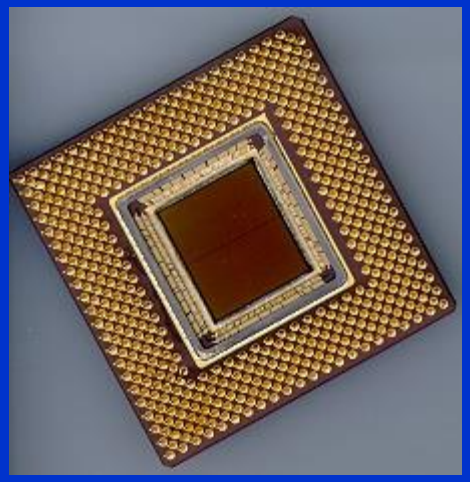


ECE 477 Digital Systems Senior Design Project

Module 9

Documentation Standards

Reading Assignment: Wakerly *DDPP* – 3rd Ed., pp. 311-337; 4th Ed., pp. 342-370

Instructional Objectives:

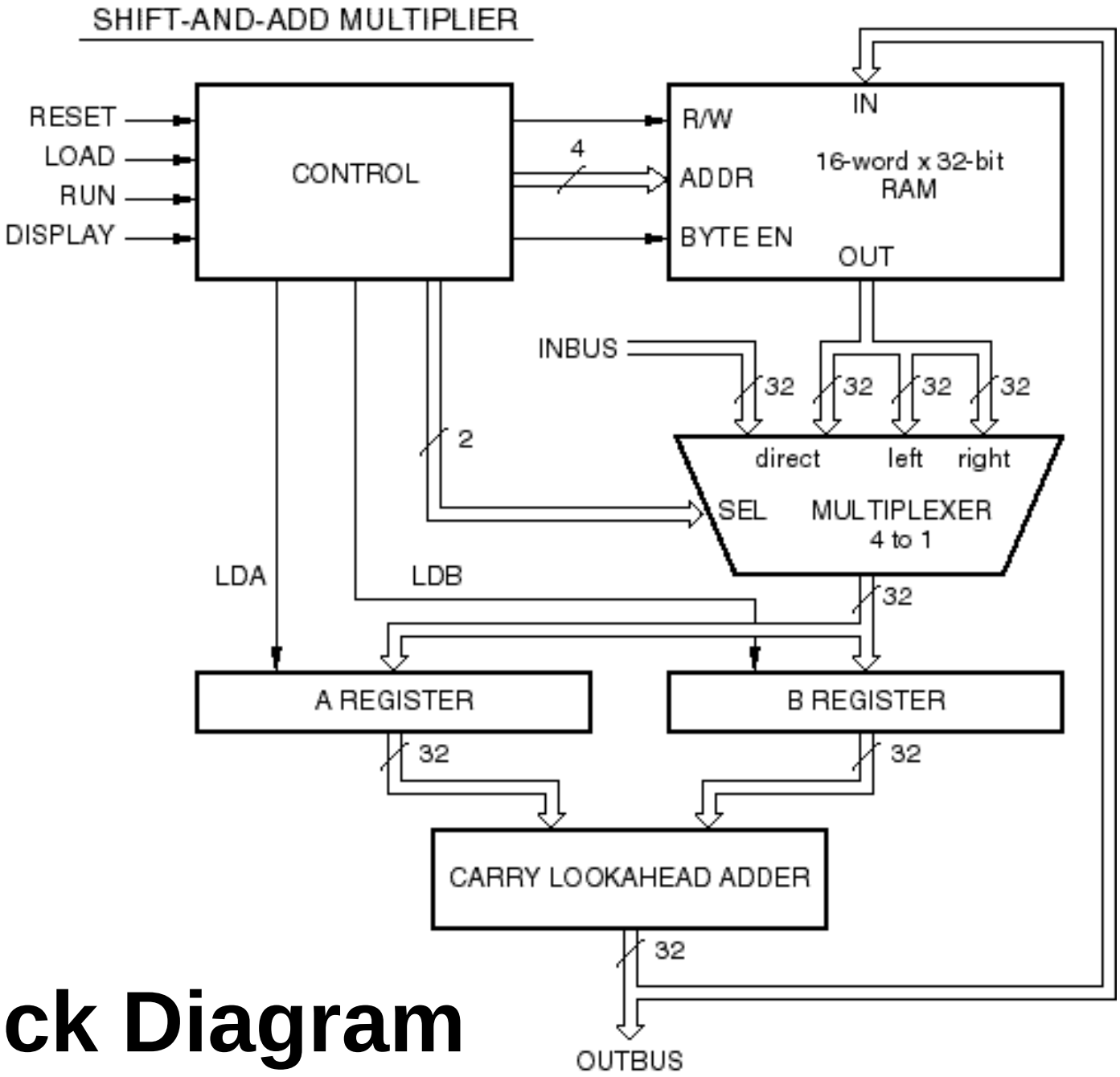
- **To review the foundations of structured digital system design**
- **To review combinational logic design documentation standards**
- **To review how to draw timing diagrams and use timing specifications**

Outline

- Documentation standards
- Active levels for pins
- Drawing layout
- Timing diagrams and specifications

