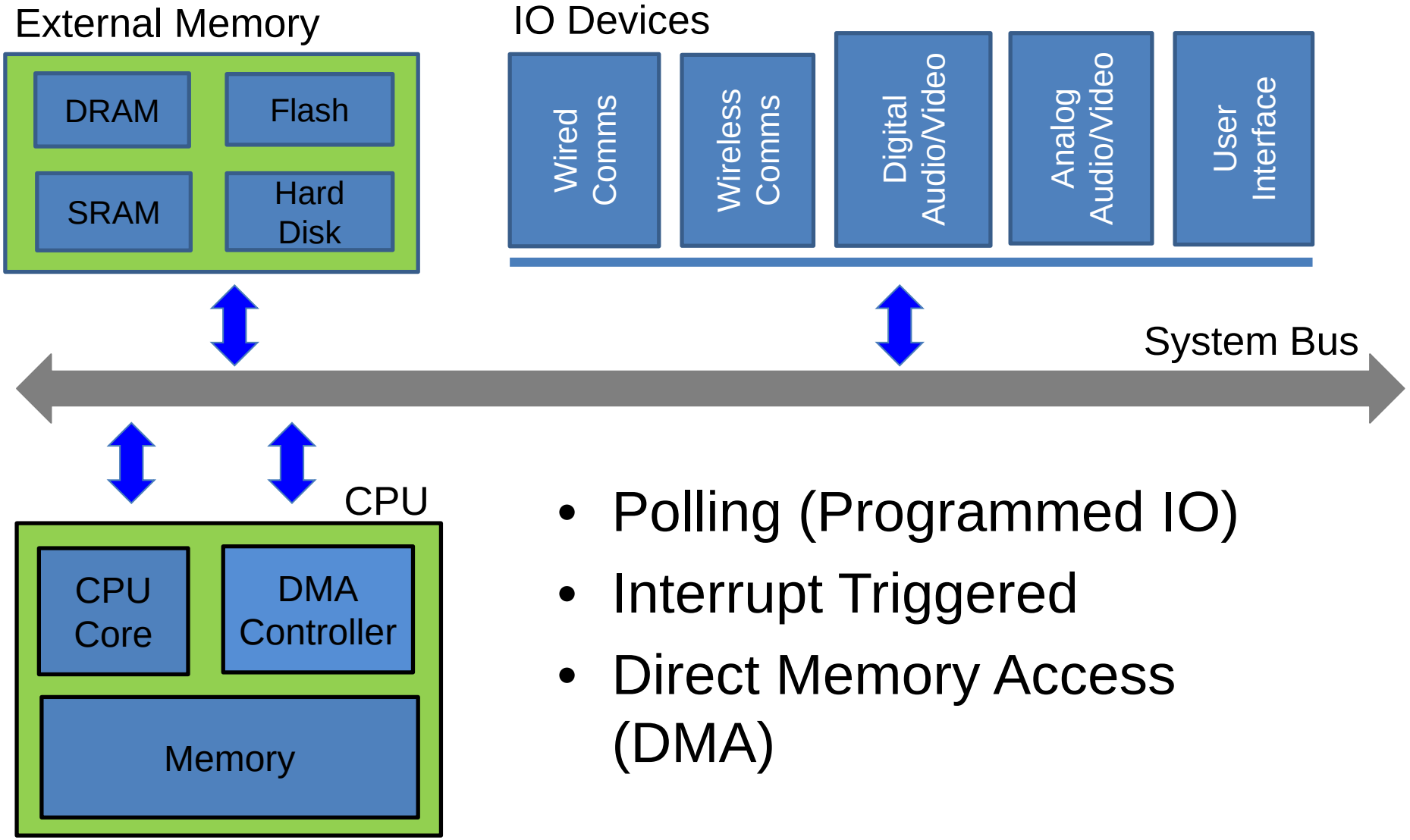
CE1006/CZ1006
Computer Organisation and Architecture

Polling, Interrupts and DMA Mechanism (Part 1)

