

# COL216

# Computer Architecture

Input/Output – 1

14th March 2022

# Diversity of Input/Output devices

Communicate with

- Human
- Computer
- Physical systems
- Storage

Variations in terms of

- direction
- speed
- data type . . .



# Communication with humans

- Key board, mouse, joy stick, tablet, touch, gesture
  - very slow,  $10^1\text{-}10^2$  bytes per sec
- Documents: Scanners, printers, plotters
  - $8\text{-}16$  ppm,  $300\text{-}1200$  dpi,  $10^2\text{-}10^3$  KB/sec
- Sound (voice/music): mikes, speakers
  - $10^1\text{-}10^3$  KB/sec
- Graphic displays
  - mega pixels, refresh rate,  $10^1\text{-}10^3$  MB/sec

# Communication with machines

- Wired network controllers

- $10^1\text{-}10^3 \text{ MB/sec}$

- Wireless network controllers

- ethernet, blue tooth, zigbee etc

- $10^0\text{-}10^1 \text{ MB/sec}$

- Modems

- $10^1\text{-}10^3 \text{ KB/sec}$

# Communication with physical systems

## ■ Sensors

- pressure, temperature, humidity, flow, level, position, speed, proximity, gas
- audio, visual

## ■ Actuators

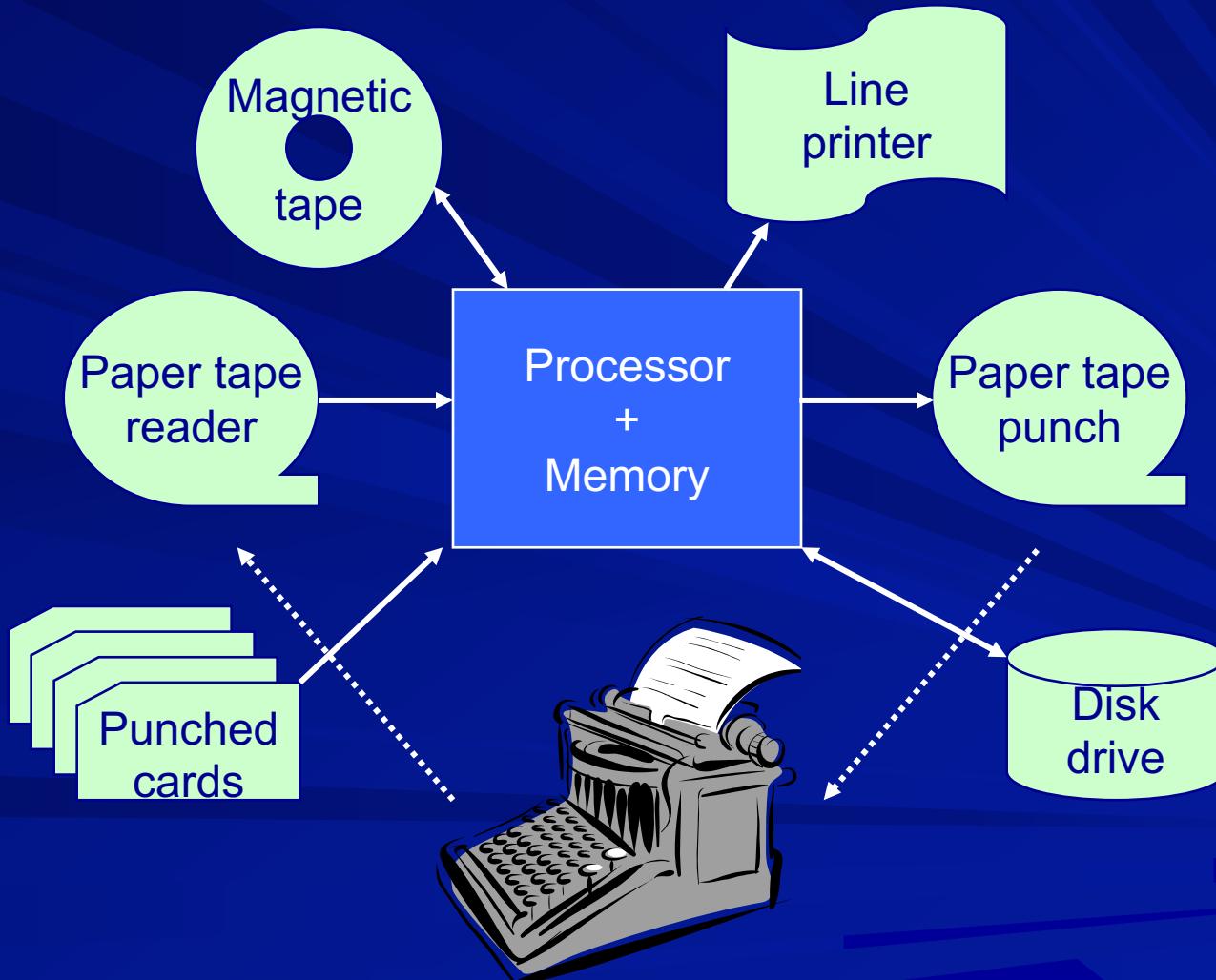
- switches, valves, solenoid, motors

Signal forms: on/off, analog, pulse train etc

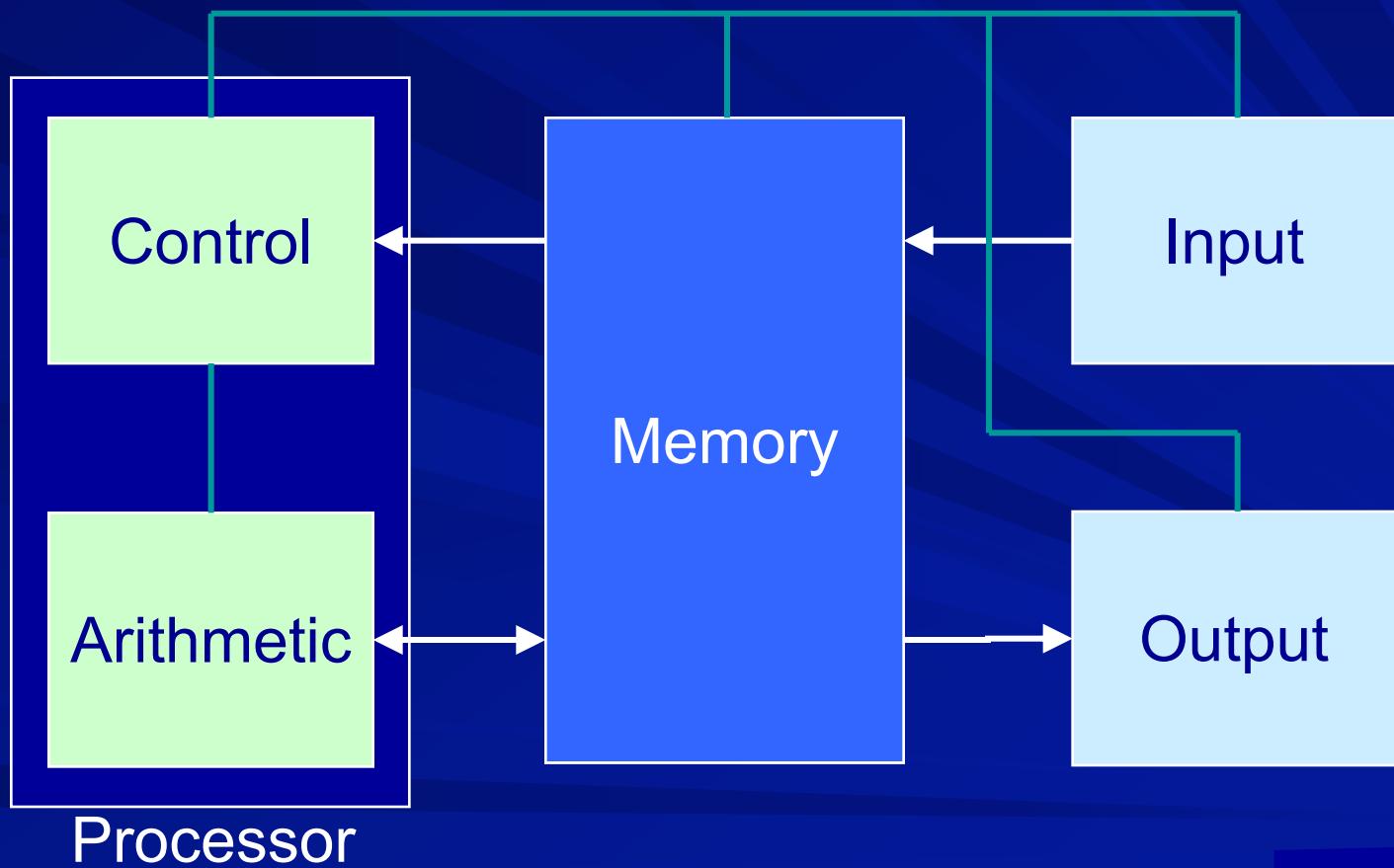
# Storage devices

- Magnetic
  - disks, tapes
- Optical
  - CDROM, DVD, Blue Ray
- Semi-conductor
  - Nor Flash, Nand Flash