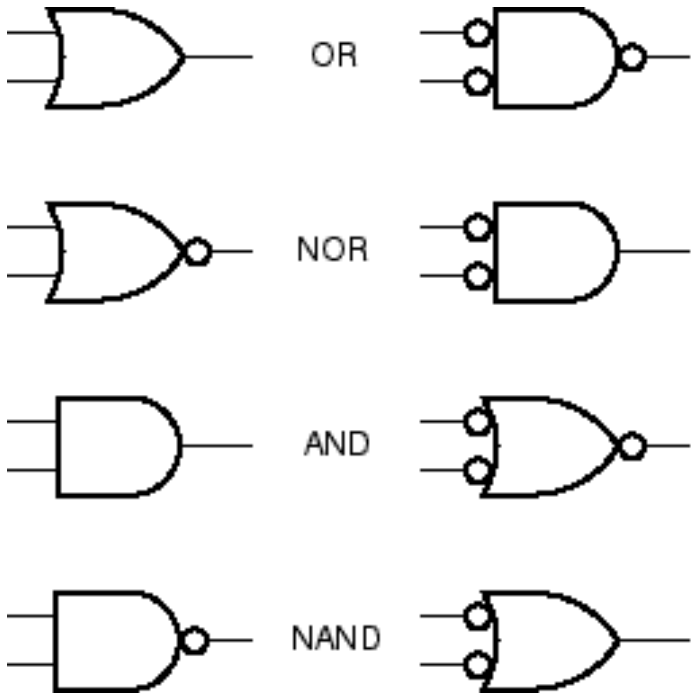
Documentation Standards

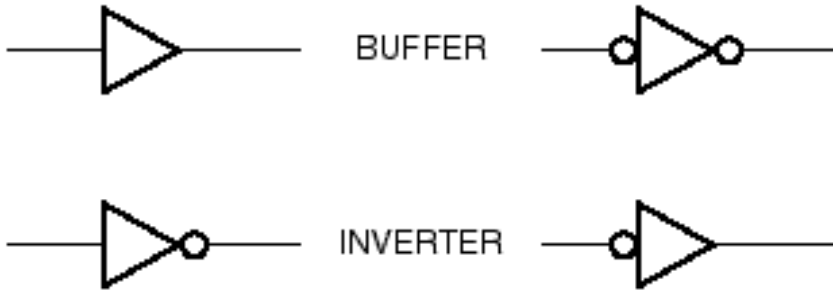
- Good documentation is essential for correct design and efficient maintenance
- A documentation package should contain:
  - abstract
  - specification
  - block diagram
  - schematic diagram
  - PCB layout diagram
  - timing diagram
  - structured logic device description
  - circuit description (narrative text)
  - software description (narrative text)



Block Diagram

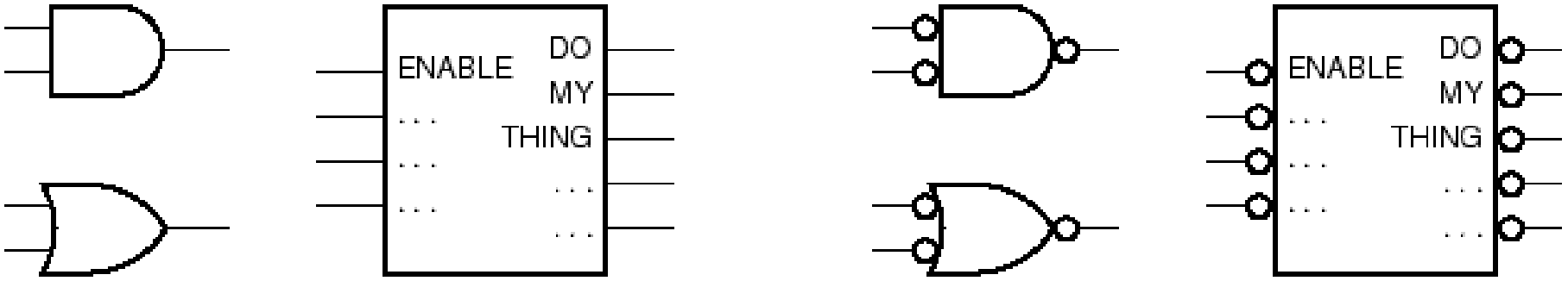
Equivalent Gate Symbols





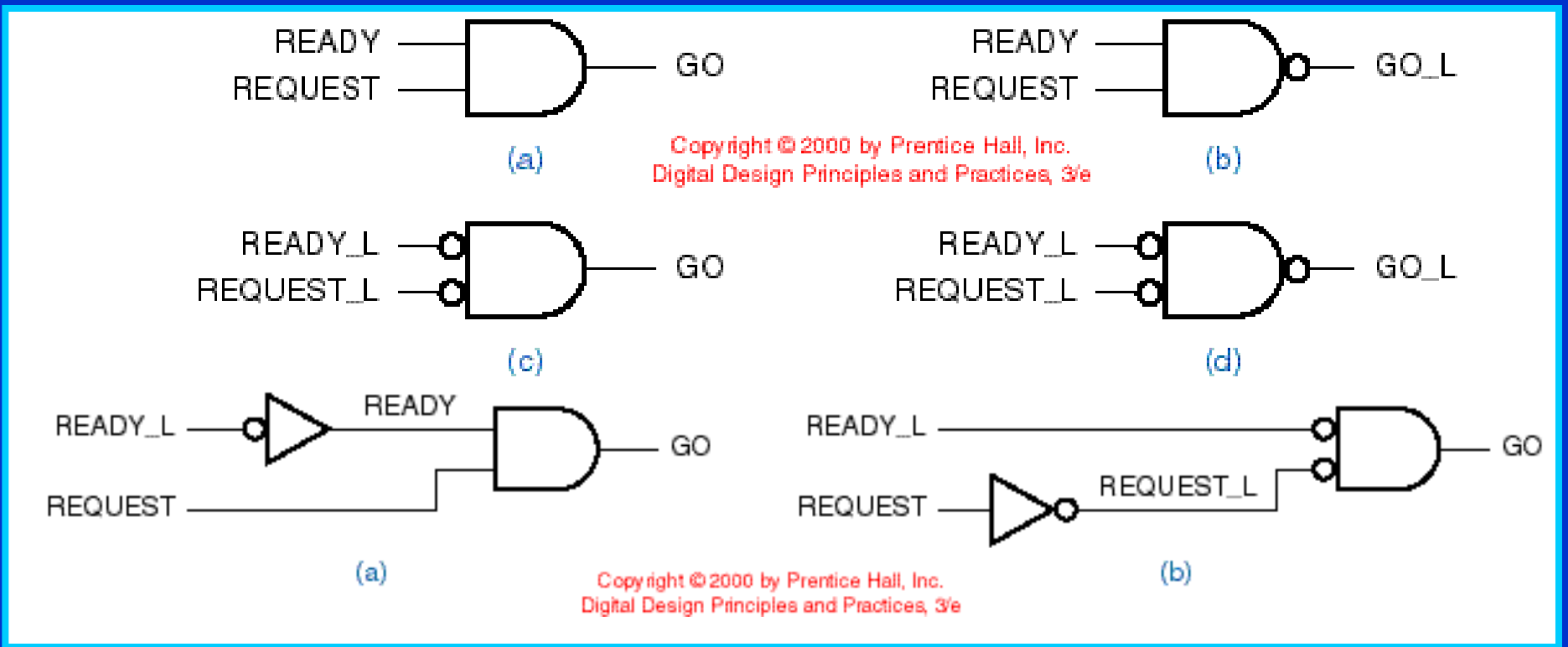
Active Levels for Pins

- When we draw a logic symbol, we think of a function being performed "inside" that symbolic outline
- Use ***inversion bubbles*** to indicate pins that are ***active low***



