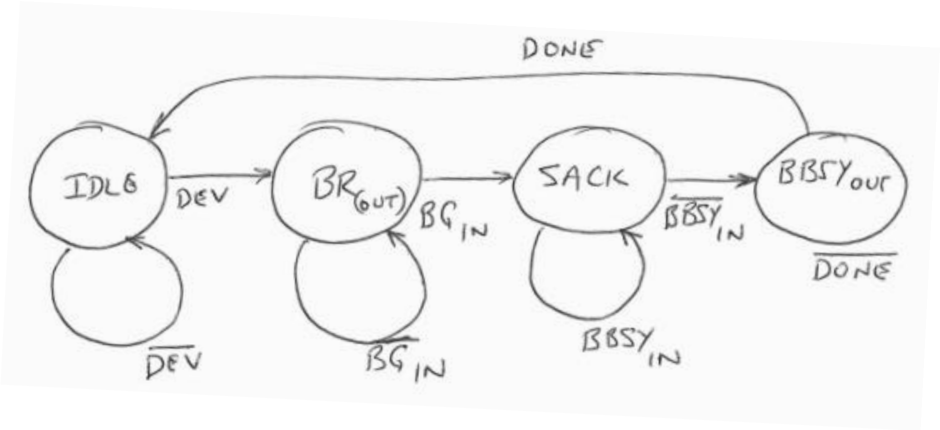
- *Synch vs Asynch*
  - *Asynch (slow)*
    - *Handshaking*
    - *No clock*
    - *Everything explicit*
  - *Synch (fast)*
    - *Clock → Most things are implicit*
    - *Very fast, but must be short*
- *Signals*
  - *Three types: Address, Data, Control*
    - *Multiplexed address, data*
- *Arbitration*
  - *Central: Priority Arbitration Unit*
  - *Distributed: my "dinner table" analogy*
- *Transfer*

An Asynchronous I/O System

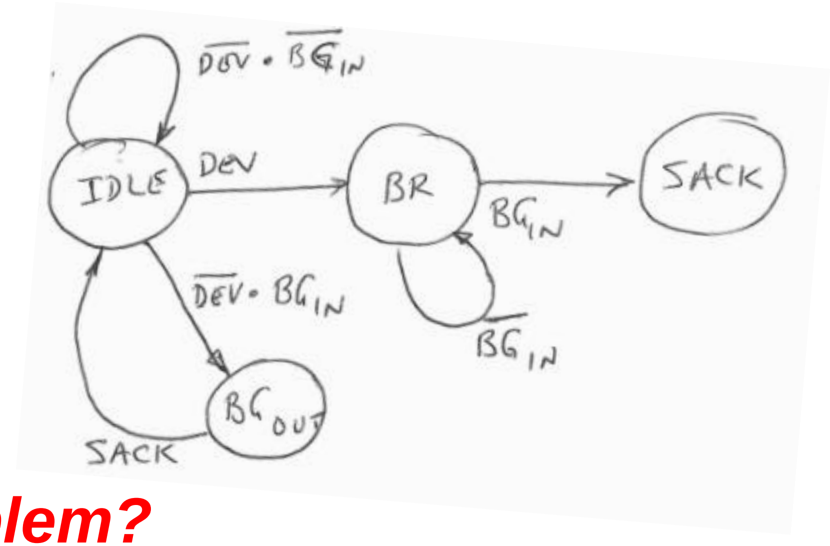


Arbitration

- ***The concept:***



- ***If the device does not want the bus:***



Is there a problem?

