Computer Organization and Architecture

13 Bus

Tongwei Ren

Nov. 19, 2019

南京大學
NANJING UNIVERSITY

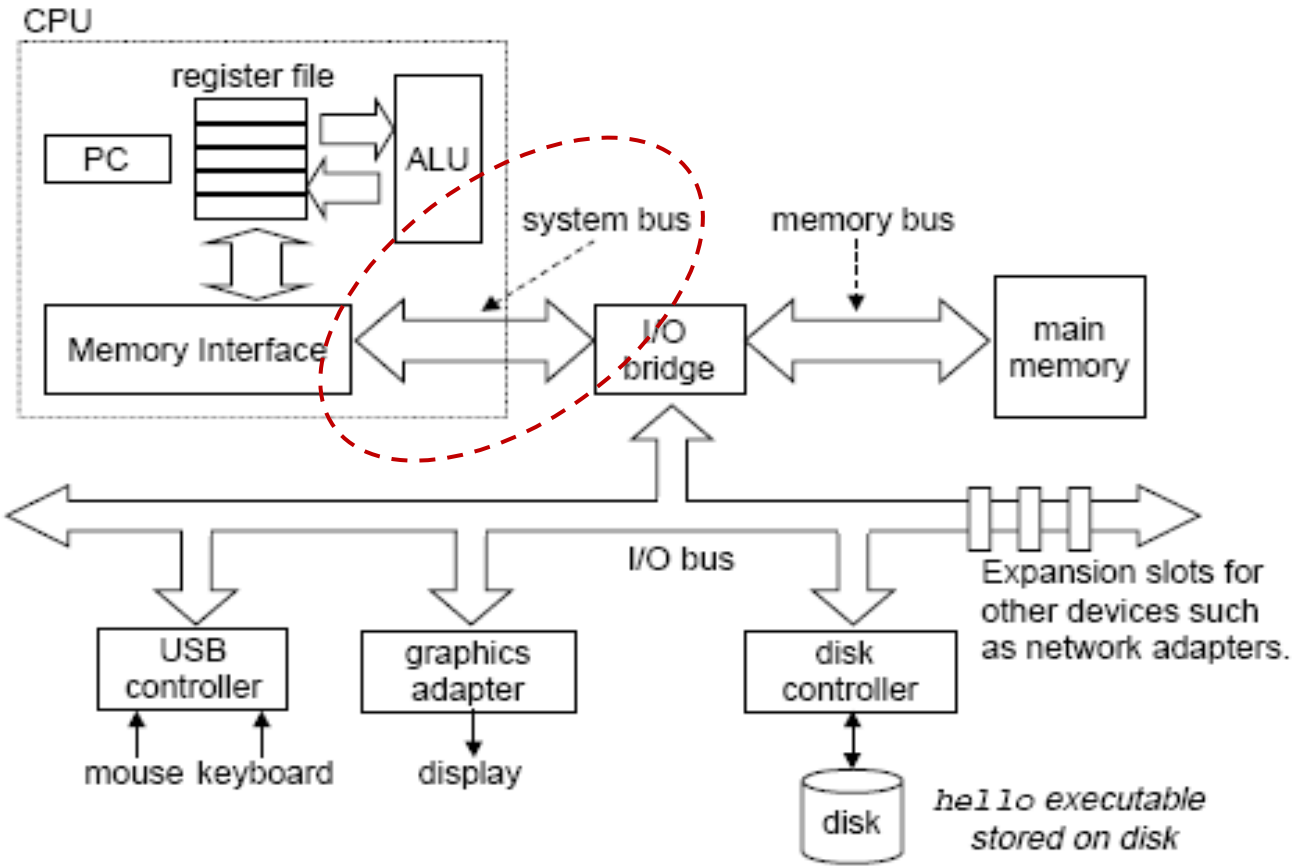
Review

- Memory management
- Exchange and overlap technique
  - Partitioning and paging
- Virtual memory
  - Page based VM, segment based VM, segment and page based VM

Type

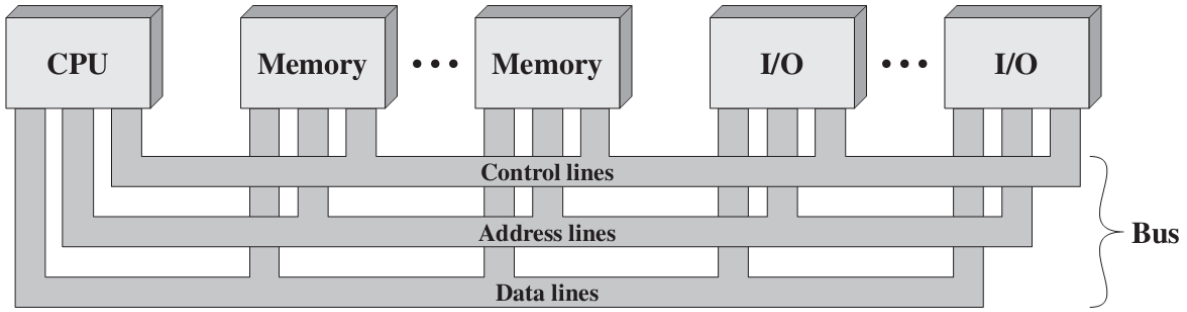
- Chip inner bus
  - Connect the parts inner a chip
  - E.g. connect registers, ALU and other parts in CPU
- **System bus**
  - Connect CPU, memory, I/O controller and other functional devices
- Communication bus
  - Connect host and I/O devices or connect different computer systems

“System Bus” in Intel Architecture



Component

- Bus lines can be classified into three functional groups
  - Data lines: move data between system modules
    - The number of data lines determines the size of data can be transferred at one time
  - Address lines: designate the source or destination of the data on the data bus and address I/O ports
    - The number of address lines determines the size of addressing space



Component (cont.)

- Control lines: control the access to and the use of the data and address lines
  - Clock: bus synchronization
  - Bus request: the device sending the signal requests bus
  - Bus grant: the device receiving the signal can use bus
  - Interrupt request: some interrupt requests
  - Interrupt acknowledge: some interrupt request is permitted
  - Memory read: read data from memory to bus
  - Memory write: write data from bus to memory
  - I/O read: read data from I/O port to bus
  - I/O write: write data from bus to I/O port

Design Element

- Bus type
  - Dedicated, multiplexed
- Arbitration
  - Centralized, distributed
- Timing
  - Synchronous, asynchronous, semi-synchronous, split transaction
- Bus bandwidth and data transfer rate
- Bus hierarchy
  - Single bus, double bus, multiple bus

Bus Type

- Dedicated bus: permanently assigned either to one function or to a physical subset of computer components
  - Advantage: high throughput for there is less bus contention
  - Disadvantage: the increased size and cost
- Multiplexed bus: use the same lines for multiple purposes
  - Advantage: fewer lines to save space and cost
  - Disadvantage: need more complex circuitry within each module and potentially reduce performance for sharing