Active Levels for Pins

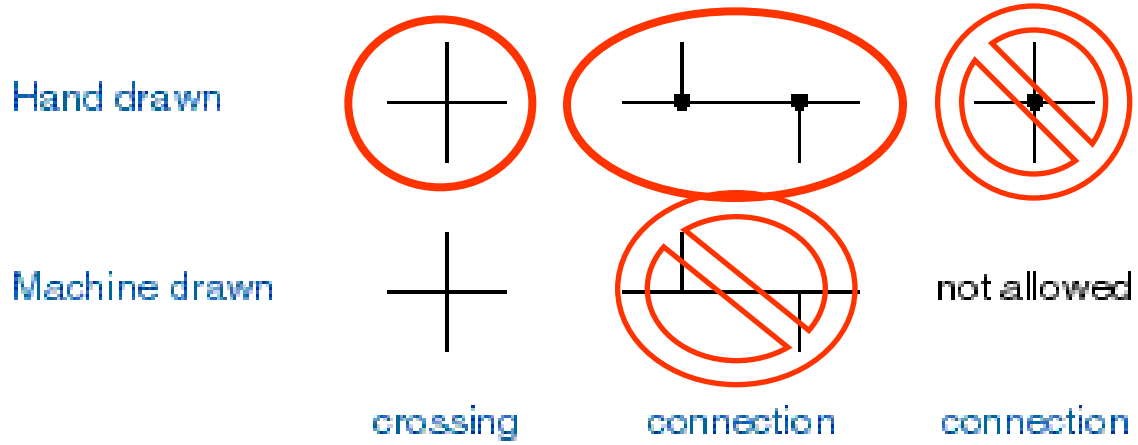
- **It is very helpful to use net aliases with "_L" suffix to indicate active low for those signals that are active low**



Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

Schematic Guidelines

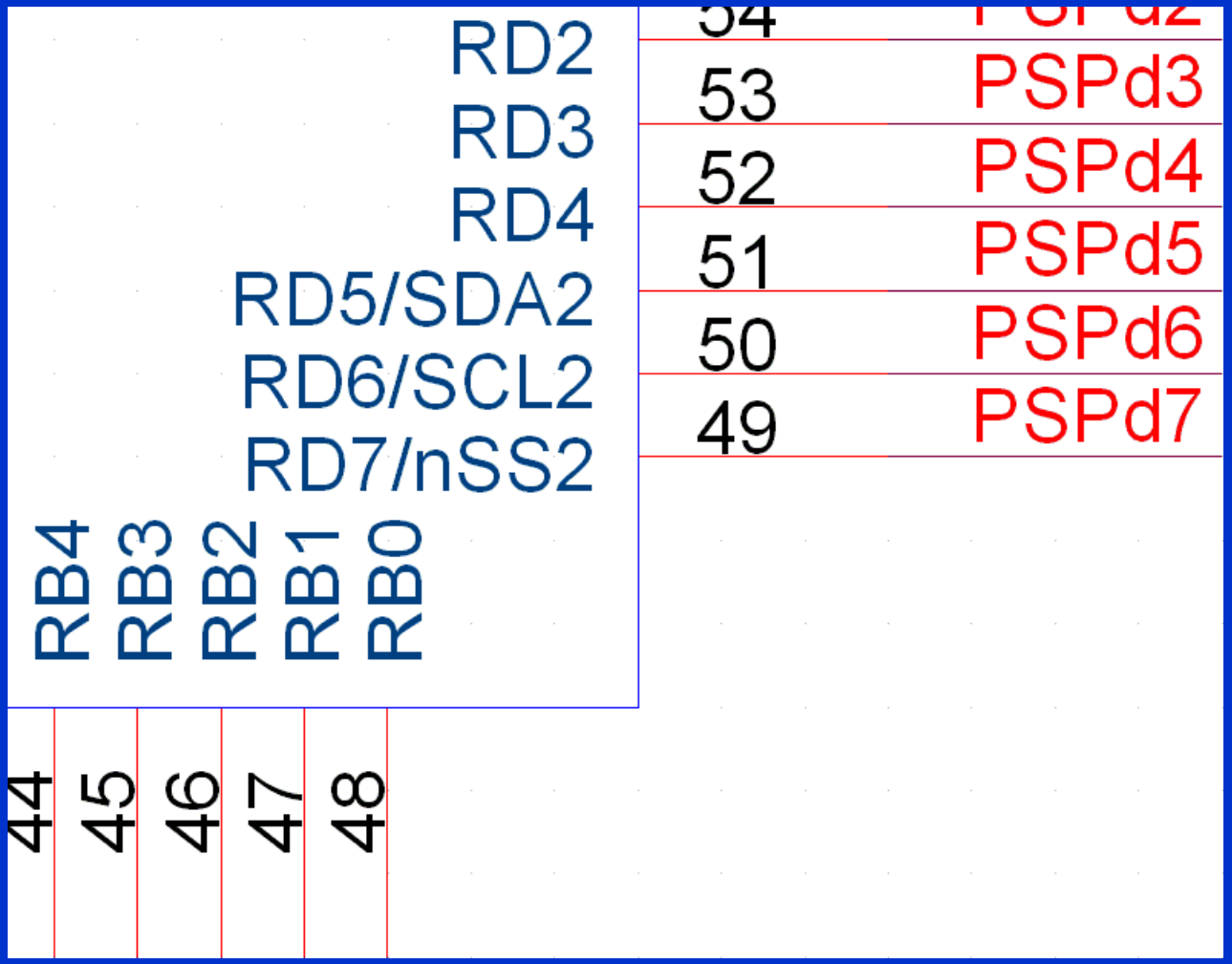
- Logic diagrams and schematics should be drawn with gates in their "normal" orientation (inputs on left, outputs on right)
- A complete schematic page should be drawn with system inputs on the left and outputs on the right
- Line crossings and connections



Schematic Guidelines (2)