A Race Condition

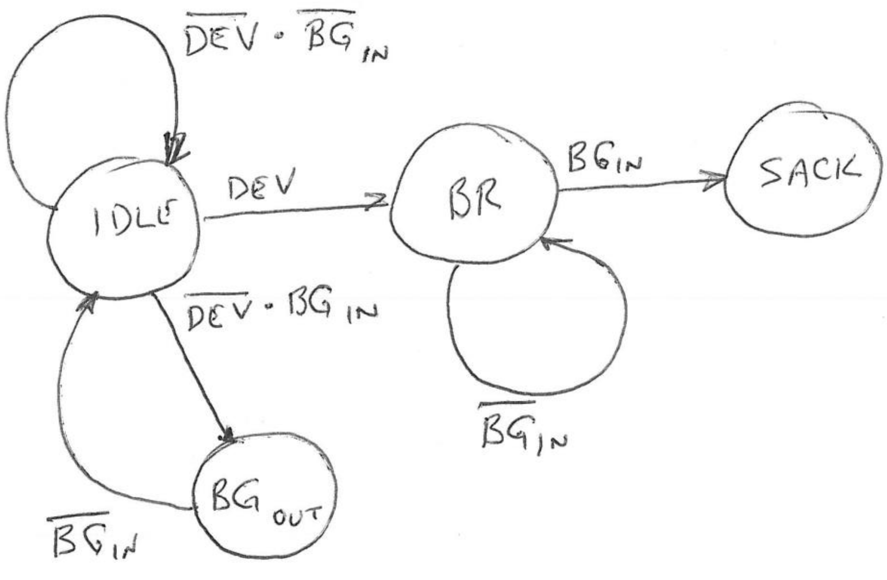
- ***Consider the following:***
  - ***The PAU asserts the BG signal***
  - ***Device A does not want the bus***
  - ***Controller A passes it on***
  - ***Controller B wants it, asserts SACK***
  - ***Controller A sees SACK, returns to Idle***

- ***What if:***
  - ***Device A wants the bus before PAU negated BG***
  - ***Controller A goes to BR and, since BG is still asserted***
  - ***Controller A goes to SACK***

- ***How do we fix it?***

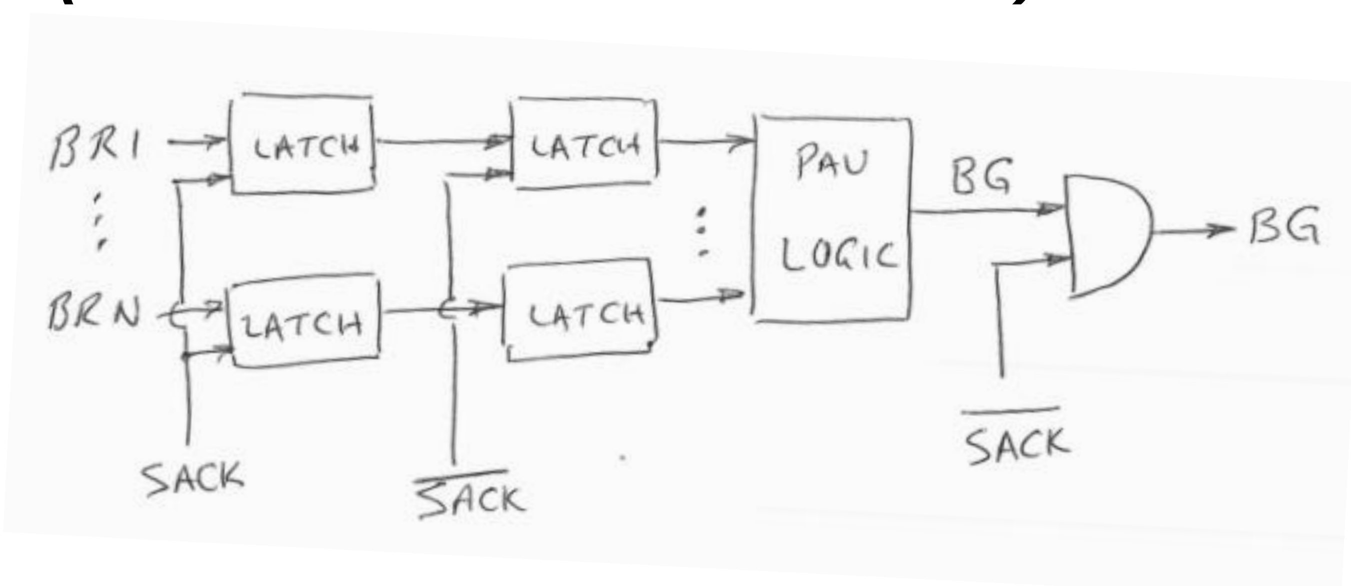
The fix!

