Input & Output

CS2053 Computer Architecture

Computer Science & Engineering

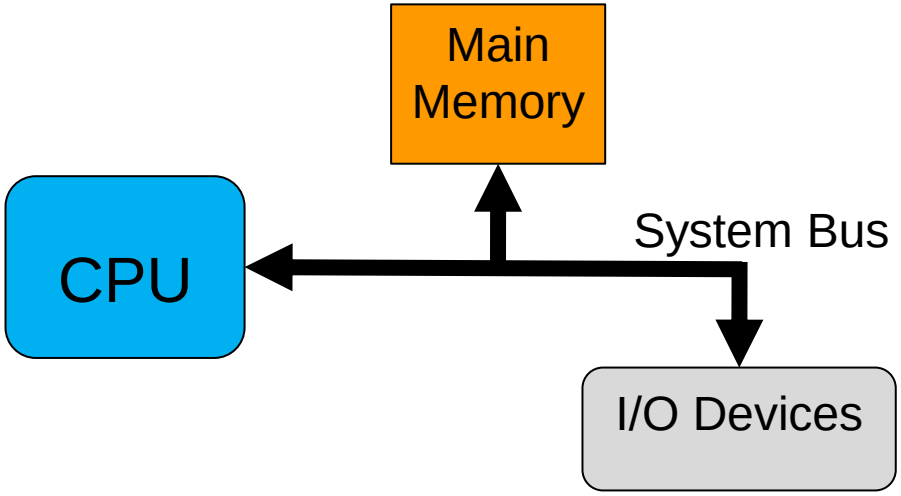
University of Moratuwa

Dr. Sulochana Sooriyaarachchi

Dr. Chathuranga Hettiarachchi

Acknowledgement: Dr. Dilum Bandara

Input & Output

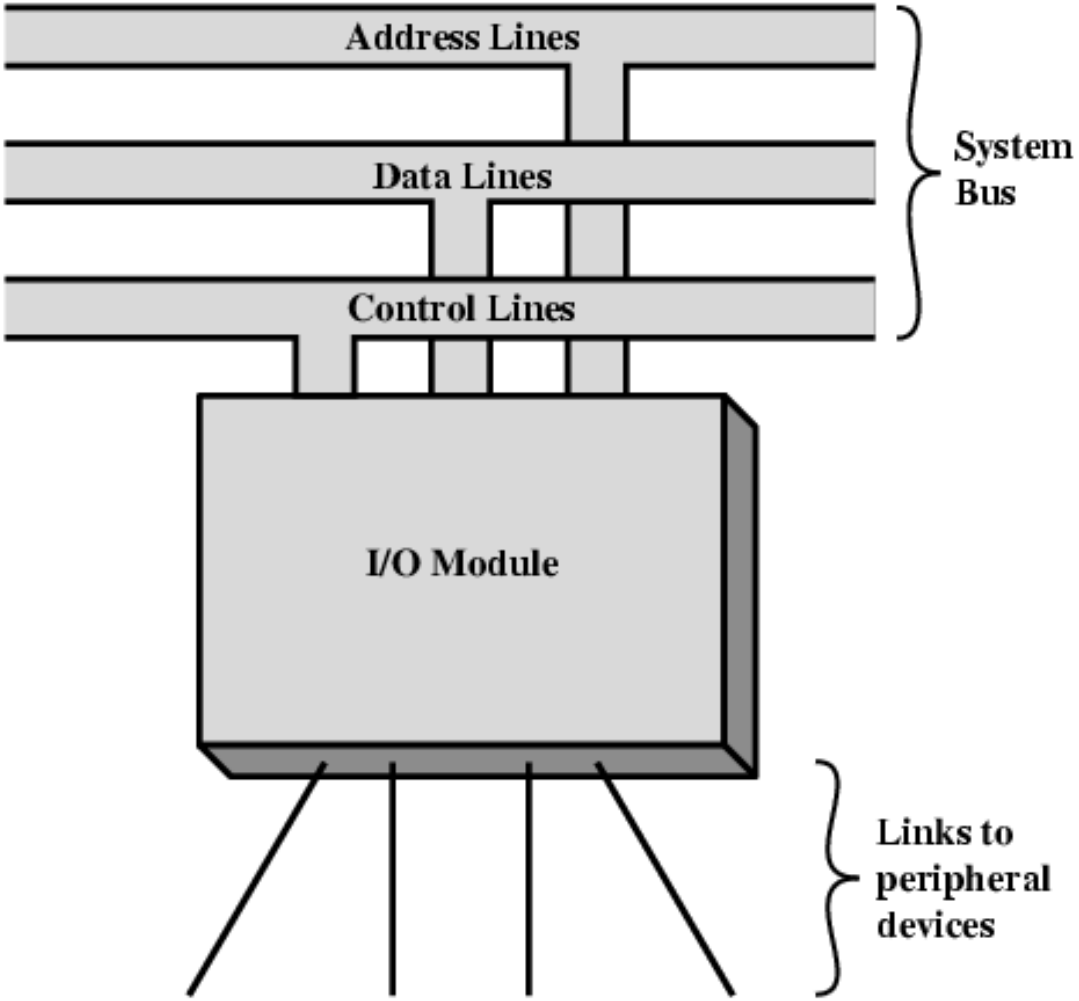


- Wide variety of peripherals
  - Different volumes of data, in different formats, & different speeds
- All slower than CPU & RAM
- Controlled via I/O modules/controllers
  - Interface to CPU & Memory
  - Interface to 1 or more peripherals

External Devices

- Interact with humans
  - Monitor, printer, keyboard, mouse
- Machine readable
  - Monitoring & control
  - e.g., process scheduling, CPU/casing temperature monitoring, fan speed control
- Communication
  - Network Interface Card (NIC)
  - Modems
  - Dongles – Bluetooth, Wi-Fi, 3G, 4G