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

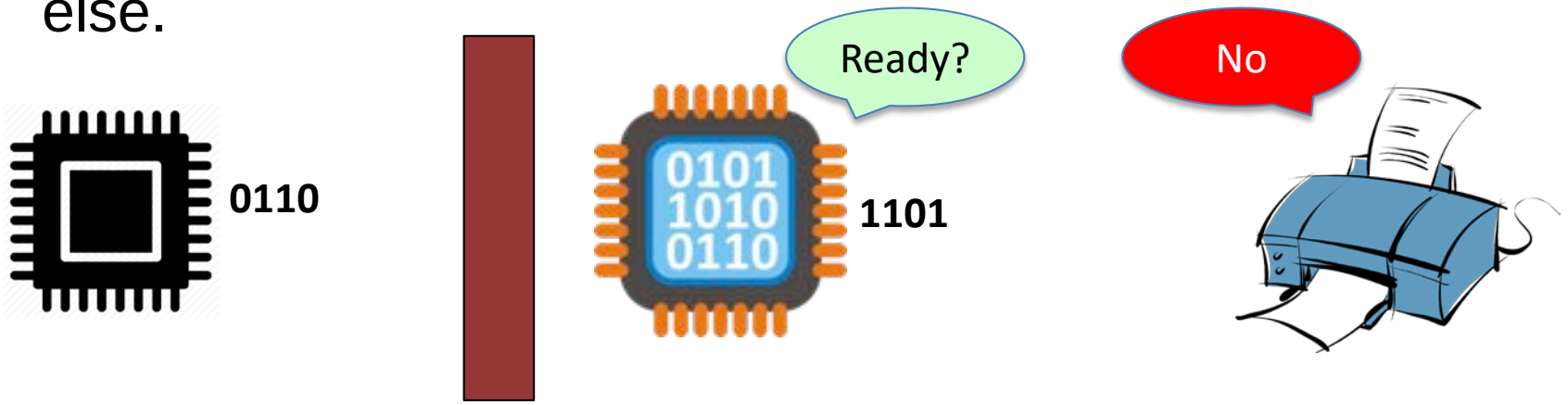
Data Transfer Mechanism



- Polling (Programmed IO)
- Interrupt Triggered
- Direct Memory Access (DMA)

