مدينة زويل للعلوم والتكنولوجيا
Zewail City of Science and Technology

COMMUNICATION AND INFORMATION ENGINEERING

CIE 314
Embedded Systems Fundamentals

Lecture #4
Memory and IO subsystem organization

Instructor:

Dr. Ahmad El-Banna

© Ahmad El-Banna

SPRING 2017

Agenda

Memory Organization

- Types - Program and Data Memory
- Architectures - Data Exchange
- Memory Map

I/O Subsystem Organization

- Interfaces (serial vs. parallel)
- Timing diagrams

Design Tips

ZEWAIL CITY
ESTABLISHED 2000

3.5 MEMORY ORGANIZATION

- Data Address: Little and Big Endian Conventions
- Program and Data Memory
- Von Neumann and Harvard Architectures
- Memory and CPU Data Exchange: An Example
- Memory Map
- MSP430 Memory Organization

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

MEMORY STRUCTURE

- The memory subsystem is organized as an array of cells or locations
- Each memory location is identified by an address
- The contents of a memory cell is a word
  - A memory word may store instructions or data

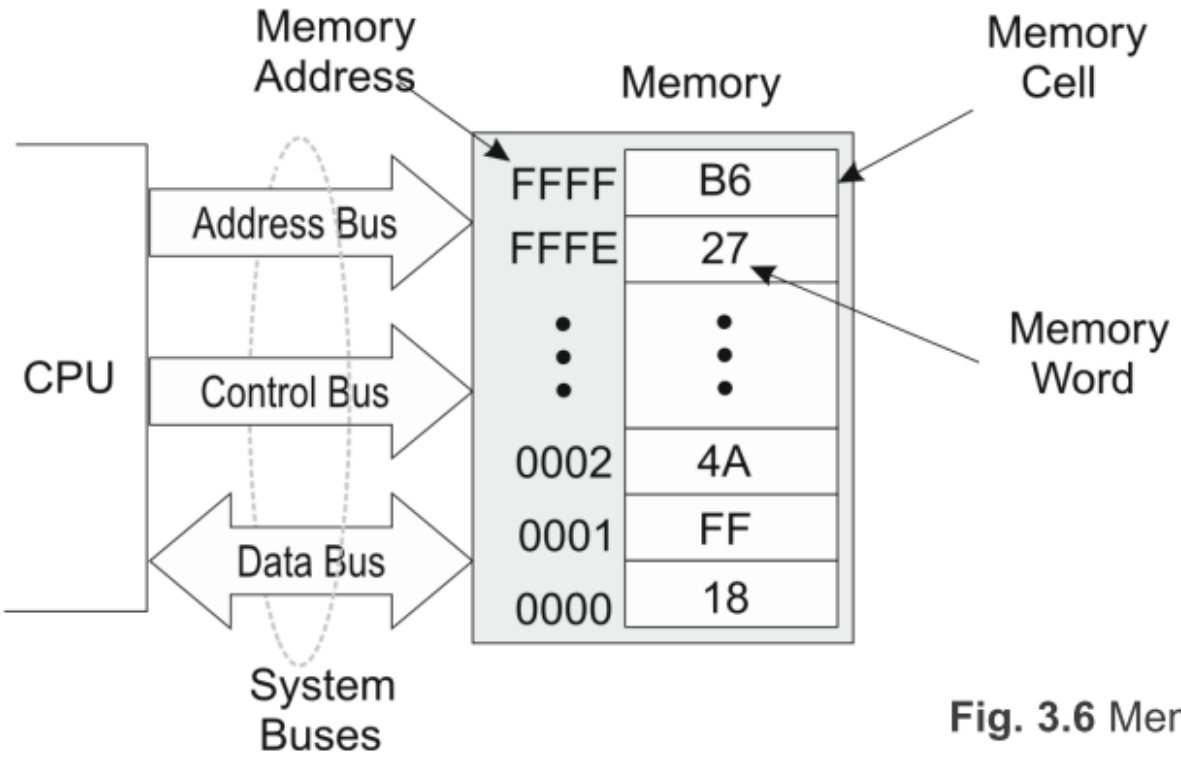


Fig. 3.6 Memory structure

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

MEMORY TYPES

Storage	Memory	In-system Writable	Comments
Nonvolatile	Masked ROM	No	Non programmable
	OTPROM	No	One time programmable with programming device
	EPROM	No	Erasable and programmable with external device
	EEPROM	Yes	Slow to erase/write. Not advisable to write during program execution. Requires higher voltage.
	Flash	Yes	Similar to EEPROM
	FRAM	Yes	Fast to write at low voltage
Volatile	Static RAM	Yes	Fastest to write/read
	DRAM	Yes	Fast to write/read

Fig. 3.7 Memory types

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

16-BIT MEMORY ARRAY

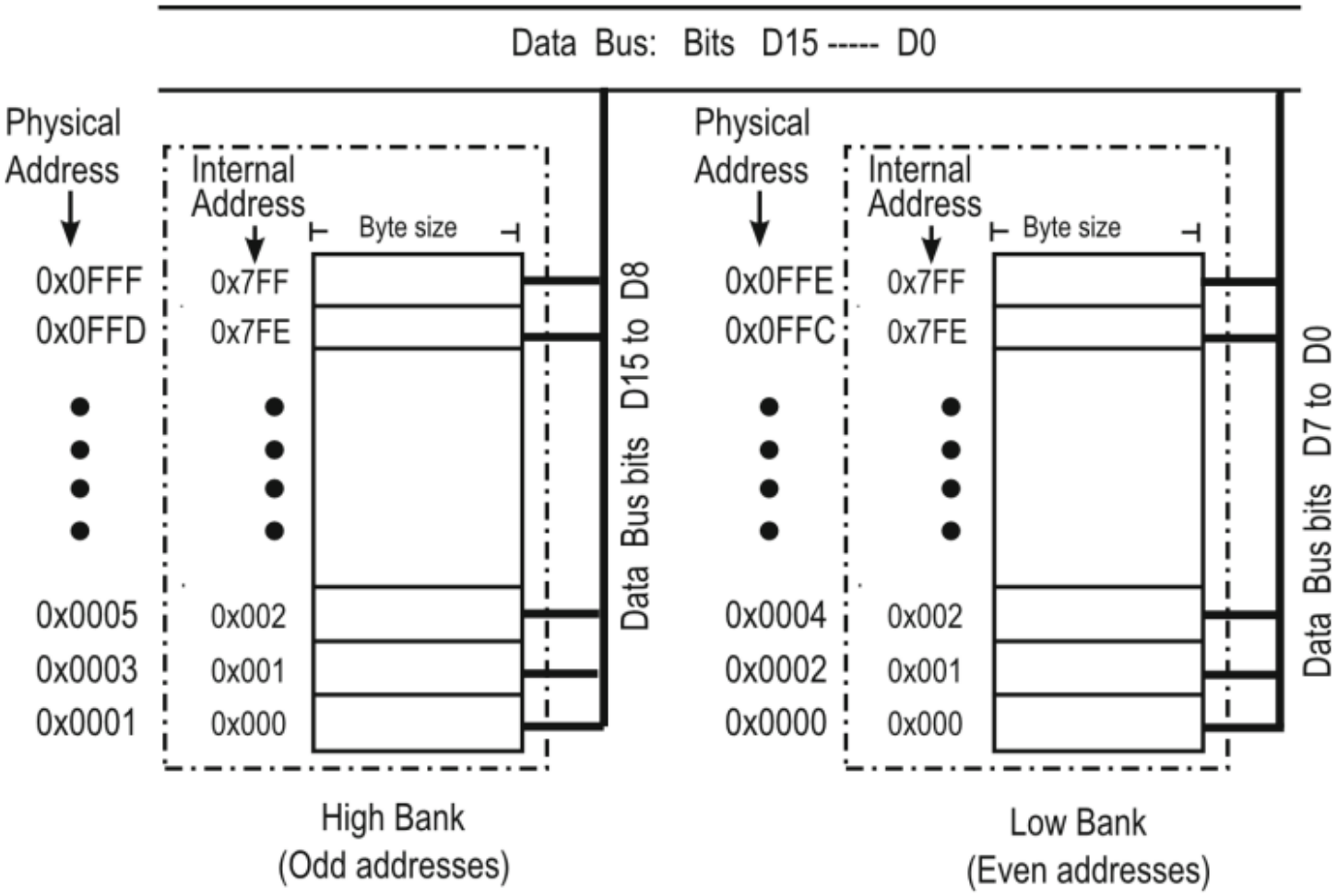


Fig. 3.8 Example of bank connection

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

ENDIANNESS

- Big Endian:
  - Data is stored with the most significant byte in the lowest address and the least significant byte in the highest address
- Little Endian
  - Data is stored the the least significant byte in the lowest address and the most significant byte in the highest address