- ***We do not return to IDLE when we see SACK***
  - ***PAU may still be asserting BG***
- ***We wait until PAU stops asserting BG***
  - ***Then it is safe to return to IDLE***
- ***The fix:***

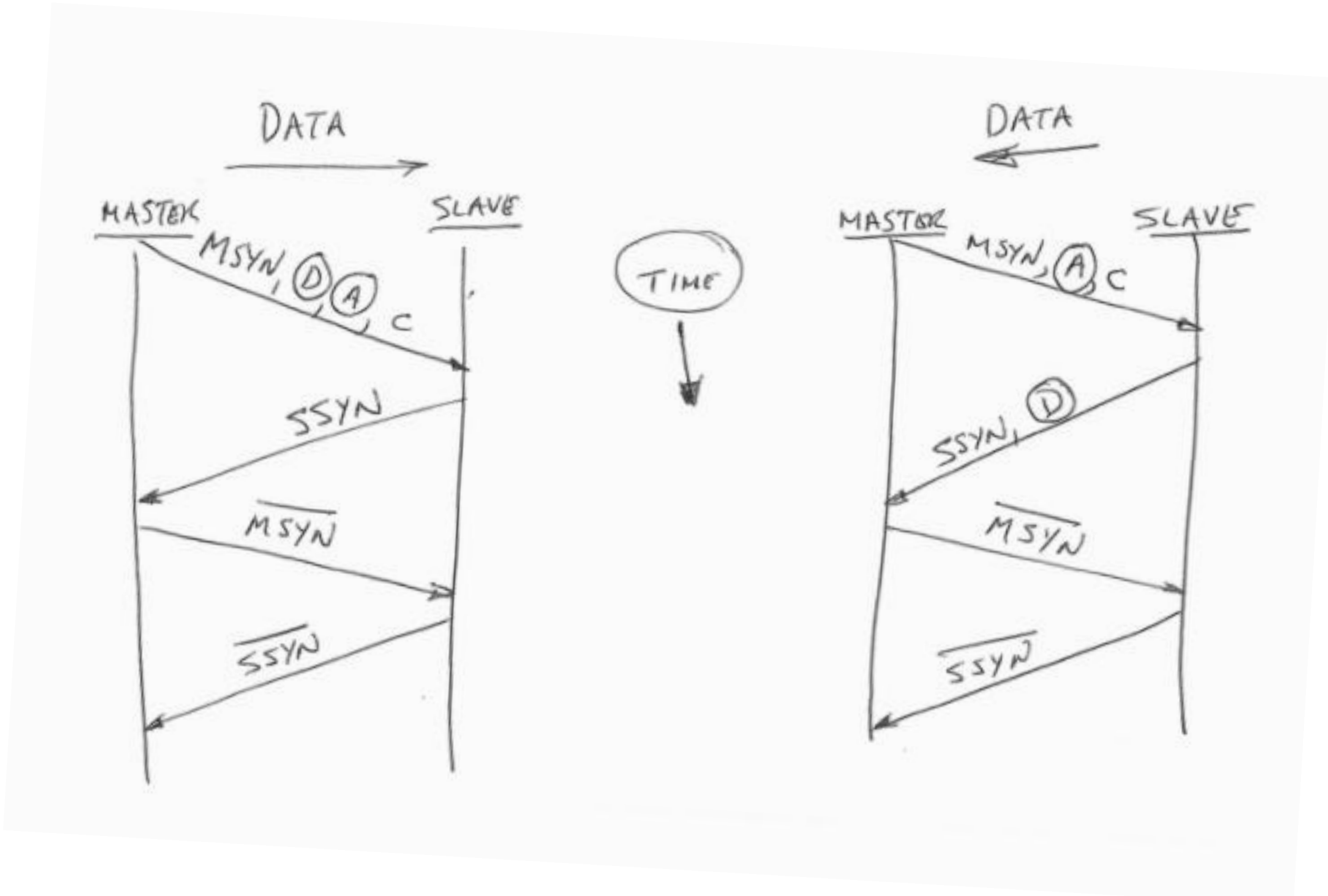


What if a higher priority request comes in AFTER the PAU has issued BG?

- ***How do we keep PAU from issuing higher BG***
  - ***Disable new requests to PAU at start of bus cycle (Bus master, asserts BBSY, negates SACK)***
  - ***Enable requests to PAU at end of arbitration (Next bus master asserts SACK)***



The Transfer

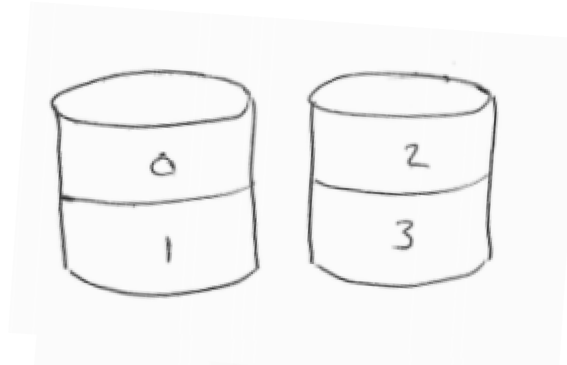


Redundant Array of Interdependent Disks (RAID)