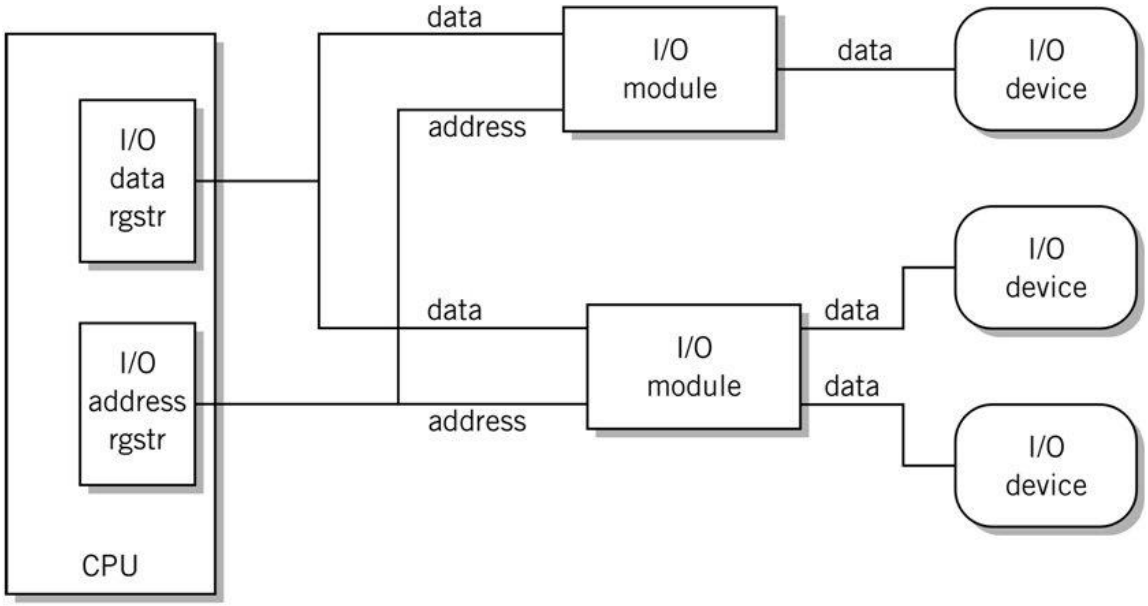
Generic Model of I/O Module



Source: William Stallings, Computer Organization and Architecture, 8th Edition

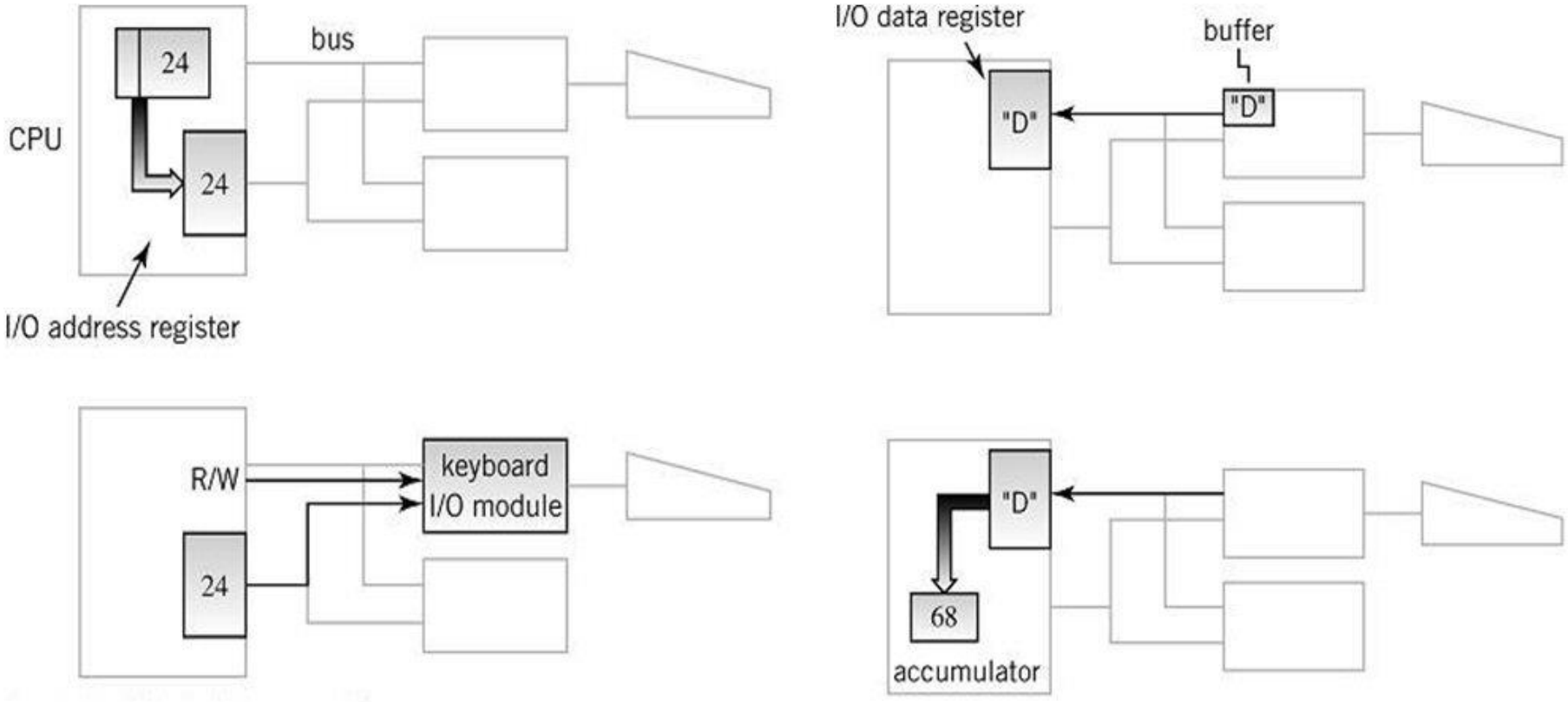
I/O Module Functions

- Control & Timing
- CPU communication
- Device communication
- Data buffering
- Error detection