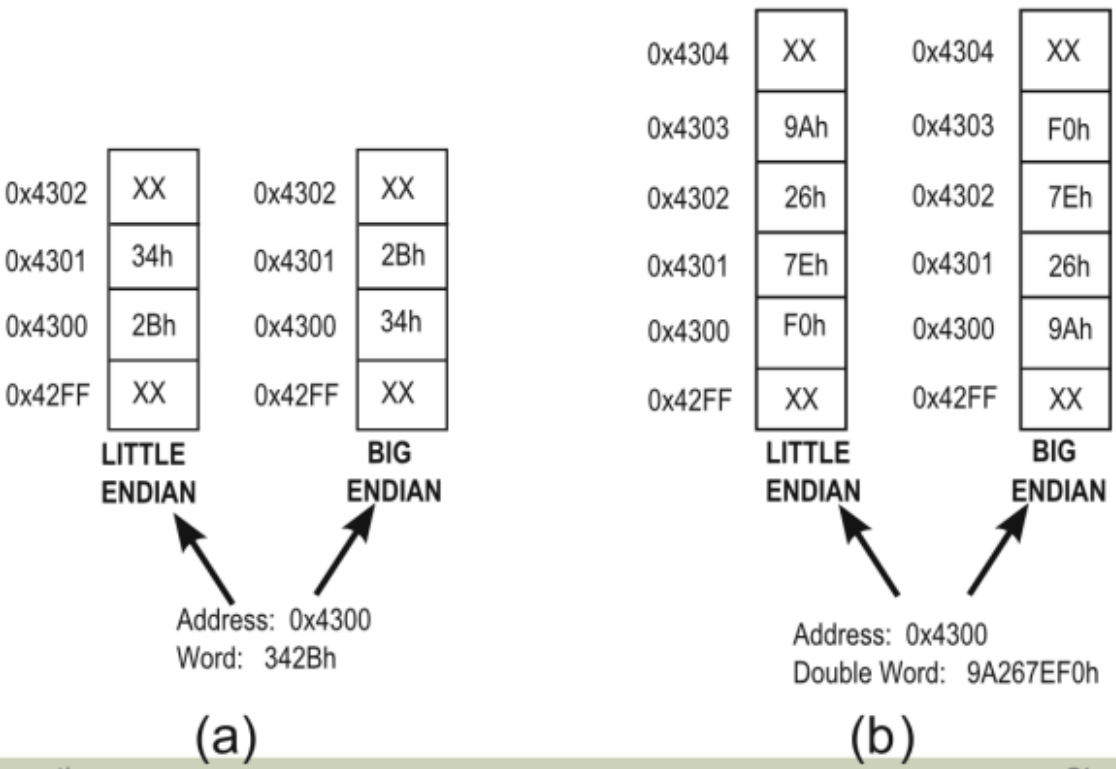


Fig. 3.9 Big and Little Endian conventions for (**a**) a word ([4300 h = 3428 h]) and (**b**) a double-word ([4300 h] = [9A267EF0h])

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

MCU ARCHITECTURES

- Von Neumann
  - A single set of buses for accessing both programs and data and a single address space
- Harvard
  - Uses separate buses for accessing programs and data and has separate address spaces

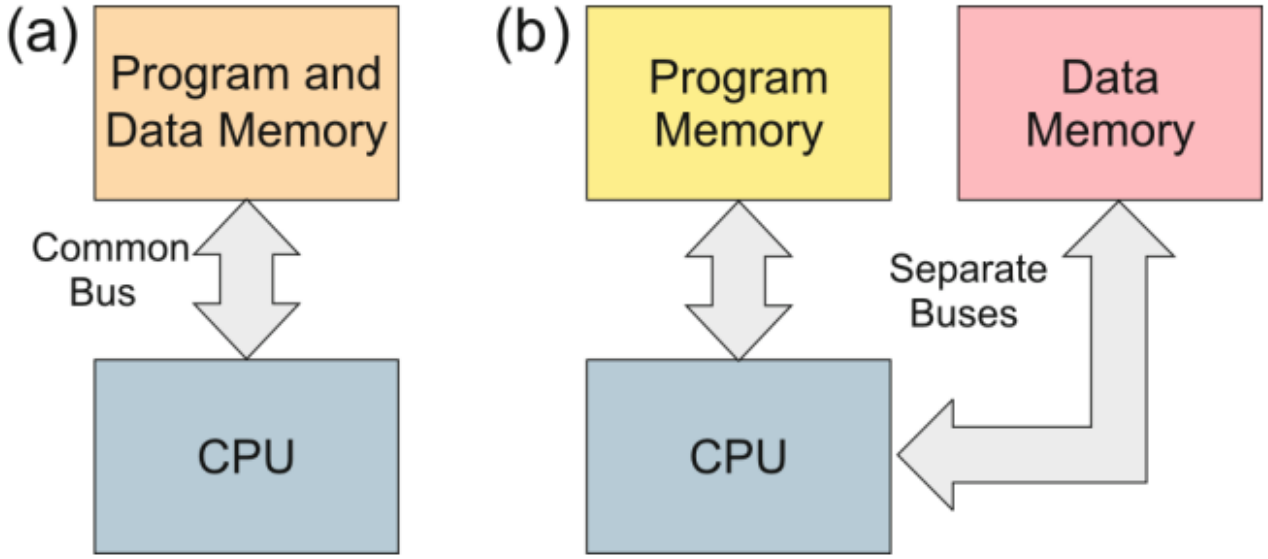


Fig. 3.10 Topological difference between (**a**) von Neumann and (**b**) Harvard architectures

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

CONCEPTUAL MEMORY CONNECTION

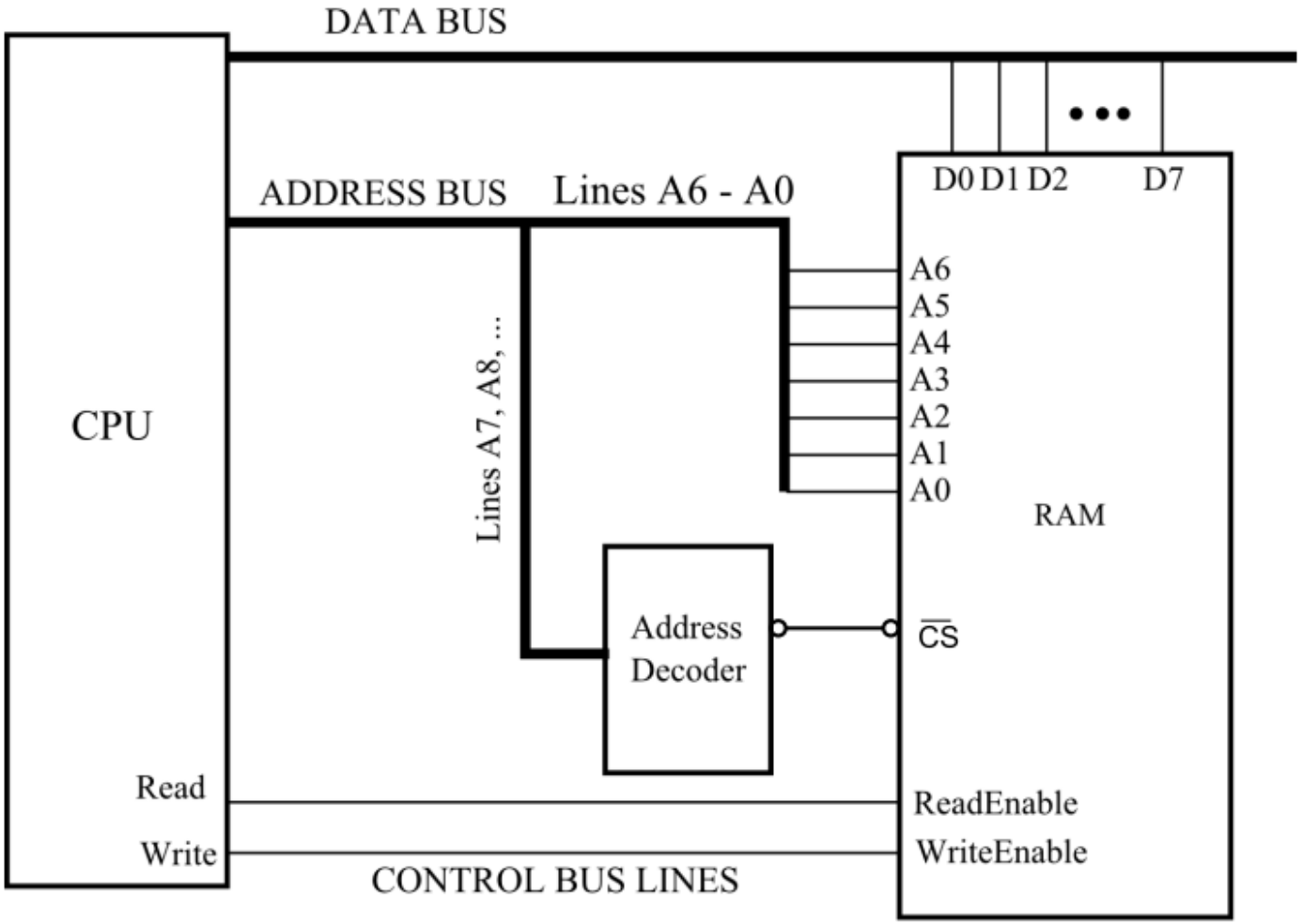


Fig. 3.11 CPU to memory connection: a conceptual scheme

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

TWO BANK DECODER (EXAMPLE 3.6)