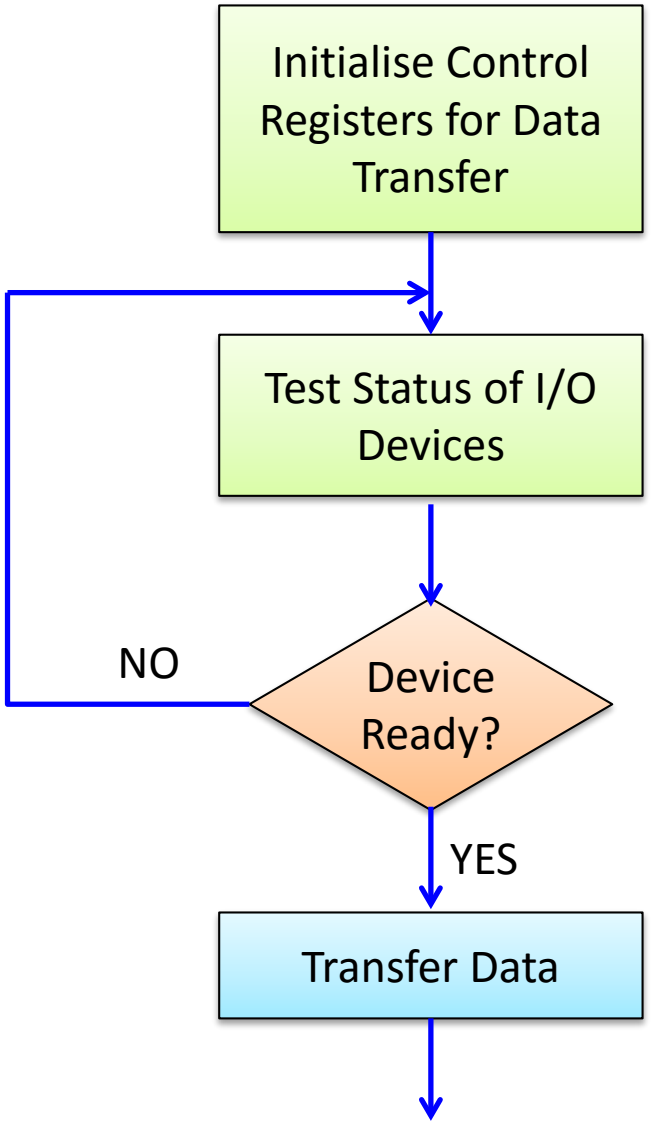
Polling Technique

- Also known as Programmed I/O
- CPU polled a certain IO port continuously for data or readiness of the port to perform a data transaction.
  - For example, the CPU continuously polls the printer port to see if the printer is ready to accept data.
  - If it is ready, the CPU writes a data byte to the printer port. Else, it waits.
- CPU has full control and dedicate 100% of its resource in the whole data transfer process and does nothing else.



Polling Technique - Flowchart

- CPU performance all the necessary initialisation.
- CPU poll the I/O device for its readiness to perform data transfer.
- If I/O device is not ready, CPU continue to wait in the loop to check if device is ready.
- If device is ready, CPU make the data transfer and exit the loop.



Pros/Cons of Polling Technique

- Advantages
  - Minimum hardware interface circuitry between I/O device and processor.
  - Programmer has complete control over the entire process.
  - Easiest method to test and debug.
- Disadvantages
  - Since the CPU waits in a loop, it cannot perform any other task until data transfer is completed.
  - Inefficient use of CPU resources.
  - Program execution of CPU held up while waiting for I/O device to get ready.

CE1006/CZ1006
Computer Organisation and Architecture

Polling, Interrupts and DMA Mechanism (Part 2)

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

Polling vs Interrupt

- Boiling Water Analogy

Polling:
Check every few minutes

Interrupt:
Listen for the whistle

Interrupt Triggered Data Transfer

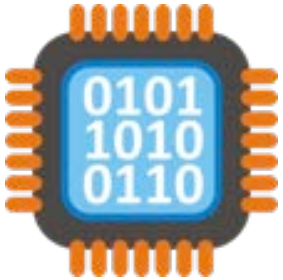
Main Routine

```
main  MOVS R0, #0
      MOV  R2, #12

Loop  ROL  R1
      JNC  skip
      INC  R0

skip  .....
```

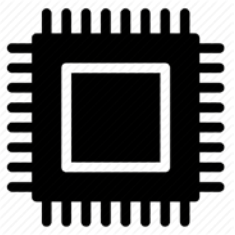
I/O Device #3
Interrupt Service
Routine



INTERRUPT REQUEST

DATA TRANSFER

I/O Device #3



Interrupt Control Flow

- A signal from I/O device notifying CPU that some event has occur and asking for CPU's (immediate) attention.
- If the CPU decide to service the interrupt, it'll suspend its current program temporarily.
- CPU then proceed to a execute a special routine know as the Interrupt Service Routine (ISR) linked to the particular I/O device that has triggered the interrupt.

Interrupt Service Routine

- The ISR will perform some operation e.g. Data Transfer.
- ISR is typically a very short routine so as not to suspend the main program for too long.
- After servicing the ISR, CPU will return to the previous program and continue from it branched off.
- It is possible for CPU to receive multiple interrupt requests simultaneously since it typically interface to multiple devices. In this case, some arbitration scheme has to be designed to decide which interrupt to service first (priority, first-come-first-serve etc).

Pros/Cons of Interrupt Triggered Technique

- Advantages
  - Efficient use of CPU resources as it does not need to monitor I/O device status.
  - CPU can continue with other tasks between interrupts.
  - Allows prioritisation and pre-emption.
- Disadvantages
  - More hardware interface circuitry required between I/O device and processor.
  - Program is slightly more complex and difficult to debug.