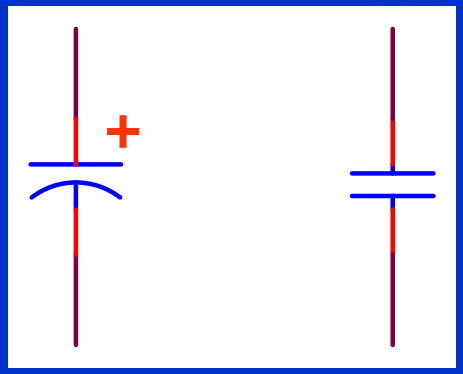
- Pin names on custom footprints should be descriptive, but not so long that they overlap other pin names (quad packs…)
- Either physical or logical arrangement is okay, but try to be consistent throughout document.
- Use power / ground symbols
- For power, show net alias on schematic

Schematic Guidelines (3)

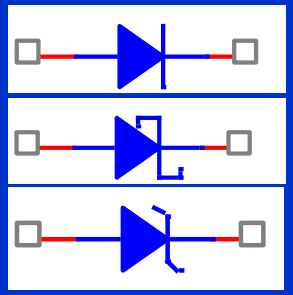


Schematic Guidelines (3)

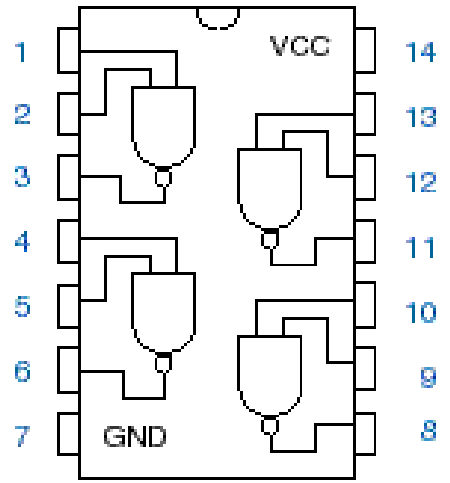
- **Polarized / non-polarized capacitors**
  - Value
  - Tolerance
  - Voltage
  - Footprint

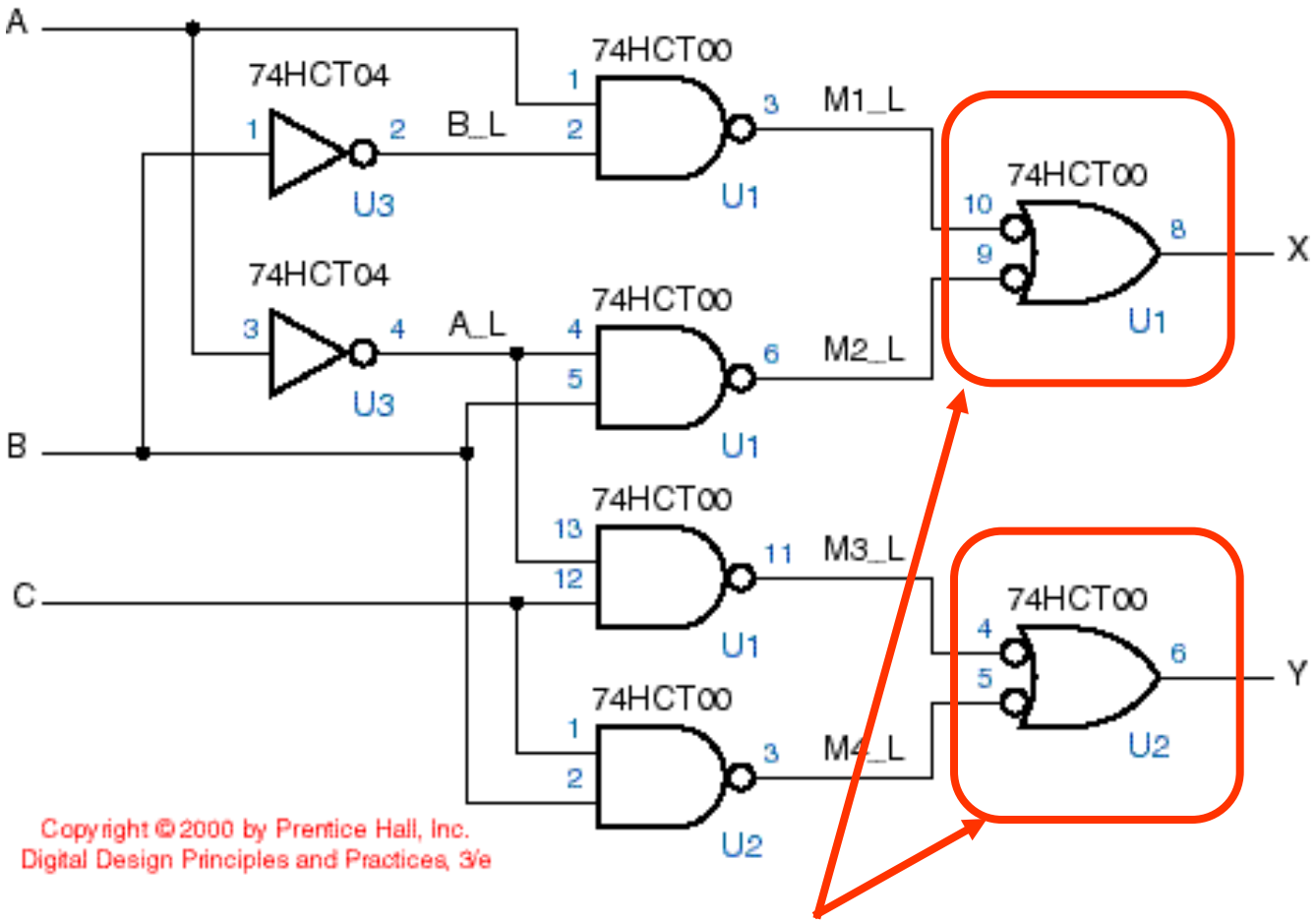


- **Diode types**
  - PN / Ge, etc.
  - Schottky
  - Zener
  - **designate by part #**