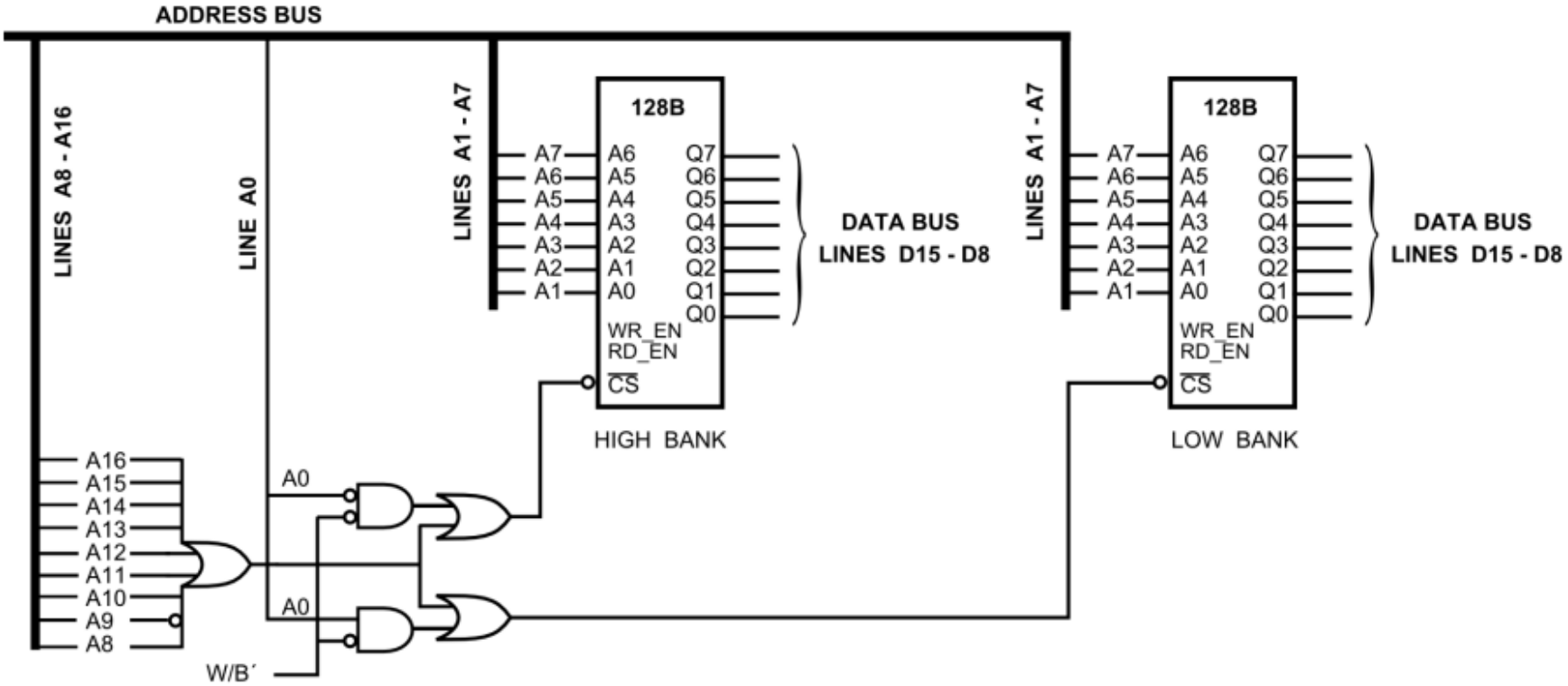


Fig. 3.12 256B memory bank example

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

Example 3.6 *Figure 3.12 illustrates an example of the connection illustrated by Fig. 3.8. It shows two 128B memory modules connected to a 16-bit data bus and driven by a 16-bit address bus and a control signal* ***W/B'****. Since each block has only eight I/O terminals, two of them are required to cover the data bus lines. The control signal* ***W/B'*** *indicates if the CPU is reading or writing a word (W/B' = 1) or a byte (W/B' = 0). Determine the range of addresses for this memory bank, and what addresses are for each block.*

Solution: *The blocks are activated when a low signal appears at the $\overline{CS}$ terminal, which are connected to OR outputs of the encoder subsystem. If the output of the eight input OR is high, the system is disconnected from the data bus. Hence, we need A15 = A14 = A13 = A12 = A11 = A10 = 0, A9 = 1, and A8 = 0 to ensure that the blocks can be activated. The address lines A7 to A1 activate the internal address lines of the activated RAM block.*

Now, if signal W/B' = 1 then both blocks are enabled, irrespectively of A0's value. If W/B' = 0 then A0 = 0 activates the low bank RAM and A0 = 1 the high bank RAM. Therefore, the range of addresses covered by the memory bank is as follows:

Range: From 0000 0010 0000 0000B to 0000 0010 1111 1111B, i.e., 0200h to 02FFh.

Low bank: Activated when A0 = 0, meaning even addresses in the range, that is: 0200h, 0202h, 0204h ..., 02FEh.

High bank: Activated when A0 = 1, meaning odd addresses in range: 0201h, 0203, ..., 02FFh.

MEMORY MAP EXAMPLE

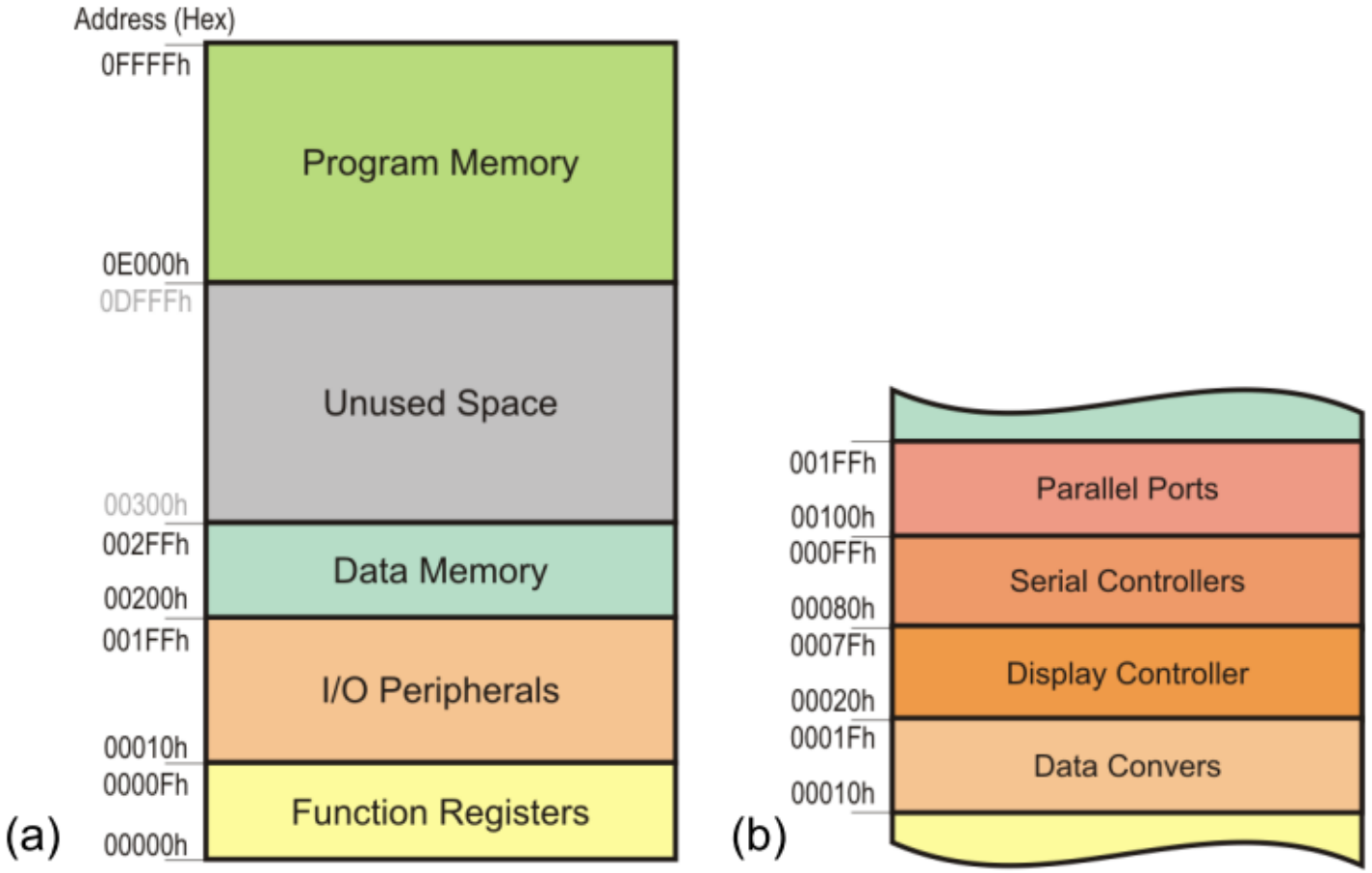


Fig. 3.13 Example memory map for a microcomputer with a 16-bit address bus. **a** Global memory map, **b** Partial memory map.