Arbitration

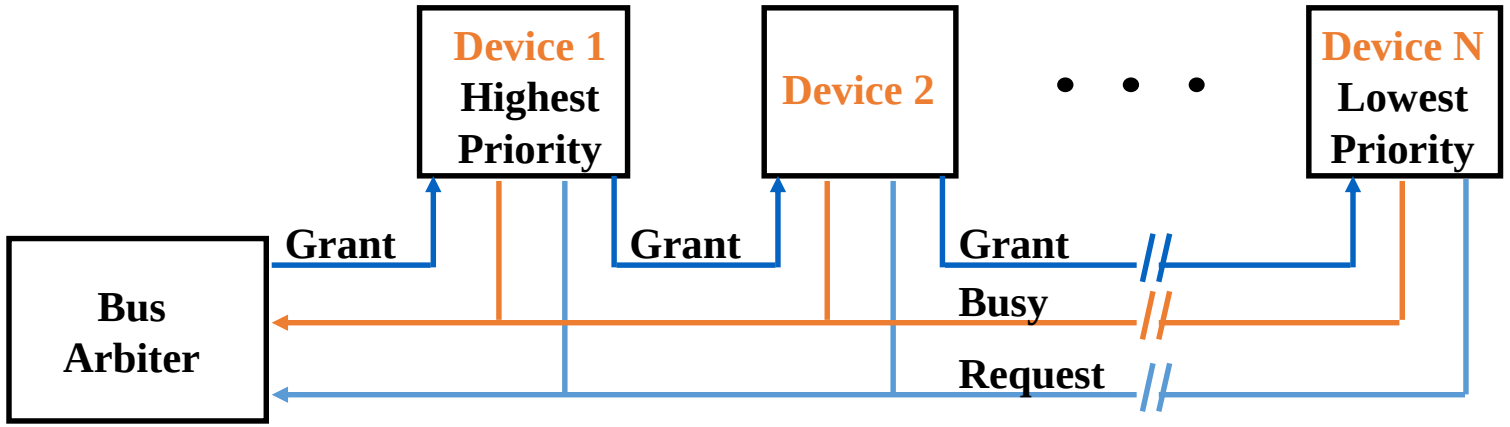
- A bus can be listened by multiple devices but only sent information by one device
- Bus arbitration
  - Choose one device by some strategy when multiple devices want to communicate with bus
- Balance
  - Priority: the device with higher priority should be serviced first
  - Fairness: the devices with the lowest priority cannot be delayed forever

Arbitration Type

- Centralized scheme: a single hardware device, referred to as a bus controller or arbiter, is responsible for allocating time on the bus
  - Daisy chain
  - Query by a counter
  - Independently request
- Distributed scheme: each module contains access control logic and the modules act together to share the bus
  - Self selection
  - Collision detection

Daisy Chain

- All the devices are connected serially, and grant is delivered from the highest priority device to the lowest priority device
- If one device receives the grant and has bus request, the device sets the bus busy and the grant is not further delivered

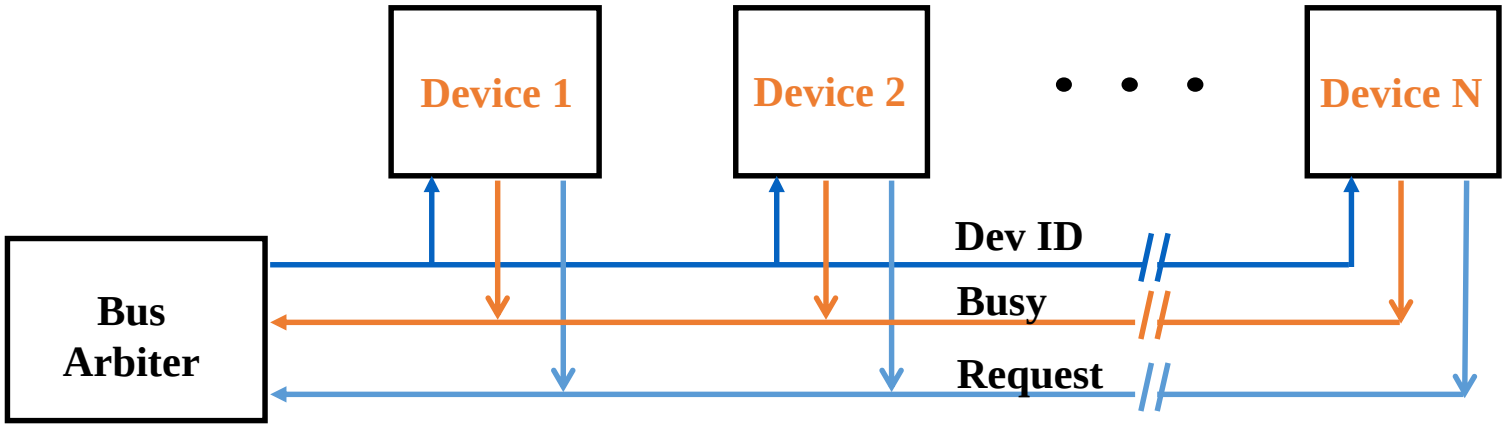


Daisy Chain (cont.)

- Advantage
  - Simple to determine the priority
  - Flexible to add devices
- Disadvantage
  - Cannot ensure the fairness
  - Sensitive to circuit fault
  - Limit the bus speed

Query By A Counter

- Remove bus grant line, and use devices ID lines
- If the bus is not busy, the bus arbiter sends the count through device ID lines
- If one device requests the bus whose ID equals to the current count, the arbiters stops counting and the device sets the bus busy



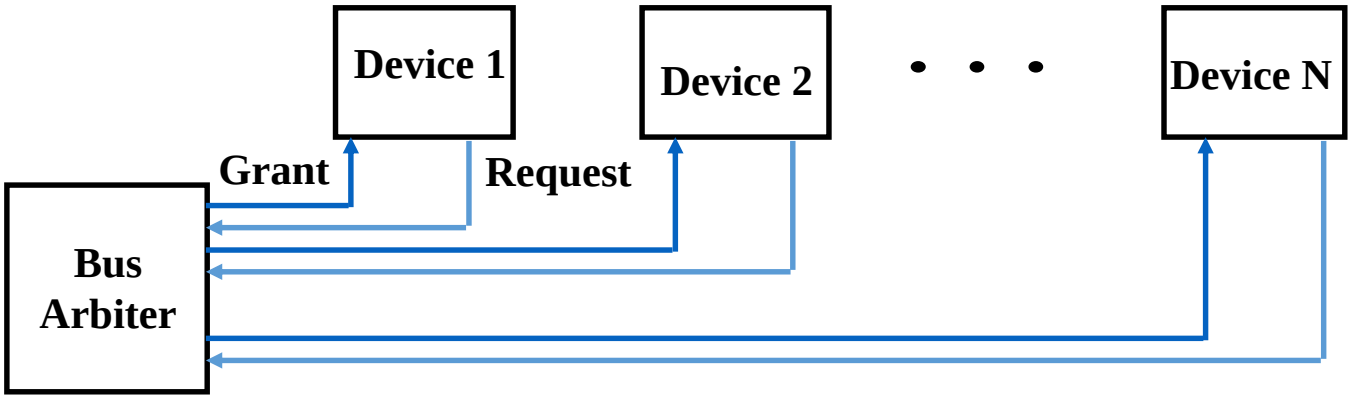
[倪安松, 141250095]

Query By A Counter (cont.)