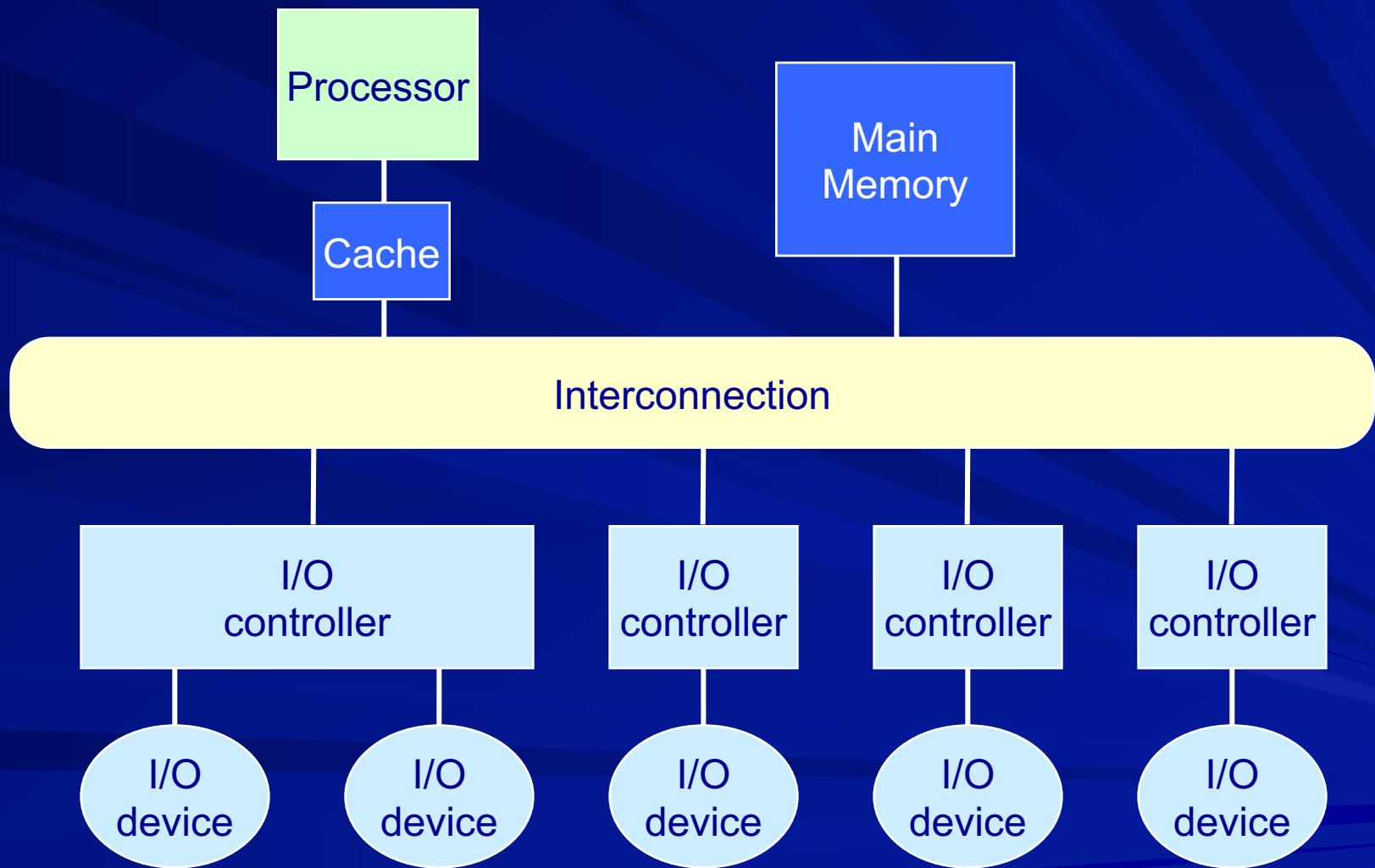
# Peripherals circa 1970 : ICL 1909



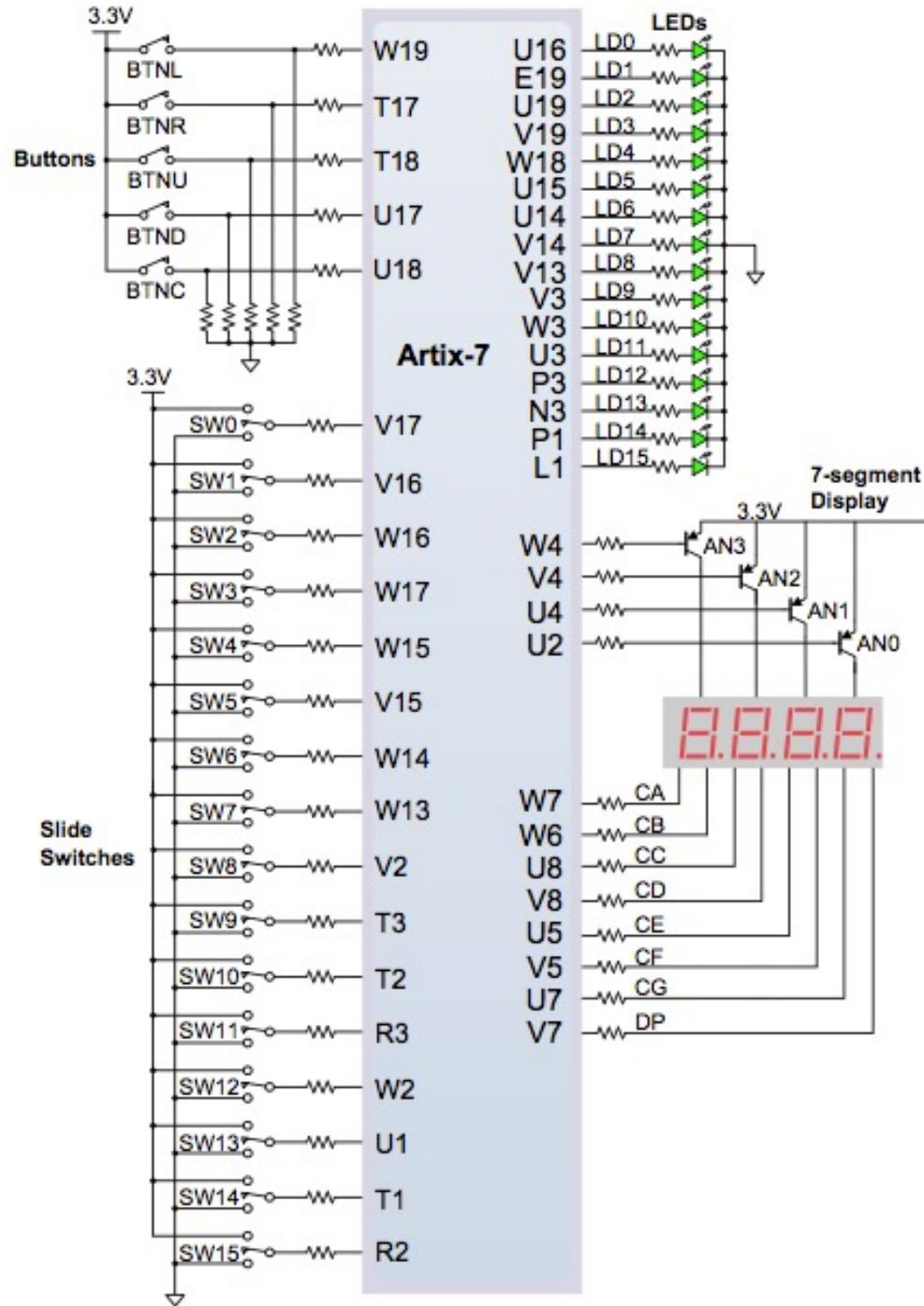
# Computer System: Simplified Block Diagram