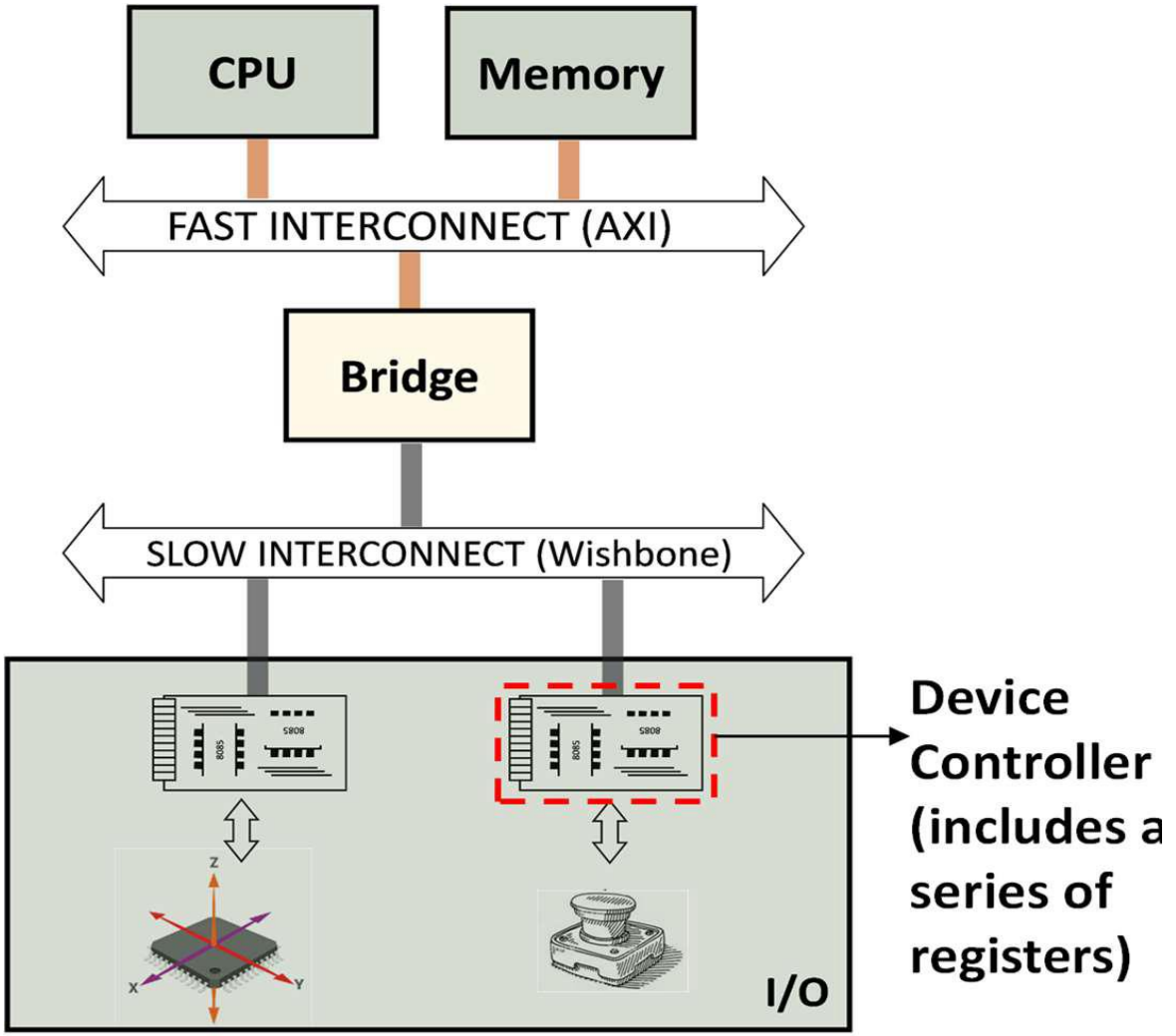
Source: http://lilt.ilstu.edu/cheri/itk254/sc/inputoutput/InputOutput_print.html