- Advantage
  - Flexible to determine the device priorities by using different initial counts
    - Fixed priority: 0
    - Fair priority: the ID following the device used the bus
  - Not sensitive to circuit fault
- Disadvantage
  - Add the device ID lines
  - Require to decode and compare device ID signal

Independently Request

- Each device has its bus request line and bus grant line
- When one device request the bus, it sends the request signal to bus arbiter through the bus request line
- Bus arbiter determines which device can use the bus
  - Determination strategy: fixed priority, fair daisy chain, LRU, FIFO, …

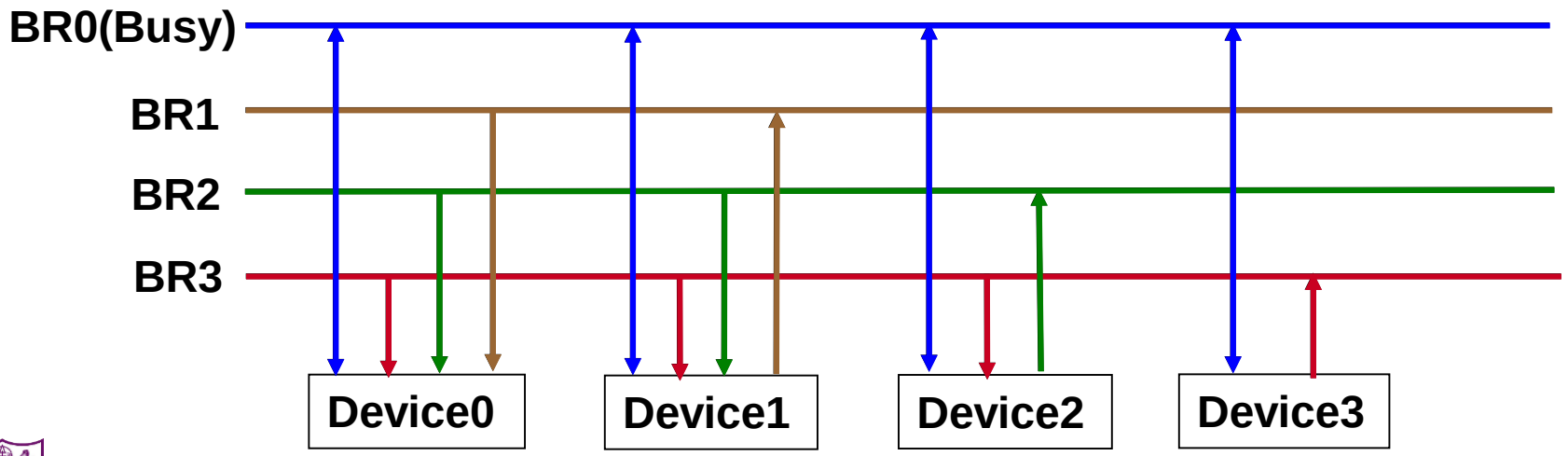


Independently Request (cont.)

- Advantage
  - Fast response
  - Programming priority
- Disadvantage
  - Complex control logic
  - More control lines

Self Selection

- Fixed priority
- Each device sends the request on its bus request line
  - The lowest priority device doesn't have request line
- Each device independently determines whether it is the device with the highest priority



Collision Detection

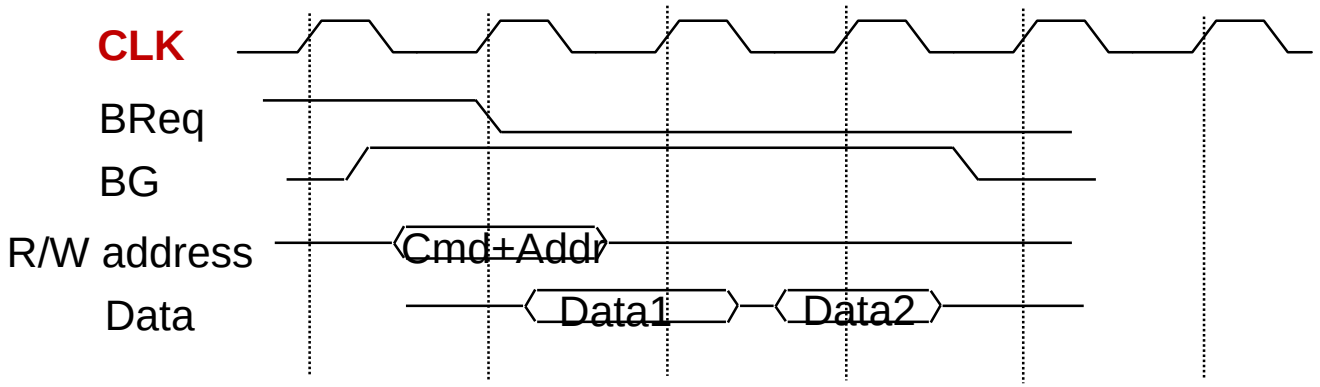
- When one device wants to use the bus, it checks whether the bus is busy
  - If bus is not busy, the device uses the bus
- Collision: if two devices found the bus is not busy, they may use the bus at the same time
  - In data transfer, the device will listen to the bus and check whether exists a collision
  - If a collision happens, all the devices using bus will stop data transfer and request the bus after a random duration

Timing

- The way in which events are coordinated on the bus
  - Determine the start and end time of each bus transaction
- Type
  - Synchronous timing: the occurrence of events on the bus is determined by a clock
  - Asynchronous timing: the occurrence of one event on a bus follows and depends on the occurrence of a previous event
  - Semi-Synchronous: combine synchronous timing (clock) and asynchronous timing (handshaking)
  - Split bus transaction: release the bus when the devices prepare data

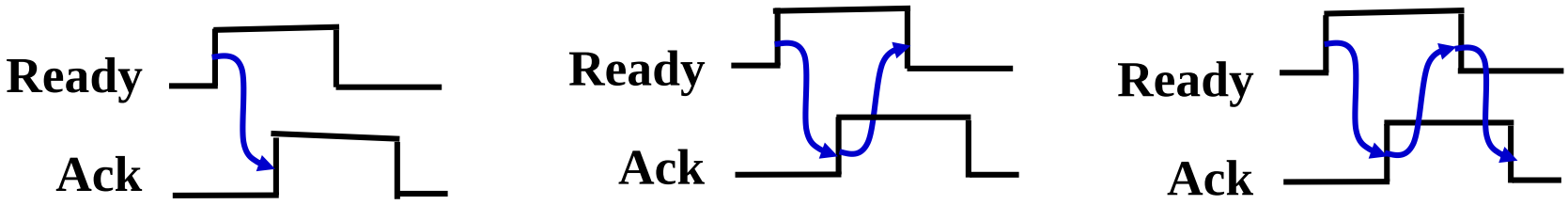
Synchronous

- Bus transaction: address + data + … + data
- Advantage
  - Simpler to implement and test
- Disadvantage
  - All the device share the same clock
  - Bus length is limited for clock skew