CE1006/CZ1006
Computer Organisation and Architecture

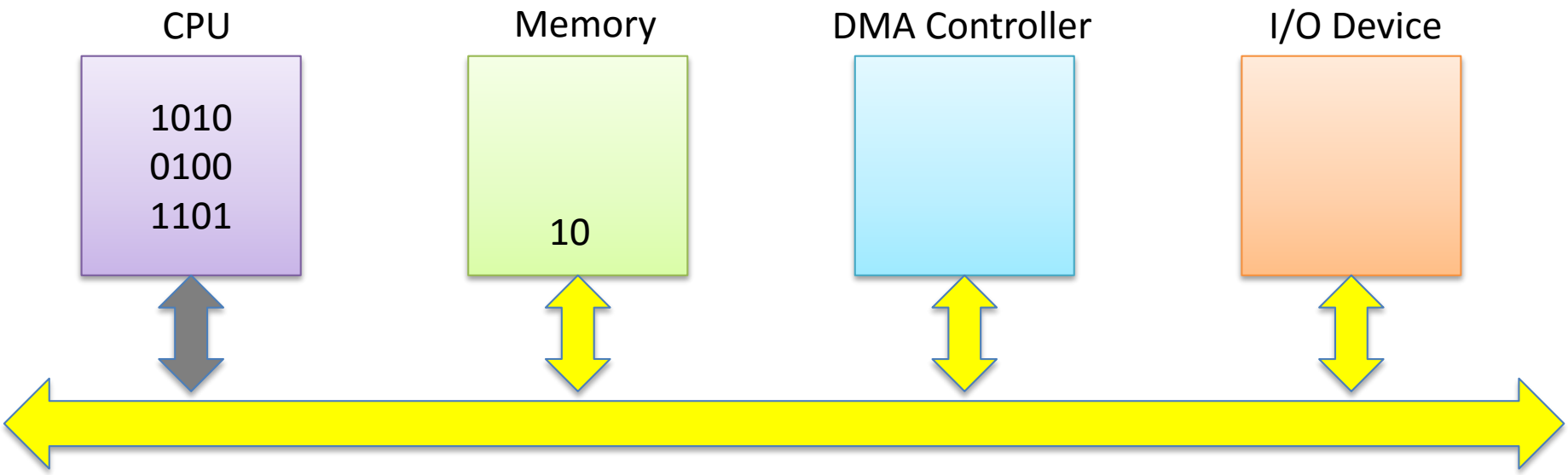
Polling, Interrupts and DMA Mechanism (Part 3)

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

Optimising CPU resources in Data Transfer

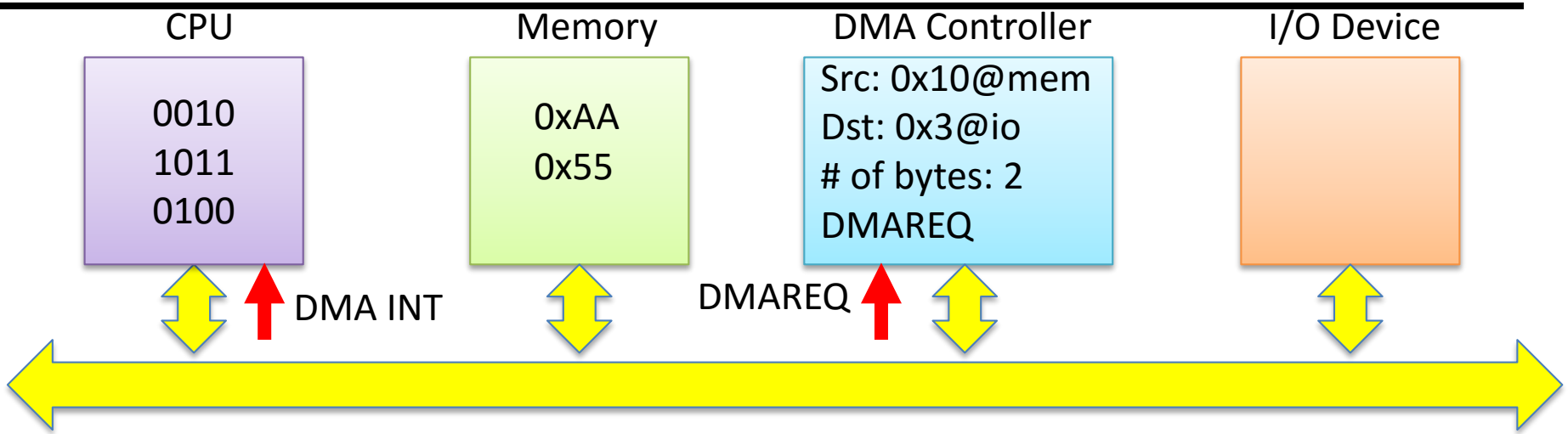
- Both Programmed I/O and Interrupt Triggered I/O requires CPU to perform the actual data transfer.
- This becomes increasing inefficient as amount of data increase as CPU has to spend most of its time moving data.
- As such, CPU has less time to perform algorithm processing.
- DMA controller is added to relief CPU of the data transfer task.
- DMA controller has dedicated hardware that could move data more efficiently than CPU.
- Depending on the situation, it is possible for data transfer via DMA and CPU execution to occur simultaneously.