# More realistic block diagram



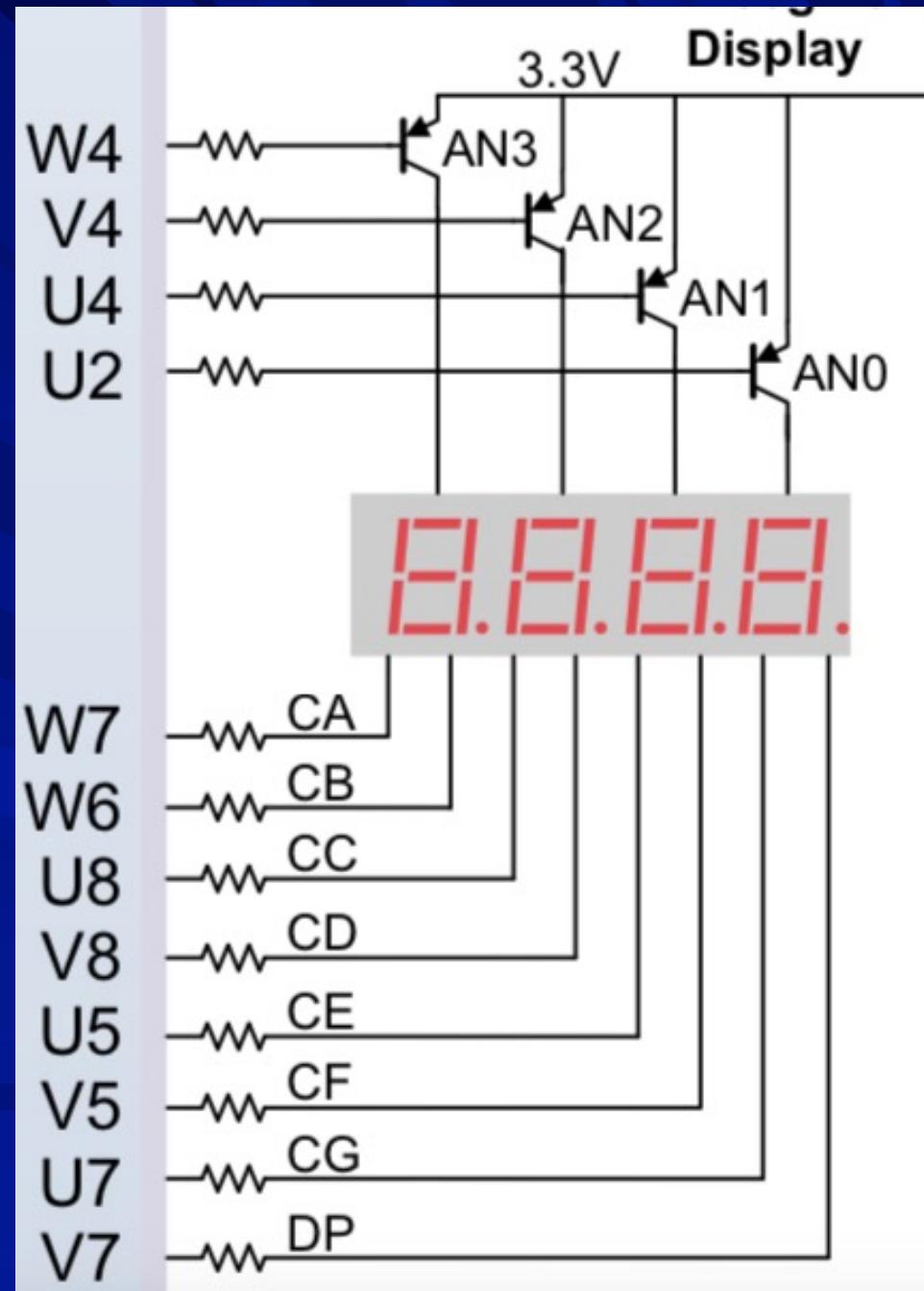
# Switches and displays on BASYS3 board



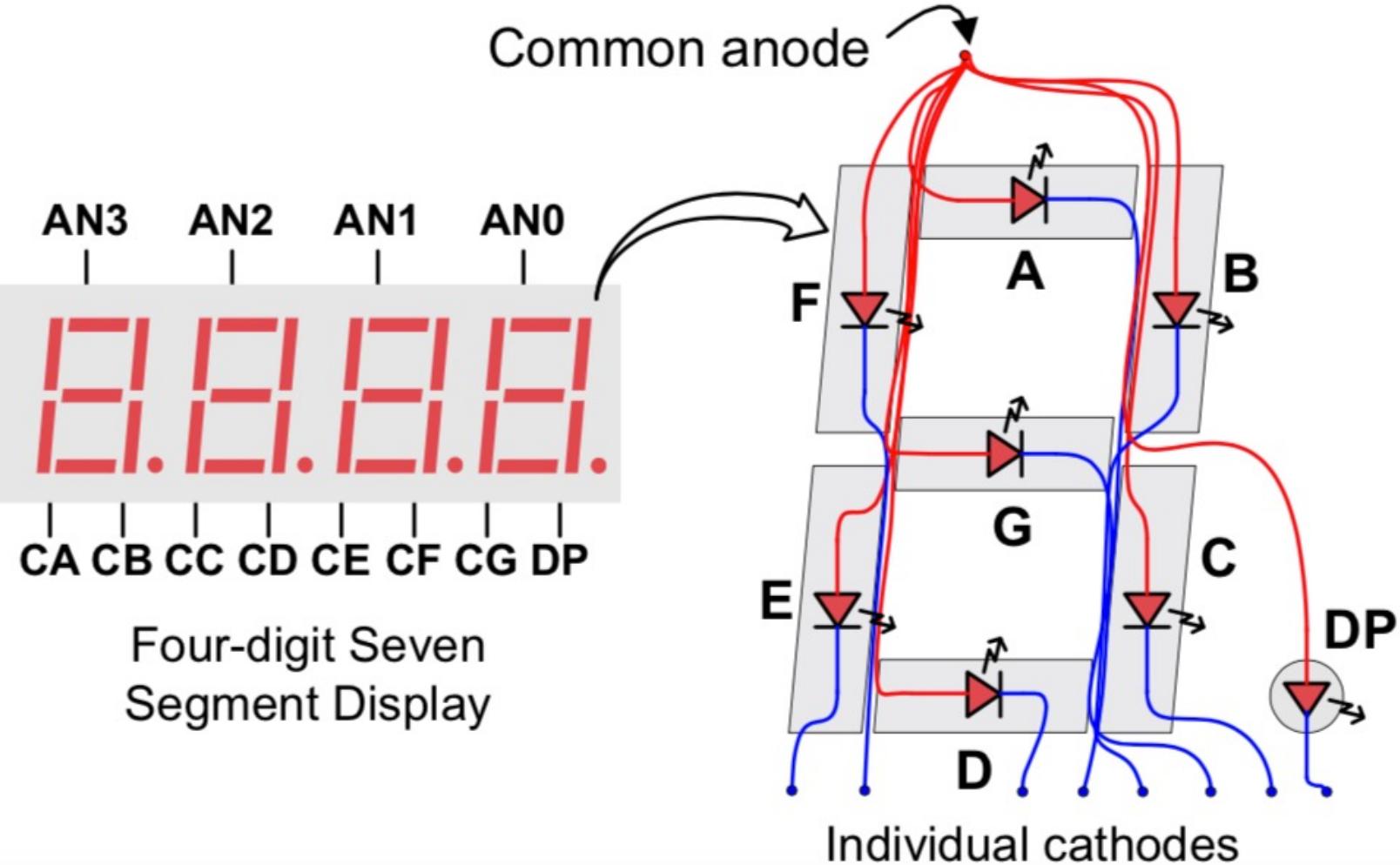
# 16 Button Key-pad



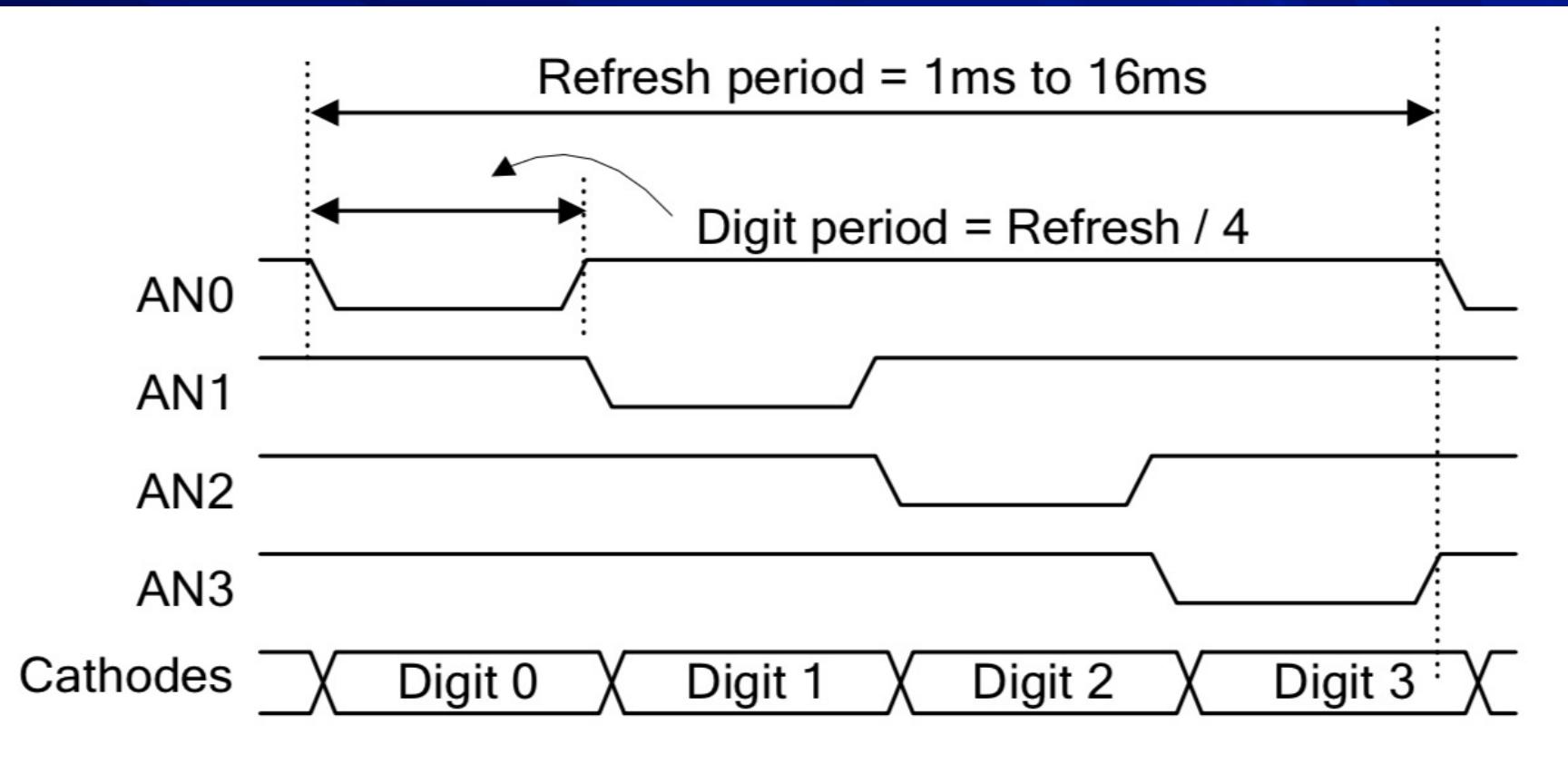
# 4-digit Display on BASYS 3 Board



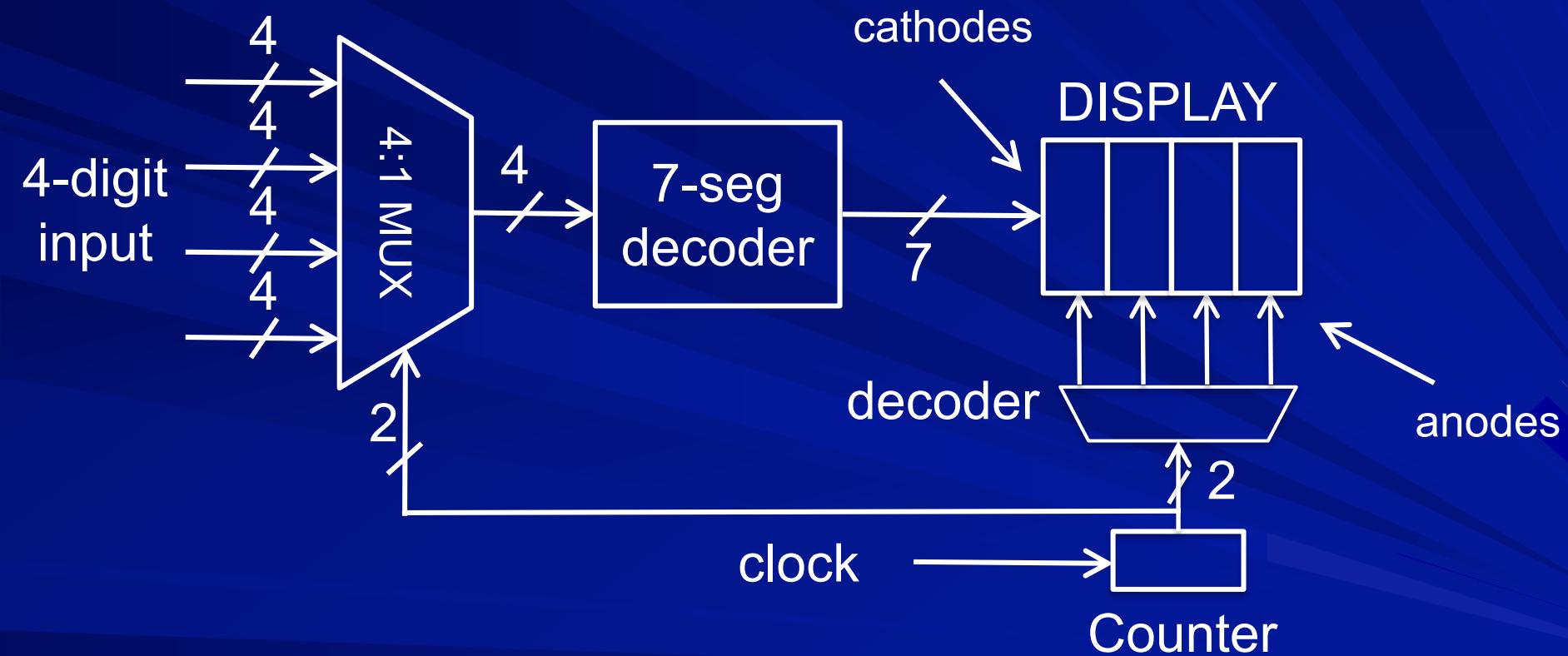