Asynchronous

- Bus transaction: address + data + … + data
- Use handshaking protocol

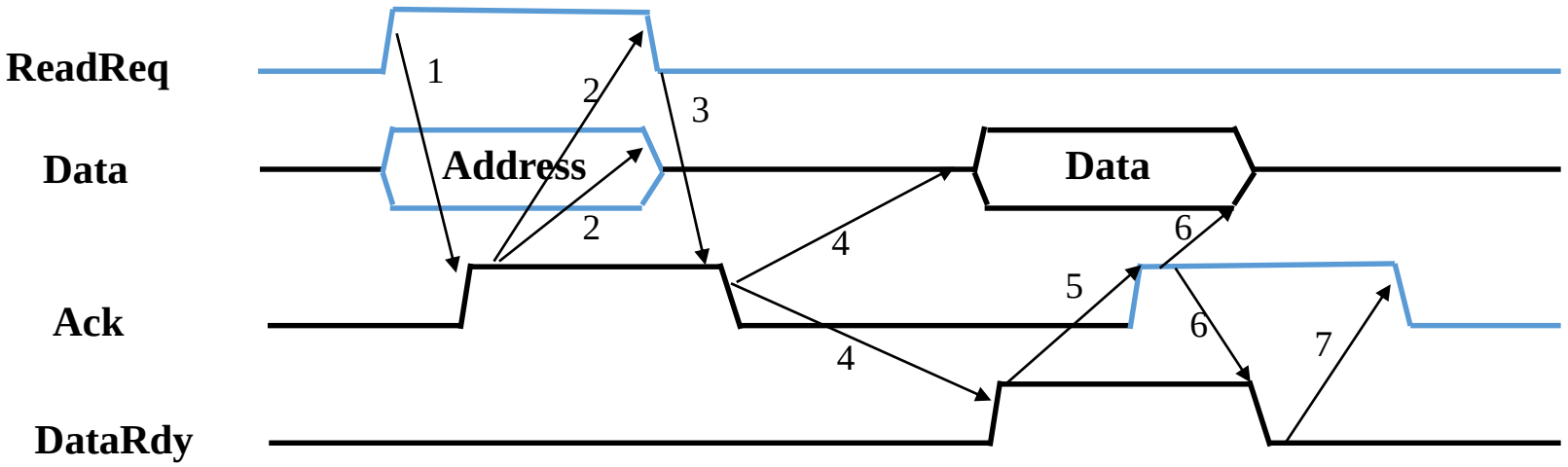


No interlock | Semi-interlock | Interlock

- Advantage
  - Flexible to the devices with different speeds
- Disadvantage
  - Sensitive to noise
  - Complex interface logic

Asynchronous (cont.)

- Handshaking example



1 CPU puts address and sets **ReadReq line** ➔ **Memory reads address**

2 Memory reads address and sets **Ack line** ➔ **CPU releases address lines and ReadReq line**

3 CPU release address lines and **ReadReq line** ➔ **Memory releases Ack line**

4 Memory releases **Ack line** ➔ **Memory puts data on data lines**

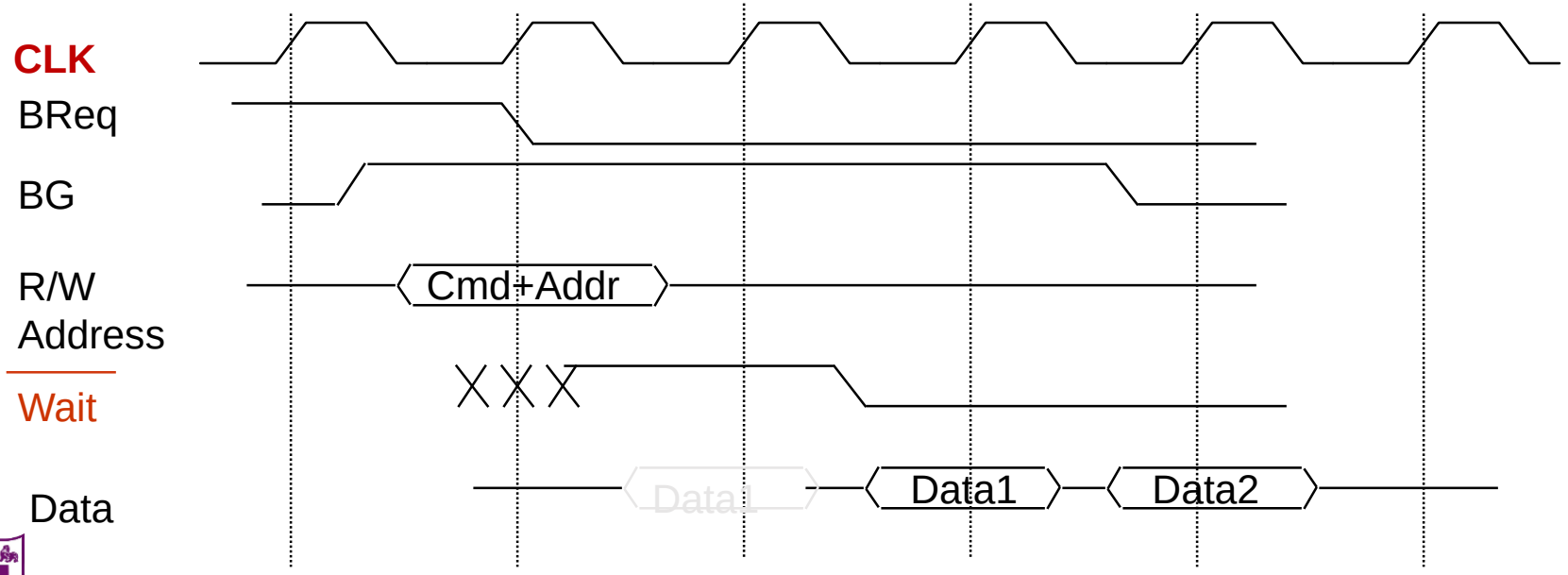
5 Memory puts data on data lines and sets **DataRdy line** ➔ **CPU reads data**