DMA Controller (DMAC)



- DMAC is a Data Bus Controller module that performs data transfer independent of CPU, between
  - Memory and memory
  - I/O Peripheral and I/O peripheral
  - Memory and I/O Peripheral
- Generates address and initiates read/write operation between devices mentioned above.

Basic DMA Process



- DMA Configuration Parameters initialisation (by CPU to DMAC)
  - Source Address
  - Destination Address
  - Amount of data to transfer
  - DMA Trigger signal (DMAREQ, interrupt, software bit etc)
- CPU proceed with its own task while DMAC wait for DMA Trigger Signal.
- Once DMA Trigger occurs, DMAC request to take over system bus.
- DMAC transfer data according to configuration.
- DMAC notify CPU of data completion (typically via interrupts) and release system bus.

DMA Mode of Operation

- Different processors has different DMAC design, typically with slight variance in terms of how they transfer data and the data type (Audio, Video etc) they are optimised for.

- In general, there are 3 basic modes of transfer.
  - Burst
  - Cycle Stealing
  - Transparent

- Do note that you may encounter DMAC design that varies from the basic mode of operation discussed here but general concept will still apply.

Fetch-n-Deposit vs FlyBy DMA

- Most DMAC design uses internal buffers to manage the data transfer (Fetch and Deposit DMA).
  - Data is read into the DMAC before being transferred to the destination.

- One special type of DMA allows data to be transferred directly from the source to the destination. Its know as Fly-By DMA. More on that later.

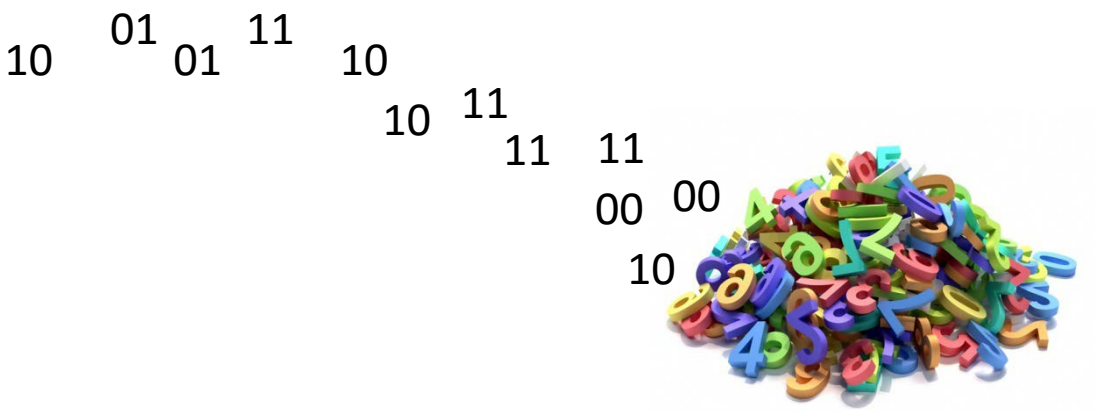
CE1006/CZ1006
Computer Organisation and Architecture

Polling, Interrupts and DMA Mechanism (Part 4)

Oh Hong Lye
Lecturer
SCSE, Nanyang Technological University.

Burst Mode

- Also known as Block Transfer Mode.
- The DMA controller gains control of the data bus and transfer a block of data before returning control of the bus to the CPU.
- CPU may continue to operate as long as it does not need access the particular data bus that DMAC is using. E.g. when it is taking data from the cache.
- If CPU needs to access the data bus, CPU may be suspended till DMAC has completed its data transfer.
- Fast data transfer rate but may render the CPU inactive for extended period of time.