# 7-segment Display



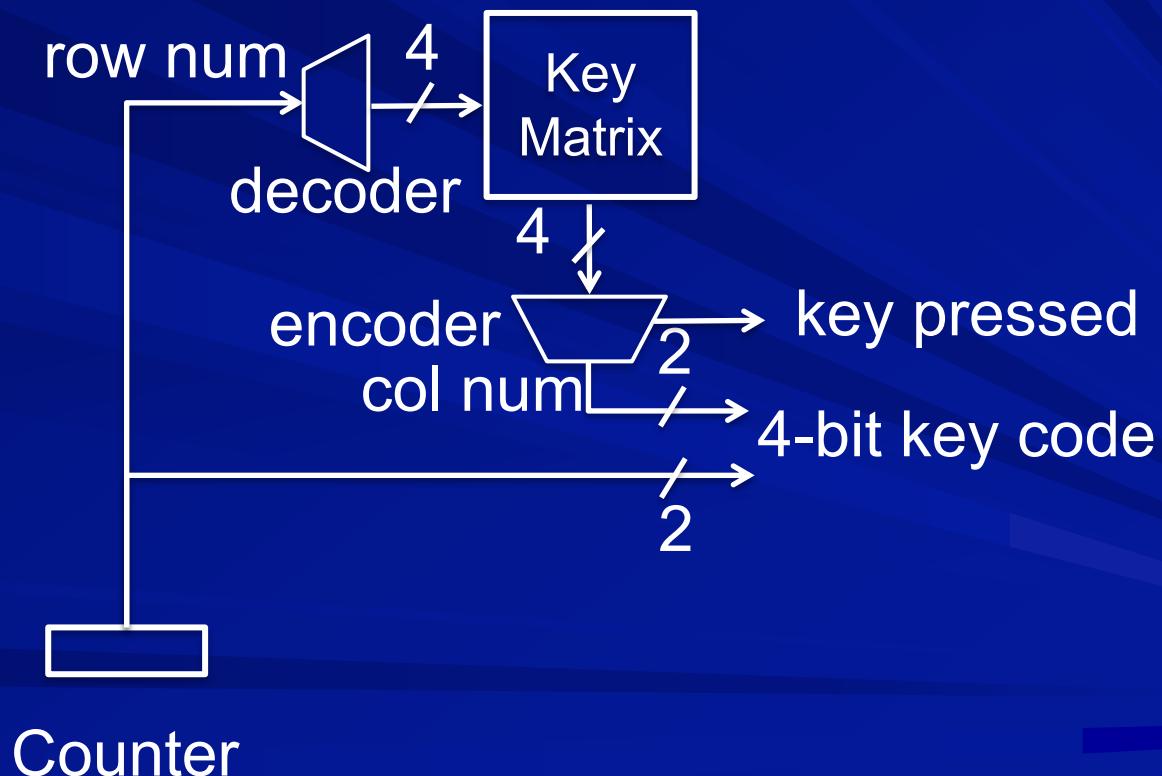
# Refresh timings



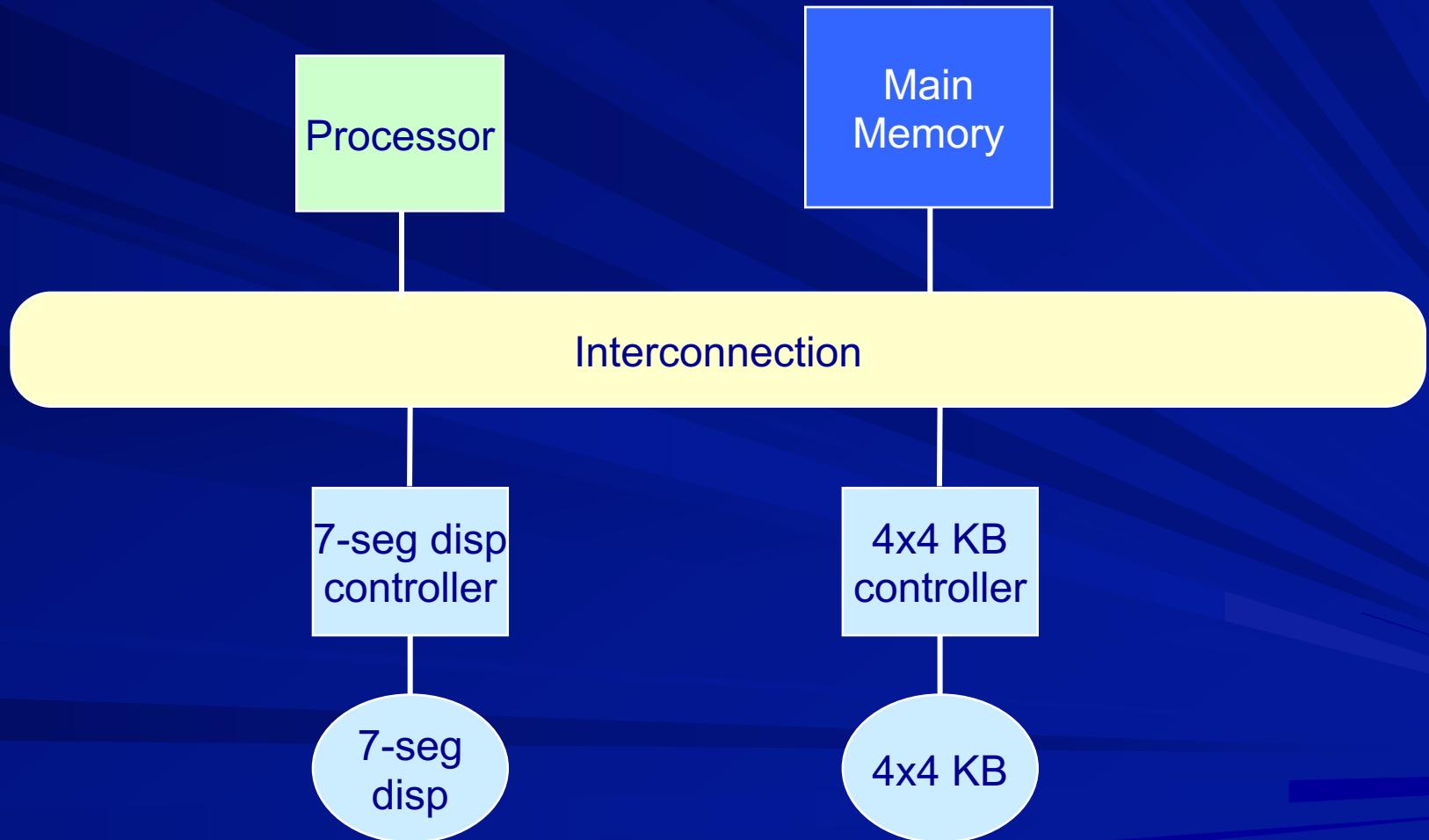
# 7-segment Display Controller



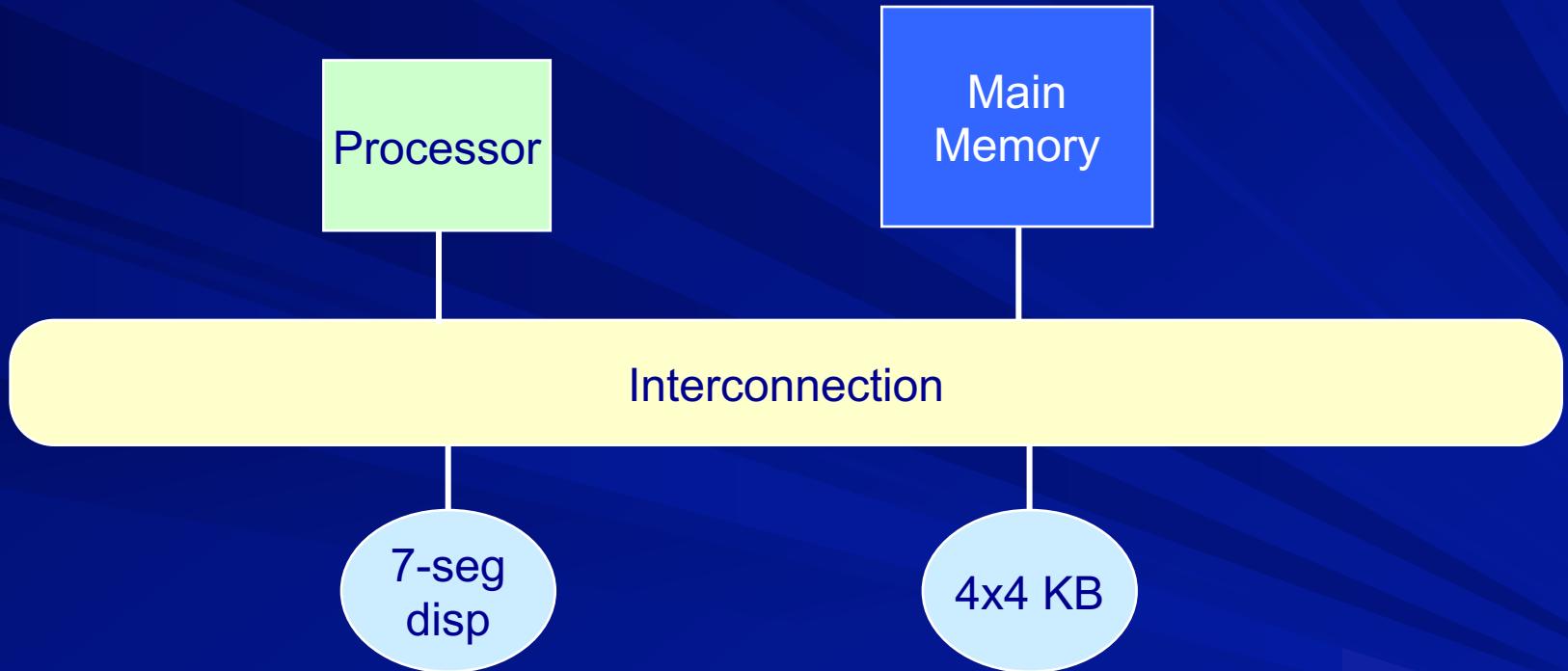
# 4 x 4 Keyboard Encoder



# Block diagram



# Block diagram



Low level control by software

# Instructions for I/O

Use SWI instruction/system call

Require instruction for I/O inside handler

- Special input/output instructions

- specify device
  - specify register/memory location

=> Dedicated I/O

- Use LDR, STR instructions

- use devices as memory locations

=> Memory mapped I/O

# I/O address space

- Memory mapped I/O
  - part of memory address space reserved for I/O
  - usual load/store instructions are used for I/O
  - memory ignores these addresses
- Dedicated I/O
  - separate address space for I/O - much smaller than the memory space
  - control signal from processor indicates which space is being addressed
  - separate instructions are required