6 CPU reads data and sets **Ack line** ➔ **Memory release data lines and DataRdy line**

7 Memory releases data and **dataRdy line** ➔ **CPU releases Ack lines**

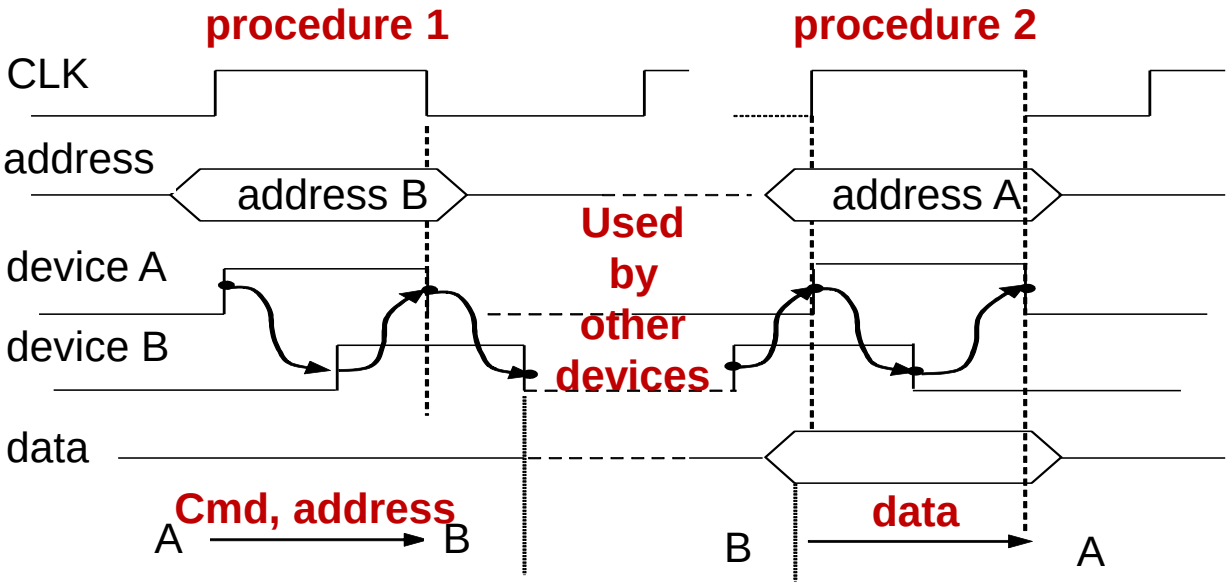
Semi-Synchronous

- To reduce the influence of noise, use clock in asynchronous timing
  - Ready and reply signals are valid in the rising edges of clock
- Combine the advantage of synchronous and asynchronous timing



Split Bus Transaction

- Split one bus transaction into two procedures
- Advantage: increase the bus utilization
- Disadvantage: increase the duration of each bus transaction and system complexity



Bus Bandwidth and Data Transfer Rate

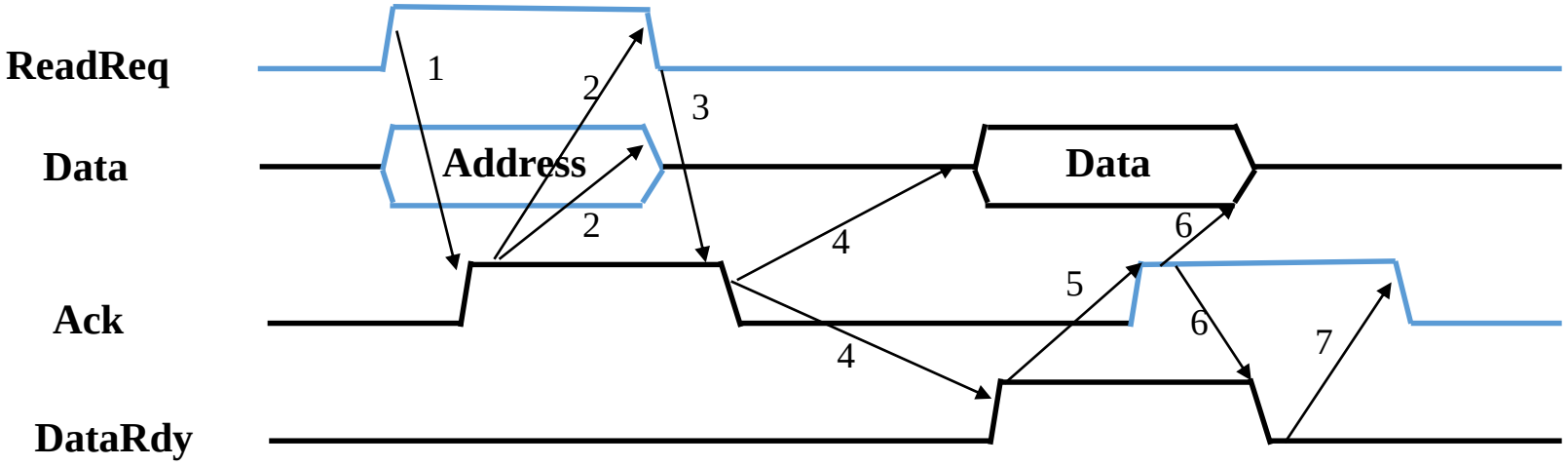
- Bus bandwidth: the maximum data transfer rate of bus
  - Don’t consider bus arbitration, address transfer, …
- Data transfer rate
  - Consider address transfer, …
- Bus width: the number of lines to compose a bus
  - The wider the data bus, the greater the number of bits transferred at one time
  - The wider the address bus, the greater the range of locations that can be referenced

Data Transfer Rate of Synchronous Bus and Asynchronous Bus

- Assume the clock cycle of synchronous bus is 50ns and each transfer requires one clock cycle, asynchronous bus requires 40ns for each handshaking. Both buses have 32 bits width, and the data prepare time of memory is 200ns. Calculate the data transfer rates of the two buses when read one word (32 bits) from memory.
- Synchronous bus
  - Sends command and address to memory: 50ns
  - Memory prepares data: 200ns
  - Transfers data to CPU: 50ns
  - Data transfer rate = 32bit / (50 + 200 + 50)ns = 106.7Mbps

Data Transfer Rate of Synchronous Bus and Asynchronous Bus (cont.)

- Asynchronous bus
  - Step 1: 40ns
  - Step 2, 3, 4: max(40ns*3, 200ns) = 200ns
  - Step 5, 6, 7: 40ns * 3 = 120ns
  - Data transfer rate = 32bit / (40 + 200 + 120)ns = 88.9Mbps



Data Transfer Rate of Synchronous Bus and Asynchronous Bus (cont.)