I/O Module Functions – Example

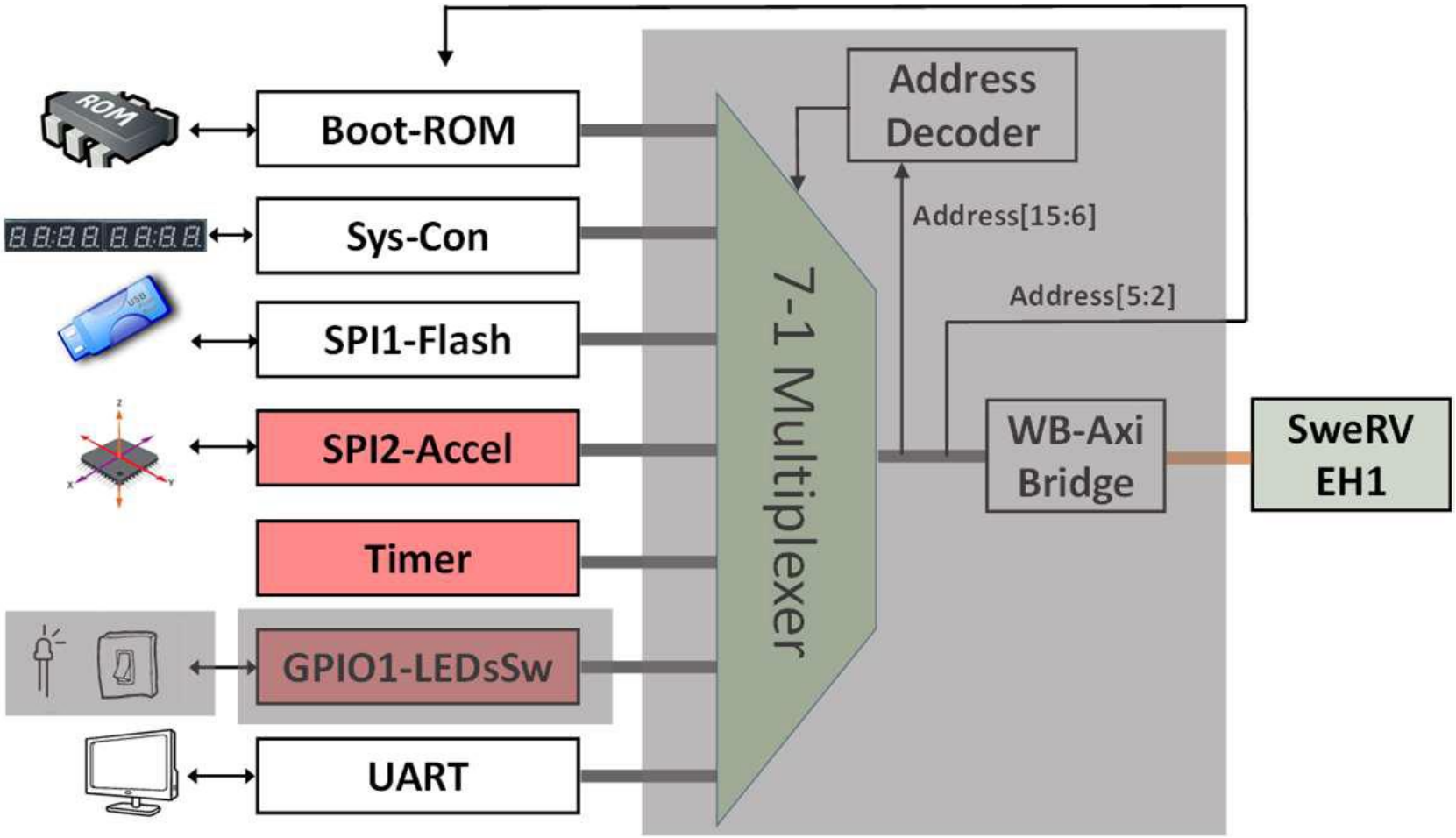


Source: http://lilt.ilstu.edu/cheri/itk254/sc/inputoutput/InputOutput_print.html

SweRVolfX


CPU
Memory
FAST INTERCONNECT (AXI)
Bridge
SLOW INTERCONNECT (Wishbone)
Device Controller (includes a series of registers)
I/O

SweRVolfX IO at low level



Input Output Techniques

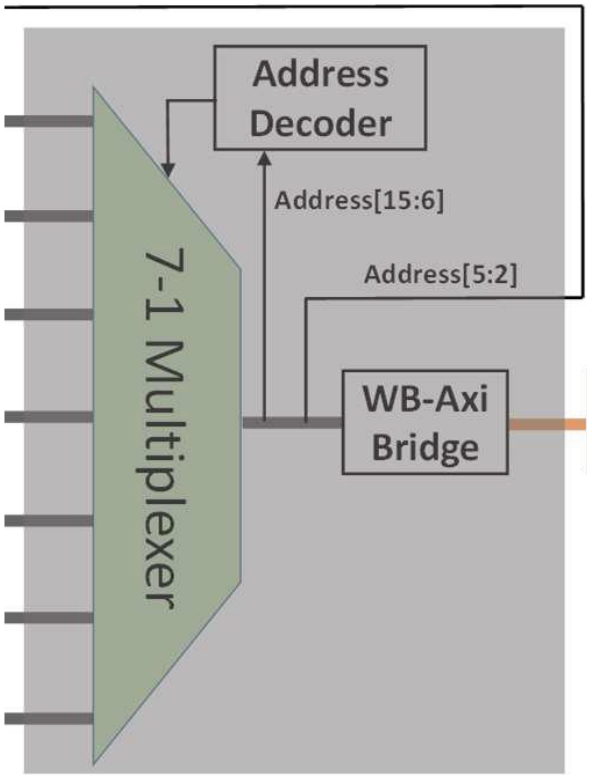
- Programmed I/O
  - Polling
- Interrupt driven I/O
- Direct Memory Access (DMA)

Programmed I/O

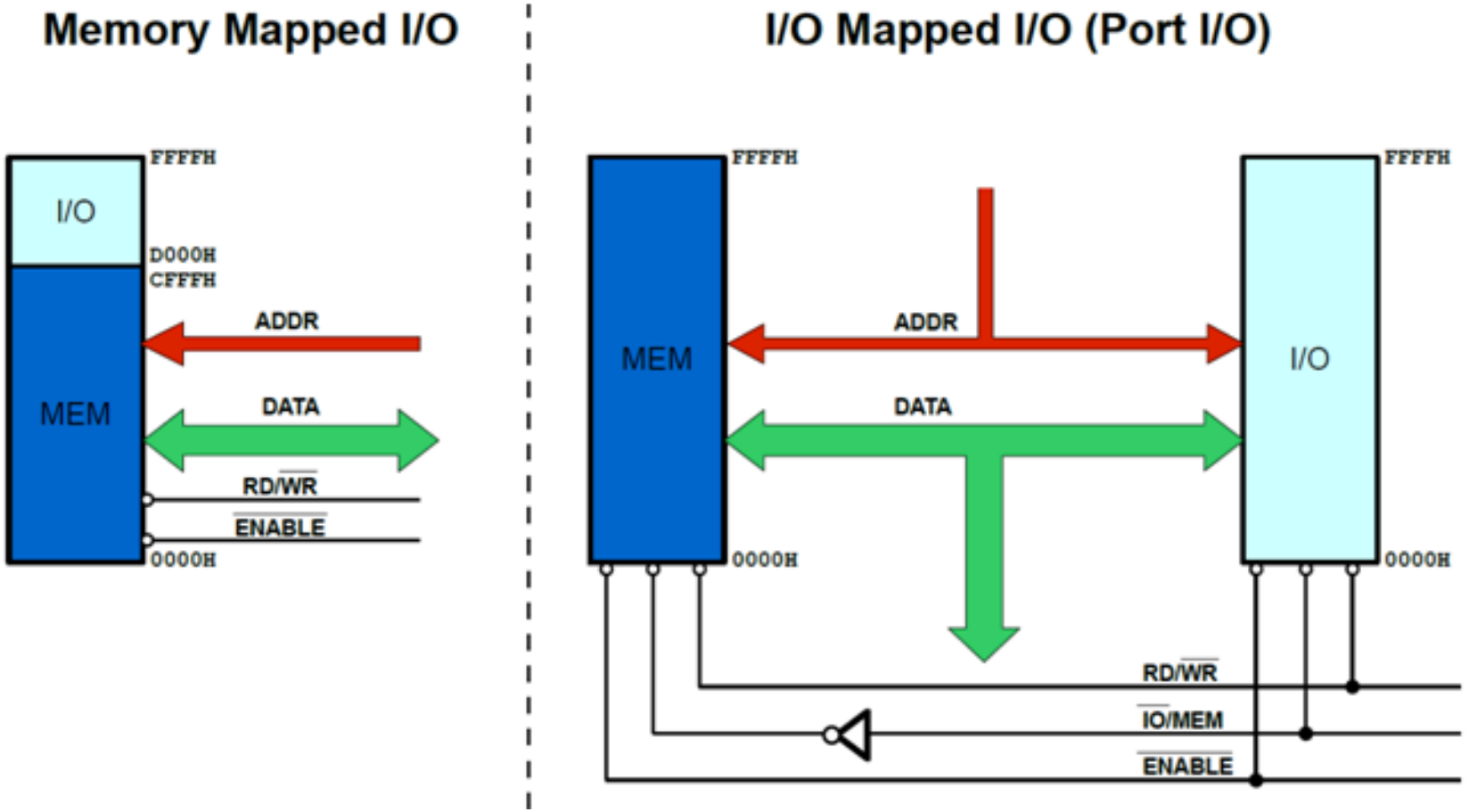
- CPU has direct control over I/O
  - Continuously sense status – Poll
  - Read/write commands
  - Transferring data
- CPU waits for I/O module to complete operation
- Wastes CPU time