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

16 AND 20-BIT MSP430 MEMORY MAPS

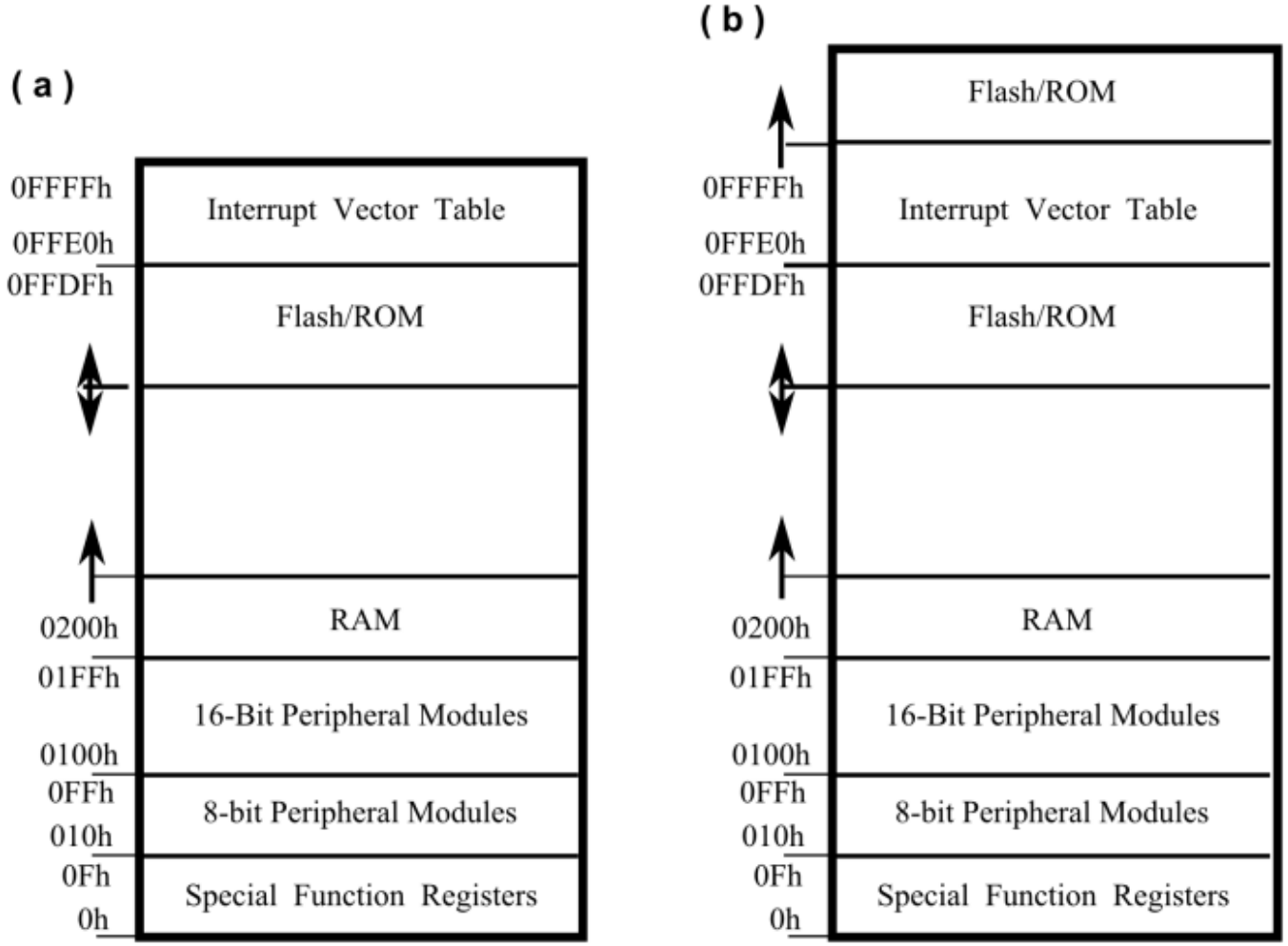


Fig. 3.14 MSP430 global memory maps (*Courtesy of Texas Instruments, Inc*). **a** 16-bit address bus memory map, **b** 20-bit address bus memory map

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

3.6 I/O SUBSYSTEM ORGANIZATION

- Anatomy of an I/O Interface
- Parallel Versus Serial I/O Interfaces
- I/O and CPU Interaction
- Timing Diagrams
- MSP430 I/O Subsystem and Peripherals

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

I/O SUBSYSTEM

- All the components other than the CPU & memory connected to the system buses
  - Timers & Watchdog timers
  - Communication interfaces
  - Analog to Digital Converter (ADC)
  - Digital to Analog Converter (DAC)
  - Development peripherals
- Its organization resembles that of memory
- Memory mapped I/O
  - Address space inside the memory space, uses same instructions used to access memory
- I/O mapped I/O
  - Separate address space, instructions, and signals for I/O (rarely used in modern processors)

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

CONCEPTUAL I/O INTERFACE

- Serves as a bridge between a device & the buses
- Has one or more registers each with their own addresses
  - Data, control, and status registers
- Accessed like memory (read/write)

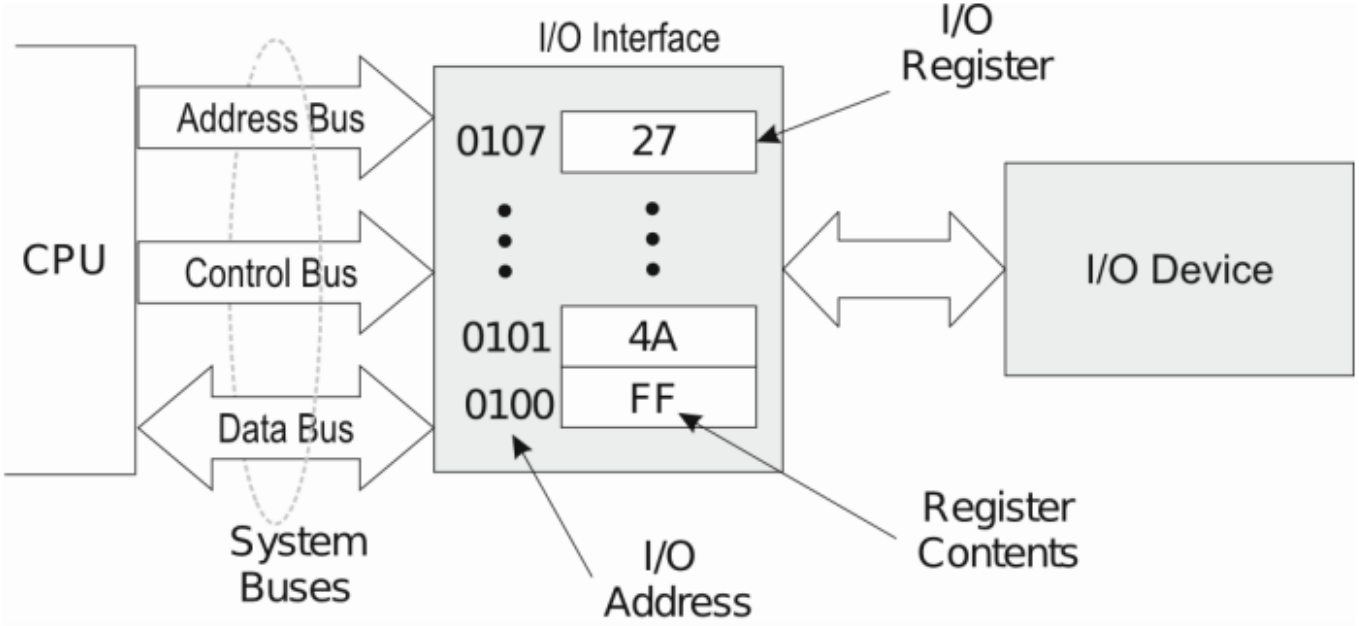


Fig. 3.15 Structure of an input/output (I/O) interface

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

I/O INTERFACE ANATOMY

- Includes lines to connect to the system buses, I/O device connection lines, and a set of internal registers
- Internal register types
  - Control: to configure the operation of the device & interface
  - Status: to allow inquiries about the device & interface status
  - Data: for exchanging data with the device