Cycle Stealing

- DMAC releases the data bus after transferring one word of data.
- Depending on processor design, DMAC tends to execute the data transfer
  - Between CPU instructions
  - Between Pipeline stages
- CPU may be suspended if it need to access the data bus but suspend time is shorter as only one word is transferred at one time.
- Transfer rate is not as fast as that in Burst Mode but will not cause the CPU to be inactive for long period of time.
- Used in application which requires CPU to be responsive e.g. real-time security status monitoring.

Between CPU Instr

CPU	Instr1	Instr2		Instr3		Instr4
DMAC			Data		Data	

Between Pipeline Stages

CPU	Fetch Instr	Decode Instr		Fetch Operand	Exe Instr	
DMAC			Data			Data

Transparent Mode

- DMAC transfer data only when CPU is not using the data bus.
- Zero impact to CPU performance in terms of data bus access.
- Potentially the slowest transfer rate among the three modes.
- More complex hardware needed to detect when CPU is not using the data bus.

Comparison of DMA Transfer Modes

- Burst
  - Allow faster data transfer
  - CPU may be suspended for extended period of time
  - May not be suitable for Real-time application
  - Suitable for application where transfer is bursty e.g. HDD file transfer.

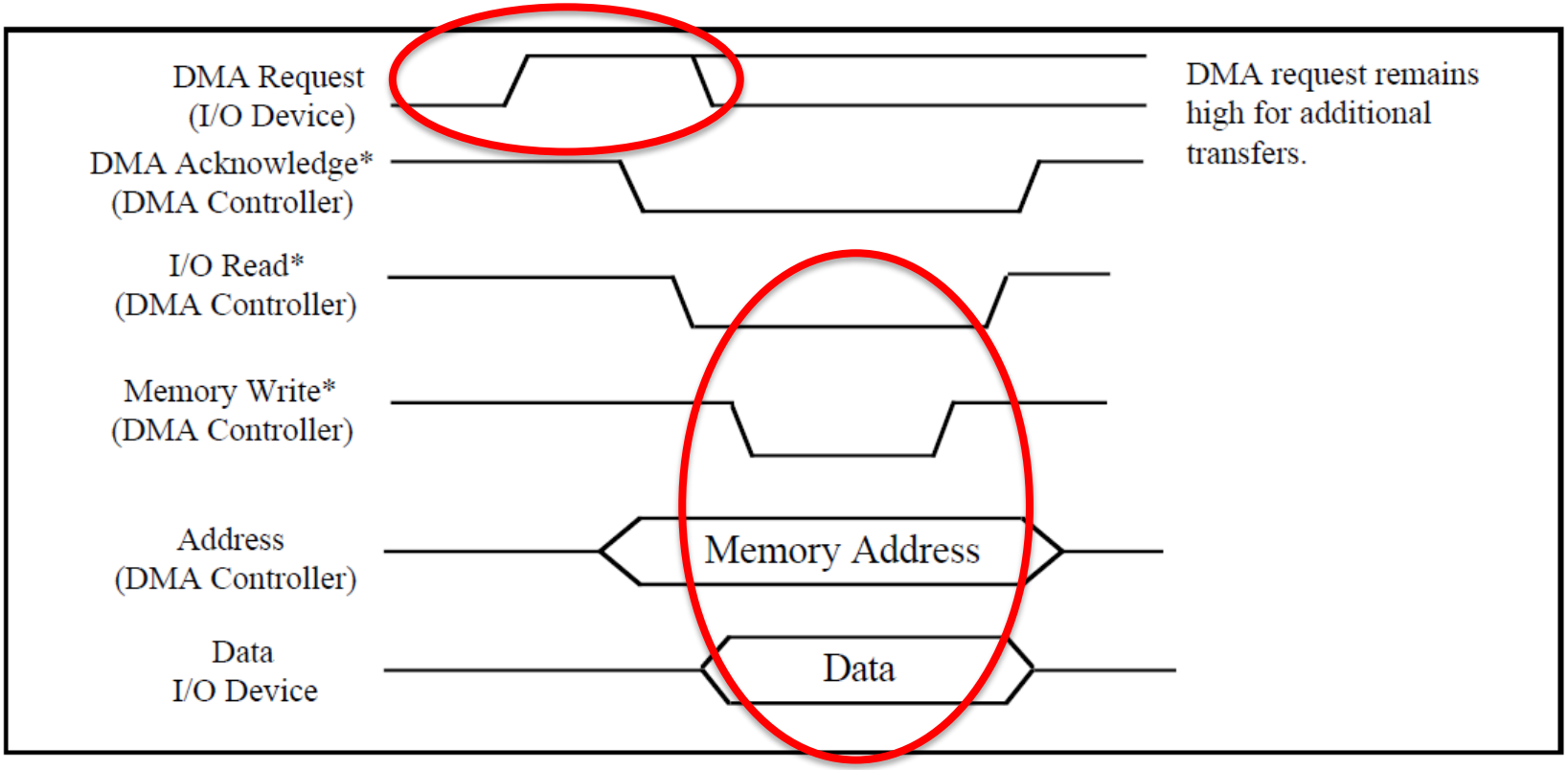
- Cycle Stealing
  - Interleaving DMA data transfer with CPU instructions allow CPU to continue executing its program while DMAC is doing the data transfer.
  - CPU performance lower due to interleaving of DMA transfers, but more responsive than in Burst mode.
  - Slower data transfer rate than Burst mode.
  - Suitable for Real-time application.