Pin Diagram and Schematic





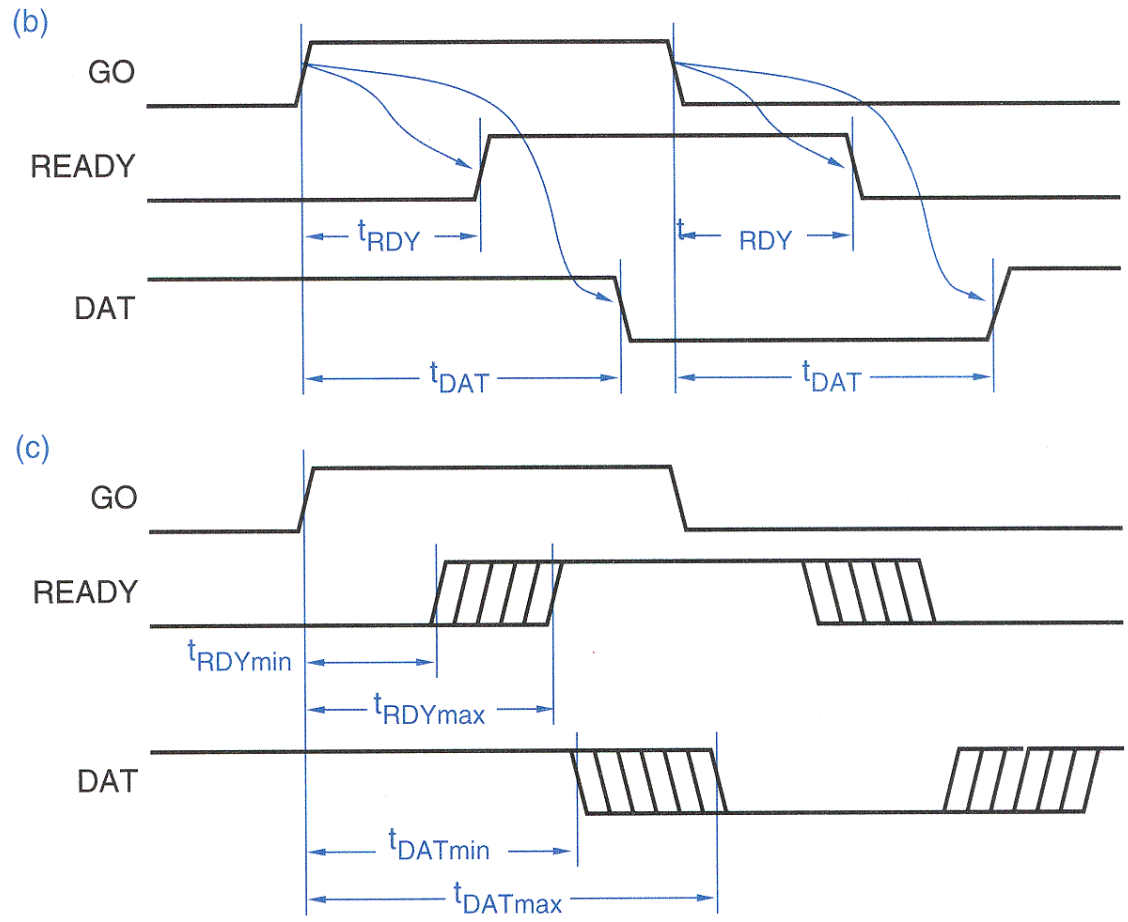
In ORCAD, these are called the "ALT" symbols

Timing Diagrams

- A ***timing diagram*** **illustrates the logical behavior of signals in a digital circuit as a function of time**
- **They are used to explain the timing relationship among signals within a system and to define the timing requirements of external signals applied to the system**
- **Arrows are sometimes drawn to show *causality* (which input transitions cause which output transitions)**