Addressing I/O Devices

- Under programmed I/O data transfer is like memory access
  - Use LOAD/STORE instructions
- Each device is given a unique identifier
  - Address[15:6]
- CPU commands contain identifier (address)
- Registers within I/O controller are identified by
  - Address[5:2]



I/O Mapping



Source: http://me-lrt.de/memory-map-port-isolated-input

I/O Mapping (Cont.)

Memory Mapped I/O

- Devices & memory share same address space
- I/O looks just like memory read/write
- No special commands for I/O
  - Large selection of memory access commands

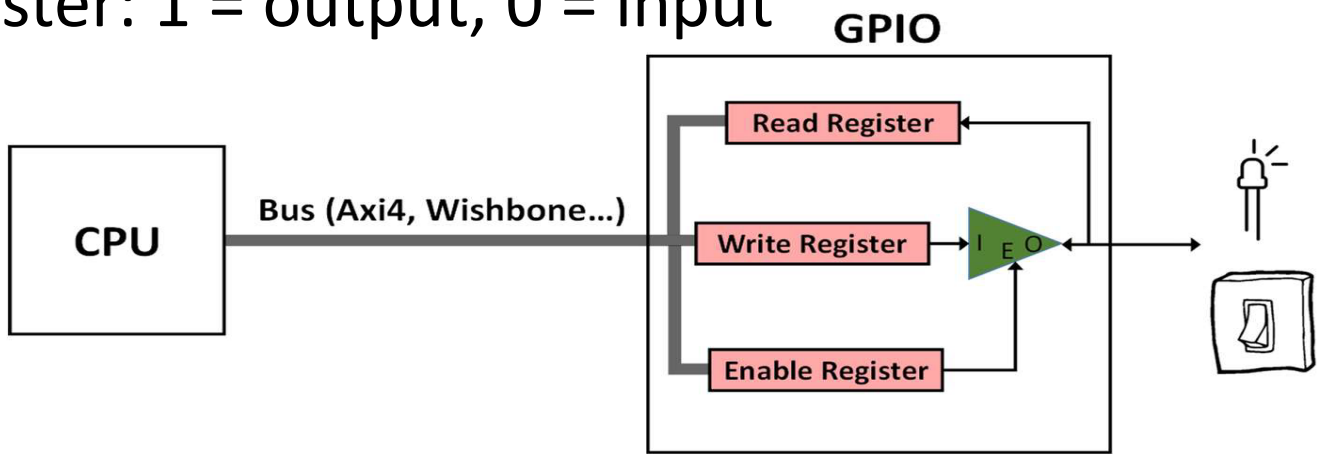
Isolated I/O

- Separate address spaces
- Need I/O or memory select lines
- Special commands for I/O
  - Limited set

E.g. Memory mapped - GPIO

Three memory-mapped registers:

- Read Register: value read from pin
- Write Register: value to write to pin
- Enable Register: 1 = output, 0 = input