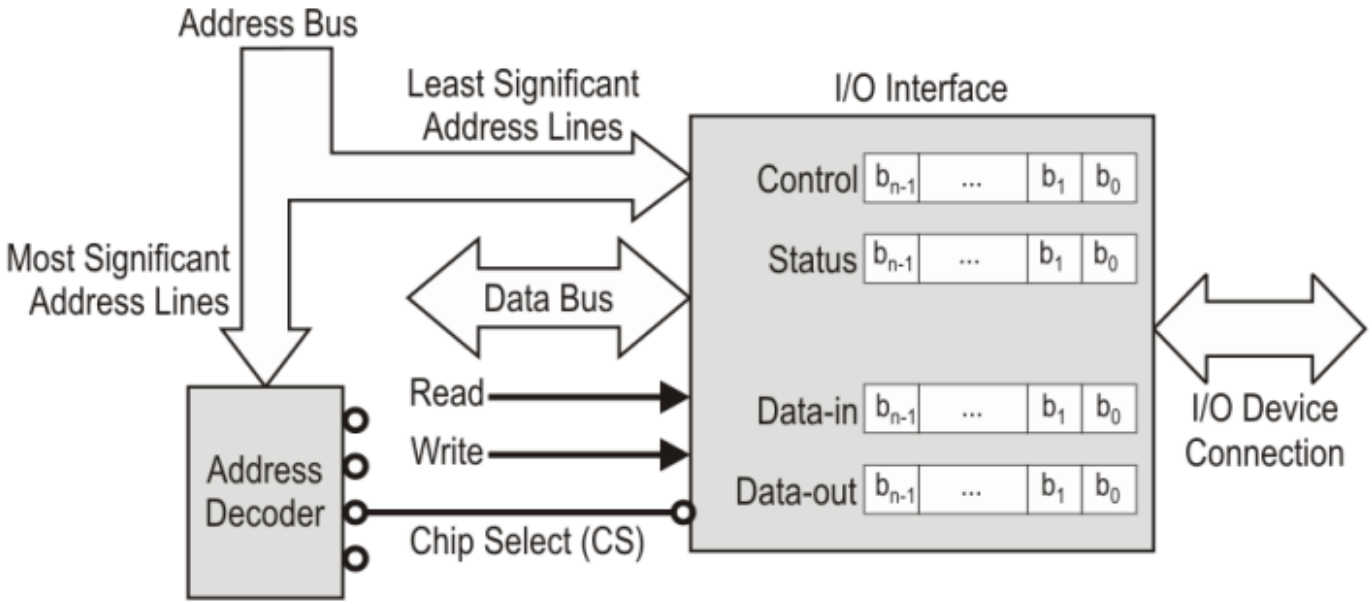


Fig. 3.16 Anatomy of an input/output (I/O) interface

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

EXAMPLE 3.7 INTERFACE EXAMPLE

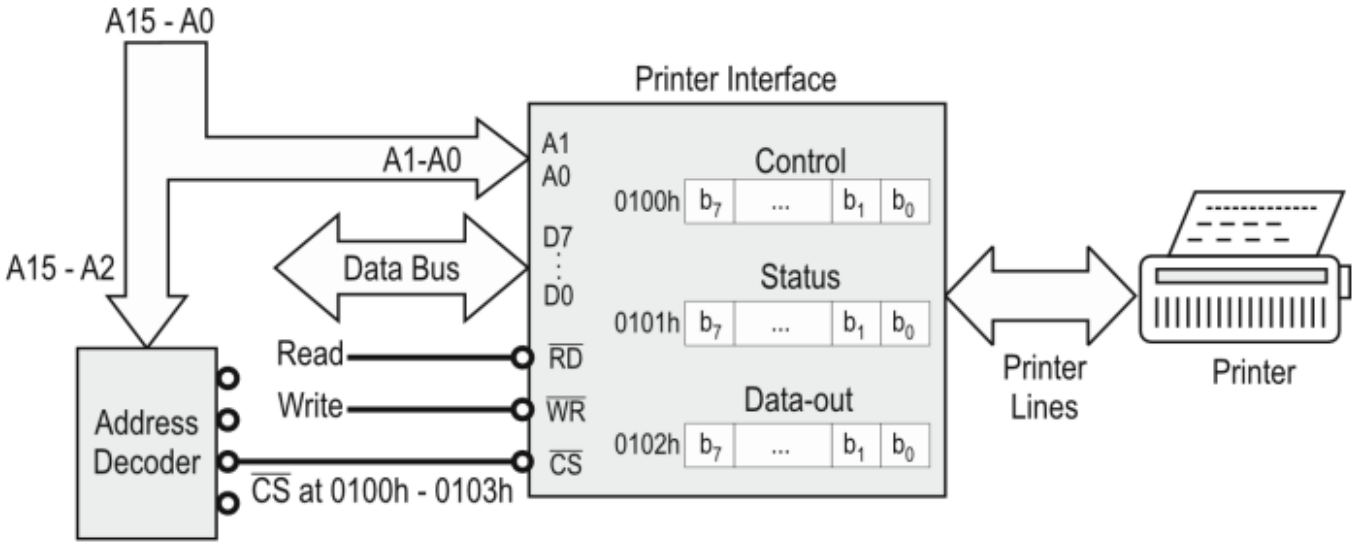


Fig. 3.17 Connection of printer interface to CPU buses and printer itself

Table 3.3 Internal addresses for sample printer interface

A1	A0	Register
0	0	Control
0	1	Status
1	0	Data-out
1	1	Not used

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

EXAMPLE 3.7 INTERFACE REGISTERS

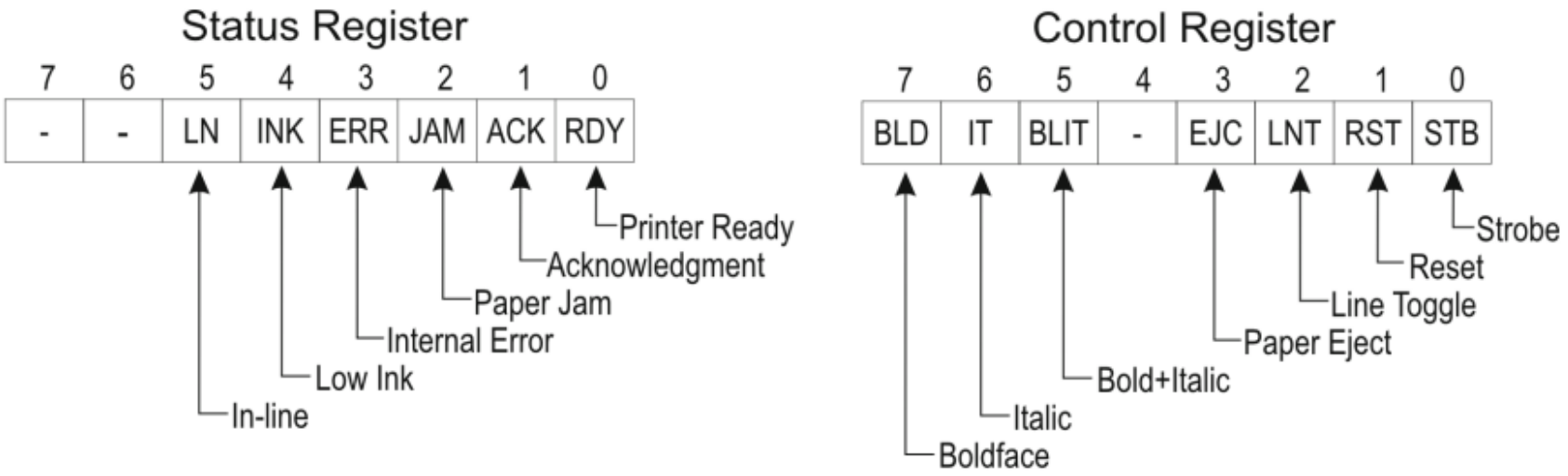


Fig. 3.18 Status and control registers in printer interface example

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

I/O PIN HARDWARE CONFIGURATION

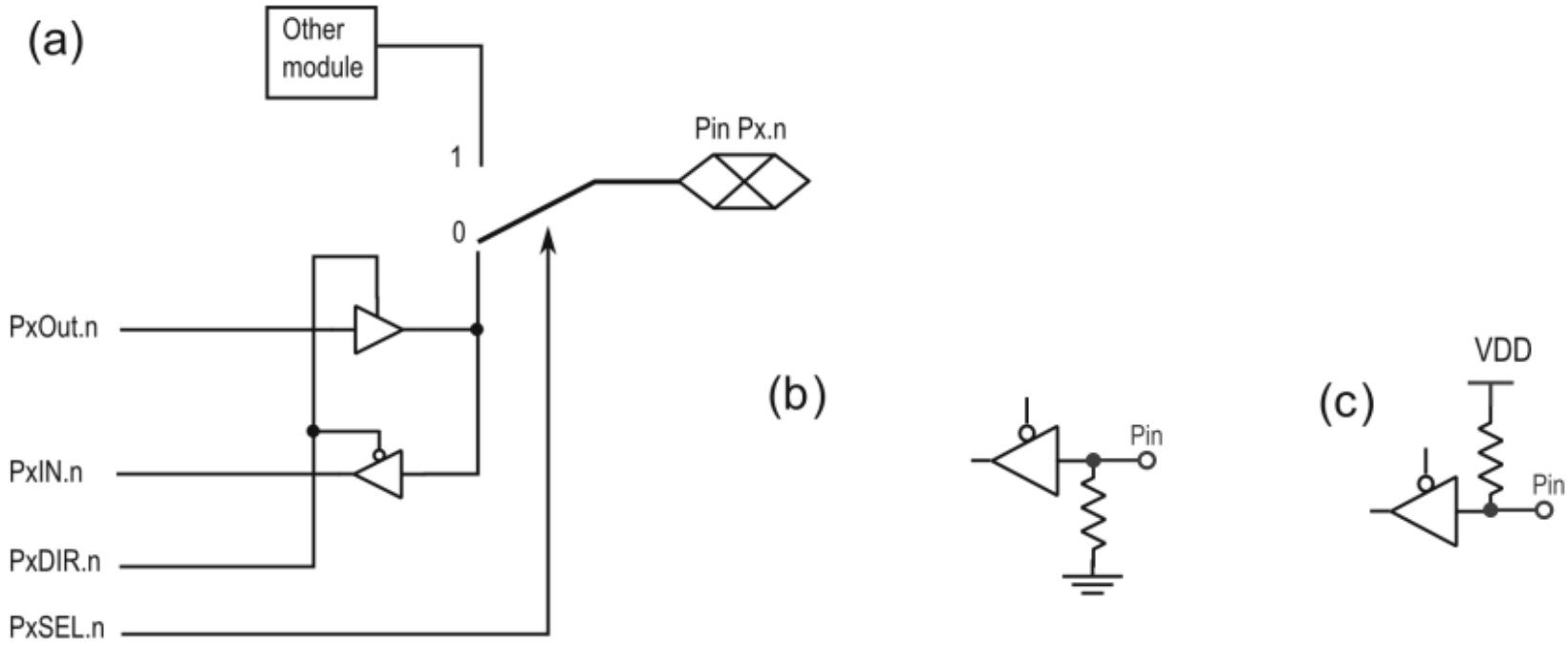


Fig. 3.19 Basic IO Pin hardware configuration: **a** Basic I/O register's functions; **b** pull-down resistor for inputs; **c** pull-up resistors

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

TIMING DIAGRAMS