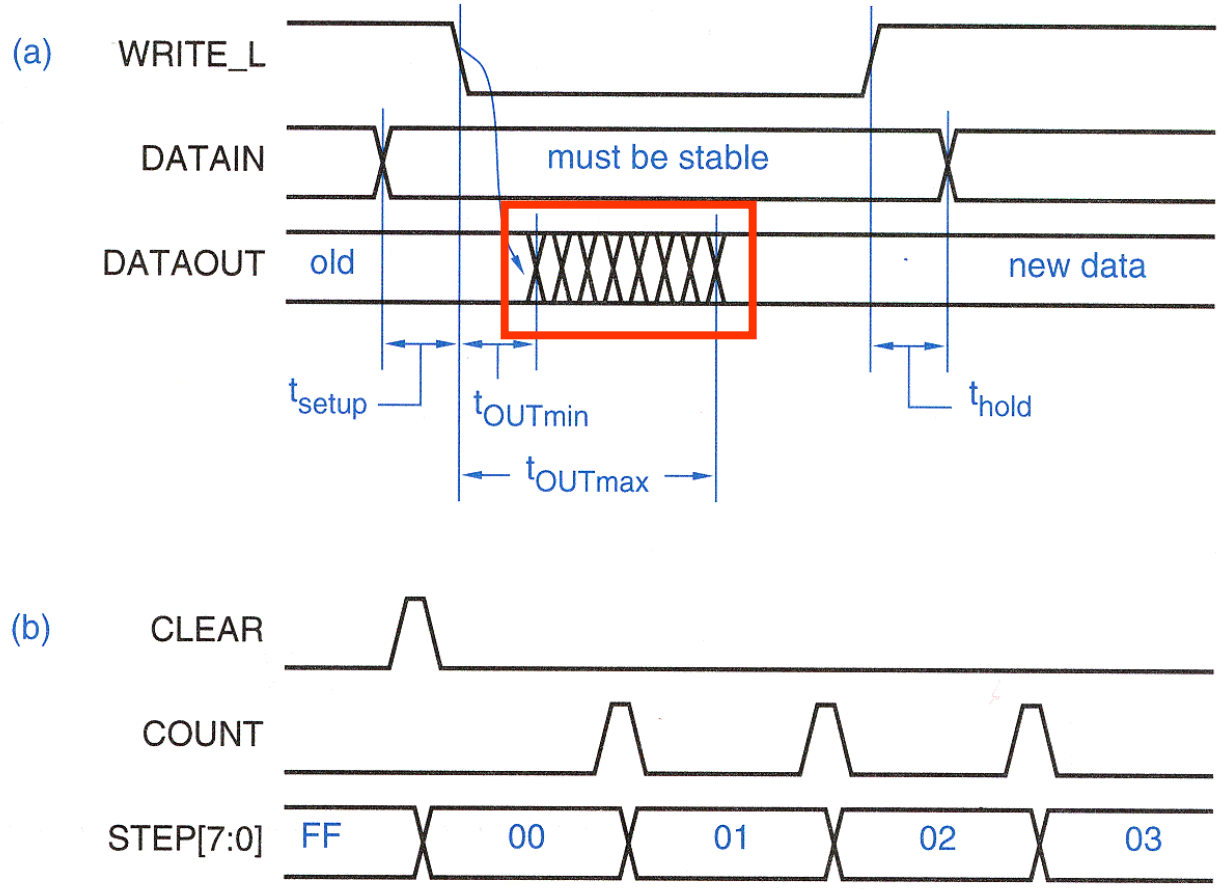
Timing Diagrams

- **Combinational circuit timing diagram**



Timing Diagrams

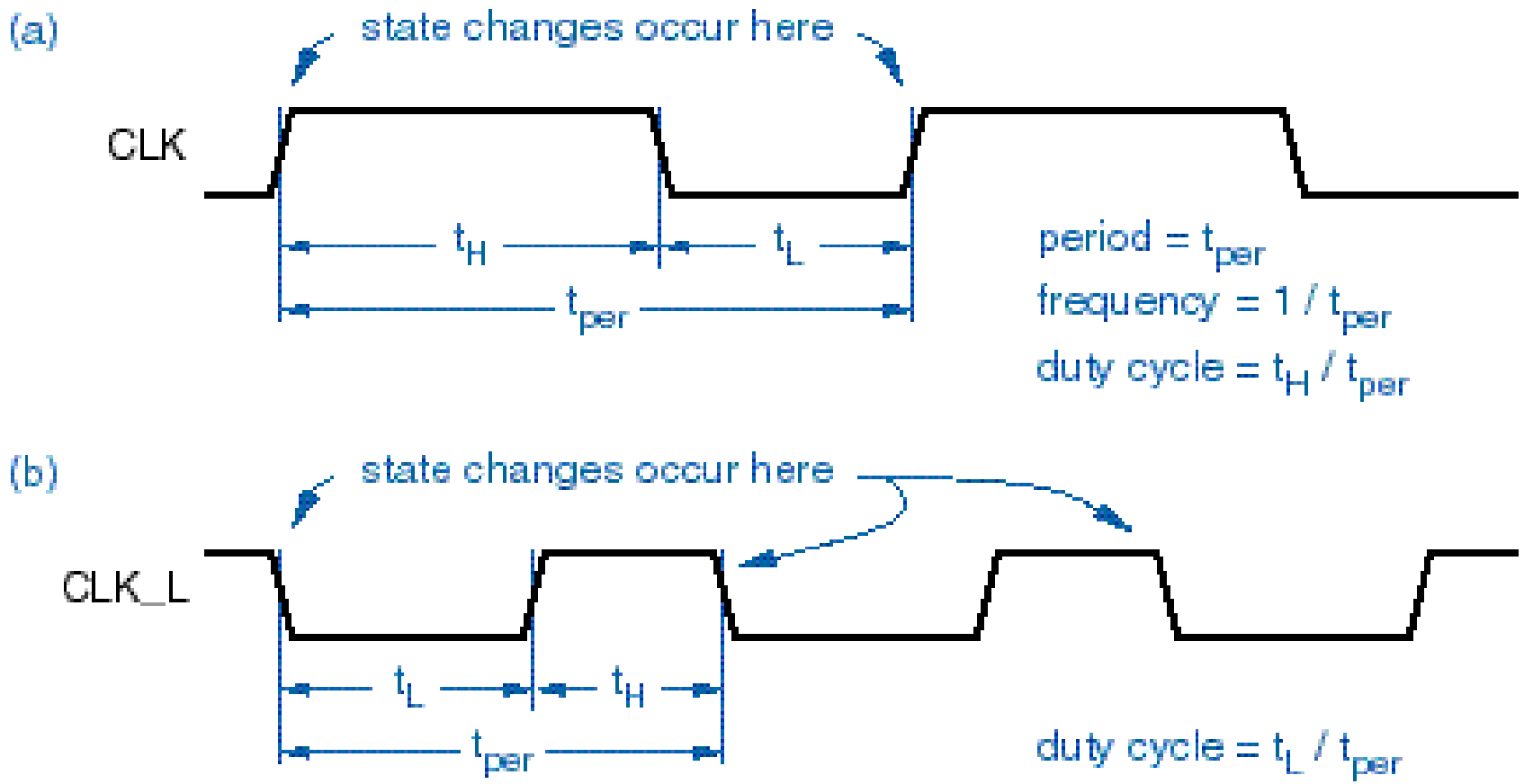
- **Timing diagrams for "data" signals**



Timing Specifications

- A combinational circuit with many inputs and outputs has many different paths
- Each path in a combinational circuit may have a different propagation delay
- The propagation delay when an output changes from **LOW** to **HIGH** (t_{PLH}) ***may be different*** than the delay when it changes from **HIGH** to **LOW** (t_{PHL})
- Timing specifications in data sheets usually provide minimum, typical, and maximum values for each propagation delay path