- Assume the clock cycle of synchronous bus is 50ns and each transfer requires one clock cycle, asynchronous bus requires 40ns for each handshaking. Both buses have 32 bits width, and the data prepare time of memory is **230**ns. Calculate the data transfer rates of the two buses when read one word (32 bits) from memory.
- Synchronous bus
  - Data transfer rate = 32bit / (50 + **250** + 50)ns = 91.4Mbps
- Asynchronous bus
  - Data transfer rate = 32bit / (40 + **230** + 120)ns = 82.1Mbps

[吴超月, 131250168]

Data Transfer Rate with Different Data Block Size

- Assume a system has the following characteristic:
    1. It supports to access the blocks with the size of 4 to 16 words (32 bits per word)
    2. Synchronous bus has the 64 bits width and 200MHz clock frequency, which requires one clock cycle to transfer address or 64 bits data
    3. There are two idle clock cycles between two bus transactions
    4. The access time of memory requires 200ns for the first 4 words, and then reduces to 20ns per 4 words
    5. When the previous data is transferred on bus, memory is reading the following data at the same time

If read 256 words, calculate the data transfer rate, transfer time and bus transactions number per second when transfer 4 words and 16 words each time, respectively

Data Transfer Rate with Different Data Block Size (cont.)

- Transfer 4 words each time
  - Bus transaction: address + 4 words data
    - Address transfer: 1 cycle
    - Data read: 200ns (40 cycles)
    - Data transfer: 2 cycles
    - Idle: 2 clock cycles
  - Total: 256 / 4 * (1 + 40 + 2 + 2) = 2880 cycles
  - Transfer time = 2880 * 5 = 14400ns
  - Bus transaction number per second
    = 64 * 1s / 14400ns = 4.44M
  - Data transfer rate = 256 * 4B / 14400ns = 568.9Mbps

Data Transfer Rate with Different Data Block Size (cont.)

- Transfer 16 words each time
  - Bus transaction: address + 16 words data
    - Address transfer: 1 cycle
    - Data read (first 4 words): 200ns (40 cycles)
    - Data transfer: 2 clock cycles **(read the following 4 words)**
    - Idle: 2 clock cycles **(finish the following 4 words read)**
  - Total: 256 / 16 * (1 + 40 + 3 * max(4, 2) + 2 + 2) = 912 cycles
  - Transfer time = 912 * 5 = 4560ns
  - Bus transaction number per second
    = 16 * 1s / 4560ns = 3.51M
  - Data transfer rate = 256 * 4B / 4560ns = 1796.5Mbps

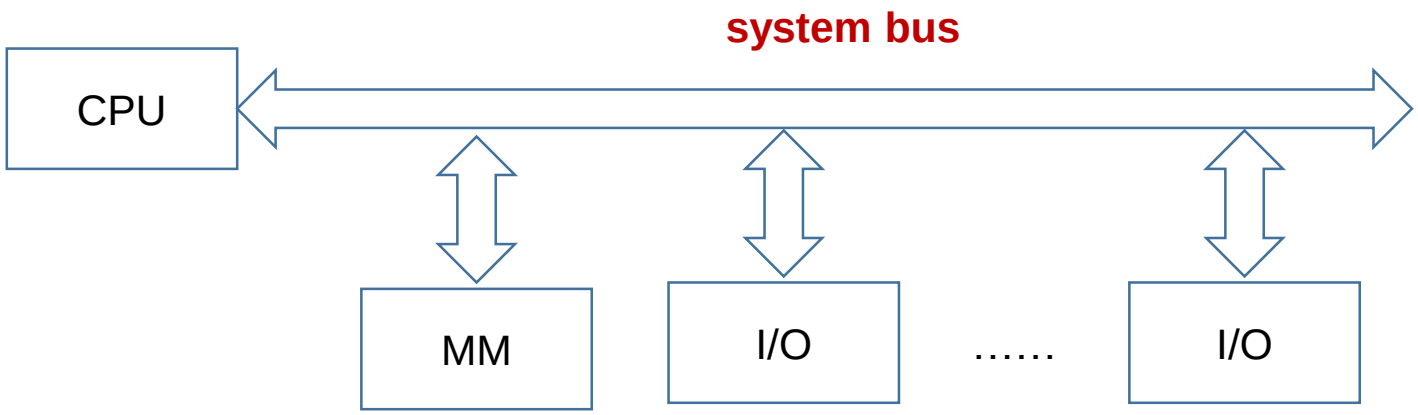
[刘璟, 121250083]

Increase Bandwidth Of Synchronous Bus

- Increase clock frequency
- Increase data bus width
  - Transfer more data each time (cost: more bus lines)
- Block transfers
  - Transfer address once and transfer data (cost: high complexity)
- Split bus transaction
  - Increase system bandwidth (cost: high complexity, enlarge the duration of each transaction)
- Separate address lines and data lines
  - Transfer address and data simultaneously (cost: more bus lines, high complexity)

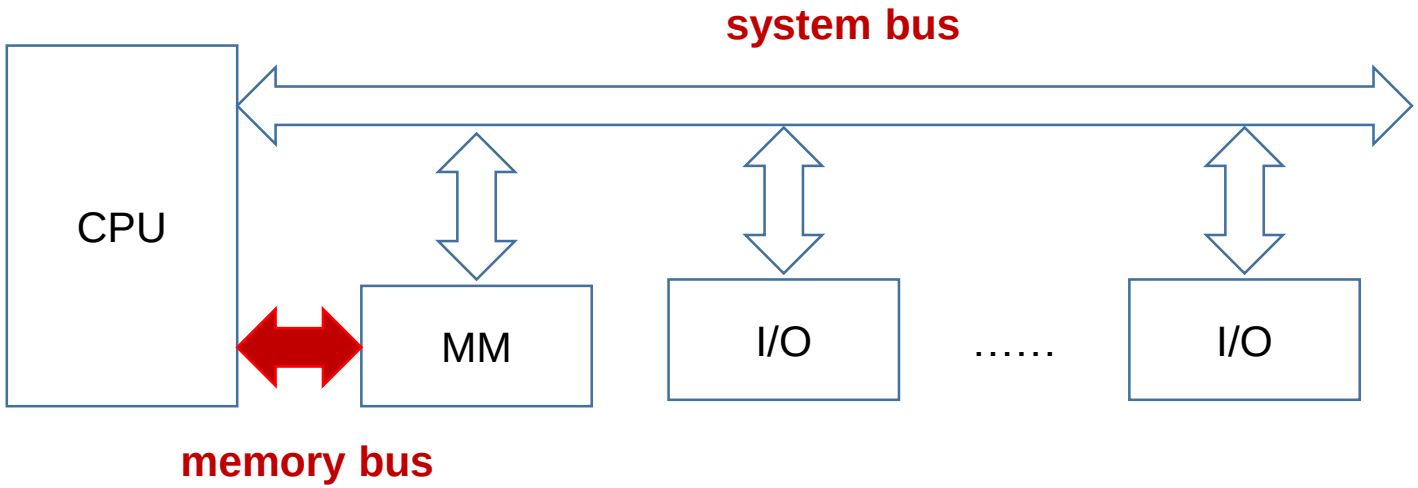
Bus Hierarchy

- Single bus hierarchy
  - CPU, memory and I/O modules attach to **system bus**
  - Advantage: simple, easy to extend
  - Disadvantage: bus is the bottleneck