- The CPU and hardware components communicate exchanging signals sent over the system buses
- Signal exchanges require protocols
  - Indicate the times required for the steps in the process
  - Settling of states
  - Delays
  - Order of signals activation, etc.
- Timing diagrams are the most common way to define a signal protocol

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

TIMING DIAGRAM CONVENTIONS

Single Line and Bus Timings

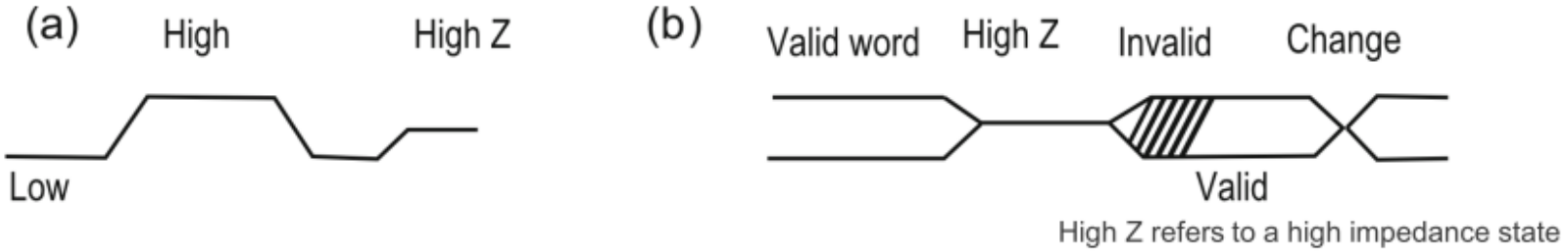


Fig. 3.20 : Definitions of timing conventions. **(a)** Single signal timing, **(b)** Bus timing

MPU Read and Write Timing Diagrams

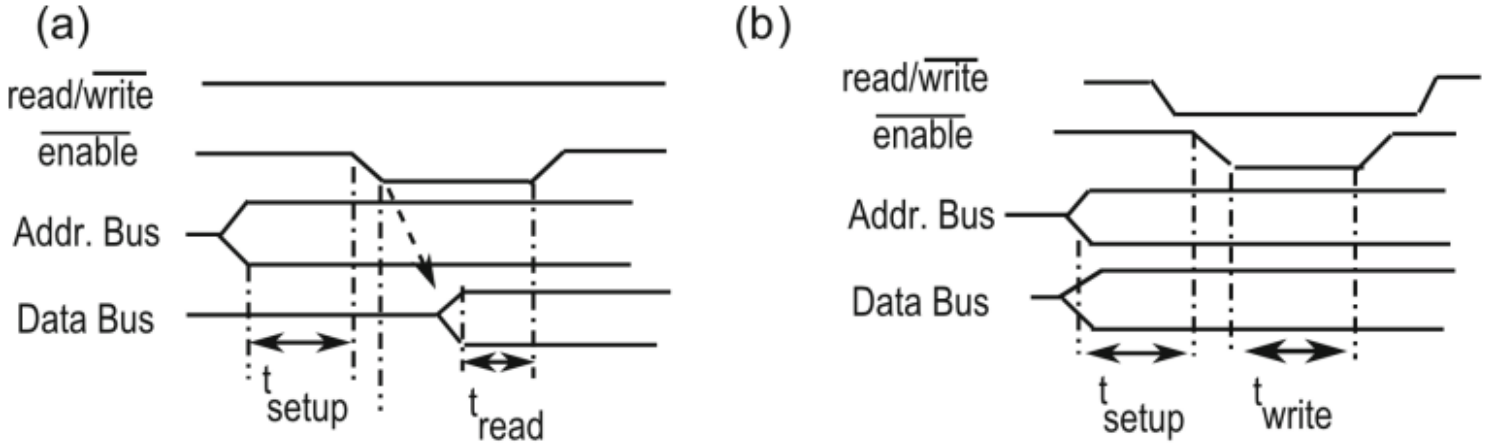


Fig. 3.21 Example of simplified write &read timing diagrams. **(a)** Read timing, **(b)** Write timin

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

© 2014 by M. Jiménez, R. Palomera, & I. Couvertier

DESIGN TIPS

مدينة زويل
ZEWAIL CITY
ESTABLISHED 200

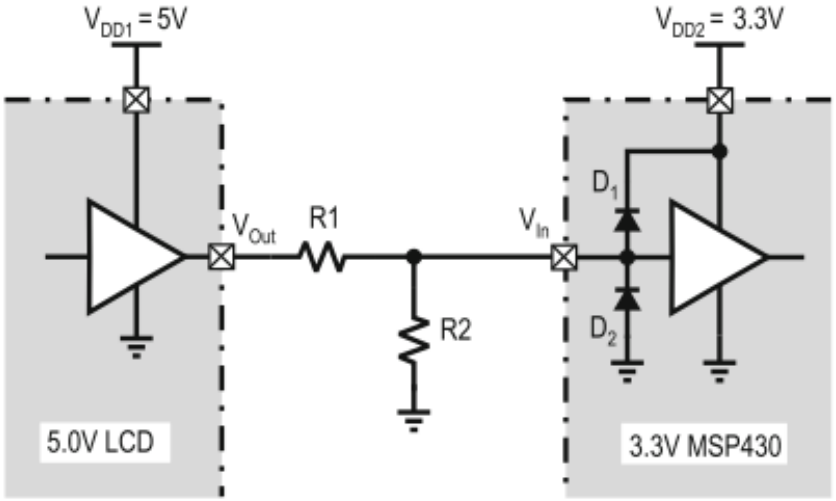
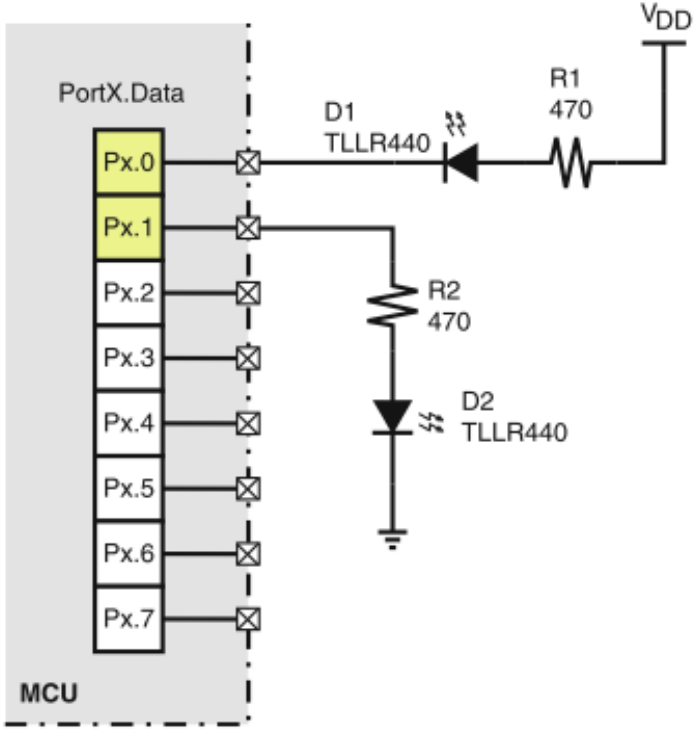