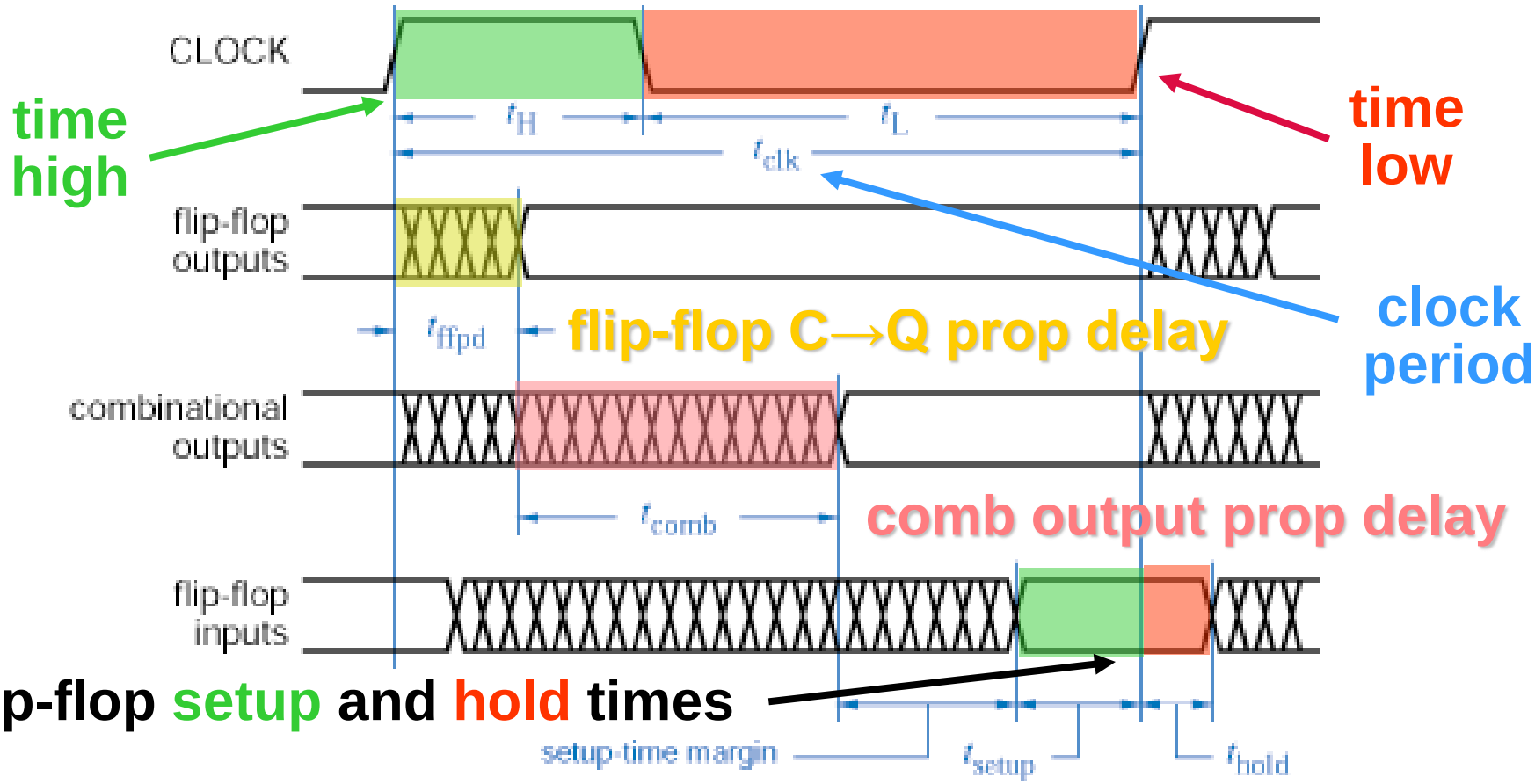
Clock Signals (active high/low)



Timing Diagrams and Specifications

- **For synchronous systems, timing diagrams can be used to show the relationship between the clock and various input, output, and internal signals**

clock frequency (f) = $1/t_{clk}$ duty cycle = $t_H/(t_H+t_L)$



External Memory Timing (NE64)

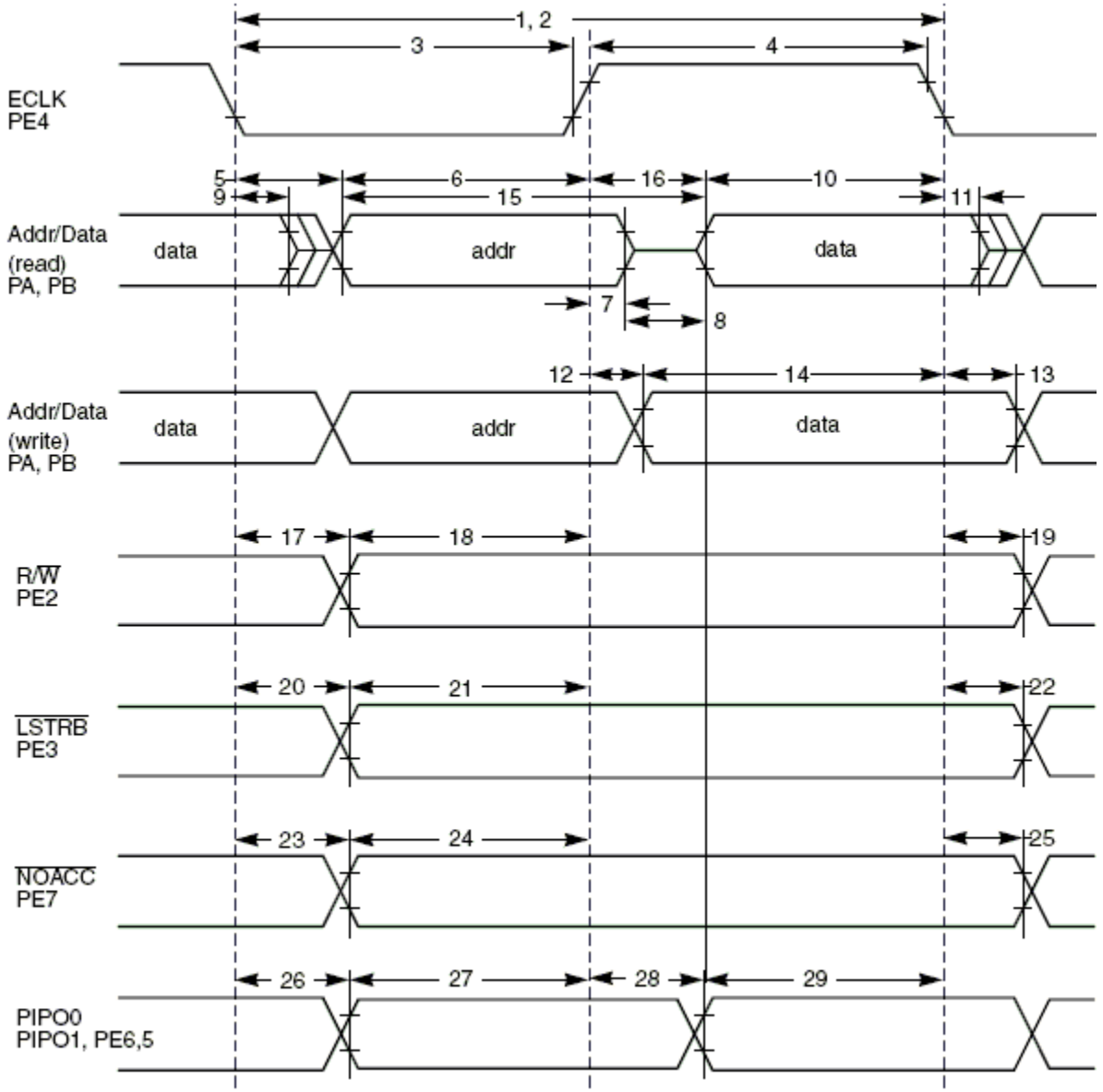
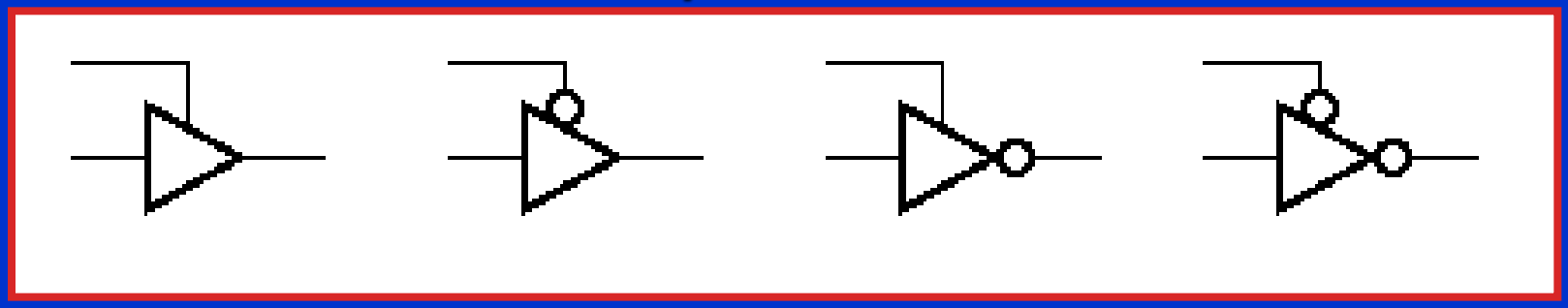