# Addressing I/O devices

Each I/O device (controller) viewed as a set of registers or ports by CPU

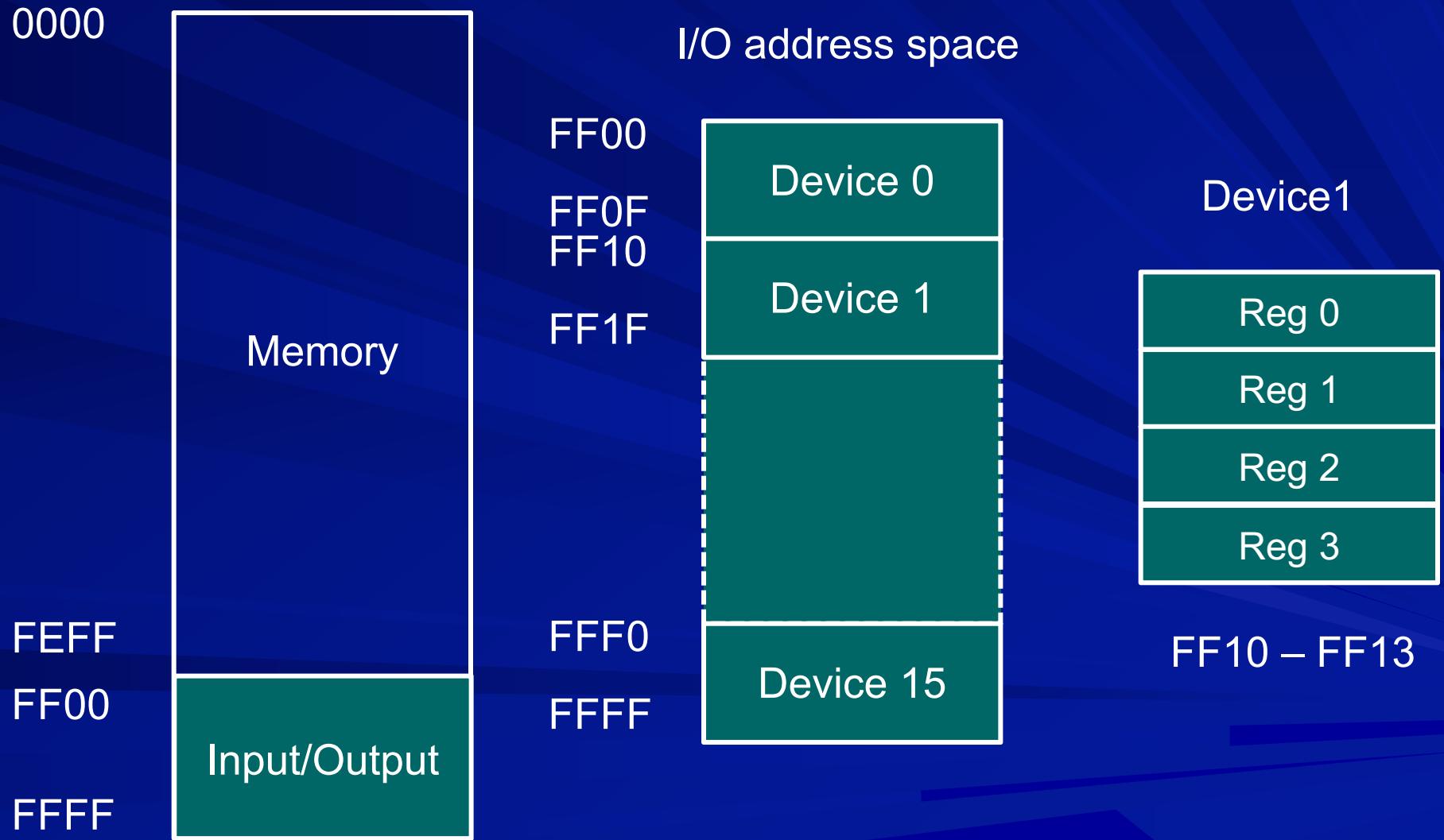
CPU can use these registers to

- write commands and parameters
- read status
- read/write data

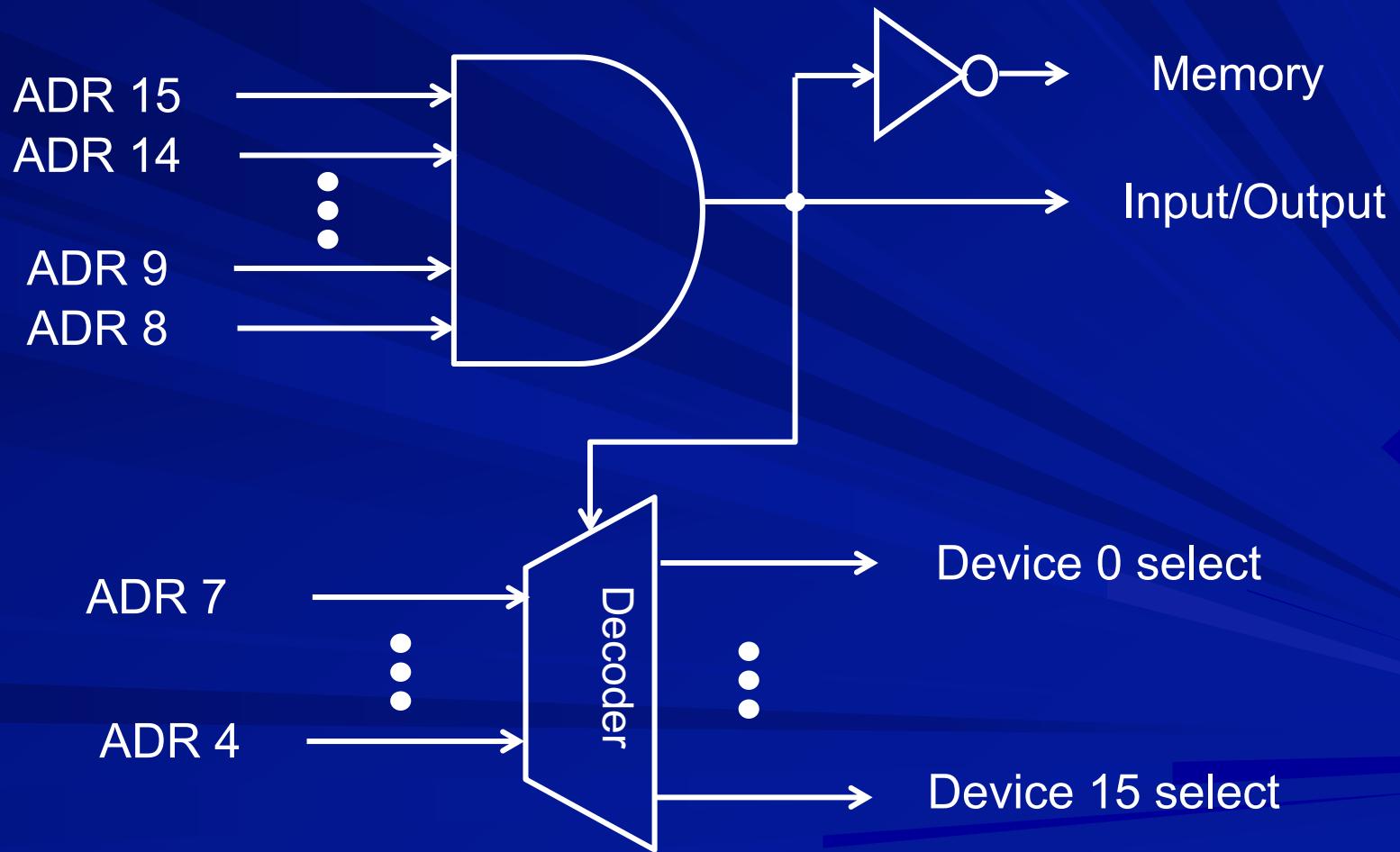
These registers are addressable

- Memory mapped I/O
- Dedicated I/O