Fig. 8.8 A practical 5.0–3.3 V level shifter

Fig. 8.31 Driving LEDs with I/O pins



Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

ZEWAIL CITY ESTABLISHED 2000

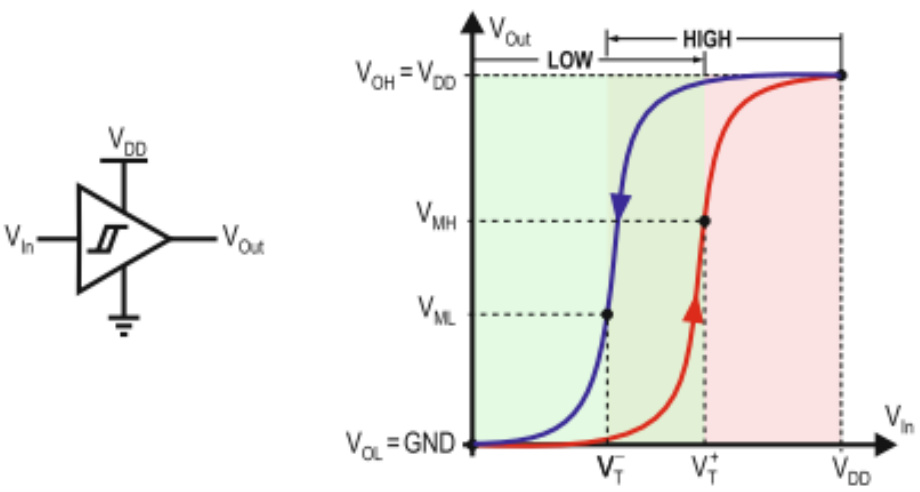


Fig. 8.11 Symbol and VTC of a Schmitt-trigger buffer

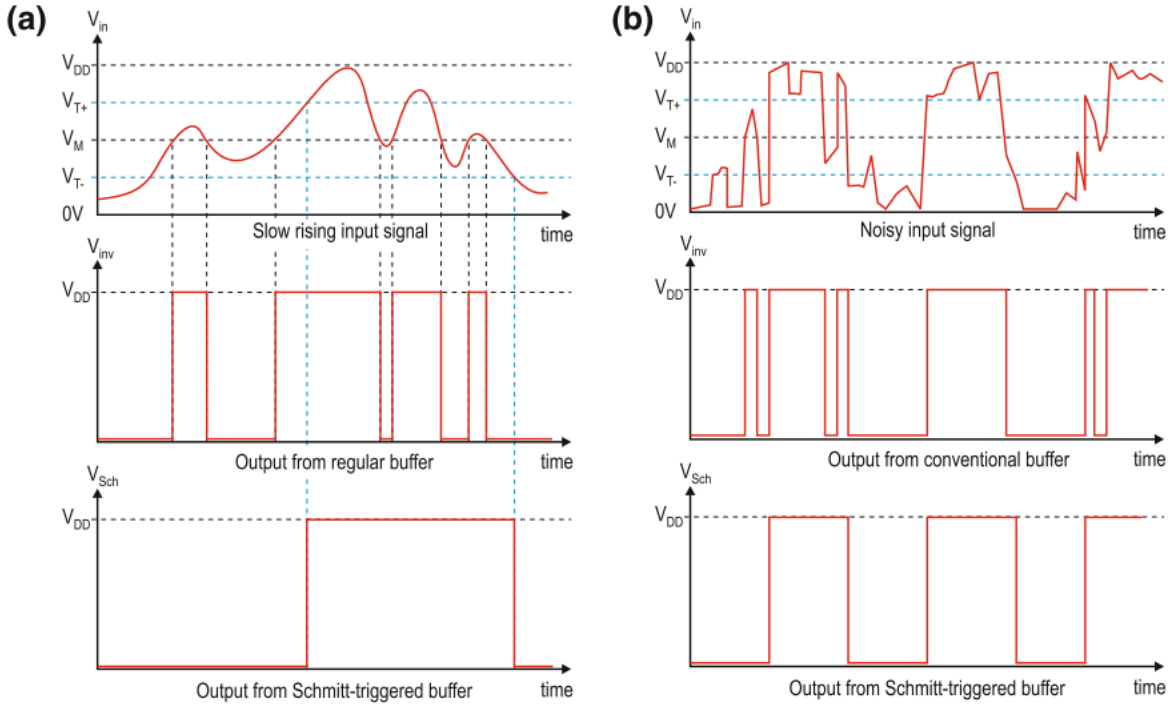
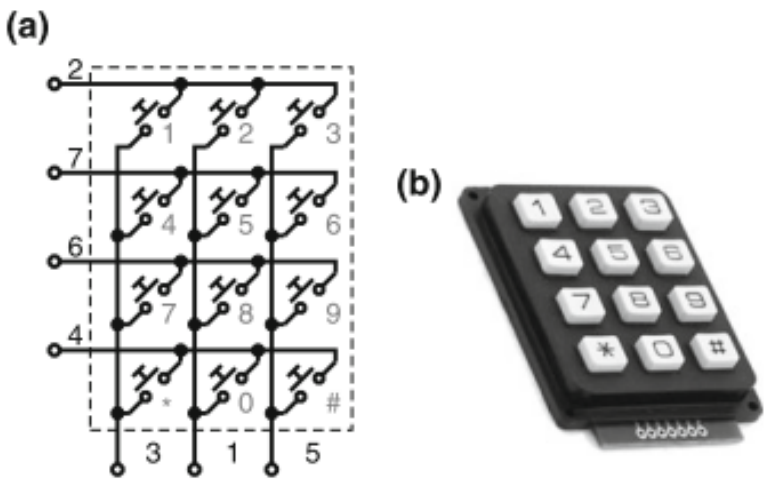


Fig. 8.12 Schmitt-trigger effect on noisy and slow rising inputs. **a** Response to slowly changing input, **b** Response to noisy input

ZEWAIL CITY
ESTABLISHED 2000

Fig. 8.26 A 3-by-4 keypad showing the internal arrangement and appearance. **a** Switch arrangement, **b** Actual keypad



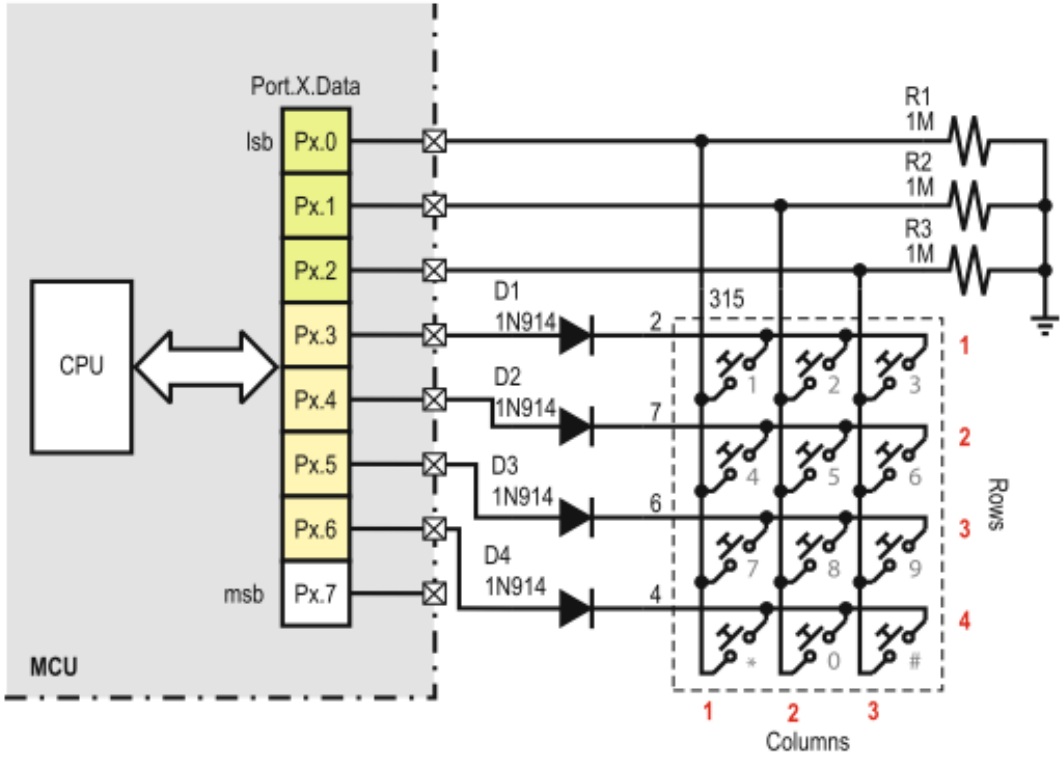


Fig. 8.27 Interfacing a 3-by-4 keypad to a GPIO port

How to scan?