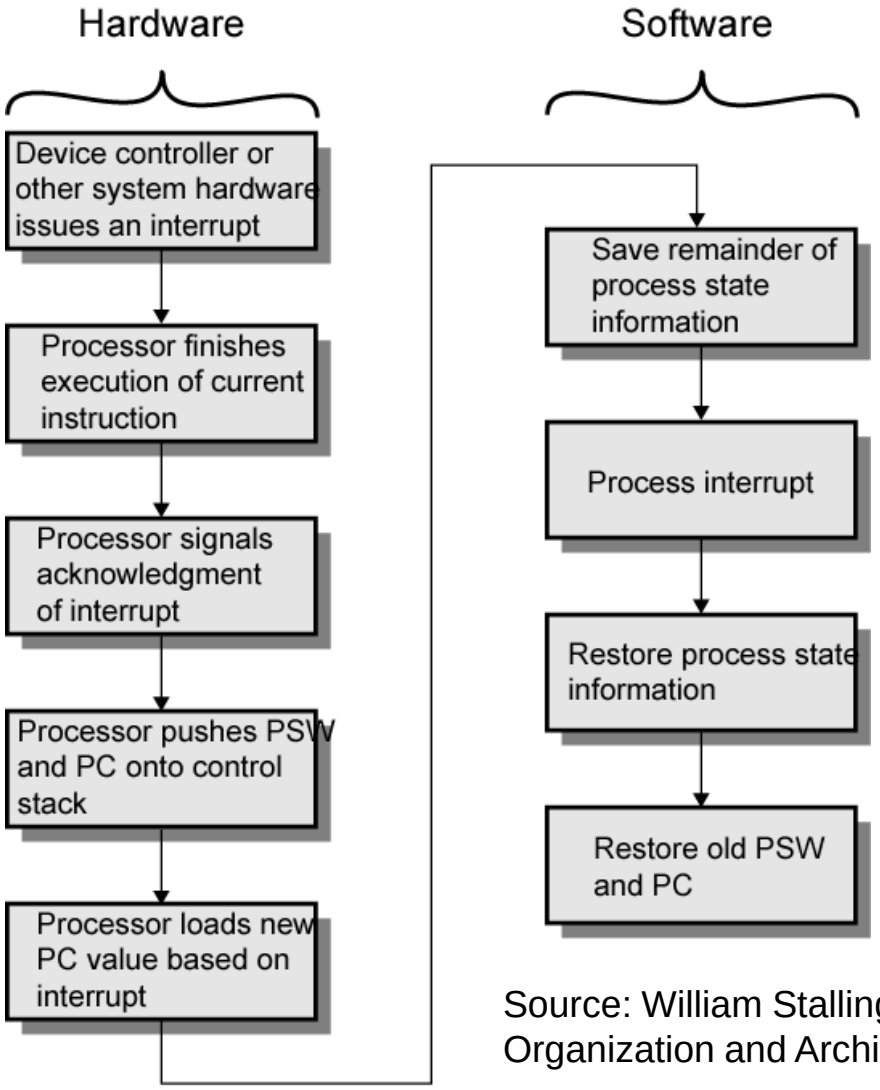


Register	Memory-Mapped Address
Read Register	0x80001400
Write Register	0x80001404
Enable Register	0x80001408

Interrupt Driven I/O

- No repeated CPU checking of device
  - No waiting
  - CPU does its own work
- I/O module interrupts CPU when ready
- Steps
  - CPU issues read command
  - I/O module gets data from peripheral whilst CPU does other work
  - I/O module interrupts CPU
  - CPU requests data
  - I/O module transfers data

Interrupt Processing (Cont.)