Figure 18-25. General External Bus Timing

Tri-State Buffers

- **Three-state (“tri-state”) devices allow multiple sources to share a single “party line” (provided only one device “talks” on the line at a time)**

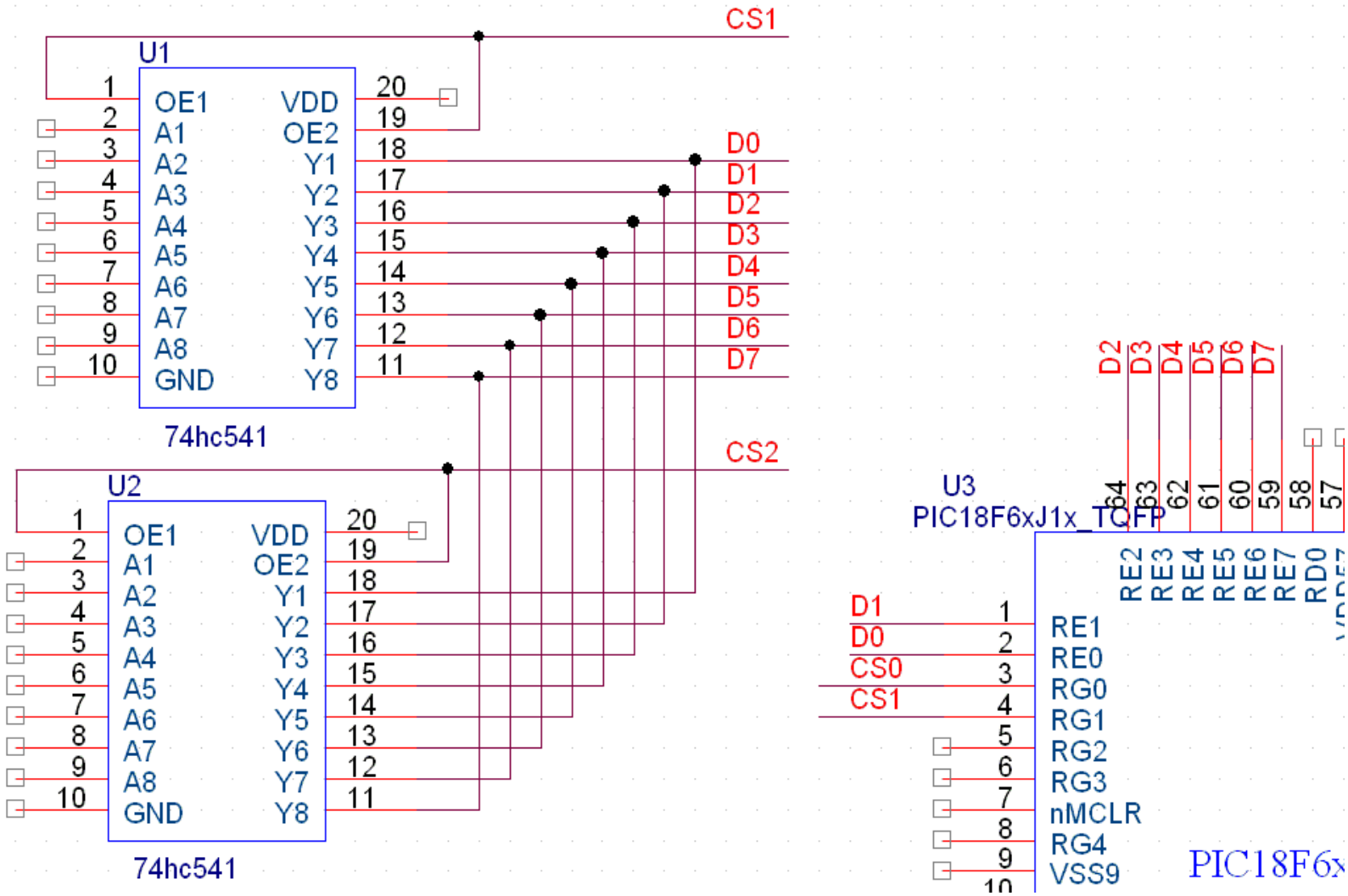


<u>Basic variations</u>: The buffer may be inverting or non-inverting, and the tri-state enable can either be active low or active high

Tri-State Buffers

- Three-state devices are typically designed so that they ***go into*** the Hi-Z (high impedance) state faster than they ***come out of*** the Hi-Z state (i.e., t_{pLZ} and t_{pHZ} are both less than t_{pZL} and t_{pZH})
- Given this "rule", if one tri-state device is disabled and another tri-state device is enabled simultaneously, then the first device will get ***off*** the bus before the second one gets ***on*** – this prevents ***fighting*** (a.k.a. ***bus contention***)

Why we need tri-state buffers:



Tri-State Buffers

- In practice it is difficult to ensure that the enable inputs of different tri-state devices change “simultaneously”
- The only ***safe*** way to use tri-state devices is to design control logic that ***guarantees*** a ***dead time*** on the bus line during which none of the buffers is driving it