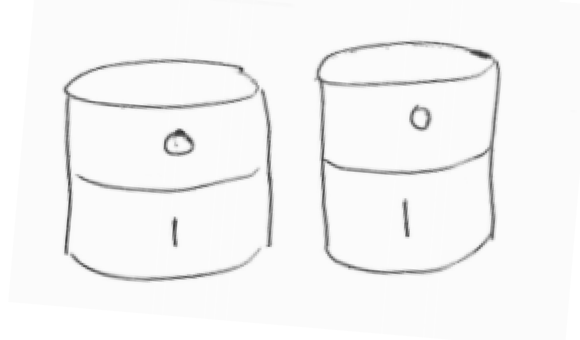
- ***The soul of RAID: performance plus redundancy***
  - ***Introduced by Norman Ouchi (IBM), US Patent granted in 1978***
  - ***Acronym by Gibson, Katz, Patterson (UC Berkeley), 1988***
- ***The meaning of I in RAID***
  - ***Initially Inexpensive, until they realized it was not inexpensive***
  - ***Then independent, except the disks are not independent***
  - ***I suggest "Interdependent" !***
- ***The various levels***
  - ***RAID 0: Vanilla -- Coarse, No redundancy***
  - ***RAID 1: Mirroring – Coarse, Redundancy***
  - ***RAID 2: ECC – Fine, ECC***
  - ***RAID 3: Parity – Fine, Parity disk***
  - ***RAID 4: Coarse parity – Coarse, Parity disk***
  - ***RAID 5: The preferred model – Fine, no parity disk***
  - ***RAID 6: More than one mechanism for error checking***

The RAID levels

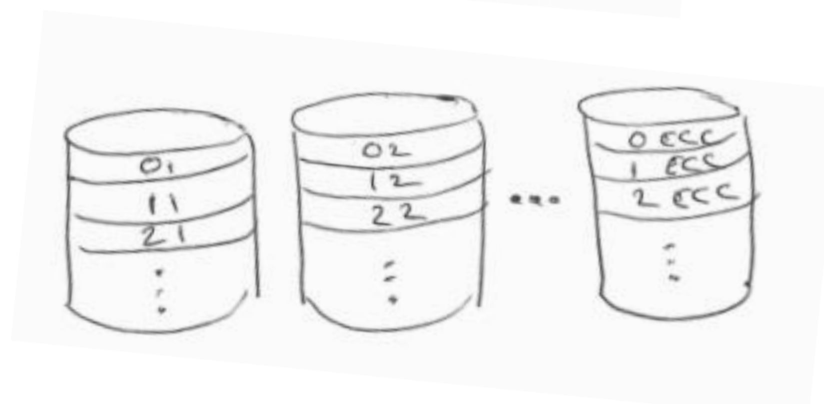
- ***RAID 0:***
  - ***Coarse***
  - ***No Redundancy***



- ***RAID 1:***
  - ***Coarse***
  - ***Redundancy***

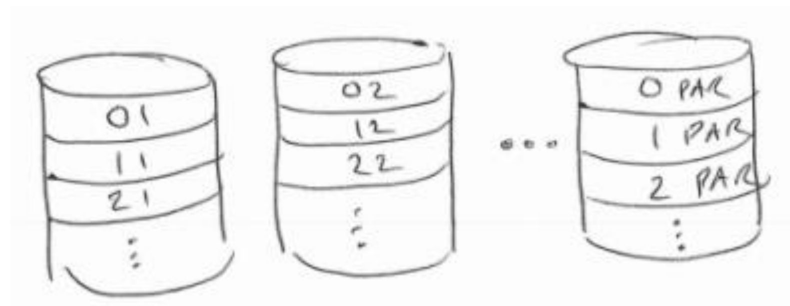


- ***RAID 2:***
  - ***Fine***
  - ***ECC***



The RAID levels (continued)

- ***RAID 3:***
  - ***Fine***
  - ***Parity Disk***



- ***RAID 4:***
  - ***Coarse***
  - ***Parity Disk***

- ***RAID 5:***
  - ***Fine***
  - ***No Parity Disk***

Danke!