Source: William Stallings, Computer Organization and Architecture, 8th Edition

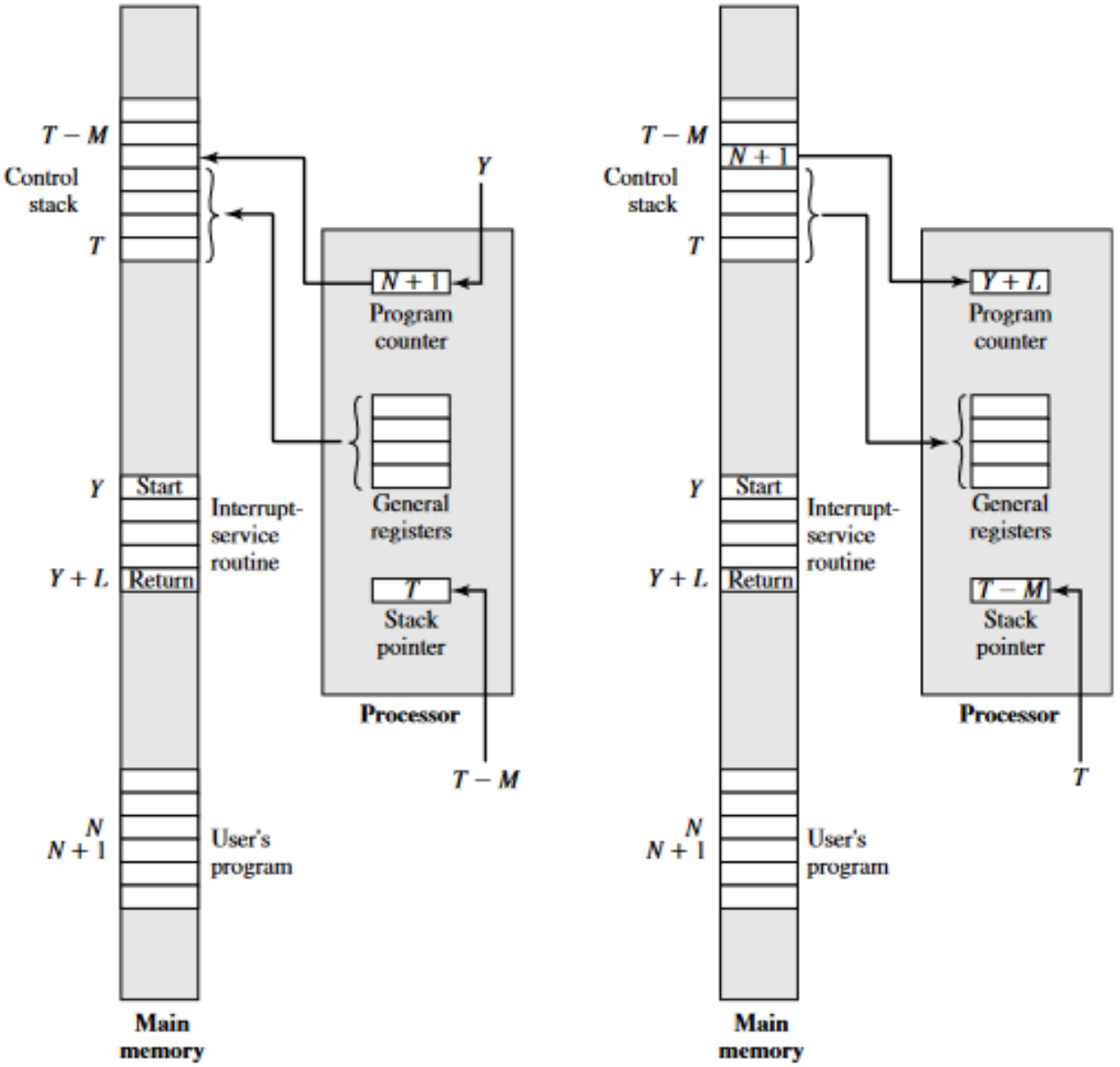
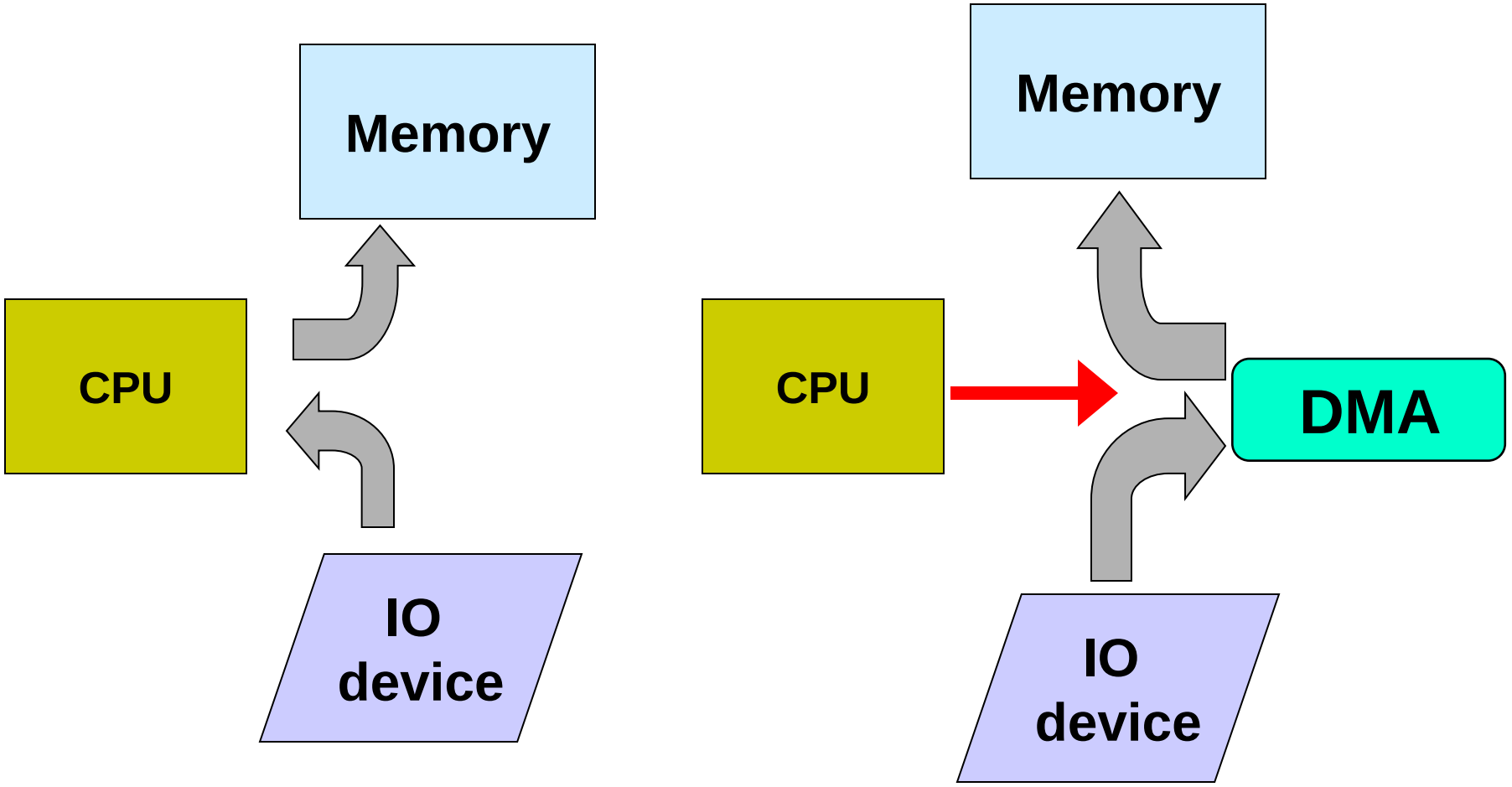


Figure 7.7 Changes in Memory and Registers for an Interrupt

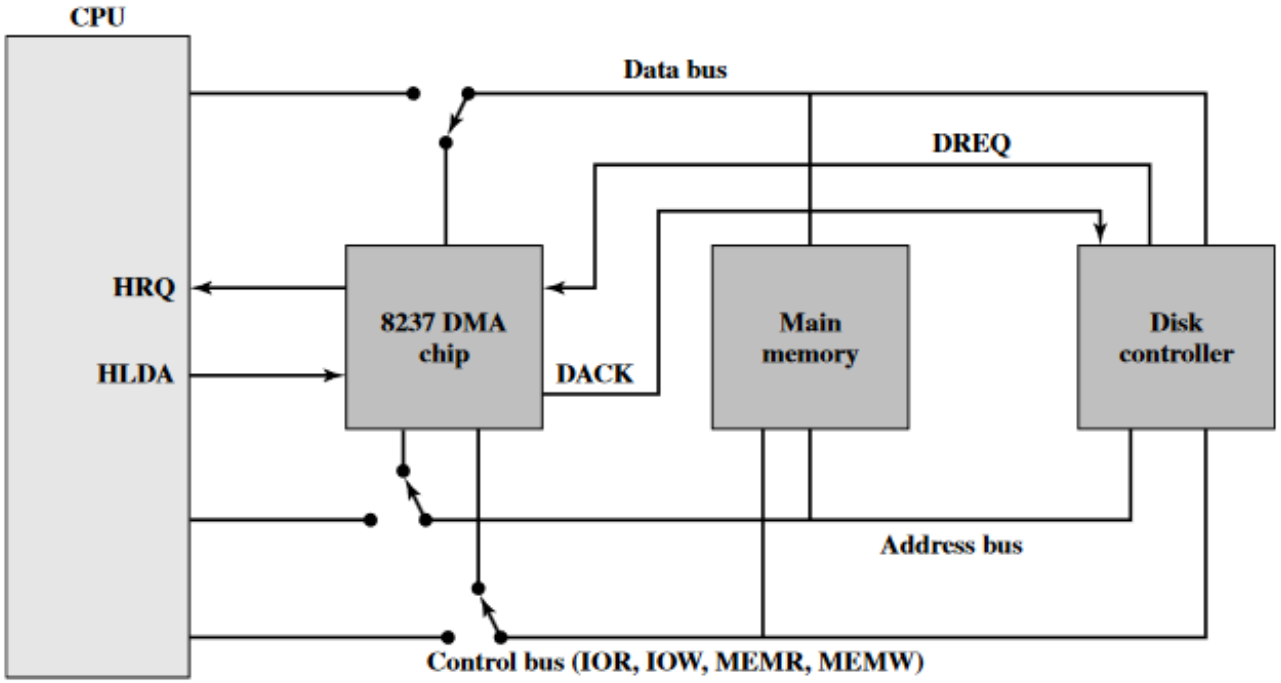
Direct Memory Access (DMA)