Flowchart in Fig. 8.28

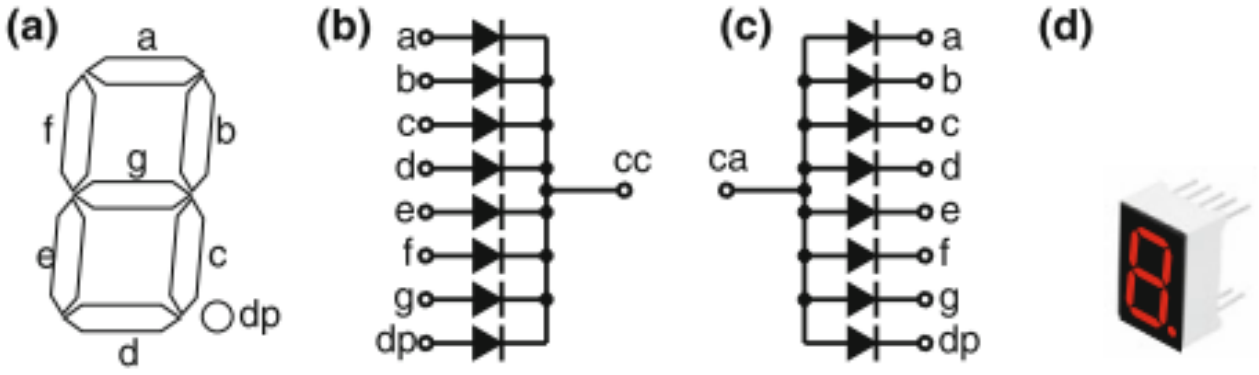


Fig. 8.32 Ordering of individual LEDS in a 7-segment display. **a** Segment arrangement, **b** Common cathode, **c** Common anode, **d** Actual Component

Fig. 8.33 Directly driving a 7-segment display module from a GPIO port

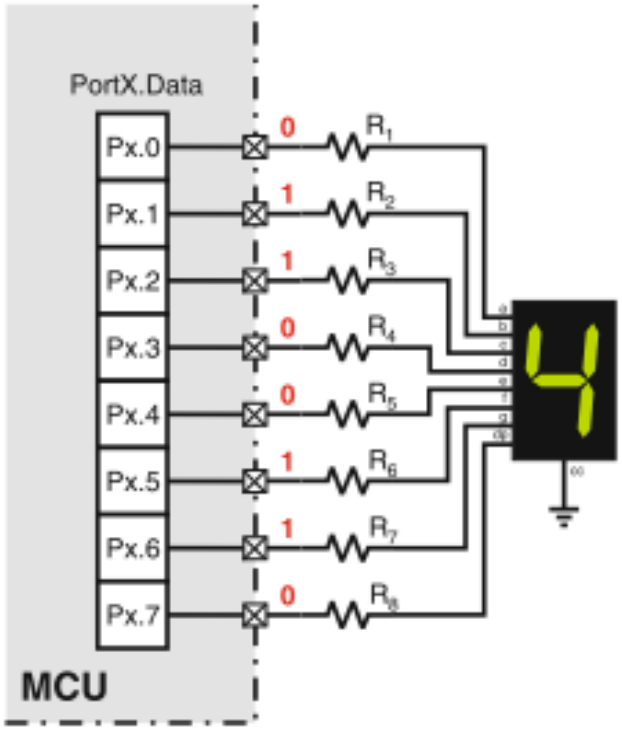
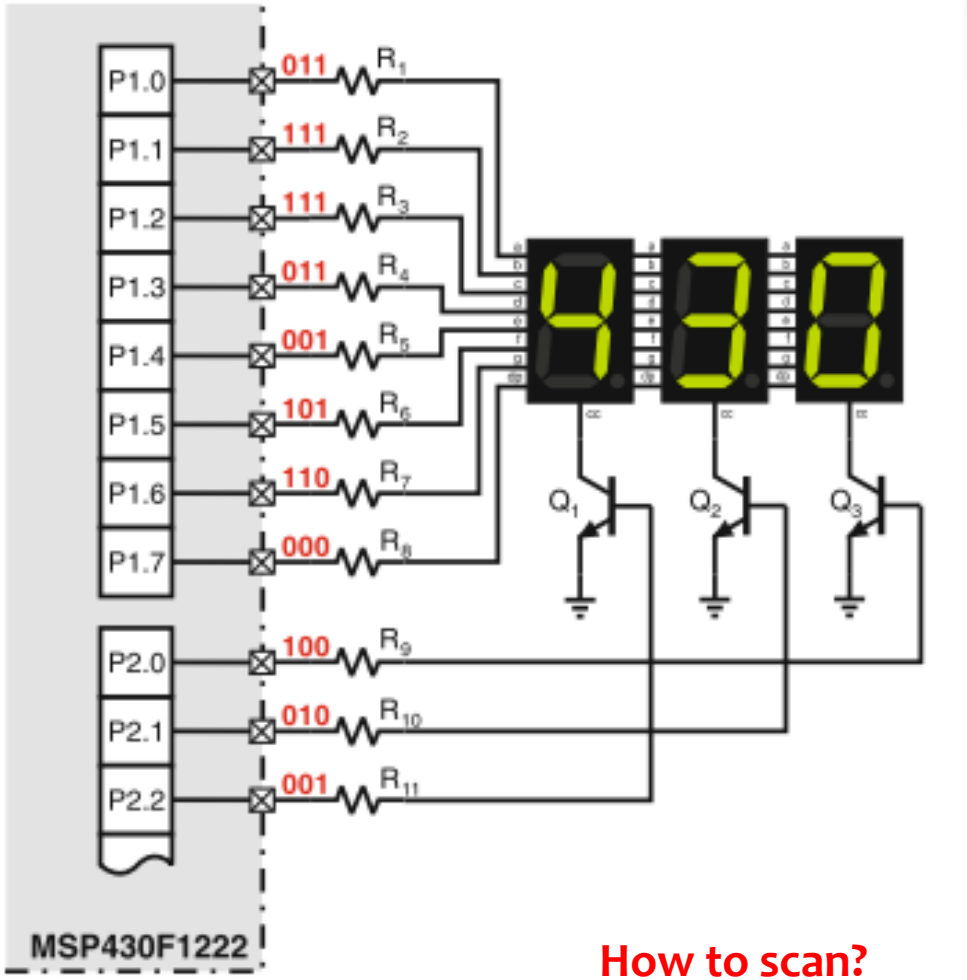


Fig. 8.34 Multi-digit seven-segment display arrangement



How to scan?

Flowchart in Fig. 8.35

ZEWAIL CITY ESTABLISHED 2000

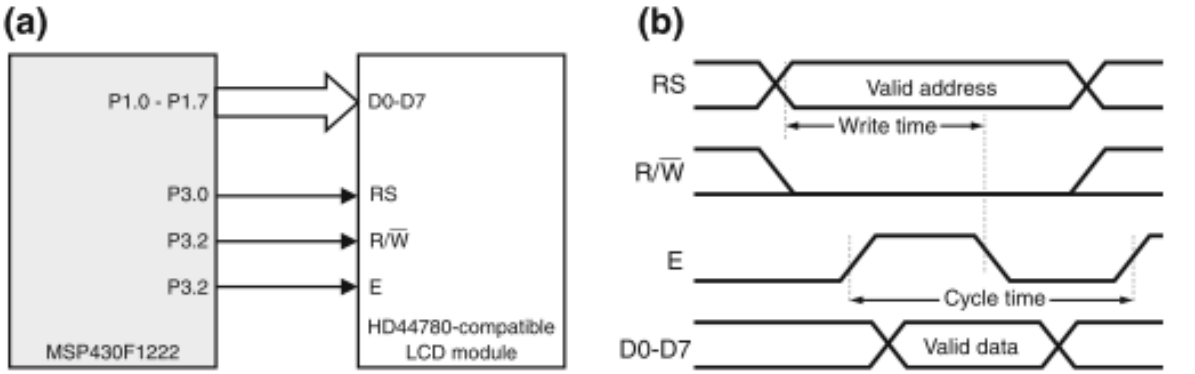


Fig. 8.45 Connecting an alphanumeric LCD module through GPIO lines. **a** GPIO connection, **b** signal timing

Table 8.1 Signals in an HD44780-compatible alphanumeric LCD module

Signal	Type	Description
RS	Control	Command/Data: '0' = Command, '1' = Data
$R/\overline{W}$	Control	Read/Write line: '0' = Write, '1' = Read
E	Control	Enable strobe
D0 – D7	Data	Bidirectional data lines

Fig. 8.46 A common 2 × 16 alphanumeric LCD module

Embedded Sys. Fundamentals Spring 17 © Ahmad El-Banna

ZEWAIL CITY

Assignment#2

1. Differentiate between a digital camera & an audio amplifier devices and state which one of them is considered an embedded system application.
2. Determine the Carry, Zero, Negative, and Overflow flags after the operation A3h + B2h.
3. Solve problem 1.14 in your textbook.
   - **Introduction to Embedded Systems,** Springer 2014 by Manuel Jiménez et al.

- Deadline:
  - due next Wednesday, 8 March.

- For more details, refer to:
  - Chapter 3,8 at **Introduction to Embedded Systems,** Springer 2014 by Manuel Jiménez et al.

- The lecture is available online at:
  - *http://bu.edu.eg/staff/ahmad.elbanna-courses*

- For inquires, send to:
  - ahmad.elbanna@feng.bu.edu.eg