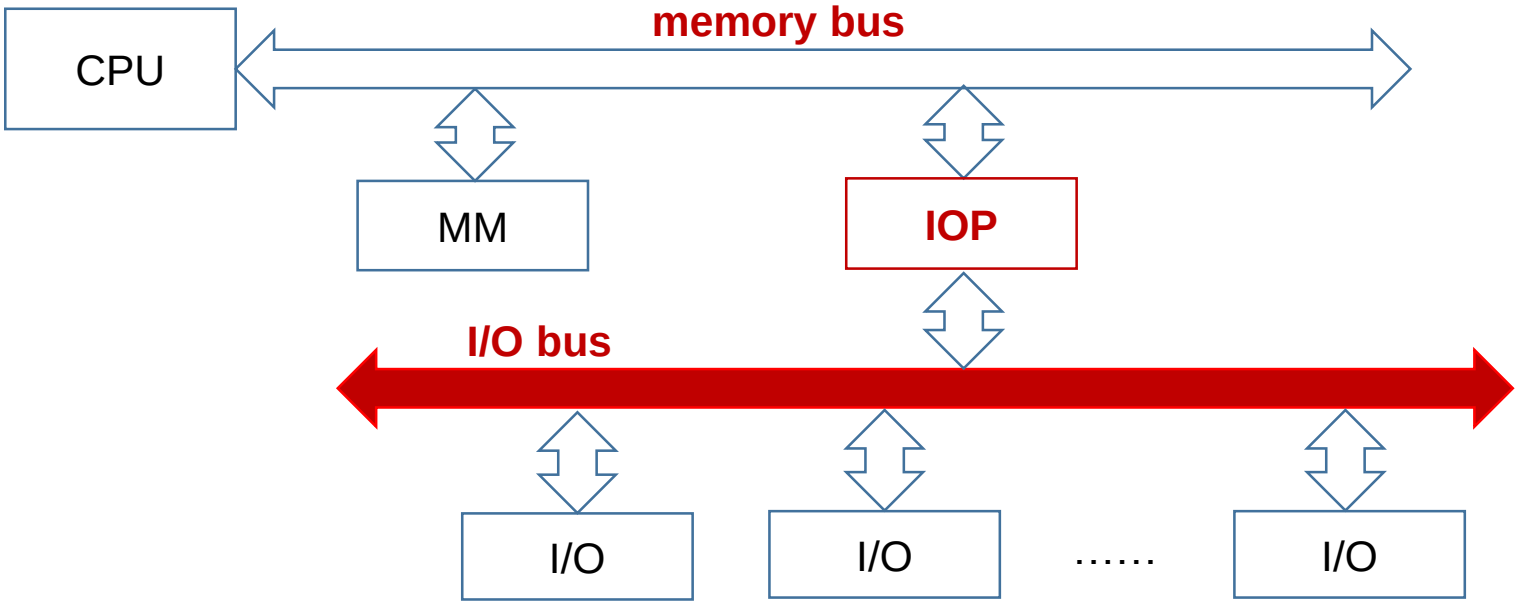
Bus Hierarchy (cont.)

- Double bus hierarchy
  - Add a **memory bus** between CPU and memory
  - Advantage: increase the transfer efficiency between CPU and memory, and reduce the burden of system bus



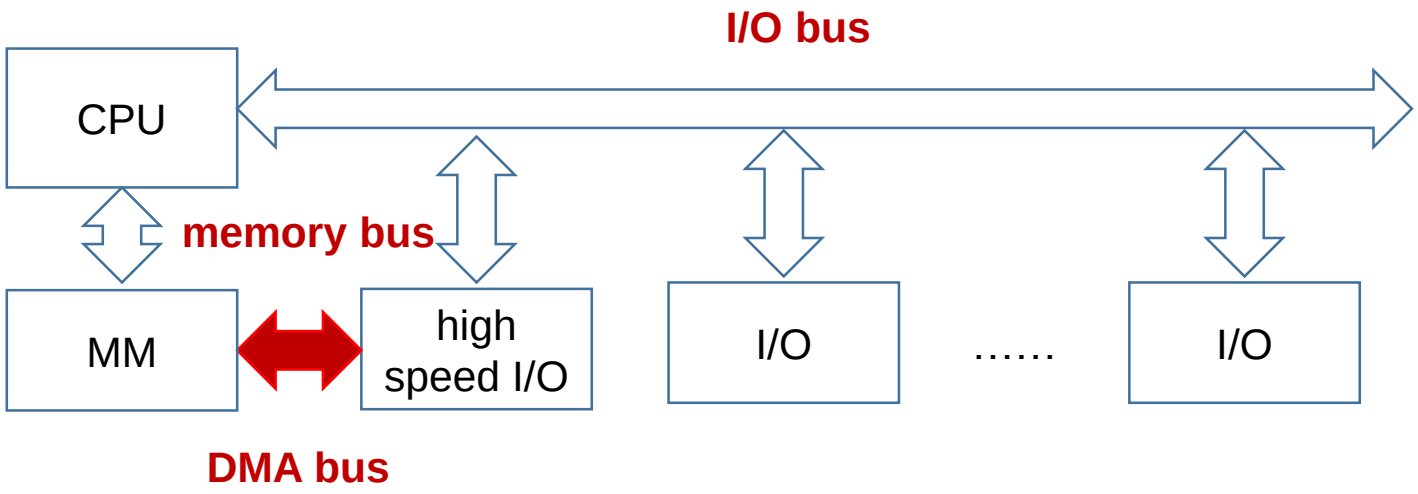
Bus Hierarchy (cont.)

- Double bus hierarchy (cont.)
  - Split system bus to memory bus and **I/O bus** with **IOP** (input/output processer)
  - Advantage: reduce the burden of I/O