- Programmed & interrupt driven I/O require active CPU intervention
  - Transfer rate is limited
  - CPU is tied up in data transfer
- DMA is the answer
  - Additional module (hardware) on bus
  - DMA controller takes over from CPU for I/O
    - Provide a way of bypassing CPU when transferring data between memory & IO

DMA (Cont.)



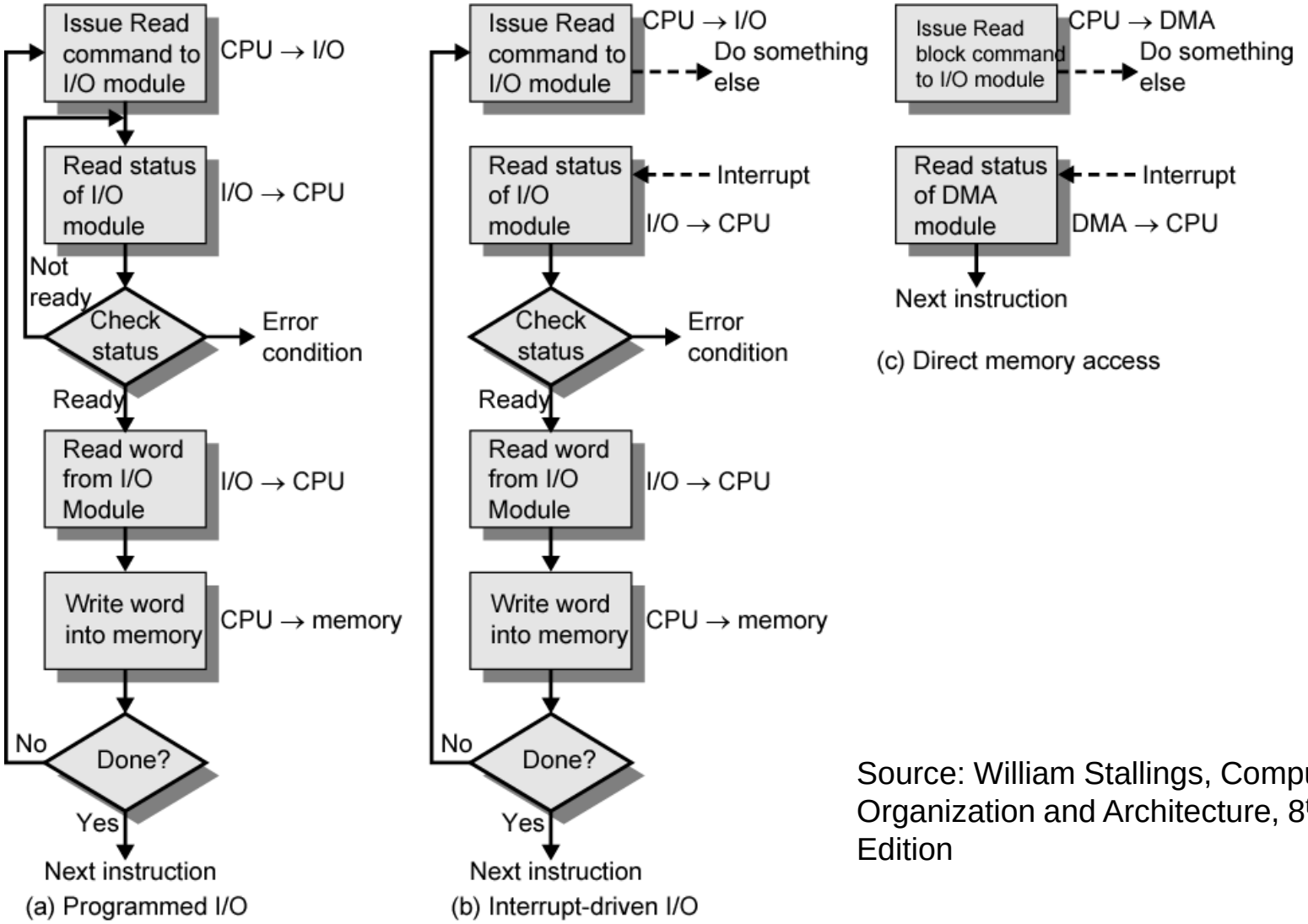
Example bus configuration



DACK = DMA acknowledge
DREQ = DMA request
HLDA = HOLD acknowledge
HRQ = HOLD request

Figure 7.14 8237 DMA Usage of System Bus

3 Techniques for Input of a Block of Data



Source: William Stallings, Computer Organization and Architecture, 8th Edition

Blocks of a Computer

Further Reading

Chapter 7 : Computer Organization and Architecture : Designing for Performance (8th Edition), by William Stallings

THANK YOU