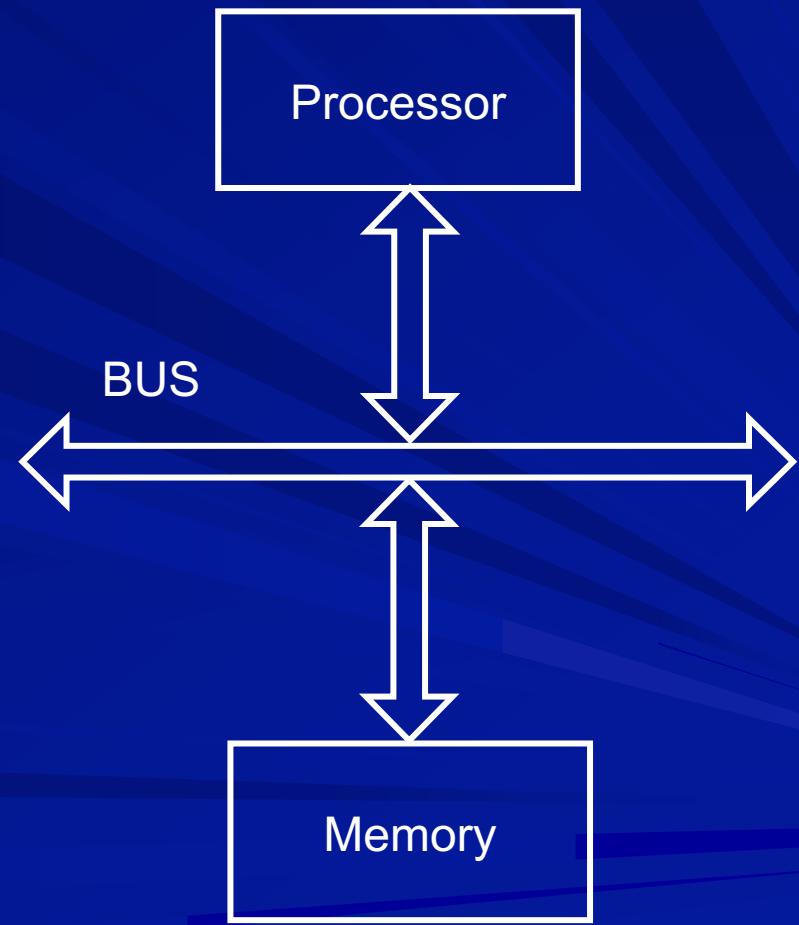
# Address Map Example



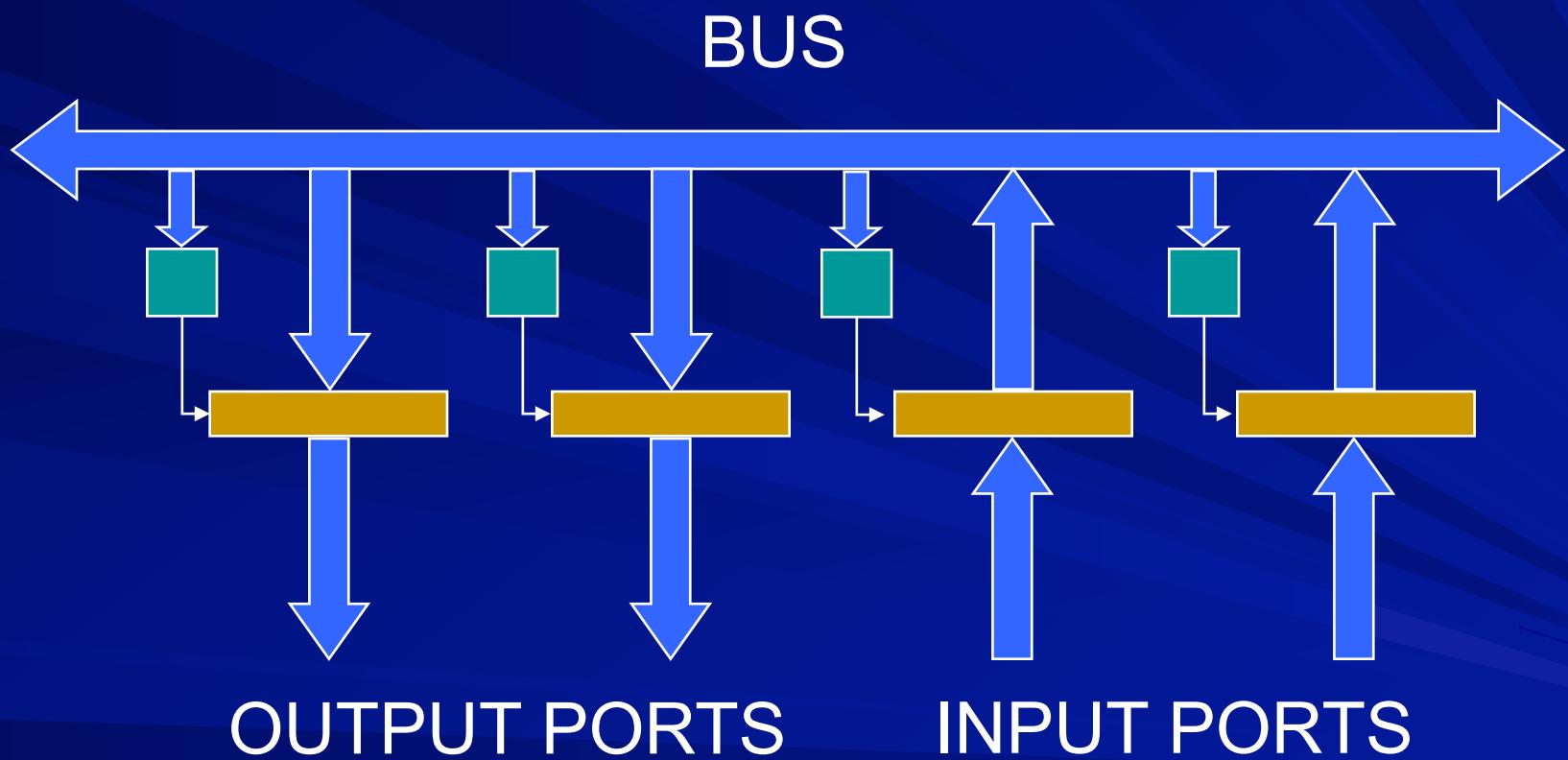
# Decoding Address



# Interconnection using a bus



# INPUT / OUTPUT PORTS



# Ports for Disp, KB controllers



# Ports for raw Disp, KB

Display

out	anode pattern
out	cathode pattern

refresh required

Key Board

out	row pattern
in	column pattern

repeated scan required