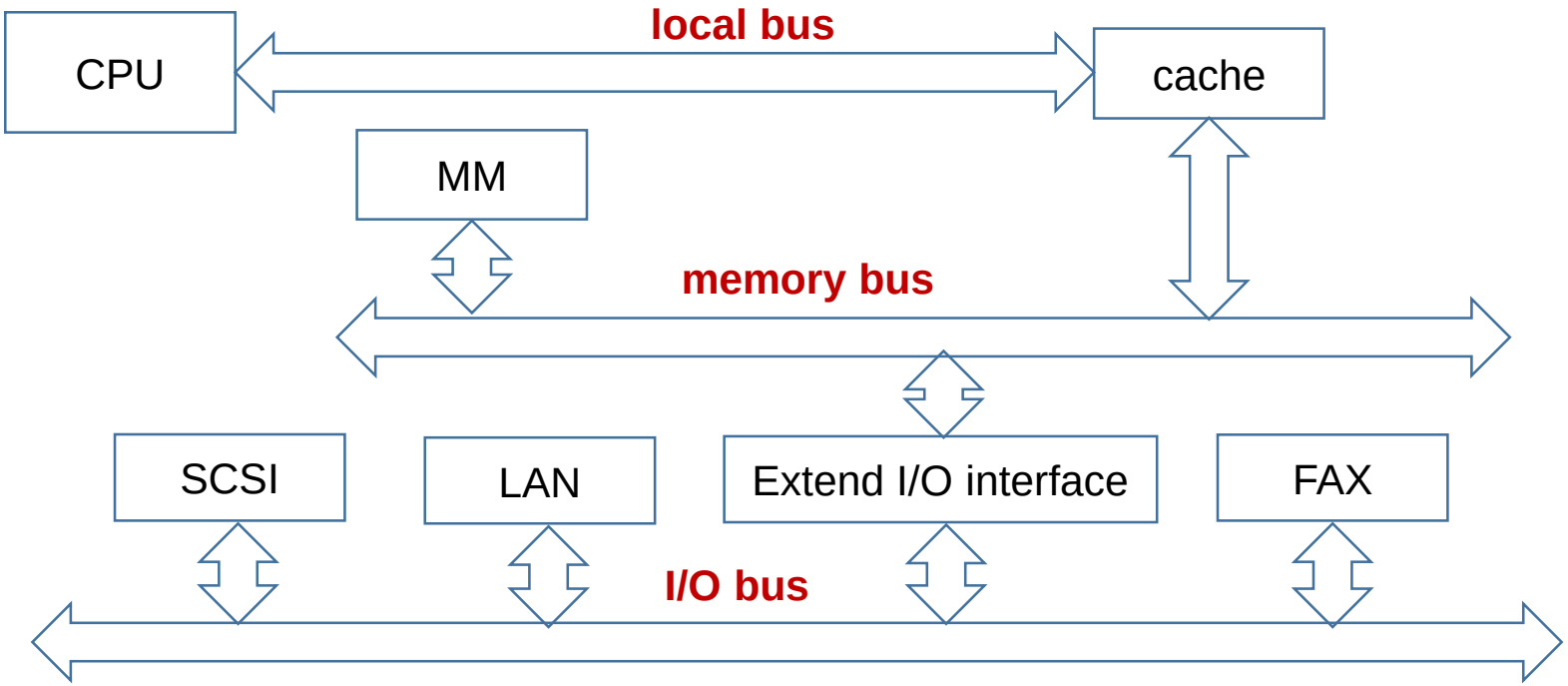
Bus Hierarchy (cont.)

- Multiple bus hierarchy
  - Split system bus to memory bus and I/O bus, and add **DMA bus**
  - Advantage: increase I/O efficiency



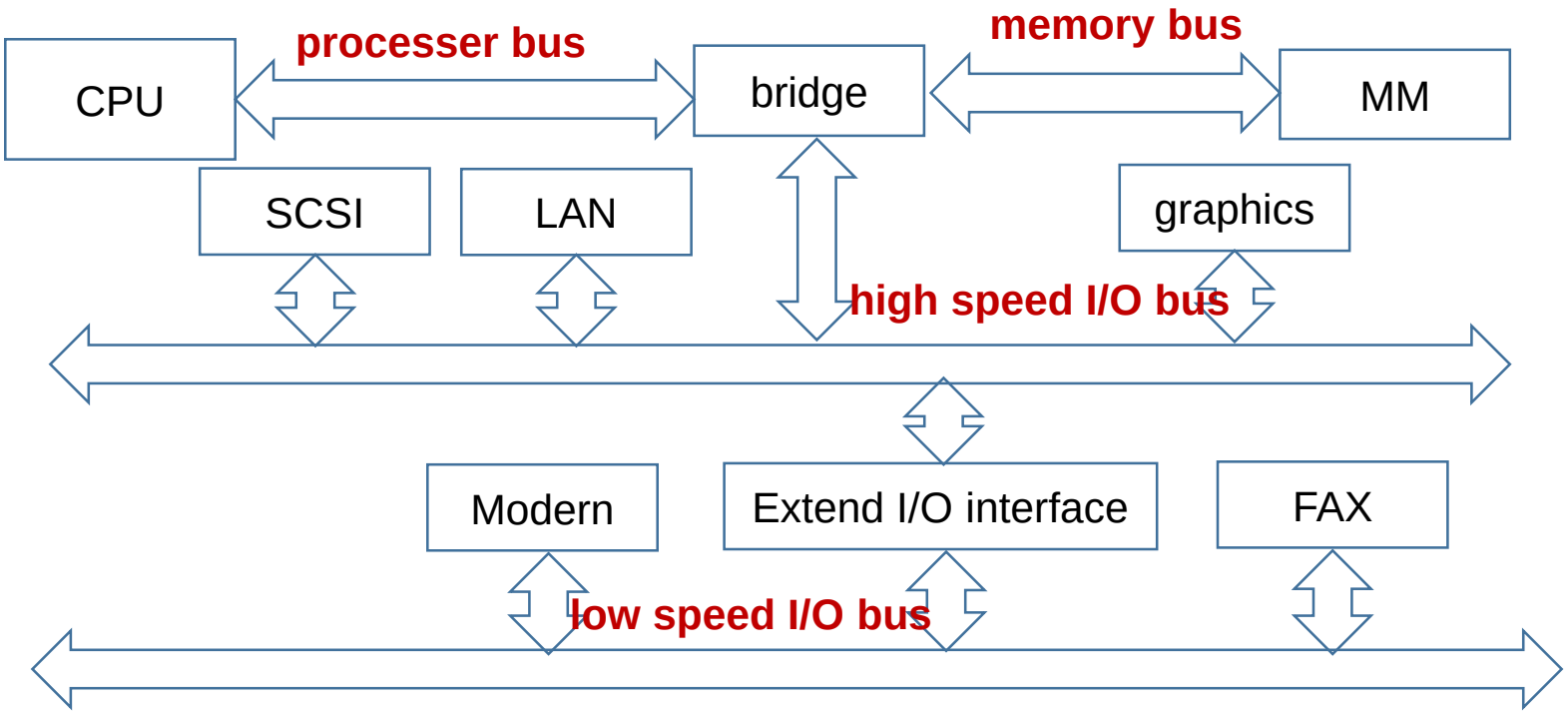
Bus Hierarchy (cont.)

- Multiple bus hierarchy (cont.)
  - Add a **local bus** to connect CPU and cache
  - Advantage: separate CPU with I/O communication



Bus Hierarchy (cont.)

- Multiple bus hierarchy (cont.)
  - Add a **high speed I/O bus** to connect high speed devices
  - Advantage: increase I/O communication efficiency



Summary

- Bus
  - Concept, characteristic, type, component
- Design Element
  - Bus type: dedicated, multiplexed
  - Arbitration: centralized, distributed
  - Timing: synchronous, asynchronous, semi-synchronous, split bus transaction
  - Bus bandwidth and data transfer rate
  - Bus hierarchy: single bus, double bus, multiple bus

Thank You

rentw@nju.edu.en

南京大學
NANJING UNIVERSITY