Comparison of DMA Transfer Modes

- Transparent
  - Does not affect CPU performance at all.
  - Slowest data transfer rate but best CPU respond
  - More complex hardware needed to detect when CPU is not using the bus.

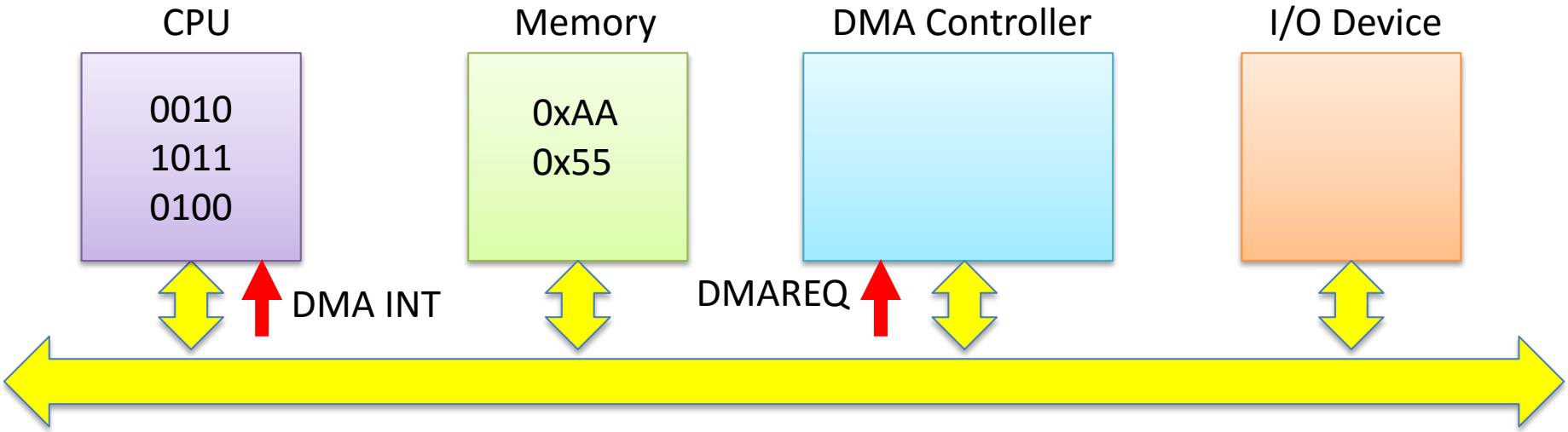
Fly-By DMA

- Data is transferred from the source to the destination directly (in the same cycle) without passing through the DMA Controller.
- One example below (8237 DMA Controller in Intel PC Platform)



[Fig Source]Harvey AF et al, DMA Fundamentals on Various PC Platforms, National Instruments Application Note 011, 1991

FlyBy DMA



- DMAC issue the memory read and I/O write command in the same cycle. Providing the control and address information below simultaneously to the data bus
  - Memory Read Strobe and Memory Read Address
  - I/O write strobe